



HAL
open science

Mesure de la difficulté des jeux vidéo

Guillaume Levieux

► **To cite this version:**

Guillaume Levieux. Mesure de la difficulté des jeux vidéo. Interface homme-machine [cs.HC]. Conservatoire national des arts et métiers - CNAM, 2011. Français. NNT : 2011CNAM0761 . tel-00612657

HAL Id: tel-00612657

<https://theses.hal.science/tel-00612657v1>

Submitted on 29 Jul 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

le **cnam**

CONSERVATOIRE NATIONAL DES
ARTS ET MÉTIERS

Ecole Doctorale Informatique, Télécommunications et Electronique
Equipe ILJ - Groupe MIM - Laboratoire CEDRIC

THESE présentée par :

Guillaume Levieux

soutenue le : 9 Mai 2011

pour obtenir le grade de :

Docteur du Conservatoire National des Arts et Métiers
Informatique / Multimédia

Mesure de la difficulté dans les jeux vidéo.

Thèse dirigée par :

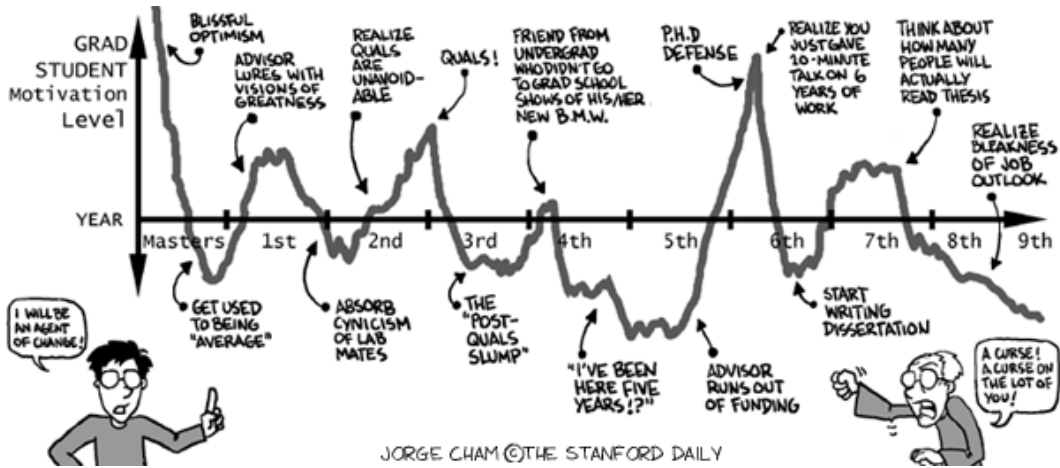
NATKIN Stéphane
APONTE Maria Virginia

Rapporteurs :

COURCHESNE Luc
ESTRAILLIER Pascal
TIJUS Charles

Jury :

APONTE Maria Virginia
BUENDIA Axel
COURCHESNE Luc
CUBAUD Pierre - Président du Jury
ESTRAILLIER Pascal
NATKIN Stéphane
TIJUS Charles



Courbe de difficulté d'une thèse - Jorge Cham.

Remerciements

Je tiens à remercier ceux qui m'ont aidé tout au long de cette thèse, et pour commencer, ceux sans qui je ne l'aurais jamais débutée. Je tiens à marquer tout d'abord ma profonde affection pour tout ce que représente le Conservatoire National des Arts et Métiers. J'ai en effet suivi une scolarité banale jusqu'au secondaire, puis naturellement intéressé par les jeux vidéo et motivé par l'achat d'un ordinateur, je me suis inscrit en IUT d'informatique. A l'issue de ces deux années, mon stage m'a permis d'être embauché, et mes études auraient logiquement dû s'arrêter là.

J'ai alors eu la chance de croiser mon ancien professeur de C++, Mr Delobel, au milieu d'un supermarché. Il m'a donné un vrai conseil gratuit et personnel : je le revois me vanter avec passion les mérites du CNAM au milieu des boites de conserves, et j'en ai été profondément touché. C'est la première personne que je souhaite remercier, sans qui je n'aurais jamais poursuivi mes études.

Je remercie donc l'ensemble des enseignants du CEDRIC qui m'ont eu comme élève au CNAM. Je me suis doucement rapproché des média interactifs par plaisir, attiré par les UV de conception d'applications multimédia, et les connaissances originales et variées qu'on m'y a transmis. De fil en aiguille, j'ai pu découvrir qu'il était non seulement possible d'utiliser deux webcams pour simuler un jeu vidéo de hockey sur table, mais qu'on pouvait également s'y consacrer complètement en tant que chercheur. Mes encadrants de l'époque, Alexandre Topol et Jérôme Dupire, m'ont alors dispensé le second conseil qui m'ait fait réviser mes objectifs de carrière : entrer en thèse au sein du CEDRIC. Je ne l'aurais jamais envisagé sans eux, et je les remercie de m'y avoir amené.

J'ai eu la chance, durant ma thèse, de bénéficier d'un duo d'encadrants particulièrement efficace et adapté à mon mode de fonctionnement. Mon directeur, Stéphane Natkin, a su me donner la liberté nécessaire à ce qu'Eric Gressier a appelé un jour *l'apprentissage de l'incertitude*. Il a su me diriger et m'ouvrir la voie sans jamais m'obliger, et j'ai pu bénéficier du luxe ultime de me sentir parfois perdu, en sachant toujours que notre prochain entretien me remettrait sur les rails. Sous son impulsion, j'ai eu la chance de pouvoir élargir mon horizon scientifique et de vivre ma thèse comme un apprentissage passionnant. J'ai de plus bénéficié du suivi de ma co-directrice, Maria Virginia Aponte, qui a su m'aider à garder un esprit synthétique, et à développer nos idées tout en les rassemblant de manière toujours

plus construite. Je les remercie tous les deux pour leur délicate attention, et pour m'avoir permis de vivre cette thèse dans de parfaites conditions.

Je remercie bien sûr également ma compagne, Hélène, pour m'avoir soutenu, rassuré et stabilisé. Tu as su m'extirper doucement des mondes virtuels pour m'ouvrir à tant de choses qui m'échappaient jusqu'alors, et cette thèse est l'étape d'un trajet que j'ai avant tout parcouru avec toi. Merci d'avoir été là et de m'avoir autant donné.

Je remercie pour finir mes parents. J'ai eu mon enfance délicieusement bercée d'une ambiance *labo*, entouré d'éprouvettes et de filtre-toupies, et ce n'est finalement sûrement pas un hasard si j'ai suivi cette voie.

Table des matières

Remerciements	3
Resumé de la thèse	13
1 Introduction	17
1.1 Vocabulaire	21
1.1.1 Design de jeu vidéo	21
1.1.2 La boucle de jeu	22
1.1.3 Difficulté et complexité	23
2 La difficulté élément de gameplay	25
2.1 Définition générale	26
2.2 Manipulation de la difficulté	29
2.2.1 Courbe de difficulté	30
2.2.2 Scénarisation	32
2.2.2.1 Scénarisation et recherche	33
2.2.3 Equilibrage dynamique	36
2.2.3.1 Equilibrage dynamique et recherche	38
2.3 Evaluation de la difficulté	40
2.3.1 Les tests de jouabilité	40
2.3.2 Test par joueur synthétique	42
2.4 Synthèse	43
3 Les différentes formes de difficulté	45
3.1 Difficulté sensorielle - Les jeux d'aventure	46
3.2 Difficulté logique - Les jeux de stratégie	49
3.3 Difficulté motrice - Les jeux d'actions	53
3.4 Synthèse des différentes dimensions de difficulté.	55
3.5 Etude de sous types	56
3.5.1 Collin McRae Dirt	56

3.5.2	Everest Poker	58
3.5.3	SimCity 3000	59
3.5.4	Fly For Fun	60
3.6	Conclusion	61
4	La difficulté du point de vue du joueur	63
4.1	Approche expérimentale	64
4.1.1	Challenge, curiosité et fantaisie	64
4.1.2	Hard Fun et Easy Fun	67
4.1.3	Autodétermination	67
4.1.4	Choix et compétition	69
4.1.5	Playground et Pacman	69
4.1.6	Difficulté et premiers instants de jeu	72
4.1.7	Synthèse	72
4.2	Approche théorique	73
4.2.1	Renforcement, regret et dissonance cognitive	74
4.2.2	L'attribution causale	77
4.2.3	Sentiment d'efficacité personnelle	79
4.2.4	L'expérience optimale	81
4.2.5	Difficulté et émotions	84
4.3	Synthèse	85
5	Premières mesures de difficulté	89
5.1	Analyse de la difficulté et joueur synthétique	89
5.2	Premières expériences	90
5.2.1	Le cube-serpent	90
5.2.2	Skywar	92
5.2.3	Pacman	97
5.3	Limitations d'un joueur synthétique	99
6	Mesure de la difficulté	103
6.1	Objectif	104
6.2	Les différentes variables	104
6.2.1	L'effort nécessaire	105
6.2.1.1	Variables impliquées	106
6.2.1.2	Mesurabilité des variables	108
6.2.2	Perception	109
6.2.2.1	Variables impliquées	110
6.2.2.2	Mesurabilité des variables	110

6.2.2.3	Hypothèses	110
6.2.3	Décision	111
6.2.3.1	Variables impliquées	111
6.2.3.2	Mesurabilité des variables	112
6.2.3.3	Hypothèses	113
6.2.4	Action	114
6.2.4.1	Variables impliquées	114
6.2.4.2	Mesurabilité des variables	115
6.3	Modèle simplifié	115
6.4	Les challenges	117
6.5	Mesure de la performance	118
6.6	Mesure du niveau du joueur	119
6.7	Equation générale de la difficulté	121
6.8	Trace d'évènements	123
6.9	Calcul du niveau du joueur	124
6.10	Calcul de la difficulté	126
6.11	Apprentissage du joueur	128
6.11.1	Correspondance entre capacités locales	128
6.12	Temps et difficulté	129
6.13	Durée d'un challenge et théorie de la fiabilité	130
6.14	Tracer la courbe de difficulté	131
6.15	Synthèse	132
7	Modéliser le parcours du joueur	135
7.1	Objectif	135
7.2	Challenges et construction de la difficulté	137
7.3	Formalisations existantes	138
7.4	Modèle de scénario	141
7.4.1	Interface avec le moteur de jeu	141
7.4.2	Les challenges	143
7.4.3	Hypergraphe de scénario	145
7.4.4	Calcul de l'état du graphe	147
7.5	Synthèse	148
8	Logiciel de mesure de la difficulté	151
8.1	Objectif	151
8.2	Algorithme général de fonctionnement du logiciel	152
8.3	Interface	154
8.3.1	Edition du scénario	155

8.3.1.1	Mise à jour de l'état du jeu	156
8.3.1.2	Préconditions des challenges	157
8.3.2	Edition des algorithmes de calcul des capacités	158
8.3.3	Courbe de difficulté	160
8.4	Synthèse	161
9	Expérimentations	163
9.1	Objectifs de l'expérience	164
9.2	Dispositif expérimental	165
9.2.1	Choix matériels et logiciels	165
9.2.2	Gameplay et courbes de difficulté	167
9.2.3	Déroulement de l'expérience	171
9.2.4	Données recueillies	172
9.3	Résultats	173
9.3.1	Evaluation des capacités du joueur et difficulté	173
9.3.1.1	Analyse détaillée du challenge 4	175
9.3.1.2	Analyse de l'ensemble des challenges	179
9.3.2	Difficulté calculée et difficulté théorique	182
9.3.3	Difficulté calculée et difficulté subjective	184
9.3.4	Plaisir de jouer et niveau de difficulté	185
9.3.5	Plaisir de jouer et courbe de difficulté	189
9.4	Discussion et Conclusion	190
10	Conclusion	193
	Glossaire	200
	Bibliographie	201

Table des figures

1.1	La courbe de difficulté d'un jeu vidéo	17
1.2	Tétris (Nintendo)	23
2.1	Courbes de difficulté	31
2.2	Super Mario World (Nintendo)	31
2.3	The Legend of Zelda : A Link to the Past (Nintendo)	32
2.4	Fallout 3 (Bethesda)	34
2.5	Unreal Tournament 3 (Epic Games / Infogrames)	36
2.6	Pure (Black Rock Studio / Disney Interactive)	37
2.7	Battlefield Heroes (Electronic Arts / Dice)	41
3.1	Indiana Jones et le destin de l'Atlantis (Lucas Art)	46
3.2	Aces of The Galaxy (Artech Studios - Sierra)	48
3.3	Heroes of Might and Magic 3 (The 3DO Company)	49
3.4	Starcraft (Blizzard)	50
3.5	Arbre de jeu	52
3.6	Doom (Id Software)	53
3.7	Dimensions de difficulté.	56
3.8	Collin McRae Dirt (Codemasters)	56
3.9	Everest Poker (Everest Gaming)	58
3.10	SimCity 3000 (EA Games)	59
3.11	Fly For Fun (Aeonsoft / Gala-Net)	60
4.1	Univers ludiques de Yannakakis et al	70
4.2	Lien entre plaisir du joueur et dimensions de complexité.	73
4.3	Boîte de skinner et jeu vidéo	74
4.4	Le flow : équilibre entre challenge et capacités [Csikszentmihalyi 91].	82
4.5	Le flow et la difficulté relative.	83
5.1	Cube-serpent	90

5.2	Section rigide d'un cube-serpent.	91
5.3	Temps d'exécution pour découvrir la solution de cube-serpents.	92
5.4	Skywar (MotionTwin)	93
5.5	Simulateur de SkyWar	94
5.6	Résultats obtenus grâce au <i>build harpies</i>	96
5.7	Pacman chassé par Blinky.	98
5.8	Score de Pacman à différentes vitesses	99
6.1	Relations entre les variables du modèle de difficulté.	105
6.2	Relation entre niveau du joueur et effort nécessaire pour réussir le challenge.	106
6.3	Perception de l'effort.	110
6.4	Détermination de l'effort fourni.	111
6.5	Action du joueur.	114
6.6	Modèle simplifié de difficulté.	116
6.7	Automate d'un challenge.	117
6.8	Réseau bayésien : causalité entre capacités et performance.	122
7.1	Trois temps du développement d'un processus [Bremond 74]	139
7.2	Interface entre jeu est scénario.	142
7.3	Automate d'un challenge.	144
7.4	Exemple de lien	146
8.1	Fonctionnement du logiciel	153
8.2	Logiciel de calcul de la difficulté	155
8.3	Représentation d'un challenge	155
8.4	Exemple de lien entre challenges	156
8.5	Probabilité d'échec en fonction de la précision au tir.	160
8.6	Courbe de difficulté.	161
9.1	L'ennemi du joueur	166
9.2	Liste des challenges.	168
9.3	Courbe de difficulté relative et théorique de type <i>suivi</i>	169
9.4	Courbe de difficulté de type <i>progression puis oscillation</i>	169
9.5	Difficulté théorique d'un challenge.	170
9.6	Questionnaire sur la difficulté et le plaisir du joueur	171
9.7	Liste des événements.	172
9.8	Age des participants	173
9.9	Nombre de challenges joués par joueur	174
9.10	Quantité de données par challenge.	175
9.11	Distributions des variables et tests de normalité (Shapiro/Wilks).	176

9.12	Tableau de corrélation capacités et résultat (Coefficient de Pearson).	176
9.13	Tableau de contingence entre précision et résultat.	177
9.14	Classes de niveau pour la capacité Tirer	178
9.15	Régression linéaire entre précision et difficulté	179
9.16	Corrélation capacités et résultat par challenge (Coefficient de Pearson). . . .	180
9.17	Predictions de difficulté pour chaque résultat obtenu sur données de validation.	181
9.18	Distributions des prédictions de difficulté pour chaque difficulté théorique. . .	182
9.19	Distributions des prédictions de difficulté pour chaque difficulté théorique (niveau du joueur réév	
9.20	Tableau de corrélation de la difficulté subjective.	184
9.21	Distributions des prédictions de difficulté pour chaque niveau difficulté subjective.	185
9.22	ρ entre plaisir de joueur et difficulté.	186
9.23	Distributions des difficultés subjectives pour chaque niveau de plaisir.	186
9.24	Distributions des difficultés calculées pour chaque niveau de plaisir.	187
9.25	Régression linéaire entre écarts de difficulté objective et plaisir.	188
9.26	Plaisir du joueur pour chaque type de courbe de difficulté.	189
9.27	Distributions du niveau de plaisir pour chaque type de courbe de difficulté. . .	190

Resumé de la thèse

Cette thèse a pour objectif d'étudier la notion de difficulté dans les jeux vidéo. Plus spécifiquement, nous cherchons à mettre au point une méthode et un outil de mesure de la difficulté d'un **gameplay**¹. Le contrôle de la progression de la difficulté joue un rôle prépondérant dans la construction d'un jeu vidéo. Les théoriciens du jeu lui accordent une grande importance, et de nombreuses études démontrent le lien entre difficulté et plaisir du joueur. Cependant, il n'existe aucune définition généralement acceptée de la notion de difficulté dans un jeu et à fortiori aucune méthode de mesure de la difficulté applicable à tous les jeux. Les principes de construction d'un jeu à partir de la progression corrélée de l'apprentissage et de la difficulté sont empiriques et difficilement vérifiables expérimentalement. La méthode de mesure couramment employée consiste à déterminer une heuristique propre à chaque jeu, dont l'interprétation peut être délicate et qui n'autorise pas de comparaison entre différents jeux. Nous souhaitons proposer une méthode qui s'appuie sur les principes fondamentaux que partagent tous les gameplay, de manière à obtenir une évaluation générique et plus facile à interpréter de la difficulté d'un jeu vidéo.

L'état de l'art porte, dans un premier temps, sur la notion de difficulté telle que considérée dans la pratique et la théorie de la conception des jeux. Dans une seconde partie nous examinons les travaux portant sur l'aspect psychologique de la notion de difficulté et sa relation au plaisir de jouer.

Ceci nous amène à proposer de définir la difficulté comme l'effort fourni par le joueur pour atteindre ses objectifs. Nous précisons ce point de vue en nous concentrant tour à tour sur différents types de difficulté. Nous nous inspirons d'un modèle général de traitement de l'information pour proposer de distinguer les difficultés selon plusieurs catégories : *sensorielle*, *logique* et *motrice*. Cette catégorisation a pour but d'aider le game designer à comprendre la difficulté de son gameplay, en l'examinant au travers de chacun de ces trois prismes.

Par la suite, nous analysons un ensemble d'hypothèses concernant la forme optimale d'une courbe de difficulté, c'est à dire celle qui contribue le plus fortement au plaisir du joueur. La courbe de difficulté décrit l'évolution de la difficulté dans le temps au cours d'une partie. Le game designer, concepteur du jeu vidéo, travaille cette courbe pour contribuer à fournir une expérience de jeu optimale, et les hypothèses que nous avançons peuvent l'aider dans cette tâche.

Nous explorons ensuite plusieurs méthodes de mesure de la difficulté basées sur la complexité des algorithmes d'un joueur synthétique et montrons leur intérêt particulier dans la compréhension de la difficulté d'un gameplay. Ces méthodes de test basées sur l'utilisation d'un joueur synthétique permettent de détecter les déséquilibres d'un gameplay, mais ne sauraient simuler la complexité du système cognitif du joueur. Les valeurs calculées par ces méthodes ne permettent donc pas de déduire la difficulté réelle d'un jeu vidéo, c'est à dire l'effort que devra fournir un joueur humain.

A partir de notre étude bibliographique et à l'issue des premières expérimentations, nous proposons une définition formelle de la difficulté d'un jeu vidéo qui s'appuie sur deux principes fondamentaux. Tout d'abord, tout gameplay est décomposable en une suite de challenges pour lesquels il existe un instant t au delà duquel le joueur a réussi ou échoué. Ensuite, un gameplay exploite un nombre fini de *capacités* chez le joueur. Ces capacités sont en nombre restreint, mesurables à court terme à partir d'une trace des événements du jeu, et dépendent du style général du gameplay. Le nombre de challenges peut être très élevé, et correspond aux différentes situations où le game designer met à l'épreuve les capacités du joueur. Nous proposons de mesurer la difficulté d'un jeu comme la probabilité d'échec du joueur à chaque challenge, connaissant les capacités qu'il a démontré et que nous avons pu mesurer pendant la session de jeu. Cette mesure est applicable à tout gameplay respectant nos principes fondamentaux.

La première étape d'une analyse de la difficulté consiste donc à découper le gameplay en un ensemble de challenges ordonnés. C'est ce que nous appelons le scénario du jeu. Nous proposons un langage qui permet de représenter graphiquement la suite de challenges ainsi que les conditions qui gouvernent leur présentation au joueur.

Nous présentons ensuite un logiciel complet de mesure de la difficulté, qui exploite le langage de représentation du scénario mais permet également de spécifier les algorithmes de mesure des capacités du joueur et d'interpréter les traces enregistrées lors des expériences, ainsi que de présenter graphiquement les résultats.

L'ensemble de ce travail a fait l'objet d'une expérience menée dans le cadre d'un jeu de tir à la première personne. Nous avons construit un niveau de jeu basé sur des hypothèses réalistes et classiques d'apprentissage des capacités et de progression de la difficulté. Nous avons ensuite instrumenté ce jeu à l'aide de notre logiciel. Deux expériences réalisées par des ergonomes et une mesure aussi rigoureuse que possible ont été menées. Nous montrons tout d'abord la cohérence de notre définition : il est possible de mesurer les capacités du joueur, et ces capacités sont effectivement liées à la difficulté d'un challenge. Nous montrons également son utilité : les hypothèses faites lors de la conception d'un jeu ne sont pas toujours validées expérimentalement, et la mesure de la difficulté permet d'améliorer cette étape de conception. Nous montrons également qu'à partir d'un modèle linéaire du lien entre capacité du joueur et probabilité d'échec, il est possible de prédire le résultat d'un challenge avec une bonne précision. Enfin une enquête qualitative établit que des raisonnements trop simples

sur la relation entre le plaisir de jouer et progression de la difficulté ne sont pas toujours exacts.

Ce travail peut être étendu de nombreuses façons. D'un point de vue pratique notre logiciel peut être amélioré et intégré dans un moteur de jeu et d'autres expériences peuvent être menées. D'un point de vue théorique, la mesure des capacités et de la difficulté en distinguant le sensoriel, le logique et le moteur peuvent être affinées. Enfin ce travail peut conduire à terme à la définition de principes d'auto adaptation de la difficulté aux capacités du joueur.

Chapitre 1

Introduction

Sommaire

1.1	Vocabulaire	21
1.1.1	Design de jeu vidéo	21
1.1.2	La boucle de jeu	22
1.1.3	Difficulté et complexité	23

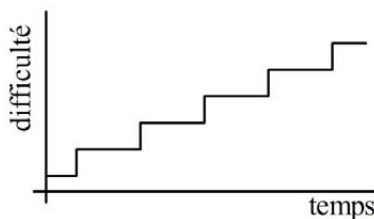


FIGURE 1.1 – La courbe de difficulté d’un jeu vidéo

Cette première figure illustre la problématique à l’origine de ce travail de thèse. C’est le schéma qu’a tracé mon directeur de thèse, en introduction de notre toute première discussion sur ce qu’a été mon sujet de préoccupation pendant ces trois années. Ce fut parfois, au milieu du flou qui, je crois, caractérise l’apprentissage de la recherche pour tout jeune thésard, la seule balise intacte vers laquelle je savais devoir concentrer tous mes efforts : la courbe de difficulté.

Cette courbe est d’une importance capitale pour tout concepteur de jeu. Elle exprime la progression de la difficulté d’un jeu en fonction du temps, et comme nous le montrerons, la difficulté est au coeur du game design et fortement liée au plaisir du joueur. Il n’existe cependant à ce jour aucune méthode qui permette d’évaluer cette courbe sans ambiguïté, pour n’importe quel gameplay. La méthode couramment employée consiste à concevoir une mesure

heuristique, complexe à interpréter et qui ne permet pas de comparaisons entre gameplay. Dans cette thèse, notre objectif général consiste à proposer une mesure générique et facile d'interprétation de la difficulté d'un gameplay. Une telle méthode serait particulièrement utile aux **game designer**¹ et **level designer**², dont une des tâches les plus importantes consiste à sculpter une courbe de difficulté qui maximise le plaisir du joueur.

Implicitement, la plupart des joueurs connaissent le principe d'évolution de la difficulté : au fur et à mesure qu'un joueur progresse dans l'univers du jeu, il sait que son personnage ainsi que ses ennemis ne seront que plus gros, plus forts et mieux armés jusqu'au dénouement final. Comme l'explique Ralph Koster, tout jeu est voué à perdre sa saveur, une fois son contenu entièrement découvert et ses obstacles franchis par le joueur [Koster 05b]. C'est un exercice auquel se plie le game designer avec bienveillance, ses énigmes sont faites pour être percées à jour, ses niveaux explorés et son intelligence artificielle vaincue. Au fur et à mesure que le joueur comprend la dynamique du jeu, cette dernière se complexifie et lui offre de nouvelles possibilités d'exploration, d'apprentissage, jusqu'à épuiser tout son potentiel.

La courbe de difficulté traduit un mécanisme général, connu de tous et particulièrement important, mais comme de nombreux aspect du game design, aussi limpide en apparence que complexe à définir précisément. Chacun reconnaît par exemple d'une manière intuitive une activité ludique, et sa définition peut sembler évidente. Néanmoins, une définition précise et universelle du jeu se révèle complexe et un sujet de discussion ouverte parmi les théoriciens du game design. Il en va de même pour la difficulté. N'importe quel joueur sait dire si une section précise d'un jeu lui semble facile ou difficile, mais à notre connaissance, personne n'a proposé de méthode de mesure générale de la difficulté d'un jeu vidéo. La méthode évidente proposée jusqu'alors consiste à s'appuyer sur le game designer pour obtenir des métriques spécifiques à son game design : nombre d'ennemis, points de vies, taille du barillet du revolver, puis à combiner linéairement ces métriques.

Si cette approche semble satisfaisante à première vue, elle ne propose en réalité que de placer le problème de la mesure sous la responsabilité du game designer. Pourtant, comme tout artiste, le game designer travaille de manière itérative, intuitive, et il n'est absolument pas évident que pour construire un bon game design, il lui soit nécessaire d'être capable d'en formaliser le contenu. Une expérience vidéoludique émerge des règles du jeu, et si le game designer a l'intuition que l'interaction de ces règles sera plaisante, il lui faudra néanmoins prototyper et mettre à l'épreuve son game design pour tester et ressentir l'expérience qu'il procure. Le game designer n'a pas besoin d'avoir conscience de toutes les variables qu'il manipule lorsqu'il suit son intuition et construit un gameplay, de la même manière qu'un musicien, un peintre ou un scénariste n'ont pas forcément besoin de formaliser le fonctionnement de leur art pour l'exercer. L'identification des variables de difficulté n'est donc pas forcément une opération évidente pour le game designer, bien qu'on puisse attribuer à ce dernier la paternité du gameplay. La mesure de la courbe de difficulté d'un jeu vidéo reste donc, d'une manière générale, un problème entier.

Pourtant, s'il n'est pas nécessaire de fournir une mesure objective de la courbe de difficulté pour élaborer un game design, tout game designer peut par contre bénéficier des apports d'un travail théorique sur ce sujet. La théorie du game design fournit un langage commun aux différents développeurs de jeux, elle leur permet de disposer d'objets théoriques sur lesquels s'appuyer pour guider leur recherche. La courbe de difficulté semble être une abstraction particulièrement pertinente des propriétés d'un gameplay. Elle est identifiée par les théoriciens du jeu comme un principe de base de construction d'un jeu vidéo, et comme nous le verrons par la suite, de nombreux mécanismes liés au plaisir du joueur semblent reposer sur sa progression. Il semble donc nécessaire et particulièrement intéressant d'aller au delà d'une définition basique et fortement contextuelle de la difficulté d'un jeu vidéo.

La difficulté d'un jeu est donc un concept aujourd'hui défini de manière implicite, mais commune à l'expérience que procure tout gameplay et donc d'une grande valeur théorique pour la conception d'expériences vidéo ludiques. Il semble donc nécessaire de fournir une définition théorique plus précise de la difficulté d'un jeu vidéo, d'établir et de valider des méthodes de mesure ainsi que d'étudier plus précisément sa relation avec l'expérience vécue par le joueur.

Nous débutons cette thèse au chapitre 2 par une étude des travaux de différents game designer et théoriciens du jeu vidéo. Nous cherchons en premier lieu à montrer le lien profond qui unit difficulté et jeu vidéo et à en fournir une définition générale, en étudiant directement les travaux qui s'attachent à définir la notion même de jeu. Cette première étape, théorique et bibliographique, permettra d'offrir une synthèse de différents travaux réalisés dans le cadre d'une mesure de la difficulté d'un gameplay.

Au chapitre 3, nous décrivons ensuite plus précisément les différentes formes de difficulté, à partir des types les plus communs de jeu vidéo et en nous basant sur un modèle général de traitement de l'information. Nous étudions différents gameplay, et montrons de quelle manière ils exploitent ces différents types de difficulté.

Nous abordons ensuite la difficulté du point de vue du joueur au chapitre 4. Nous cherchons à comprendre de quelle manière la difficulté influe sur la psychologie du joueur et favorise son plaisir au cours d'une expérience ludique. Cette approche nous permet de mieux comprendre l'importance de la difficulté dans un gameplay, ainsi que de proposer certaines hypothèses quant à la forme optimale d'une courbe de difficulté, c'est à dire la forme à même de susciter le plus grand plaisir chez le joueur.

Au chapitre 5, nous présentons plusieurs expériences, au cours desquelles nous avons mesuré la difficulté de différents gameplays par le biais d'un joueur synthétique. Dans ce cas, mesurer la difficulté revient à mesurer la complexité en temps de l'algorithme qui remplace le joueur. Ces mesures confirment l'utilité d'une mesure de la difficulté, mais également les faiblesses d'une approche algorithmique ainsi que la nécessité d'une approche plus générale. En effet, l'approche algorithmique ne permet pas de tenir compte de la complexité du système cognitif des joueurs. Également, les méthodes de mesure couramment employées s'appuient

sur une heuristique proposée par le game designer, fonction de certaines variables du jeu, dont l'interprétation peut s'avérer particulièrement délicate et donc aboutir à une évaluation possiblement erronée de la difficulté.

Nous proposons donc par la suite, au chapitre 6, une définition générale de la difficulté, qui s'appuie sur une base théorique stable et d'interprétation directe et indiscutable. Nous adoptons pour principe que la plupart des gameplay sont décomposables en une suite de **challenges**³, et que pour chacun d'entre eux, il existe un instant t au delà duquel le game designer est en mesure de déterminer si le joueur a réussi ou échoué à atteindre les objectifs qui lui sont fixés. Nous montrons également que la difficulté varie en fonction des capacités du joueur, et que certaines de ces capacités sont mesurables à partir d'un traitement algorithmique de traces d'évènements du jeu. Nous définissons donc finalement la difficulté comme la probabilité d'échec du joueur à un challenge, étant donné son niveau, c'est à dire les capacités dont il a fait preuve au cours de ce challenge d'après nos mesures. Evaluer la difficulté d'un challenge consiste à évaluer statistiquement cette probabilité, au moyen d'une méthode que nous détaillons. Nous définissons également un ensemble de conditions à respecter lors de la mise en place d'un playtest, de manière à limiter l'impact de différentes variables pouvant perturber la mesure.

Par la suite, nos travaux présentés chapitre 7 ont consisté à mettre au point l'interface et le logiciel nécessaires à l'analyse d'un gameplay. La première étape d'une analyse de gameplay consiste à définir la suite de challenges qui le compose, c'est à dire, au sens où nous l'entendons dans cette thèse, le scénario du jeu. Nous proposons donc un langage graphique qui permet de représenter les challenges du jeu et les conditions logiques qui gouvernent l'ordre avec lequel ils seront présentés au joueur.

Dans un second temps, au chapitre 8, nous présentons plus généralement le reste de l'application, qui permet de définir les algorithmes de calcul du niveau du joueur, ainsi que de traiter des traces d'évènements, enregistrées au cours d'une partie, pour effectuer les différents calculs de difficulté.

Pour finir, dans le chapitre 9, nous détaillons les résultats d'une expérimentation, au cours de laquelle nous avons exploité notre modèle pour calculer la difficulté d'un gameplay. Nous montrons qu'il est effectivement possible, au moins pour le gameplay étudié, de définir et de mesurer les capacités du joueur, et de mettre en évidence le lien qui unit le niveau du joueur et le résultat qu'il obtient. Nous étudions le niveau de précision que notre modèle permet d'atteindre, et discutons de ses faiblesses et améliorations envisagées.

Avant de débiter la revue de littérature, nous terminons ce chapitre par une première définition de vocabulaire classique employé dans l'étude théorique des activités ludiques.

1.1 Vocabulaire

Avant toute chose, il est nécessaire d'introduire rapidement une partie du vocabulaire employé par la suite. Le game design est un domaine particulier, au vocabulaire largement influencé par des termes anglo-saxons dont la traduction française serait aussi artificielle que déroutante. D'autres termes nécessitent d'être précisément définis, afin de gagner en clarté pour le reste de notre propos. Le vocabulaire absent de cette première partie sera défini à la volée et figurera dans le glossaire en fin de document.

1.1.1 Design de jeu vidéo

Robin Hunicke, de l'université du Northwestern, propose de décomposer un jeu vidéo en trois niveaux : sa *mécanique*, sa *dynamique* et son *esthétique* [Hunicke 04].

La *mécanique* d'un jeu représente l'ensemble de ses composants logiciels, pris hors de leur contexte. Par exemple, la plupart des **First Person Shooter**⁴ récents s'appuient sur des outils de simulation physique, d'animation procédurale, de calcul d'effets sonores en temps réel. Ces outils pourraient être exploités dans d'autres cadres, par exemple dans la réalisation de films d'animation, et constituent support mécanique d'un jeu vidéo.

La *dynamique* d'un jeu vidéo s'appuie sur sa mécanique, et exprime le comportement des outils pour un jeu vidéo particulier. Toujours dans un First Person Shooter, la règle qui détermine que tirer sur une caisse avec un fusil à pompe déclenche son explosion et l'apparition d'un bonus fait partie de la dynamique du jeu. La dynamique exploite le moteur physique pour calculer l'impact de l'arme et la trajectoire du bonus, elle implante les *règles du jeu* par le truchement de la mécanique.

L'*esthétique* d'un jeu décrit l'impact émotionnel de la dynamique sur le joueur. Lorsque la caisse détruite libère un bonus, cette récompense apportée au joueur a pour résultat un impact émotionnel positif.

Cette modélisation du jeu vidéo permet de mettre en perspective les différents métiers relatifs à leur création. En effet, parmi les concepteurs d'un jeu, tous ne sont pas préoccupés par les problématiques de difficulté. Les programmeurs, par exemple, se concentrent sur la mécanique du jeu. Ils conçoivent les outils d'édition et les différents blocs logiciels chargés de faire fonctionner le jeu. Leur rapport avec la difficulté est lointain : ils doivent fournir des composants les plus malléables possibles, capables de créer une grande variété de dynamiques différentes. Mais ces composants logiciels sont ensuite exploités par d'autres corps de métiers, plus spécialisés dans la conception de la dynamique du jeu. Les graphistes et designers sonores ont une approche plus esthétique, et leurs créations contribuent fortement à l'impact émotionnel d'un jeu vidéo. Par contre, cet impact ne dépend pas d'un mécanisme lié à la difficulté : les graphistes et designers sonores n'interviennent pas directement dans la conception de la dynamique du jeu.

La dynamique du jeu concerne principalement les *game designers* et les *level designers*, qui sont respectivement à l'origine de ses principes de fonctionnement et de leur implantation tout au long du jeu⁵. Le *game designer*, ou *designer du jeu*, conçoit le *gameplay* et donc la dynamique du jeu. Il détermine les règles du jeu, c'est-à-dire le fonctionnement de son univers, l'ensemble des actions auxquelles le joueur a accès ainsi que leur impact sur les objets du jeu. Le *level designer*, de son côté, a la tâche de traduire les choix du *game designer* de manière concrète sur l'ensemble du jeu. Il va utiliser les outils fournis par les programmeurs et les ressources créées par les graphistes et *sound designers* pour construire l'univers du jeu, en veillant à ce que le parcours du joueur dans cet univers respecte les choix du *game designer*. Le *game designer* a donc une vision décontextualisée du *gameplay*, dont il détermine les principes. Le *level designer* en a une vision plus contextualisée. La littérature propre à ces deux univers, *level design* et *game design*, va nous être particulièrement utile pour comprendre la nature et le fonctionnement de la difficulté des jeux vidéo.

1.1.2 La boucle de jeu

Dans un jeu, le joueur est face à des objets, avec lesquels il interagit : les *objets du jeu*. L'ensemble de ces objets constitue l'*univers du jeu*. Dans un jeu vidéo, certains objets sont représentés à l'écran et le joueur peut ainsi observer une partie de l'univers. Les objets peuvent être dynamiques, c'est-à-dire que leurs propriétés peuvent varier au cours du temps. Elles varient alors en respectant un comportement bien précis, dicté par le *game designer* au travers des *règles du jeu*.

Le joueur a la possibilité de modifier certaines propriétés des objets du jeu, au moyen d'*actions*. Ces actions sont elles aussi soumises aux règles du jeu : le créateur du jeu a construit chacune de ces actions en définissant précisément leur impact sur les propriétés des objets du jeu. Une manière d'identifier une action consiste à chercher à décomposer le comportement du joueur. Une fois atteint que le comportement du joueur ne peut plus être décomposée en sous parties ayant un sens pour le système de jeu, alors nous avons identifié une action [Koster 05a].

L'*état du jeu* correspond à la valeur des propriétés des objets du jeu, à un instant donné. C'est un instantané de l'univers du jeu.

La *boucle de jeu* correspond au calcul répétitif d'un état du jeu E_{n+1} à partir de l'état courant E_n . A chaque tour de boucle, le système met à jour les objets dynamiques du jeu, et vérifie si le joueur n'a pas effectué une action. Si c'est le cas, l'effet de l'action est appliqué aux objets concernés. Ces modifications aboutissent au calcul d'un nouvel état du jeu.

Certaines des règles du jeu sont chargées d'évaluer les transitions d'un état E_n à un état E_{n+1} , et peuvent distribuer au joueur récompenses et punitions en fonction de cette

5. *Design* vient du latin *designare*, définit dans le Gaffiot comme « *marquer (d'une manière distinctive), représenter, dessiner* ».

évaluation. Le *score* est une représentation cumulée de ces évaluations.

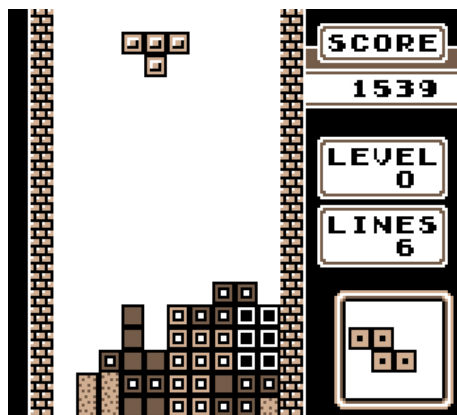


FIGURE 1.2 – Tétris (Nintendo)

Par exemple, dans un jeu comme Tétris, les pièces qui tombent du haut vers le bas de l'écran sont des objets du jeu (Figure 1.2). Les règles du jeu déterminent leur vitesse de chute, leur manière de s'empiler et de disparaître. Le joueur a accès à différentes actions : déplacer une pièce, la faire tourner ou la faire descendre plus rapidement. Ces actions lui permettent uniquement de manipuler la pièce en train de tomber, et seulement dans certaines directions, tout en respectant les contraintes imposées par les pièces déjà empilées.

A chaque tour de la boucle de jeu, la position de la pièce qui chute est recalculée, et si le joueur a demandé une rotation, un déplacement ou une accélération, celles ci sont appliquées. Lorsque la pièce entre en collision avec les autres pièces empilées, le système vérifie si une ligne complète, de la largeur de l'écran, est formée. Si c'est le cas, la ligne disparaît. De plus, le système reconnaît alors une transition entre états qu'il évalue de manière positive. En récompense, le joueur est crédité d'un certain nombre de points. Par contre, si la pile de pièces dépasse une hauteur maximale, cette transition est évaluée de manière particulièrement négative et le joueur perd tous les points accumulés jusque là et sa partie est terminée.

1.1.3 Difficulté et complexité

Une dernière précision sémantique nécessaire concerne la distinction entre les termes de difficulté et de complexité. Le Larousse définit *complexe* comme « *Qui contient plusieurs parties ou plusieurs éléments combinés d'une manière qui n'est pas immédiatement claire pour l'esprit* » et définit *difficile* comme « *Qui n'est pas facile à réaliser, qui exige des efforts importants ; ardu, malaisé* ».

On remarque que la difficulté est une notion plus large que celle de complexité. La complexité qualifie la difficulté de comprendre la manière dont sont combinés plusieurs éléments. La difficulté englobe d'autres formes d'effort, comme l'effort physique pur par exemple : courir un marathon est une tâche difficile à réaliser, mais de faible complexité. Dans son emploi, la complexité peut donc directement qualifier un objet, une structure complexe par exemple, puisqu'elle décrit la difficulté qu'on a à comprendre son fonctionnement. Par contre, la notion de difficulté suppose qu'on précise une activité, comme pour la difficulté de porter une charge lourde par exemple.

En ce qui nous concerne, l'effort du joueur, s'il est confortablement installé et convenablement alimenté en café et petits gâteaux, est de nature essentiellement cognitive. La seule ressource que le joueur peut choisir d'investir dans sa tâche est son attention, que Mihaly Csikszentmihalyi décrit comme de l'énergie psychique [Csikszentmihalyi 91]. On peut ainsi dire que dans un jeu vidéo, un challenge d'une certaine difficulté demande une certaine quantité d'attention pour être résolu, et cette quantité d'attention dépend directement de la complexité du problème.

Nous nous autorisons donc à utiliser sans distinction les termes de complexité et de difficulté. Certes, le joueur interagit avec l'univers au moyen d'une interface et nous verrons qu'il doit être capable de la manipuler avec précision. Mais l'effort fourni tient plus à la précision du geste qu'à l'effort purement musculaire fourni par le joueur et, nous nous autoriserons donc à utiliser aussi bien le terme de complexité que de difficulté pour cette partie purement motrice de l'activité du joueur.

Les différents termes que nous venons de définir sont fondamentaux à l'étude des jeux vidéo et de leur gameplay. Ils vont nous permettre d'analyser précisément les différents types de jeu vidéo, et de comprendre leur manière respective d'exploiter la notion de difficulté.

Chapitre 2

La difficulté élément de gameplay

Sommaire

2.1	Définition générale	26
2.2	Manipulation de la difficulté	29
2.2.1	Courbe de difficulté	30
2.2.2	Scénarisation	32
2.2.3	Equilibrage dynamique	36
2.3	Evaluation de la difficulté	40
2.3.1	Les tests de jouabilité	40
2.3.2	Test par joueur synthétique	42
2.4	Synthèse	43

Dans ce premier chapitre, nous étudions la difficulté en tant qu'élément de gameplay, c'est-à-dire en tant que partie intégrante de n'importe quel jeu vidéo. Nous souhaitons tout d'abord faire ressortir, par une approche très générale, la place fondamentale qu'occupe la difficulté dans toute activité ludique. Nous proposerons de définir la difficulté d'un jeu comme l'effort que fournit le joueur pour atteindre ses objectifs, effort mis en scène par le gameplay.

Dans un second temps, nous étudierons de quelle manière, aujourd'hui, chercheurs et game designers approchent cette notion épineuse, afin d'être tout autant capable de manipuler que d'évaluer la difficulté d'un jeu vidéo.

Pour aborder le thème de la difficulté dans les jeux vidéo, une première étape consiste donc à examiner les définitions générales qu'ont proposées différents auteurs. Nous disposons tous d'une définition intuitive du jeu, qu'il convient de préciser et de compléter par une étude de ce qu'offre la littérature à ce sujet. Plusieurs auteurs ont réfléchi à la notion de jeu, par exemple dans son impact sur nos sociétés, ou avec l'objectif d'établir un vocabulaire commun

et de permettre ainsi un meilleur partage des connaissances relatives à la conception des jeux. L'étude de ces travaux va nous permettre de poser un cadre théorique et de faire ressortir si, d'une manière générale et purement descriptive, la difficulté joue bien un rôle important dans l'activité ludique.

2.1 Définition générale

Plusieurs auteurs ont proposé des définitions du jeu, aussi bien sous une forme très générale, qu'en s'intéressant plus spécifiquement au jeu vidéo. Ces définitions sont intéressantes et nécessaires : en étudiant le rôle de la difficulté dans la conception d'un jeu vidéo, on pénètre au coeur du jeu pour en examiner les rouages. L'image intuitive que nous avons tous du jeu montre alors ses limites. Nous sommes en effet tous capables d'identifier une activité ludique, mais décrire sa forme générale avec justesse pour en faire un objet théorique manipulable se révèle beaucoup plus difficile. Ces définitions permettent de mettre le doigt sur certaines des propriétés principales du jeu, et leur étude doit nous permettre de déterminer si la difficulté en fait partie.

Johan Huizinga, historien néerlandais, a rédigé un essai sur la fonction sociale du jeu [Huizinga 51]. Il y propose une définition précise et intuitive :

« Sous l'angle de la forme, on peut donc, en bref, définir le jeu comme une action libre, sentie comme "fictive" et située en dehors de la vie courante, capable néanmoins d'absorber totalement le joueur ; une action dénuée de tout intérêt matériel et de toute utilité ; qui s'accomplit en un temps et dans un espace expressément circonscrits, se déroule avec ordre selon des règles données, et suscite dans la vie des relations de groupes s'entourant volontiers de mystère ou accentuant par le déguisement leur étrangeté vis-à-vis du monde habituel. »

Cette première définition pose certaines des caractéristiques essentielles du jeu. Tout d'abord, Huizinga fait ressortir l'aspect libre, gratuit du jeu. Il remarque que le joueur est engagé dans une « *action libre* » : il joue parce que le système de jeu lui apporte quelque chose, et non parce qu'on lui impose de jouer. Cette liberté est de plus renforcée par un autre aspect du jeu : l'action du joueur y est « *dénuée de tout intérêt matériel et de toute utilité* ». Ces caractéristiques placent le créateur de jeu vidéo à une place singulière parmi les développeurs de logiciels. Il doit fournir une application qui n'est pas un outil pour exploiter une machine à des fins extérieures au logiciel, mais un objet qui porte ses propres objectifs, et donc auquel les utilisateurs sont totalement libres d'adhérer. Cette position peut bien sûr être nuancée, en rappelant par exemple l'aspect social du jeu, au coeur de l'ouvrage de Johan Huizinga et à la fin de cette définition. Le jeu suscite des relations de groupes et ces relations, qui perdurent souvent hors de l'univers du jeu¹, peuvent constituer un objectif

1. « *La communauté joueuse accuse une tendance générale à la permanence, même une fois le jeu ter-*

atteint au moyen du jeu. Néanmoins, si le jeu peut être employé comme un outil, une des ses propriétés fondamentales reste de pouvoir obligatoirement être envisagé *per se*.

Huizinga met également en avant le décalage des jeux par rapport au réel, à la vie courante : ces activités s'accomplissent « *en un temps et dans un espace expressément circonscrits, [...] avec un ordre et selon des règles données* » et présentent une « *étrangeté vis-à-vis du monde habituel* ». Toujours dans le même ouvrage, Huizinga écrit cette jolie formule : « *L'extravagance et l'extase constituent les deux pôles de l'ambiance ludique.* » Le jeu est donc le cadre d'un rêve, il nous extirpe du réel et du quotidien et propose un monde parallèle, au sein d'un espace et d'un temps particuliers, avec ses règles, ses objectifs et sa propre échelle de valeurs.

En ce qui nous concerne plus précisément, c'est à dire au regard de la difficulté, cette définition fait ressortir une caractéristique fondamentale du jeu : pour qu'une activité soit considérée comme ludique, elle doit être autonome. Cette activité doit être coupée du monde réel, tant dans son exécution qui suit ses propres règles, que par le fait que son intérêt réside principalement dans son exécution, et non par exemple dans un gain matériel ou tout autre forme de rétribution qui la rattacherait à la vie courante. Ce premier point nous éloigne de la notion de difficulté plus qu'il nous en rapproche. Le dictionnaire Larousse définit *difficile* comme « *Qui n'est pas facile à réaliser, qui exige des efforts importants ; ardu, malaisé* ». En quoi une activité libre et gratuite serait elle pratiquée si elle exige un effort, est ardue, malaisée ? Johan Huizinga précise que le joueur est intensément absorbé, ce qui pourrait traduire un effort, mais d'autres activités libres et gratuites comme regarder un film ou faire une sieste nous absorbent intensément sans pour autant être ardues ou malaisées.

Si cette première définition semble écarter le jeu de la difficulté, elle souligne en même temps un point essentiel. Si le jeu demande bel et bien un effort au joueur, c'est que cette difficulté contribue activement à renforcer son expérience. On joue dans un espace et un temps choisis, et selon des règles choisies : si jouer demande un effort alors cet effort est volontaire car uniquement déterminé par des règles qu'il suffirait de modifier. Cependant, la définition de Johan Huizinga ne précise pas si l'investissement du joueur suppose un effort de sa part, il est donc nécessaire de se pencher sur d'autres définitions pour préciser ce point.

Plusieurs autres auteurs, parmi lesquels Roger Caillois, Bernard Suits, Avedon et Sutton, Chris Crawford, David Kelley, Katie Salen et Eric Zimmerman ont proposé leur propre définition du jeu. Ces définitions, ainsi que celle de Johan Huizinga, ont été rassemblées, analysées et synthétisées par Jesper Juul [Juul 03]. Il propose ainsi la définition suivante :

« *Le jeu est un système formel basé sur des règles, avec un résultat variable et quantifiable, où différents résultats se voient attribuer différentes valeurs, le joueur exerçant un effort pour influencer ce résultat, auquel il est attaché, et dont les conséquences sont négociables et optionnelles.* »

miné. » (p.30)

Cette définition aborde le jeu d'une manière plus formelle. La notion d'autonomie a en partie disparue mais subsiste dans l'idée d'un système formel basé sur des règles, mais par contre, cette définition précise qu'un jeu fournit un résultat quantifiable auquel le joueur est attaché et qu'il fournit un effort pour l'influencer. En plaçant au coeur de sa définition l'attachement du joueur au résultat et l'effort qu'il fournit, Jesper Juul confirme l'importance de la difficulté dans l'expérience offerte par un jeu. En effet, la notion de difficulté d'un jeu est synonyme d'effort fourni par le joueur, et Jesper Juul considère que cet effort est un trait caractéristique de l'ambiance ludique en le faisant figurer dans sa définition.

Ces définitions pointent la difficulté comme un rouage essentiel du jeu vidéo : la définition de Jesper Juul précise que le joueur fournit un effort et se heurte donc à un niveau de difficulté, et la définition de Johan Huizinga insiste sur l'autonomie de l'activité ludique, donc sur le fait que cette difficulté est utile, volontaire, car totalement maîtrisable puisque le jeu est autonome.

La littérature propre au game design, c'est à dire à la conception de jeux vidéo, souligne également la notion de difficulté comme prépondérante. Chris Crawford explique qu'un jeu suppose un but donné au joueur et des obstacles qui l'empêchent d'atteindre ce but [Crawford 84]. Il insiste sur le fait que la notion de conflit, donc d'effort et de difficulté, est un élément nécessaire et indispensable.

Ces définitions posent les traits principaux du jeu, et mettent en évidence l'importance de la difficulté dans cette activité. Bien sûr, leur objectif est uniquement descriptif et ces définitions n'expliquent en aucun cas en quoi la difficulté joue un rôle important vis à vis du jeu, ce sur quoi nous reviendrons par la suite. Elles se contentent de faire ressortir la difficulté comme essentielle à la plupart des expériences de jeu. Elles sont également forcément incomplètes : la définition d'un concept aussi général et partagé que le jeu est complexe, et ses frontières sont suffisamment floues pour qu'aucune définition basée sur une accumulation de propriétés ne puisse englober l'ensemble des activités ludiques. Mais ces définitions, et à plus forte raison la synthèse de Jesper Juul, construite comme une synthèse des définitions précédentes, nous donnent une vision prototypale du jeu suffisante pour débiter notre analyse théorique.

Pour finir, cette première approche permet d'adapter la définition générale de la difficulté au cadre du jeu vidéo et d'en proposer une plus adaptée à notre propos :

« La difficulté d'un jeu est la mesure des efforts successifs que doit fournir un joueur pour atteindre les différents objectifs qui lui sont assignés par le jeu. »

Cette définition de la difficulté implique plusieurs corollaires. Tout d'abord, la difficulté est liée aux buts du joueur et aux obstacles que le gameplay oppose à ces buts. Ensuite, si la difficulté est une mesure de l'effort du joueur, elle dépend donc de chaque joueur. La difficulté est avant tout relative aux savoirs et compétences de chaque joueur.

Cette première approche est très générale et fait ressortir l'importance de la difficulté pour le jeu et nous a permis d'en proposer une définition générale. Nous poursuivons ce

chapitre par l'étude de travaux de designers et de chercheurs consacrés à la manipulation et la mesure de la difficulté dans les jeux vidéo. Cette revue de littérature permet de mieux comprendre l'utilité, tant pour la recherche que pour l'industrie, d'une méthode générale de mesure de la difficulté dans les jeux vidéo.

2.2 Manipulation de la difficulté

Manipuler la difficulté d'un jeu vidéo est une tâche complexe [Rollings 03, Boutros 08, Douville 09]. Andrew Rollings et Ernest Adams abordent la problématique de la difficulté d'un gameplay par la notion d'*équilibre* d'un jeu vidéo, qu'ils définissent de la façon suivante : « *Un jeu équilibré est un jeu où le talent du joueur est le facteur déterminant de sa réussite.* ». Cette définition ne décrit pas le niveau de difficulté d'un jeu, mais explique que cette difficulté doit être cohérente avec les capacités du joueur : plus il joue, plus il gagne de l'expérience et donc plus la difficulté globale du jeu doit diminuer. Si ce n'est pas le cas, le joueur se bat contre un système indépendant de ses capacités, comme un jeu de hasard pur par exemple, et ce jeu est considéré comme déséquilibré. Face à un jeu de loto, peu importe l'expérience du joueur, ses chances de réussite sont toujours les mêmes.

Cette définition implique également une idée plus précise que la simple corrélation entre l'effort du joueur et sa réussite, elle implique aussi une quantification de cette réussite, c'est à dire un niveau de difficulté donné. Le premier niveau d'équilibre consiste en un choix structurel général : les actions du joueur doivent avoir un impact sur la dynamique du jeu, et cet impact doit être plus ou moins reproductible, pour que le joueur puisse s'exercer et gagner de l'expérience. Mais l'équilibre d'un jeu demande surtout un nombre considérable d'ajustements minutieux. Un jeu est conçu pour un joueur type, avec un talent donné, et sa difficulté doit être précisément calibrée. L'équilibre d'un First Person Shooter sur Wii, par exemple, sera très différent de celui d'un First Person Shooter sur PC, même si les principes généraux de gameplay peuvent paraître similaires. Le joueur type de Wii est un joueur occasionnel, dont l'investissement et donc la marge de progression sont plus limités, alors qu'un joueur type de First Person Shooter sur PC est plus impliqué et recherche une difficulté plus élevée. L'équilibre va donc au delà de la stricte définition de Rollings et Adams : le talent du joueur doit être le facteur déterminant de sa réussite, mais surtout l'effort à fournir pour réussir doit être borné : suffisant pour donner sa valeur à la réussite, mais d'un niveau qui permette au joueur d'échapper le plus souvent à l'échec et au découragement.

Comme nous l'avons décrit précédemment, la difficulté d'un jeu vidéo n'est pas une donnée stable, mais suit un processus en constante d'évolution. En effet, le niveau du joueur varie du fait de son apprentissage et donc le jeu doit s'adapter en permanence pour fournir une valeur de difficulté optimale. Ce problème particulier de l'adaptation de la difficulté est principalement pris en compte par deux méthodes de game design, scénarisation et adaptation

dynamique, qui suscitent chacune de nombreuses recherches, et bénéficieraient particulièrement d'une méthode de mesure générale de la difficulté dans les jeux vidéo. Dans les sections suivantes, nous présentons tout d'abord plus précisément la courbe de difficulté des jeux vidéos, qui décrit le processus d'évolution de la difficulté. Par la suite, nous présentons une revue de littérature des méthodes qui permettent de maîtriser cette courbe.

2.2.1 Courbe de difficulté

Depuis le début de ce document, la difficulté est définie en fonction d'un joueur moyen, le joueur cible du vidéo. A chaque fois que nous avons abordé la difficulté d'un jeu, c'était en référence implicite à ce joueur moyen, car la difficulté d'un jeu n'est pas une propriété du jeu, mais la valeur d'une relation entre un jeu et un joueur. Or en jouant, le joueur progresse. Il explore l'univers du jeu, développe des heuristiques pour prévoir les conséquences de ces actions et augmente ses capacités de coordination oculo-manuelle ainsi que sa rapidité à exécuter certain types d'actions. La difficulté d'un jeu est donc par essence variable, et tend à diminuer au cours du temps.

Il est donc nécessaire de définir plus précisément la difficulté afin de prendre en compte plus clairement son aspect relationnel, en introduisant les notions de difficulté *absolue* et *relative* :

- **La difficulté absolue d'un jeu** décrit l'effort que doit fournir un joueur type, aux capacités statiques, pour atteindre les objectifs que son gameplay propose.
- **La difficulté relative d'un jeu** décrit l'effort que doit fournir le joueur, dont les capacités évoluent tout au long du jeu, pour atteindre les objectifs que son gameplay propose.

Pour maintenir un certain niveau de difficulté relative, il est donc nécessaire d'augmenter la difficulté absolue d'un jeu, en fonction de l'évolution des capacités du joueur. Ed Byrne, Level Designer, explique ainsi qu'un jeu suit une courbe de difficulté particulière [Byrne 04]. Dans sa plus simple expression, cette courbe est en plateaux si on représente la difficulté absolue du jeu ou en dents de scie si on trace la difficulté à laquelle un joueur actif est confronté (Figure 2.1). Chaque accroissement de la difficulté correspond à une modification du gameplay : la difficulté augmente si l'on resserre les contraintes d'exécution d'une action, si on ajoute un nouvel objet qui augmente la complexité du système ou si on découvre une nouvelle partie de l'univers à explorer. La difficulté tend ensuite à diminuer pour le joueur actif, qui travaille son habileté, s'attache à comprendre les implications de l'apparition d'un nouvel objet ou explore l'univers du jeu.

On peut isoler deux techniques principales de construction d'une courbe de difficulté, dont l'utilisation dépend principalement du type de jeu envisagé. On peut en effet distinguer deux grandes familles de jeu vidéo, les jeux de *progression* ou d'*émergence* [Juul 03] [Rollings 03].

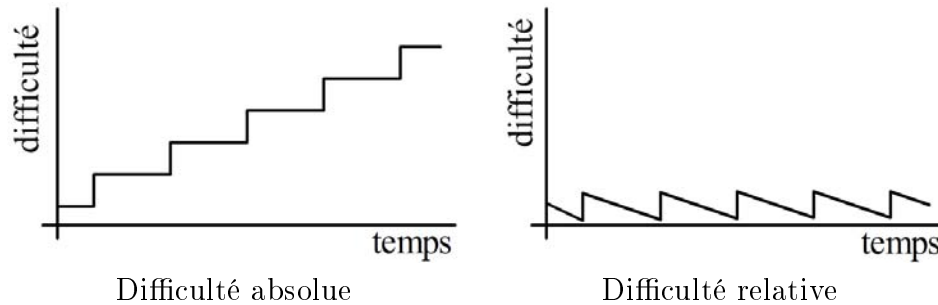


FIGURE 2.1 – Courbes de difficulté

La distinction repose sur la technique de design employée. Dans un jeu de progression, les designers ont fixé le plus précisément possible le parcours du joueur. L'ensemble des situations auxquelles il va se confronter ont été réfléchies, calibrées, et explicitement encodées dans le contenu du jeu. C'est ce qu'on appelle *scénariser* un gameplay. De plus, les jeux qui développent un contenu narratif sont typiquement des jeux de progression, le joueur découvre une histoire écrite par un scénariste ce qui suppose un déroulement précis des événements. La plupart des jeux de plateformes (Figure 2.2) sont des jeux de progression : le parcours du joueur y est contraint est finement organisé.



FIGURE 2.2 – Super Mario World (Nintendo)

A l'inverse, dans un jeu émergent, le designer ne peut pas prévoir le contenu de chaque partie. Un match de foot ou une partie d'échecs, par exemple, ne sont pas scénarisés. Quelques règles définissent le comportement des objets du jeu et chaque partie émerge de l'application de ces règles et de l'interaction des objets du jeu.

Plus précisément, progression et émergence ne sont pas deux catégories distinctes mais les extrémités d'un même axe, qui décrit le niveau de contrôle du designer sur l'expérience du joueur. En ce qui concerne la manipulation de la difficulté, plus le game designer a prévu le parcours du joueur et plus il lui est possible de définir précisément sa courbe de difficulté,

comme présenté dans la section suivante. Dans un jeu qualifié d'émergent, le designer a moins de contrôle sur le déroulement des événements et des algorithmes génériques seront plus appropriés, comme décrit dans la section 2.2.3.

2.2.2 Scénarisation

En tant qu'outil de manipulation de la difficulté, la scénarisation est vue ici comme une modification arbitraire et planifiée de la difficulté d'un jeu vidéo. La scénarisation permet de moduler finement la difficulté du jeu pour prendre en compte les capacités du joueur.

La plupart des **RTS**² proposent par exemple un mode scénarisé. Le joueur débute avec un faible nombre d'unités et doit atteindre des objectifs à court terme face à une intelligence artificielle bridée. Au fur et à mesure, le joueur progresse dans le scénario et on lui permet d'utiliser plus d'unités, pour atteindre des objectifs à plus long terme et face à un adversaire toujours plus efficace. La scénarisation permet de découper l'apprentissage du joueur, en s'assurant de son niveau à chaque étape. De cette manière la difficulté du jeu reste équilibrée tout au long de son parcours, et il est possible d'amener le joueur à exploiter tout le potentiel de difficulté du jeu. Pour le jeu de go par exemple, de nombreux sites internet proposent un apprentissage scénarisé, sous la forme de problème à résoudre. Ces problèmes posent un objectif à court terme clairement défini (e.g. sauver tel groupe de pierres) et permettent ainsi au joueur d'aborder des situations de jeu qu'il serait plus difficile de travailler lors d'une partie complète.



FIGURE 2.3 – The Legend of Zelda : A Link to the Past (Nintendo)

La scénarisation permet de manipuler le gameplay au cours du temps. Par exemple, dans un jeu d'aventure comme Zelda (Figure 2.3), les joueur découvre petit à petit des objets qui lui permettent d'accéder à de nouvelles sections de la carte. L'univers du jeu s'étend donc régulièrement, à peine le joueur a t'il exploré une partie de l'univers qu'une autre se déverrouille et demande à être explorée. De même, les ennemis que rencontre le joueur doivent

être vaincus au moyen de stratégies, déterminées arbitrairement par le game designer, que le joueur doit découvrir. A chaque nouvelle zone correspond de nouveaux types d'ennemis que le joueur doit apprendre à vaincre.

Dans *Zelda*, la progression de la difficulté du gameplay est rythmée par le déblocage de nouvelles zones et objets de l'univers accessible au joueur. Chaque nouvelle partie du jeu apporte de nouvelles énigmes dont le joueur doit découvrir la solution. Mais contrairement à un jeu comme *Indiana Jones* (Figure 3.1) où chaque découverte du joueur peut être en général immédiatement oubliée, la découverte du maniement des objets dans *Zelda* permet au joueur de se construire une base de règles. Plus il progresse dans le jeu, et plus le nombre d'objets à combiner pour résoudre les énigmes sera important et donc plus la difficulté absolue de chaque énigme augmente. Le joueur développe ainsi un réel sentiment de progression, en constatant que son investissement passé lui permet de résoudre des énigmes qu'il n'aurait jamais pu résoudre au début du jeu.

La scénarisation est donc une manière artificielle d'encadrer l'apprentissage du joueur, et suppose un découpage préalable du gameplay en phases successives à la difficulté choisie. Ce type d'équilibrage demande un travail de fond conséquent. Il a l'avantage de sa franchise : le joueur sait qu'il suit un scénario et qu'on lui distille, étapes après étapes, une expérience de jeu calibrée et planifiée sur le long terme. En ce qui nous concerne, cette méthode à l'avantage de reposer sur les choix du game designer, choix pouvant être encodés dans un modèle. Un modèle de scénario pourrait permettre au game designer de représenter facilement et précisément le scénario de son jeu ainsi que de fournir un maximum d'annotations concernant la planification de la difficulté, nous permettant ainsi d'en calculer plus facilement la courbe.

La scénarisation n'a cependant pas que des aspects positifs. Cette méthode impose des contraintes et donc limite la liberté du joueur par un encadrement dont le côté artificiel n'est pas toujours apprécié. La scénarisation limite également la variabilité des parties proposées : une fois le scénario totalement exploré, le joueur a eu accès à l'ensemble du contenu proposé et le jeu n'a plus d'intérêt, ce qui par exemple n'arrivera jamais à un joueur de *go*. Un sous-domaine particulier de la recherche en intelligence artificielle s'est ainsi fixé pour objectif de réconcilier contenu scénarisé et émergent, cherchant ainsi à créer un univers proposant les avantages des deux et les inconvénients d'aucun : la narration interactive.

2.2.2.1 Scénarisation et recherche

Dans de nombreux jeux d'aventures, le joueur est soumis à deux logiques, celle du gameplay, et celle du scénario. Toutes ces règles contraignent le comportement du joueur dans l'univers du jeu : la logique du gameplay l'empêche de se déplacer en volant ou d'ouvrir une porte dont il ne possède pas la clef, la logique scénaristique peut choisir de faire mourir un personnage dont le joueur aurait préféré ne pas se séparer, ou lui interdire de poser cer-

taines questions à un personnage du jeu, car il n'a pas été prévu qu'il y réponde. Ces règles enrichissent le jeu : les règles de gameplay fournissent des buts au joueur, et des obstacles qui déterminent la valeur de ces buts. Le scénario établit l'évolution du jeu dans le temps, de manière à lui faire suivre la structure la plus riche et la plus plaisante. Mais ces règles peuvent également frustrer le joueur, en particulier lorsqu'il existe une contradiction entre les attentes du joueur et le déroulement du jeu. Bien souvent, ces incohérences surgissent du fait qu'une règle scénaristique vient rompre la cohérence du gameplay. En effet, dans la plupart des jeux d'aventure, le joueur emploie la plus grande partie de son temps à jouer, et donc est immergé dans la logique du gameplay. Les règles scénaristiques interviennent quant à elles ponctuellement, pour orienter le joueur dans la direction voulue par le scénario. Bien souvent, le joueur ne s'en aperçoit même pas, il suit un chemin sans s'apercevoir à quel point il est balisé. Mais parfois, les incohérences apparaissent de manière évidente.

Dans le jeu *Fallout 3* (Bethesda) (Figure 2.4), par exemple, le joueur peut interagir avec un grand nombre d'objets, et leur comportement répond à la logique générale d'un monde post-apocalyptique. De nombreuses règles sont alors faciles à admettre, comme le fait que les armes aient une certaine portée, qu'on puisse blesser toute créature vivante et qu'un chien irradié soit dangereux. Ces règles respectent la logique générale d'un jeu d'aventure post-apocalyptique, et le joueur les accepte une bonne fois pour toutes. Par contre, se voir interdire la mise en marche d'une machine que le joueur répare depuis plusieurs jours et dont il sait qu'elle fonctionne parfaitement, au moyen d'un message textuel signifiant « *Qu'il n'est pas encore temps* », est plus illogique et difficile à admettre. Le joueur réalise à cet instant que l'univers virtuel dans lequel il s'est immergé et dont il a accepté sa logique peut brusquement perdre sa cohérence et évoluer de manière arbitraire et imprévisible.



FIGURE 2.4 – *Fallout 3* (Bethesda)

Ces limitations parfois frustrantes des univers virtuels scénarisés sont donc à l'origine de nombreuses recherches en narration interactive, à la recherche d'univers où l'on peut « *aller n'importe où et faire n'importe quoi* » [Bates 92]. Plus précisément, l'objectif de ces chercheurs consiste à être capable de relâcher en partie la tension entre liberté du joueur et contraintes scénaristiques en adaptant si possible le scénario au comportement du joueur. Un composant logiciel particulier, le Drama Manager, est alors chargé de permettre au joueur

d'évoluer dans un scénario respectant les volontés d'un auteur, tout en contraignant le moins possible le parcours du joueur.

La gestion d'histoires par recherche³ consiste par exemple à définir les propriétés d'un bon scénario, et à utiliser ces propriétés pour rechercher le meilleur scénario intégrant le mieux possible le comportement du joueur [Bates 92], [Mateas 03], [Nelson 05], [Nelson 06], [Nelson 08], [Roberts 06], [Roberts 07], [Barber 07], [Mott 06]. Cela revient à créer un générateur de scénario, capable d'inférer la valeur d'une histoire à partir de certaines de ses propriétés (e.g. unité de lieu, causalité, conflit...). Le joueur dispose ainsi d'un espace de liberté bien plus important, et on lui refusera une action beaucoup plus rarement.

Une autre approche consiste à partir d'un scénario défini et à considérer le maintien de la cohérence de ce scénario face aux actions du joueur. L'architecture Mimesis [Young 01] ou d'IN-TALE [Riedl 06a], associée au principe de médiation de la narration⁴ [Riedl 03] représente l'histoire sous forme d'un plan annoté causalement. Si une action du joueur met en danger ces relations causales, plusieurs comportements peuvent être envisagés : *accommodation* qui consiste à modifier le plan pour en rétablir la cohérence, ou *l'intervention*, c'est à dire un échappatoire quelconque permettant de ne pas exécuter l'action du joueur. Certaines étapes du plan peuvent être fondamentales et considérées comme des buts auteurs, que le mécanisme de planification sera forcé d'intégrer dans sa solution [Riedl 09]. D'autres approches de la narration interactive, tel l'Interactive Drama Architecture (IDA) de Brian Magerko et John E. Laird [Magerko 04], spécifient également un scénario à respecter et un système d'action à utiliser pour guider le joueur vers cet espace narratif. Le but n'est plus alors de conserver une histoire cohérente mais d'agir sur l'univers pour ramener le joueur vers le scénario prédéfini.

L'ensemble des systèmes de narration interactive partagent l'objectif de manipuler les règles du jeu à l'exécution, de manière à prendre en compte l'évolution du joueur. Ces modifications peuvent s'appuyer sur n'importe quelle heuristique : conflit, causalité, unité de lieu aussi bien que difficulté du gameplay. Une méthode de mesure de la difficulté serait d'un intérêt particulier pour toutes ces méthodes, leur permettant de planifier un chemin respectant plus finement la courbe de difficulté planifiée par le game designer.

Les systèmes de narration interactive restent cependant particulièrement complexes, et sont principalement des sujets de recherche uniquement implantés dans des prototypes. Cependant, de nombreux jeux vidéo proposent des gameplays en partie émergents. Ils ne peuvent ni s'appuyer uniquement sur une scénarisation fixe de la difficulté, ni mettre en place un système de narration interactive de la complexité de ceux présentés précédemment. Ces jeux s'appuient sur des algorithmes plus simples, décrits dans la prochaine section, que nous regroupons sous le terme d'équilibrage dynamique de la difficulté.

3. SBDM, Search Bases Drama Management

4. Narrative Mediation

2.2.3 Equilibrage dynamique

L'équilibrage dynamique permet de maintenir le niveau de difficulté d'un jeu cohérent avec les capacités du joueur, d'une manière réactive, générique. Le jeu adapte sa difficulté en fonction du comportement du joueur et suit ainsi une voie différente de la scénarisation, qui impose au joueur son comportement à partir d'un cadre de difficulté prédéfini. Son emploi est par contre particulièrement délicat, car comme tout système intervenant de manière automatique sur le gameplay, il peut avoir des conséquences dramatiques s'il n'est pas utilisé de manière suffisamment subtile [Adams 08].

Les méthodes d'auto adaptation peuvent être cependant très efficaces, par exemple pour manipuler les contraintes de coordination oculo manuelle. On retrouve souvent ce type d'algorithme dans des jeux d'actions : il est facile de relâcher ou de restreindre les contraintes de réalisation d'une action en fonction des résultats du joueur sans que celui ci s'en rende compte. Il est par contre plus difficile de jouer sur d'autres aspects du gameplay : la lisibilité des informations fournies au joueur peut être manipulée en brouillant plus ou moins la vision du joueur, mais modifier la composition et l'accès du joueur à l'univers du jeu en temps réel et de manière cohérente est beaucoup plus délicat, et la logique de l'univers dépend de des relations entre les différents objets du jeu, plus compliquées à manipuler de manière instantanée.

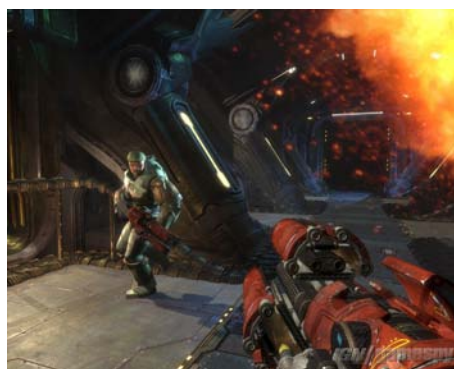


FIGURE 2.5 – Unreal Tournament 3 (Epic Games / Infogrames)

Dans un First Person Shooter comme *Unreal Tournament 3*, par exemple, l'objectif du joueur consiste à tuer le plus possible d'ennemis en mourant le moins de fois possible (Figure 2.5). Lorsque le joueur tue un ennemi, celui-ci est immédiatement ressuscité à un autre endroit du niveau. Mais avant d'être à nouveau sur le terrain, cet ennemi subit une petite modification : ses capacités sont augmentées (reflexes, précision, capacités sensorielles) et le niveau de difficulté augmente. De même, lorsque le joueur meurt, l'ennemi qui l'a tué subit une modification équivalente mais dans le sens inverse, de manière à diminuer la difficulté.

De cette manière la difficulté du jeu suit le niveau du joueur et lui évite une trop grande frustration face à des échec successifs, ou un ennui profond face à des ennemis trop poussifs. Bien sûr, avant chaque partie, le joueur peut choisir le niveau de difficulté et calibrer globalement son expérience. Mais le système se débrouille ensuite pour s'ajuster finement aux capacités du joueur, sans s'éloigner de manière trop flagrante du niveau de difficulté initial. Ces modifications se font totalement à l'insu du joueur et la seule manière d'en avoir conscience consiste à lire les scripts de l'intelligence artificielle du jeu. La liberté du joueur est ainsi préservée, tout en trichant subrepticement pour adapter précisément la difficulté.

Il aurait été également possible de manipuler la difficulté d'une manière moins subtile mais plus efficace, en modifiant par exemple directement les points de vie du joueur lorsque ceux-ci dépassent un certain seuil, comme expérimenté par Robin Hunicke dans le First Person Shooter *Case Closed* [Hunicke 05]. Cette technique d'ajustement limite simplement la probabilité qu'a le joueur de perdre s'il n'effectue pas les bonnes actions. La difficulté repose sur l'effort que doit fournir le joueur pour atteindre un objectif. Le système d'adaptation peut donc aussi bien modifier les moyens du joueur que simplement diminuer le niveau d'exigence de l'objectif.



FIGURE 2.6 – Pure (Black Rock Studio / Disney Interactive)

Il est également possible d'apporter une dose de scénarisation à un algorithme d'équilibrage dynamique, ce que propose par exemple le jeu de course *Pure* [Jimenez 09] avec un « *script de course* ».

Ce jeu étend ainsi le principe de *l'élastique*⁵ en le scénarisant. La technique de l'élastique consiste à imaginer que tous les adversaires sont attachés au joueur par un élastique. Plus ils sont loin du joueur et plus une force les oblige à s'en rapprocher. Les ennemis à la traîne sont donc plus rapides et rattrapent le joueur, et ceux en avance sont ralentis. Cette approche est une illustration parfaite de l'équilibrage dynamique et montre immédiatement ses deux faiblesses : si le joueur bride volontairement ses capacités, il verra que le jeu s'adapte et

5. Rubber band

l'illusion aura du mal à être maintenue. Si le système est parfait, le joueur va passer toute la course entouré de ces adversaires, quelque soit son comportement. L'équilibrage dynamique peut donc en quelque sorte compromettre l'équilibrage tel que définit par Andrew Rollings et Ernest Adams, c'est à dire empêcher que le joueur soit le facteur déterminant de sa réussite.

En scénarisant l'équilibrage dynamique, *Pure* choisit par exemple de désactiver l'algorithme d'équilibrage pendant les derniers tours de la course, ou crée des groupes d'adversaires en leur attribuant des capacités différentes et variables tout au long de la course, de manière à s'adapter au joueur dynamiquement, tout en respectant en partie un schéma préétabli.

2.2.3.1 Equilibrage dynamique et recherche

L'équilibrage dynamique a suscité de nombreuses recherches, en particulier parce qu'il constitue un champ d'application intéressant des diverses techniques d'intelligence artificielle. Il s'agit en effet de développer un composant capable d'optimiser certains paramètres du gameplay en temps réel, pour s'adapter au comportement du joueur.

L'apprentissage automatique est une voie particulièrement explorée dans le cadre de l'adaptation dynamique du gameplay. Ces techniques présentent en effet l'avantage de calculer de manière automatique des paramètres de gameplay avec l'objectif d'optimiser une mesure, qui peut être par exemple la difficulté du jeu. Si le joueur développe une stratégie trop efficace, en particulier s'il découvre une faille dans un gameplay émergent, alors une IA adaptative doit pouvoir le déceler et s'adapter en conséquence. Le revers de la médaille concerne bien sûr l'imprévisibilité d'un processus d'apprentissage automatique. L'algorithme peut échouer dans sa recherche d'une solution, ou développer un comportement optimal mais non désiré, parce que par exemple non crédible dans le contexte de l'univers, et ruiner l'expérience globale du joueur malgré un gameplay équilibré. La mesure à optimiser ne capture en effet qu'une toute petite parcelle de l'objectif global du game designer. C'est pourquoi l'industrie du jeu vidéo limite aujourd'hui l'apprentissage automatique à des sous problèmes d'optimisation très particuliers [Manslow 04].

La recherche dans ce domaine fournit cependant un certain nombre de tentatives fructueuses, et plusieurs familles d'algorithmes d'apprentissage ont été étudiées dans le cadre des jeux vidéo. L'apprentissage par renforcement [Sutton 98] a permis à divers équipes de générer des IA pour des jeux de combat [Andrade 05], [Graepel 04], de stratégie temps réel [Madeira 04], [Madeira 06], [Ulam 05], ou pour des First Person Shooter [Lee-Urban 08]. Une version allégée de l'apprentissage par renforcement a été développée par Pieter Spronck, le scripting dynamique, qui calcule des préférences au niveau de règles écrites par le designer. Le scripting dynamique est appliqué de manière successive à un jeu d'aventure (*Neverwinter Nights* - Bioware) et de stratégie temps réel (*Wargus*) [Spronck 05], [Spronck 06], [Spronck 08] [Timuri 07], [Ponsen 06], [Ludwig 07].

L'évolution génétique et les réseaux de neurones, utilisés séparément ou conjointement,

permettent également de créer automatiquement des IA pour divers types de gameplay, jeu d'action [Demasi 03],[Spronck 02], jeux de stratégie temps réel [Ponsen 05], [Agogino 00], des FPS [Cole 04], [Thureau 03], des jeux de sport (FIFA 99 - EA Games) [Chan 04], de puzzle (Tetris - Nitendo) [Bohm 05] ou de réalité virtuelle [Yannakakis 09], [Yannakakis 07a]. D'autres recherches portent par exemple sur des algorithmes de champs de potentiels pour les comportements stratégiques dans un FPS [Thureau 04] ou de raisonnement par cas pour RTS [Aha 05].

Toute recherche sur la génération automatique d'une IA permet par principe de modifier le niveau de difficulté en créant dynamiquement des adversaires d'un niveau donné ou en modifiant des paramètres de gameplay à la volée. Certains chercheurs ce sont directement focalisés sur cette utilisation des algorithmes d'apprentissage pour l'équilibrage dynamique du gameplay. Gustavo Andrade et al adressent par exemple directement l'adaptation de la difficulté avec l'apprentissage par renforcement [Andrade 05]. L'algorithme consiste à faire apprendre à l'IA une table de l'efficacité espérée de couples (*action, etat du jeu*), et donc lui permet d'adopter un comportement d'une efficacité choisie. Pieter Spronck a étudié diverses manières de modifier le calcul des préférences de sélections des différentes règles pour générer des comportement plus ou moins optimaux [Spronck 05]. Jeremy Ludwig a utilisé une version hiérarchique du scripting dynamique pour équilibrer le score dans un jeu de type proie vs prédateur (sous problème d'un RTS) [Ludwig 07]. Georgios Yannakakis évolue un réseau de neurones pour prédire les préférences du joueur [Yannakakis 07a] et l'utilise ensuite en temps réel pour équilibrer le gameplay [Yannakakis 09]. Dans l'ensemble de ces recherches, la mesure de difficulté repose sur une combinaison linéaire de paramètres de gameplay.

Ces différentes recherches montrent que sur des expériences réalisées en laboratoire et sur une gamme de jeux plutôt large, il est possible de manipuler des paramètres du gameplay dans le but d'en optimiser une propriété particulière. Ces techniques sont cependant limitées par leur instabilité et par l'étendue de leur champ de vision : elles assujettissent une grande partie de la dynamique du jeu à la mesure d'une des facettes de l'expérience du joueur et posent la question de la qualité d'un contenu modifié à la volée. Sera t'il possible d'encoder de manière suffisamment précise la volonté de l'auteur de manière à permettre à un algorithme de produire un contenu la respectant ? L'aspect itératif du développement de jeu vidéo tendrait pour l'instant à orienter la réponse vers la négative. Sans parler d'encoder un but esthétique, si un game designer pouvait spécifier sa vision d'un jeu à son équipe, au travers d'un document de game design, de la richesse du langage naturel et de leurs références communes au monde du jeu vidéo, le développement pourrait plus facilement s'approcher d'un cycle en V que d'une longue suite d'essai-erreurs. Le développement d'un jeu semble nécessairement lié à l'évaluation constante de son auteur et à une démarche avant tout expérimentale.

Notre objectif dans cette thèse n'est pas de mettre au point de tels mécanismes génératifs, mais d'utiliser au mieux les acquis de ces différentes recherches pour fournir aux auteurs une

vision précise de la difficulté de leur gameplay et d'en faciliter ainsi le processus créatif. Les modèles narratifs utilisés en narration interactive peuvent nous permettre d'encoder dans un modèle manipulable le scénario défini par le game designer. Un tel modèle serait utile au game designer comme outil de test, de validation et de représentation de ses courbes de difficultés.

Les différentes recherches concernant les IA adaptatives montrent qu'il est possible et utile d'explorer un gameplay à l'aide d'algorithmes d'apprentissage. Utilisés lors des phases de design, ces outils peuvent apporter des informations utiles au game designer sur la dynamique du gameplay qu'il conçoit.

La modification du gameplay n'est cependant qu'une des deux facettes de l'équilibrage d'un jeu vidéo. Une part importante de ce processus consiste à évaluer un gameplay existant pour déterminer s'il est correctement équilibré. Une méthode de mesure de la difficulté peut évidemment s'inspirer des techniques de mesure déjà utilisées par l'industrie du jeu ou proposées par la recherche. La section suivante couvre deux techniques d'évaluation du gameplay : les tests de jouabilité et l'analyse systématique du gameplay.

2.3 Evaluation de la difficulté

2.3.1 Les tests de jouabilité

Une première méthode d'évaluation de la difficulté consiste bien évidemment à recruter des joueurs pour tester le jeu en cours de développement, et à étudier ensuite soigneusement leur comportement et leur réflexions sur une portion choisie du gameplay [Davis 05], [Gomez-Martin 06].

Les inconvénients de cette méthode sont principalement son coût, sa complexité de mise en place et la subjectivité des résultats obtenus. Le jeu est en effet très souvent mis à jour au fur et à mesure que son équilibrage progresse. La portée d'un test est donc limitée dans le temps. Il s'agit de tester une version du jeu dont la durée de vie n'est que d'une itération. Il n'est pas possible de répéter ces tests à chaque itération, et l'équipe de développement reste donc avant tout la principale équipe de test.

Ces tests sont complexes à mettre en place car comme toute expérience, ils demandent une préparation minutieuse, le suivi d'un protocole, le recueil des données et l'interprétation des résultats. Il faut recruter un type de joueur cohérent avec la cible marketing du jeu, éviter les biais relatifs à toute expérience de ce type, et travailler sur un stock de donnée suffisamment conséquent pour que les résultats soient statistiquement valides.

Une fois les tests effectués, la subjectivité de leur résultat peut être un obstacle majeur à leur impact sur le développement du jeu. Les tests de jouabilité sont complexes à mettre en oeuvre et donc réalisés sur une version du jeu suffisamment aboutie pour que leur résultat soit le plus significatif possible et les choix de design les plus avancés possibles. Mais c'est

aussi à ce moment que les avis ont le plus convergé, et qu'il est le plus difficile de convaincre une équipe de développement qu'un groupe de jeunes testeurs a mis à jour une erreur de design, en particulier si la mise en évidence de cette erreur repose sur des appréciations subjectives. Néanmoins, cette étape est primordiale pour permettre à une équipe de design de poser un oeil neuf sur leur jeu, et éviter ainsi les erreurs les plus évidentes d'équilibrage. De plus, elle seule permet d'observer le comportement de vrais joueurs, toujours à même de surprendre les designers.



FIGURE 2.7 – Battlefield Heroes (Electronic Arts / Dice)

Plus particulièrement, les jeux en ligne disposent d'un gros avantage vis à vis de ce type de tests : les joueurs sont en permanence connectés au serveur, et leur comportement peut être analysé de manière transparente. La plupart de ces jeux sont d'ailleurs tout d'abord diffusés sous la forme de bêta fermées, dont l'accès est réservé au joueur avertis. Durant l'évaluation de la version bêta du First Person Shooter gratuit en ligne *Battlefield Heroes* (Figure 2.7), seuls les joueurs disposant d'une clé diffusé en nombre limité pouvaient avoir accès au jeu. Ces joueurs sont donc particulièrement motivés, et constitueront probablement le coeur de la future communauté de joueurs. A l'issue des parties, chaque joueur pouvait répondre à un questionnaire d'ordre général sur sa connaissance du jeu et, par exemple, son approche du micro paiement. Il lui était également possible de s'exprimer plus en longueur sur le forum en ligne du jeu. Le jeu disposait de deux cartes lors de l'ouverture de sa version bêta fermée, puis de quatre cartes lors de son ouverture officielle au grand public. La moitié des cartes a donc été développée en prenant régulièrement l'avis de nombreux joueurs, ce qui a permis de valider une grande partie du gameplay de *Battlefield Heroes* avant sa sortie.

Les tests de jouabilité sont donc complexes à réaliser, bien que plus faciles à mettre en place lors du développement d'un jeu en ligne. L'avis et la mesure du comportement des joueurs reste cependant une des méthodes les plus puissantes d'évaluation d'un gameplay et l'augmentation croissante des jeux en réseau, avec des consoles de salon disposant aujourd'hui elles aussi d'un accès internet, rend les techniques d'analyse de comportement des joueurs

et des paramètres de gameplay bien plus abordable, et motive d'autant la création d'outils capables d'effectuer ce type de mesures.

Néanmoins, il est également possible d'étudier un gameplay sans avoir recours à un joueur humain, ce que nous avons appelé le test par joueur synthétique, présenté dans la section suivante.

2.3.2 Test par joueur synthétique

Analyser un gameplay en faisant appel à des testeurs humains a l'avantage de reproduire le plus fidèlement possible les conditions de jeu et d'offrir aux développeurs un avis extérieur réaliste et subjectif sur le gameplay qu'ils conçoivent. Cependant, une manière directement rentable d'obtenir un avis extérieur consiste à remplacer le joueur par un algorithme doué d'un minimum d'autonomie, et d'observer son comportement. Un algorithme n'aura pas d'avis subjectif sur le plaisir de jouer et laissera de côté de nombreux aspects perceptifs. Mais si son champ d'application reste limité, le test automatique de gameplay grâce à une intelligence artificielle permet de rechercher certaines incohérences à moindre frais et aussi souvent que l'équipe le désire.

Ce type de tests intéresse tout particulièrement la recherche et plusieurs études décrivent des résultats encourageants. Tout d'abord, la réalisation d'un joueur synthétique est un problème d'intelligence artificielle intéressant : le jeu vidéo est un monde simulé où il est facile de fixer le nombre de variables observables et la complexité de la tâche. Certains jeux vidéo, comme ceux de stratégie temps réel, ont un espace d'actions et d'états particulièrement important et le développement d'une IA dans ce contexte est particulièrement intéressant. De nombreuses études concernent par exemple le développement d'une IA capable de jouer à Pacman (Namco) grâce à des techniques d'apprentissage [Bonet 01], [Lucas 05], [Gallagher 03], [Gallagher 07], [Szita 07], à Tetris (Nintendo) [Szita 06] ou au Tic Tac Toe [Soedarmadji 06].

Certaines recherches d'intéressent tout particulièrement à l'exploitation de ces techniques pour l'évaluation d'un gameplay. Neil Kirby, des laboratoires Bell, a étudié le gameplay du démineur et du sudoku [Kirby 08]. L'auteur a remplacé le joueur par une IA dont les règles de résolution sont classées par ingéniosité croissante. Une analyse du comportement de l'IA lors de la résolution de grilles a permis de montrer que la proportion d'utilisation des règles très basiques était plus importante qu'on aurait pu le prévoir de manière intuitive. Southey et al ont étudié le gameplay de FIFA 99, et utilisé une IA pour évaluer la probabilité de marquer un but dans un certain nombre de scénarios pré définis [Southey 05]. Nantes et al ont mis au point une IA capable de tester des problèmes de rendu graphiques, ici d'aliasing dans les ombres, de manière automatique [Nantes 08]. L'avantage de développer un joueur synthétique consiste à pouvoir étudier son approche directe du gameplay, un fois définie une interface entre le moteur de jeux et ce joueur synthétique [Aha 04].

Certaines techniques d'analyse systématique d'un jeu vidéo peuvent également s'appuyer

sur des représentations formelles du gameplay. De nombreuses études ont été réalisées sur la formalisation du scénario d'un jeu vidéo, présentées dans la section 2.2.3. L'intérêt d'un tel modèle réside en partie dans l'analyse automatique de propriétés qu'on peut lui appliquer. Champagnat et al proposent ainsi d'étudier directement l'impartialité (équilibre entre les chances de réussites des différents joueurs), la complexité (vitesse minimale de résolution) et le niveau de concurrence (confrontations entre le joueurs) d'un scénario, définit grâce à une représentation sous forme de logique linéaire, traduite en réseau de pétri et soumise ainsi à des tests de propriétés. Nelson et Mateas proposent de nombreuses caractéristiques d'évaluation d'un graphe de scénario [Nelson 05].

Les recherches en analyse systématique de gameplay montrent donc qu'il est possible d'étudier la difficulté d'un gameplay au moyen d'une analyse automatique. Aucune des ces recherches n'est allé jusqu'à construire la courbe de difficulté d'un gameplay à partir de telles méthodes. Il serait également intéressant d'appliquer ces méthodes à un jeu commercial connu et tenter de vérifier si des problèmes d'équilibrages peuvent être facilement mis en évidence, ce à quoi nous nous attacherons dans une section consacrée à des expériences préliminaires.

2.4 Synthèse

Nous avons tout d'abord étudié les définitions générales du jeu. Nous retenons du travail de ces théoriciens que le jeu est une activité libre qui demande un effort au joueur. Cet effort librement consenti doit donc être particulièrement utile, puisqu'il serait facile à supprimer en modifiant les règles. Cette première étape montre que la difficulté, définie comme l'effort du joueur pour atteindre un objectif donné, est un composant essentiel du jeu et mérite une étude approfondie.

Nous nous sommes ensuite focalisés sur l'évolution de la difficulté dans les jeux vidéo. En effet, les capacités du joueur sont en perpétuelle évolution, et le gameplay doit donc évoluer pour maintenir le niveau de difficulté choisi, ce qui nous a amené à préciser les concepts de courbe de difficulté *absolue* et *relative*. Nous avons ensuite étudié de quelle manière un game designer pouvait construire cette courbe de difficulté, au moyen d'un scénario ou de procédures d'équilibrage dynamique. Ces deux techniques donnent deux pistes principales quand à la réalisation d'un outil de mesure de la difficulté : permettre au game designer d'encoder facilement son scénario et de l'annoter de ces intentions en terme de difficulté, et offrir un système de mesure capable d'évaluer précisément le comportement du joueur lors des phases de jeu plus émergentes.

Enfin, nous nous sommes intéressés aux différentes techniques de mesure déjà utilisées dans l'industrie et la recherche, afin de guider l'élaboration de notre propre technique. Nous pouvons principalement retenir que les tests effectués en conditions réelles, en plus

d'être indispensable tant le game design reste un processus itératif et subjectif, seront de plus en plus accessibles grâce à l'accès de plus en plus évident des jeux à un mode connecté. Notre système pourra et devra donc appuyer sur des tests en conditions réelles. Toutefois, l'analyse systématique permet un regard supplémentaire et particulier sur un gameplay en cours d'élaboration, et ces techniques méritent également d'être explorées.

Nous avons jusqu'ici défini la difficulté comme un effort, et montré de quelle manière cet effort était construit tout au long du jeu. Dans le prochain chapitre, nous proposons une définition typologique de la difficulté des jeux vidéo, de manière à décrire plus précisément les différents aspects de l'effort produit par le joueur. Cette typologie a pour but d'apporter une grille d'analyse des gameplay, qui permette aux game designers d'identifier plus précisément les capacités dont un joueur doit faire preuve pour atteindre ses objectifs.

Chapitre 3

Les différentes formes de difficulté

Nous avons défini dans un premier temps la difficulté comme l'effort que doit fournir le joueur pour atteindre ses différents objectifs. Cette première définition est générale, et peut être complétée d'une étude plus précise des différents types de difficulté auxquelles le joueur est soumis dans la plupart des jeux commerciaux produits actuellement. Une meilleure définition de l'effort fourni par le joueur peut s'avérer utile pour tout game designer. Comprendre les différentes formes de difficulté peut aider ce dernier à manipuler à sa guise la difficulté du jeu qu'il conçoit.

Il existe plusieurs typologies des jeux vidéos [Rollings 03] [Crawford 84]. L'étude des principaux types de jeux vidéo nous a amené à extraire trois dimensions de difficulté : la difficulté *sensorielle*, la difficulté *logique* et la difficulté *motrice*. Non seulement ces dimensions correspondent au principaux types de jeux vidéo, mais elles peuvent également être rapprochées des modèles de traitement de l'information proposés en psychologie cognitive. Que le modèle de traitement de l'information postulé décrive une suite de stages discrets [Sternberg 69] ou de processus parallèles [McClelland 79], on peut considérer que le traitement de l'information débute par une analyse du stimulus, suivit de plusieurs opérations mentales et terminé par l'exécution d'une réponse. Ce modèle général est par exemple utilisé pour créer des tâches plus ou moins difficiles en éducation physique et sportive [Temprado 93], avec un *versant perceptif*, un *versant décisionnel* et un *versant moteur*. Comme nous le verrons dans cette section, ces trois phases de traitement de l'information sont pertinentes dans le cadre de l'étude de la difficulté d'un jeu vidéo, et correspondent chacune aux difficultés sensorielles, logiques et motrices.

Dans les sections suivantes, nous décrivons les différentes dimensions de difficulté, en débutant à chaque fois par l'analyse d'un type de jeu qui illustre particulièrement chacune de ces dimensions.

3.1 Difficulté sensorielle - Les jeux d'aventure

Andrew Rollings et Ernest Adams définissent les jeux d'aventure comme « *Une histoire interactive à propos d'un personnage contrôlé par le joueur* » [Rollings 03]. Toujours selon ces auteurs, les jeux d'aventure partagent les caractéristiques communes d'exploration, de collecte et de manipulation d'objets, proposent des puzzles à résoudre et attribuent une importance moindre aux scènes d'action ou de combat. *Indiana Jones et le destin de l'Atlantis* est un exemple célèbre de jeu d'aventure (Figure 3.1), du type **point and click**¹. Le joueur modifie les objets du jeu au travers d'une interface simple, sans aucune contrainte de temps. La difficulté de ce jeu, et des jeux d'aventure en général, n'est pas de faire, mais de savoir quoi faire. Dans *Indiana Jones*, par exemple, le joueur passe le plus clair de son temps à *scanner* l'écran avec sa souris à la recherche d'un objet avec lequel il puisse interagir. En effet, la plupart du décor du jeu est inerte, à l'exception de certains éléments que le joueur peut manipuler : le plus souvent un objet à ramasser et à utiliser ailleurs. La progression du jeu est ainsi rythmée de *puzzles*, c'est à dire d'énigmes plus ou moins logiques dans lesquelles le joueur doit combiner les bons objets pour supprimer un obstacle.



FIGURE 3.1 – *Indiana Jones et le destin de l'Atlantis* (Lucas Art)

La difficulté du jeu ne dépend donc pas de la précision du joueur ou de ses capacités motrices : il lui suffit de savoir pointer et cliquer pour pouvoir jouer. Elle dépend très peu de ses capacités d'inférence : la solution des énigmes repose souvent sur des actions inattendues et surprenantes (voir la série des *Monkey Island* (Lucas Art)) et le joueur progresse avant tout en essayant ce qui semble possible. Les conséquences de ces actions sont immédiates et bien souvent seule la bonne action a un impact sur l'univers du jeu. Son objectif consiste donc principalement à découvrir où se cachent les objets manipulables, l'obstacle qu'ils permettent de supprimer et leur mode d'utilisation : la clef à molette permet de démonter le radiateur, derrière lequel était cachée la clef qui ouvre la porte. Si le joueur dispose de ces informations, alors le jeu devient enfantin, il lui suffira de cliquer au bon endroit et dans le bon ordre. En effet, l'objectif du joueur dans un jeu d'aventure consiste à vivre l'aventure écrite pour lui par des scénaristes. Une aventure intéressante est souvent riche en rebondissements. Le joueur

doit sans cesse découvrir de nouveaux objets, lieux, personnages dont il ne soupçonnait pas l'existence : une aventure entièrement prévisible est rapidement monotone. Son effort est donc principalement orienté vers l'exploration, la découverte de nouvelles informations à propos de l'univers du jeu. Les jeux d'aventure mettent donc en relief une première dimension de la difficulté des jeux vidéo : l'effort fourni pour découvrir les objets du jeu et leurs propriétés.

- **Difficulté sensorielle** : La difficulté sensorielle décrit l'effort que doit fournir le joueur pour obtenir des informations nouvelles et pertinentes sur l'état de l'univers du jeu. Par informations nouvelles, on entend toute information que le joueur ne peut pas déduire des faits et règles logiques qu'il connaît déjà.

Dans un jeu d'aventure, la complexité sensorielle est donc très importante. Le joueur doit explorer un environnement dont il ne connaît rien, pour y découvrir des objets et des actions définies arbitrairement par les designers. Bien sûr de nombreux jeux d'aventure peuvent proposer des énigmes logiques et des épreuves de dextérité pure, mais la principale difficulté réside bien souvent dans l'information que le joueur doit découvrir en explorant l'univers.

La complexité sensorielle peut être mise en parallèle avec le versant perceptif des modèles de traitement de l'information. Lorsque le joueur doit fournir un effort pour enrichir sa connaissance des propriétés des objets du jeu afin d'élaborer une réponse, alors la complexité sensorielle est importante. La lecture de l'état du jeu peut être complexifiée de plusieurs manières. La situation présentée à l'écran peut être par exemple plus ou moins lisible. Certains jeux, comme les *Shoot'em up* par exemple (Figure 3.2), utilisent une abondance d'effets graphiques pour noyer l'information utile sur l'état du jeu parmi un flot d'information graphique. Même sans contrainte temporelle, la situation reste complexe à décrypter, l'affichage est ponctué d'explosions où l'on distingue alors à peine les objets du jeu. Dans l'exemple de la figure 3.2, l'embrouillage est renforcé par un monde entièrement réalisé en trois dimensions mais où le joueur suit un parcours prédéfini semblable à celui d'un véhicule de montagnes russes, ce qui noie le joueur dans un environnement sensoriellement déroutant. De la même manière, dans Collin McRae Dirt, la poussière des autres véhicules et le point de vue du joueur limitent ses capacités de lecture de l'univers du jeu. A l'autre extrême, une grille de sudoku ou un échiquier sont une représentation efficace et lisible de l'univers du jeu.



FIGURE 3.2 – Aces of The Galaxy (Artech Studios - Sierra)

La complexité sensorielle peut également s'envisager sur une échelle de temps bien plus importante. Dans un jeu d'aventure, le joueur est souvent plongé dans un univers en vue subjective dont il ne sait que très peu de choses et qu'il va devoir explorer. S'il veut acquérir des informations, il va devoir se déplacer, s'orienter dans l'espace. La présence de points d'ancrage visuels et certaines propriétés de la topologie du niveau² peuvent l'aider à s'orienter ou au contraire le dérouter. Si le joueur dispose d'une vue aérienne comme dans la plupart des RTS, sa vision des objets du jeu est plus efficace et la lecture de l'état du jeu plus évidente.

La complexité sensorielle est donc particulièrement difficile à mesurer. On pourrait fournir différentes métriques, comme le *niveau d'embrouillage de la scène* qui correspondrait au niveau d'information perturbatrice affichée à l'écran, à la *distance moyenne* entre le joueur et les différents objets du jeu dont il doit évaluer les propriétés pour être en mesure d'atteindre un objectif donné ainsi que la *lisibilité* de ces différents objets, c'est à dire leur niveau de contraste par rapport au reste de la scène. Dans les jeux d'aventures, les objets utiles de l'univers sont par exemple souvent éclairés d'une façon particulière pour permettre au joueur de les distinguer. Ces métriques apporteraient une information sur la difficulté, mais un lien direct entre cette information et la difficulté réellement ressentie par le joueur sera sûrement complexe à établir, tant le nombre de capacités cognitives qu'implique la complexité sensorielle est important : capacité d'orientation, mémoire, perception, ainsi que l'ensemble des biais et habitudes propres aux joueurs.

2. intersection à angle droits, symétrie par exemple.

3.2 Difficulté logique - Les jeux de stratégie

Chris Crawford définit les jeux de stratégie comme « *mettant l'accent sur la réflexion plus que sur l'action.* » [Crawford 84]. Entre autres définition plus militaires, le Larousse définit la *stratégie* comme « *Art de coordonner des actions, de manoeuvrer habilement pour atteindre un but* ». La caractéristique principale des jeux de stratégie concerne la difficulté de prévoir l'impact des actions du joueur. Dans un jeu d'action, l'impact des actions du joueur est clair et immédiat. Mais dans un jeu de stratégie les actions du joueur se répercutent sur l'univers du jeu en suivant une chaîne complexe de relations causes-conséquences, ce que Chris Crawford définit comme un *raisonnement séquentiel* [Crawford 03]. Le jeu de go est un exemple de jeu de stratégie par excellence. Lorsqu'un joueur pose une pierre sur le goban (le plateau de jeu), ce n'est pas uniquement pour l'effet immédiat qu'elle produit, mais pour imposer une direction à son jeu, parce qu'il a pour objectif de contrôler certaines zones et cherche à renforcer certains groupes de pierres. La valeur immédiate de ses coups, sans tenir compte de leur impact sur la suite du jeu, est d'ailleurs pratiquement toujours nulle. Excepté si le joueur capture un groupe de pierres, il ne marque aucun point en posant une pierre. Ce n'est qu'à la fin du jeu, lorsque les territoires sont délimités, que les points sont attribués et qu'une action révèle alors de manière certaine son impact global.



FIGURE 3.3 – Heroes of Might and Magic 3 (The 3DO Company)

La plupart des jeux vidéo de stratégie placent le joueur à la tête d'une armée, à l'instar de Heroes of Might and Magic (Figure 3.3). Cette armée a des capacités de production, consomme des ressources, peut produire des unités aux capacités défensives ou au contraire à utiliser pour assaillir l'ennemi. Le joueur doit prendre des décisions sur la nature de l'armée qu'il construit : quels types d'unités produire, et donc quels types de ressources récolter et quels bâtiments construire pour atteindre cet objectif ? Une fois que suffisamment d'unités d'assaut ont été construites, où et quand attaquer ? Chacun de ces choix est plus ou moins irrémédiable et influence l'ensemble de la partie. Heroes of Might and Magic est particulièrement représentatif de ce type de jeu car il se joue en tour par tour, ce que Rollings et

Adams considèrent comme une caractéristique de la version canonique du jeu de stratégie [Rollings 03]. De cette manière, chaque joueur dispose du temps qu'il souhaite pour effectuer ses actions et modifier les objets du jeu auquel il a accès. Ces modifications se font au travers d'une interface la plus claire et efficace possible, l'objectif n'est pas de tester le joueur sur ses capacités de manipulation. La rapidité du joueur reste quand même importante : même si son temps de réflexion n'est pas théoriquement borné, il l'est dans les faits et le joueur doit prendre des décisions en évaluant différentes options. Plus il est rapide, plus il considérera d'options et pourra échafauder une stratégie complexe. Mais dans l'archétype du jeu de stratégie, l'exécution de cette stratégie est facilitée au maximum et ne doit pas constituer un obstacle.



FIGURE 3.4 – Starcraft (Blizzard)

De nombreux jeux de stratégie reprennent ce principe de base pour y ajouter une composante de type action : les *jeux de stratégie temps réel*, dont Starcraft est un exemple particulièrement brillant (Figure 3.4). Comme leur nom l'indique, ces jeux évaluent toujours les capacités stratégiques du joueur, mais avec une contrainte de temps forte : les deux joueurs jouent en même temps, et non pas l'un après l'autre. De cette manière, leur capacité à manipuler rapidement et précisément les différents objets du jeu devient primordiale. En effet, ces jeux placent souvent le joueur dans le rôle d'une entité à la tête d'un peuple ou d'une armée, avec la possibilité de manipuler par exemple chaque citoyen ou chaque soldat. Pour faciliter ces manipulations, ces objets disposent d'une certaine autonomie : le joueur peut attribuer une tâche à un groupe d'objets en leur laissant la charge de s'occuper des détails. Typiquement, lorsque qu'un groupe d'unités militaires doit se déplacer d'un point à un autre de la carte, le joueur sélectionne ce groupe et clique sur sa destination. Les unités disposent alors de comportements autonomes : elles décident par exemple elles même du chemin à emprunter ou de la formation à adopter. Mais un bon joueur de jeu de stratégie temps réel se distingue en partie par sa capacité à *micromanager* ses unités. Dans l'exemple précédent, il constituera sûrement des sous groupes d'unités auquel il attribuera des objectifs différents, et surveillera le chemin emprunté pour le corriger au fur et à mesure en allant

même jusqu'à modifier la position de certaines unités pour augmenter leur efficacité, ou éviter par exemple que l'adversaire ne les remarque. L'efficacité du joueur à micromanager ses unités dépend de sa maîtrise de l'interface, de sa capacité à exécuter des actions rapidement et précisément. Starcraft propose par exemple un très grand nombre de raccourcis clavier pour la plupart des commandes effectuées avec la souris, offrant ainsi à un joueur motivé la capacité d'augmenter encore sa vitesse et sa précision.

Mais même si certains jeux de stratégie sont enrichis de composantes de type action, l'essence même de ce type de jeu sollicite avant tout les capacités d'inférence du joueur. Cet effort est tout d'abord déductif : le joueur applique les règles logiques qui gouvernent le comportement du jeu pour trouver l'action la plus efficace à exécuter. Si le joueur était capable de suites de déductions aussi interminables qu'instantanées, il pourrait, à partir des simples règles du jeu, qui régissent le comportement de chaque objet du jeu, prévoir l'ensemble des dénouements possibles pour chacun de ses choix, et choisir ainsi l'action la plus efficace. A partir règles et d'une immense capacité déductive, le joueur pourrait jouer très efficacement. Mais la complexité des jeux de stratégie est conçue pour être bien trop importante pour que ce simple fonctionnement déductif soit possible. Si les règles du jeu sont en apparence basique, à l'instar du jeu de go par exemple, il existe tellement de situations possibles qu'il est impossible de les envisager toutes pour trouver la solution optimale. En plus de capacités déductives, c'est à dire d'être capable d'utiliser une base de faits et de règles pour déterminer la conclusion qui s'impose, le joueur doit également faire preuve de capacités d'abstraction, être d'être capable d'induire de nouvelles règles, heuristiques, qui lui permettent de progresser plus rapidement vers une solution intéressante.

L'effort du joueur est donc également inductif. Le joueur doit en effet être capable, à partir de l'observation de multiples faits, d'induire les règles générales qui gouvernent l'évolution du jeu, ce que Salen et Zimmerman appellent le niveau de choix *macro* [Salen 03]. Pour pouvoir jouer efficacement, le joueur doit être capable de comprendre le fonctionnement du jeu à un niveau plus abstrait que celui des interactions locales entre objets du jeu. Il doit se construire une base de règles logiques qui ne portent pas sur le comportement basiques de simples soldats, mais sur l'impact général d'une troupe de ces soldats : ce qu'il lui coûte de les construire, leur impact sur la direction générale du jeu, étant donné le contexte général actuel. Le joueur doit être capable, à partir de la multitude de faits extrêmement localisés qu'il constate, de regrouper ces faits de manière pertinente et d'en induire de nouvelles règles. Ce niveau d'abstraction n'est pas fourni par les règles du jeu, et constitue l'intérêt principal d'un jeu de stratégie : la plupart des discussion d'un forum internet au sujet de Starcraft (Figure 3.4), par exemple, concerne cet objectif du joueur : comprendre quel comportement général adopter face à tel autre comportement général pour être en mesure de remporter la victoire.

Les jeux de stratégies sollicitent donc avant tout les capacités d'inférence du joueur, à la fois sa capacité à induire, de sa perception de l'environnement, des règles qui décrivent

le fonctionnement de l'univers du jeu, ainsi que de sa capacité à déduire, à partir de sa perception du jeu et des règles qu'il connaît, la meilleure action à effectuer. Ces aspects du raisonnement du joueur constituent la difficulté logique du jeu.

- **Difficulté logique** : La difficulté logique décrit l'effort que doit fournir le joueur pour exploiter les informations dont il dispose, c'est à dire comprendre le fonctionnement de l'univers par induction, et choisir la prochaine action à effectuer par déduction.

La difficulté logique modifie principalement la durée du versant décisionnel du modèle de traitement de l'information. Elle augmente le nombre de déductions nécessaires pour parvenir à un résultat, à partir des connaissances qu'a le joueur des objets du jeu et de leur fonctionnement, et l'oblige soit à déduire plus longuement, soit à se construire des heuristiques pour raccourcir efficacement son raisonnement.

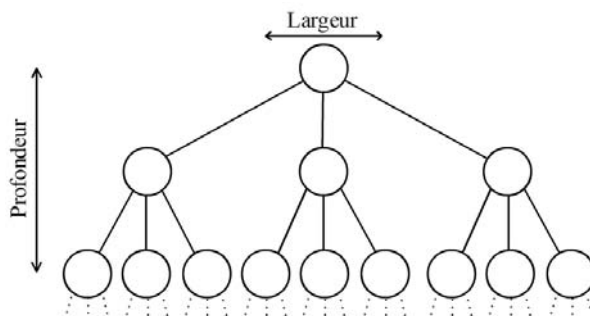


FIGURE 3.5 – Arbre de jeu

On pourrait décrire en partie la complexité logique comme la taille du raisonnement déductif que le joueur doit effectuer pour aboutir à la bonne décision. Un arbre de jeu (Figure 3.5) représente bien ce raisonnement : le nombre de mouvements possibles à chaque coup (la largeur), le nombre de coups à considérer (la profondeur), et le nombre de chemins amenant à une solution constituent des métriques intéressantes de la complexité logique. A ce titre, la complexité logique du Go est par exemple plus importante que la complexité des échecs (voir section 3.2).

Cependant, une telle métrique ne prend pas en compte la facilité avec laquelle le joueur peut se construire des heuristiques. Pour n'importe quel raisonnement, le joueur n'envisage jamais séquentiellement l'ensemble des possibilités. Certains théoriciens considèrent même que tout raisonnement est fondamentalement biaisé par des heuristiques émotionnelles, comme le neuropsychologue Antonio Damasio avec sa théorie des marqueurs somatiques [Damasio 94]. Une mesure de la complexité logique basée sur la largeur et la profondeur d'un arbre de jeu ne reflète pas complètement la difficulté logique d'un jeu. Il serait tout aussi important de comprendre l'effort nécessaire pour construire des heuristiques efficaces.

3.3 Difficulté motrice - Les jeux d'actions

La première des catégories qu'Andrew Rollings et Ernest Adams définissent est celle des *jeux d'action*, dont *Doom* est un exemple particulièrement éloquent (Figure 3.6) [Rollings 03]. Chris Crawford débute également sa taxonomie des jeux par ce type qu'il nomme jeu d'habileté / action [Crawford 84].



FIGURE 3.6 – Doom (Id Software)

Les auteurs identifient ce type de jeu comme celui des premiers jeux d'arcade, avec des règles très simples. Ils le qualifient de crispant³ et le définissent directement en fonction du type d'aptitude testé : « *Les aptitudes clés testées par le gameplay sont le temps de réaction et la coordination occulo-manuelle* ». Ces jeux demandent au joueur d'exécuter des tâches simples, mais à un rythme soutenu. Chris Crawford, game designer, identifie ce type de challenges comme *sensori moteurs* [Crawford 03].

Ce premier type de jeu semble donc particulièrement lié à la notion de difficulté puisque sa définition se base principalement sur le type d'effort demandé au joueur : vitesse et précision. L'objectif du joueur n'est pas d'échafauder des raisonnements complexes, en anticipant le déroulement du jeu à long terme. Il dispose d'un certain nombre de comportements simples, de suites d'actions à exécuter dans certains états du jeu, et son objectif consiste à reconnaître le plus rapidement possible les actions correspondant à l'état courant, puis à les exécuter le plus vite possible et sans erreur.

Par exemple le jeu Doom, qui parmi les jeux d'actions, se classe dans la catégorie reine des *jeux de tir en vue subjective*, place le joueur derrière une arme quelconque et l'envoie promener dans des couloirs toujours plus sombres, habités de créatures démoniaques et autres zombies. Les actions sont simples, il faut viser, tirer, ramasser de quoi tirer, et de temps en temps, trouver de quoi ouvrir une porte. L'objectif consiste à tirer sur tout ce qu'on peut rencontrer au détour des couloirs.

3. « *twitch game* »

On peut bien sûr isoler des éléments de Doom qui n'appartiennent pas à la définition que nous venons de faire des jeux d'action. Par exemple, lorsque le joueur échoue et recommence une partie du jeu, il a mémorisé la position de certains des ennemis qu'il a rencontré et peut alors élaborer une stratégie à plus long terme. Il ne s'agit alors plus uniquement de réagir rapidement, mais aussi, grâce à la connaissance acquise, de prévoir l'apparition des obstacles et donc d'élaborer un comportement qui maximisera les chances de réussite. De même, dans Doom, le joueur passe souvent du temps à chercher son chemin ou à trouver un passage secret dont il a entendu le son caractéristique, mais dont il ne connaît pas la position exacte. Il ne s'agit plus alors de tester ses réflexes, mais sa capacité à s'orienter dans un labyrinthe, ou à fouiller méthodiquement un espace. Cependant, si ces éléments de jeu enrichissent le gameplay de Doom, ils peuvent être considérés comme secondaires. D'un point de vue plus général, ces éléments de gameplay ne font pas partie des propriétés qu'on attribue au prototype du jeu d'action : rapidité de réaction et précision d'exécution.

Les jeux d'actions mettent en évidence une dimension supplémentaire de la difficulté d'un jeu vidéo. Dans n'importe quel jeu, le joueur dispose d'actions pour interagir avec l'univers du jeu, et les règles du jeu peuvent être plus ou moins exigeantes vis à vis de l'exécution de ces actions. Déplacer une pièce d'un jeu d'échec est trivial et ne fait pas partie du jeu à proprement parler, on peut même jouer aux échecs par l'intermédiaire d'un avatar chargé de ces manipulations. Dans un jeu d'action, l'accent est justement mis sur la manipulation.

- **Difficulté motrice** : La difficulté motrice décrit le niveau de précision spatiale et temporelle dont le joueur doit faire preuve lorsqu'il exécute une action.

La complexité motrice sollicite principalement le versant moteur du modèle de traitement de l'information. Marquer un panier à trois points au basket est principalement basé sur une complexité motrice. Le joueur sait qu'il doit imprimer une force et une direction à la balle pour qu'elle atterrisse dans le panier, mais la réalisation du bon geste demande un dosage d'une grande précision. Plus le joueur doit fournir d'effort pour effectuer ses actions dans l'univers du jeu et plus la complexité motrice est importante. Dans un jeu de billard, il y'a une grande différence entre décider de la trajectoire optimale d'une boule et lui imprimer effectivement cette trajectoire. Dans un First Person Shooter, le joueur sait que viser la tête est plus efficace que viser le torse, mais la première est plus petite que le second, et y placer précisément un curseur de visée à l'aide de la souris est bien plus difficile. En utilisant un *robot de visée*⁴ qui ajuste précisément les tirs du joueur, la complexité motrice diminue considérablement.

Une métrique de cette complexité pourrait être la *précision* temporelle et spatiale demandée au joueur, c'est à dire le rapport entre les valeurs acceptées et la plage globale de valeurs à laquelle le joueur a accès grâce aux périphériques d'entrée. On pourrait mesurer

4. aim-bot

par exemple l'évolution du taux de réussite de différents joueurs en fonction de la précision demandée pour obtenir une mesure statistique de la complexité motrice d'une action.

3.4 Synthèse des différentes dimensions de difficulté.

Dans les sections précédentes, nous avons mis en évidence trois types de difficulté, et élaboré une définition pour chacun de ces types :

- **Difficulté sensorielle** : La difficulté sensorielle décrit l'effort que doit fournir le joueur pour obtenir des informations nouvelles sur l'état de l'univers du jeu. Par informations nouvelles, on entend toute information que le joueur ne peut pas déduire des faits et règles logiques qu'il connaît déjà.
- **Difficulté logique** : La difficulté logique décrit l'effort que doit fournir le joueur pour exploiter les informations dont il dispose, c'est à dire comprendre le fonctionnement de l'univers par induction, et choisir la prochaine action à effectuer par déduction.
- **Difficulté motrice** : La difficulté motrice décrit le niveau de précision spatiale et temporelle dont le joueur doit faire preuve lorsqu'il exécute une action.

Nous proposons de résumer les différentes dimensions de difficulté dans le tableau 3.7. Pour pouvoir agir, le joueur doit construire un modèle de l'univers et l'exploiter pour prendre sa décision. Ce modèle peut être considéré comme une base de faits et de règles. Les faits décrivent les propriétés des objets du jeu que le joueur a pu observer, comme par exemple la position d'un objet à un instant donné, et la difficulté sensorielle régit l'effort que le joueur doit fournir pour les découvrir. Les règles décrivent les implications de cause à effets entre différents faits et actions : tirer sur un objet le détruit, manger un champignon fait grandir. Ces règles sont construites par induction, ainsi que par toute source d'information extrinsèque, comme la notice d'explications ou un forum internet consacré au jeu en question. Lorsque le joueur construit ces règles, il donne sens aux faits qu'il a observés et découvre comment interagir avec l'univers du jeu. Le joueur exploite ensuite son modèle du jeu, c'est à dire faits et règles, pour déduire l'action qu'il convient d'effectuer. La difficulté logique régit l'effort que doit fournir le joueur pour construire ces inductions et déductions. Il ne lui reste plus qu'à appliquer l'action choisie, en respectant les contraintes d'exécution qui lui sont propres. Ce cycle se répète indéfiniment : une fois l'action exécutée, l'univers du jeu va se trouver modifié en conséquence, et le joueur va pouvoir constater de nouveaux faits, qui vont lui permettre d'alimenter sa structure logique, qu'il exploitera à nouveau pour choisir une nouvelle action, qu'il exécutera, et ainsi de suite.

Cette première section nous a permis d'étudier trois grands types de jeux vidéo, et de définir ainsi différentes manifestations de la difficulté d'un jeu vidéo. Il existe cependant de

Dimensions de difficulté	Sensorielle	Logique	Motrice
Versant	Perception	Réflexion et décision	Action
Tâches du joueur	Explorer l'univers du jeu alimenter la base de faits qui le décrivent.	Induction de nouvelles règles à partir de la base de faits. Exploitation des règles et faits pour déduire l'action à effectuer.	Exécuter l'action en respectant les contraintes spatio- temporelles

FIGURE 3.7 – Dimensions de difficulté.

nombreux autres types de jeu, qu'il est intéressant d'étudier afin de vérifier si les trois dimensions précédentes permettent de décrire efficacement et de manière pertinente la difficulté de n'importe quel gameplay.

3.5 Etude de sous types

Dans la section précédente, nous avons abordé les plus grandes catégories de jeux vidéo. Il existe de nombreux autres sous types de jeux, détaillés par exemple par Adrew Rolling et Ernest Adams [Rollings 03]. Ces sous catégories empruntent aux types que nous avons défini précédemment, et devraient donc se soumettre facilement à une description en fonction des trois dimensions vues précédemment. A titre d'exemple, nous allons étudier quelques exemples de jeux appartenant à certaines des ces sous catégories.

3.5.1 Collin McRae Dirt



FIGURE 3.8 – Collin McRae Dirt (Codemasters)

Collin McRae Dirt (Figure 3.8) est un jeu de *sport*, plus précisément de *simulation de véhicules*. C'est une simulation car le comportement des véhicules est voué à se rapprocher le plus possible d'un comportement réel, par exemple en terme d'adhérence, de maniabilité, de puissance ou de dégâts en cas de choc. Rollings et Adams étudient les simulations de véhicule dans une catégorie particulière, ce qui se justifie étant donné le nombre de jeux appartenant strictement à cette catégorie. Mais dans le cadre de notre étude, ce type de jeu n'apporte

pas de précisions supplémentaires au regard de la difficulté, il est entièrement possible de les décrire en fonction des trois dimensions proposées précédemment :

- Complexité sensorielle : forte
- Complexité logique : faible
- Complexité motrice : forte

La difficulté de ce type de jeu repose principalement sur plusieurs aspects. Tout d'abord, le joueur doit être capable de percevoir la vitesse de son véhicule, la distance qui le sépare d'un virage ou d'un obstacle, la courbure de la route, le tout au travers de la poussière que projettent ses concurrents, avec une caméra difficilement manipulable et dans un temps très court.

Ensuite, le joueur doit élaborer une base de règles qui décrit le comportement de son véhicule. Les designers ont en effet créé un modèle physique pour cette voiture qui n'est documenté nulle part, et le joueur doit découvrir à quelle vitesse il peut aborder chaque type de virage sur chaque type de revêtement. Il existe bien sur une logique générale, le joueur s'attend à une adhérence plus réduite sur terre que sur bitume, mais il va devoir se confronter à de nombreux virages avant d'être capable d'anticiper le comportement de la voiture et de pouvoir réagir en conséquence.

Finalement, le joueur doit être capable de déclencher une action à un moment précis, avec une valeur précise (dans le cas d'un contrôleur analogique), et de réaliser des actions différentes en parallèle, comme changer de vitesse tout en freinant et en maintenant le volant dans une position précise.

Dans ce type de jeu, les difficultés sensorielles et motrices sont donc importantes. Nous considérons que la difficulté logique reste faible : la base de règles que se construit le joueur est limitée, de même que les nombre de déductions successives qu'il effectue. Si le joueur disposait d'une vision totale de l'univers du jeu et pouvait exécuter ses actions avec la précision et le timing voulu, c'est à dire si nous annulions les difficultés sensorielles et motrices, alors le joueur comprendrait très rapidement, pour chaque type de situation, la valeur exacte à appliquer pour obtenir une trajectoire optimale, et le jeu n'aurait plus aucun intérêt. La difficulté réside dans le fait que le joueur dispose de peu d'informations sur l'état du jeu, et que ses actions doivent être particulièrement précises temporellement et spatialement.

3.5.2 Everest Poker



FIGURE 3.9 – Everest Poker (Everest Gaming)

Le jeu de poker est un exemple intéressant. Les règles de ce jeu de cartes sont simples, le nombre d'actions très limité : relancer, suivre ou se coucher. A chaque tour des jetons sont misés et emportés par la meilleure main. Si un joueur perd tous ses jetons il meurt, l'objectif étant d'être le dernier en lice. Les mains distribuées aux joueurs sont issues d'un jeu battu en permanence et donc leur capacité d'action à chaque tour repose en partie sur la « chance ». Néanmoins, les probabilités sont les mêmes pour tout le monde, et la qualité d'un joueur de poker dépend d'un tout autre facteur : sa capacité à construire un modèle de ses adversaires.

En début de partie, le joueur n'a aucune information sur les autres joueurs mais au fur et à mesure, il va pouvoir déduire qui a une tendance agressive et bluffe souvent, qui relance uniquement sur une bonne main, qui se laisse emporter, qui n'a toujours pas compris les règles et reste totalement stochastique. Le poker est donc avant tout un jeu de recherche d'information, d'où la célèbre expression « *je paie pour voir* » du joueur qui n'a aucune idée du jeu de son adversaire mais qui suit (et donc paie) uniquement pour acquérir de l'information. Bien sûr pour acquérir ces informations, le joueur met au point des stratégies, tend des pièges, puis utilise ensuite ces informations pour élaborer une stratégie encore plus complexe. La complexité est donc également logique. On peut donc décrire le poker en fonction des dimensions de complexité :

- Complexité sensorielle : forte
- Complexité logique : moyenne
- Complexité motrice : faible

L'importance de la complexité sensorielle explique par exemple pourquoi les joueurs habitués à jouer sur le net peuvent avoir beaucoup de mal à jouer autour d'une vraie table :

si le jeu est à priori le même, l'interface n'a rien à voir, et les stratégies de découverte d'informations sur les autres joueurs sont alors différentes. Comme le jeu repose avant tout sur les informations qu'un joueur laisse filtrer et récupère sur ces adversaires, jouer autour d'une vraie table ou derrière un ordinateur sont deux gameplays très différents

3.5.3 SimCity 3000



FIGURE 3.10 – SimCity 3000 (EA Games)

SimCity (Figure 3.10) appartient à un autre genre de jeu de simulation, que Rollings et Adams nomment « *simulations de gestion et de construction* ». Dans SimCity, le joueur est maire d'une ville, et supervise tout son développement depuis le choix du terrain où s'installer jusqu'à la répartition entre zones commerçantes, industrielles ou résidentielles, le tracé des conduites d'eau, d'électricité et des routes, le choix des énergies et les positions des casernes de pompiers et des écoles. Toutes ces possibilités sont documentées et en nombre considérable. La complexité du jeu réside dans la capacité du joueur à prévoir l'impact de ses choix à long terme, construire par exemple un réseau routier initial suffisamment bien pensé pour supporter une montée en capacité lorsque la ville grandira, afin d'éviter les bouchons. Le mécanisme de difficulté de ce jeu peut être entièrement décrit en fonction des trois dimensions :

- Complexité sensorielle : faible
- Complexité motrice : faible
- Complexité logique : forte

Dans SimCity, le joueur dispose en général d'une grande quantité d'informations précise sur l'état actuel du jeu, et donc la complexité sensorielle est faible. Les actions sont simples à effectuer : l'écoulement du temps peut être maîtrisé, et l'interface est pensée pour offrir un contrôle intuitif et précis au joueur. La complexité logique est par contre extrêmement élevée :

à n'importe quel moment, le joueur peut influencer sur un très grand nombre de paramètres. L'évolution du jeu dépend de décisions locales sensées reproduire le comportement d'une ville réelle, et une action entraîne une chaîne de conséquences complexe que le joueur doit anticiper.

3.5.4 Fly For Fun



FIGURE 3.11 – Fly For Fun (Aeonsoft / Gala-Net)

Pour finir, Fly For Fun (Figure 3.11) est un **MMORPG**⁵ gratuit. Le joueur y effectue les tâches communes à l'ensemble des jeux du genre : il collecte des objets, élimine des ennemis et poursuit des quêtes pour obtenir des points d'expérience, utilise ces points d'expérience pour obtenir de nouvelles capacités, et se constitue un réseau social, par exemple en s'associant plus ou moins durablement à un groupe de joueurs aux intérêts communs, ou en marchandant des objets. Les jeux de rôle en ligne puisent dans l'ensemble des registres de difficulté que nous avons définis :

- Complexité sensorielle : moyenne
- Complexité logique : moyenne
- Complexité motrice : moyenne

Tout d'abord, un joueur de Fly For Fun est confronté à une composante de type action : pour gagner des points il doit exterminer des ennemis, et doit donc pour cela combiner diverses attaques et sorts de protection, avec un timing précis. Néanmoins, cette difficulté reste raisonnable et le joueur joue la plupart du temps de manière très automatique, le résultat d'un combat étant autant conditionné par les capacités de son avatar que par sa propre habileté.

La complexité logique est également moyenne : lorsqu'il combat, le joueur a accès à un nombre raisonnable d'actions, de l'ordre de la dizaine, et dont les conséquences sont

immédiates. Par contre, le joueur utilise des points d'expériences pour configurer son avatar : il peut obtenir divers sorts, augmenter certaines de ces aptitudes. Chacun de ces choix influe sur son style de jeu et sur la probabilité de remporter un combat. Le joueur doit donc respecter une certaine logique pour effectuer des choix avisés qui auront un impact à long terme sur son expérience de jeu. De même, une grande partie du succès des jeux de rôle en ligne réside dans leur côté social. Le joueur peut intégrer des groupes de joueurs, et entrer alors dans un jeu complexe de relations sociales qui peuvent aussi susciter des comportements stratégiques. Le joueur doit donc faire preuve de capacités inférentielles pour certains aspects du gameplay.

La complexité sensorielle est également moyenne : le joueur découvre une histoire et cherche à accomplir des quêtes. Il doit donc découvrir des informations sur l'univers du jeu qui ne peuvent être déduites de ces règles, comme par exemple rechercher un jouet caché dans une zone de la carte, pour le ramener ensuite à l'enfant qui l'avait perdu et obtenir une récompense.

On peut remarquer que cette difficulté est pour le joueur bien plus malléable que dans d'autres genres de jeu. Le joueur peut choisir de s'intéresser principalement aux quêtes, et ainsi limiter les dimensions logiques et motrices pour se concentrer sur l'aspect sensoriel. Il peut aussi se concentrer sur le combat et s'appuyer sur un guide stratégique pour les choix de compétences à débloquent, se concentrant ainsi uniquement sur la dimension logique. Cette adaptation du gameplay au joueur est un aspect particulier de ce type de jeu, et a par exemple amené Richard Bartle à en proposer une typologie [Bartle 96].

3.6 Conclusion

Les trois dimensions de complexité proposées permettent de caractériser la difficulté des gameplay de la plupart des jeux vidéo. Nous comprenons désormais plus précisément de quelle manière un jeu vidéo peut faire varier l'effort demandé au joueur pour atteindre les différents objectifs qu'il propose. On remarque que parmi l'ensemble des jeux vidéo développés, ces dimensions sont différemment exploitées et permettent ainsi aux joueurs de trouver un type de challenge qui leur correspond. Cette première approche est nécessaire en ce qu'elle permet de décrire ce que nous souhaitons mesurer. De cette manière, la difficulté n'est plus un concept abstrait simplement lié à l'effort du joueur mais peut être étudiée en fonction de ces trois dimensions.

Cette étude des dimensions de difficulté permet aussi d'envisager l'ampleur de la tâche : jouer à un jeu vidéo sollicite l'ensemble des capacités cognitives et motrices d'un individu. Une modélisation a priori de la difficulté, directement à partir d'une description du gameplay, sera donc forcément incomplète. Le comportement humain est complexe, soumis à de nombreux biais, dont il n'existe aujourd'hui aucun modèle complètement satisfaisant

[Noveck 07]. Ces pistes sont donc particulièrement instructives et permettent d'envisager la forme d'un modèle de la difficulté, mais soulignent la nécessité d'effectuer des tests en condition réelles, avec de vrais joueurs, pour obtenir une mesure réaliste.

Les dimensions de difficulté fournissent également une grille d'analyse de gameplay. Le game designer doit en effet identifier de quelle manière se construit la difficulté du gameplay qu'il conçoit, et donc comprendre quel effort il propose au joueur, pour mieux le modéliser. Comme nous le verrons par la suite, modéliser la difficulté demande de modéliser le joueur, et notre typologie de la difficulté peut aider le game designer dans cette tâche.

Le prochain chapitre s'intéresse à l'impact de la difficulté sur le plaisir du joueur. La théorie du game design explique que la difficulté est un paramètre important du gameplay. Mais d'autres se sont posés les mêmes questions, en étudiant directement l'impact de la difficulté sur la psychologie du joueur. Le chapitre suivant complète donc notre approche. En rassemblant plusieurs études décrivant la psychologie du joueur et son lien avec la difficulté, nous motivons plus précisément notre travail et formulons des hypothèses sur la forme la plus efficace des courbes de difficulté d'un jeu vidéo.

Chapitre 4

La difficulté du point de vue du joueur

Sommaire

4.1	Approche expérimentale	64
4.1.1	Challenge, curiosité et fantaisie	64
4.1.2	Hard Fun et Easy Fun	67
4.1.3	Autodétermination	67
4.1.4	Choix et compétition	69
4.1.5	Playground et Pacman	69
4.1.6	Difficulté et premiers instants de jeu	72
4.1.7	Synthèse	72
4.2	Approche théorique	73
4.2.1	Renforcement, regret et dissonance cognitive	74
4.2.2	L'attribution causale	77
4.2.3	Sentiment d'efficacité personnelle	79
4.2.4	L'expérience optimale	81
4.2.5	Difficulté et émotions	84
4.3	Synthèse	85

Jusqu'ici, nous avons étudié la difficulté par une approche théorique du jeu vidéo et de sa conception. Ce chapitre propose d'étudier la difficulté non plus avec un regard centré sur le jeu, mais tourné vers le joueur. Comme l'expliquent Katie Salen et Eric Zimmerman, le jeu doit être vu dans une perspective esthétique [Salen 03]. Esthétique, du latin *aesthetica*, sentir, est défini par le Larousse comme « *Qui a rapport au sentiment du beau, à sa perception.* » D'un point de vue général, l'objectif d'un jeu réside dans la gamme d'émotions qu'il nous fait ressentir, que Salen et Zimmerman rattachent au sentiment du beau.

Plusieurs auteurs ont cherché à déterminer, d'une manière générale, l'origine du plaisir d'un joueur. Certains se sont attachés à découvrir les propriétés des jeux les plus attirants et nous chercherons le lien entre ces propriétés et la difficulté. D'autres études se consacrent plus spécifiquement à comprendre de quelle manière le jeu provoque des émotions et nous chercherons à savoir en quoi la difficulté intervient dans ce mécanisme. Cette partie de notre état de l'art doit nous permettre de motiver plus profondément notre recherche, en soulignant le rôle de la difficulté vis à vis de l'expérience du joueur, mais également nous renseigner plus précisément sur la manière d'exploiter la difficulté, et donc par exemple sur certaines propriétés de bonne formation des courbes de difficulté.

4.1 Approche expérimentale

De nombreux auteurs ont cherché à comprendre le caractère particulièrement attrayant du jeu vidéo, et ce depuis le *Pacman* (Namco) des années 1980. Cette section présente différentes études empiriques réalisées dans ce but, ainsi que les modèles théoriques qui les motivent, et examine plus particulièrement les résultats obtenus sous l'angle de la relation entre difficulté et plaisir de jouer.

4.1.1 Challenge, curiosité et fantaisie

En 1980, Thomas Malone réalise une des premières études du plaisir suscité par les jeux vidéo. Le modèle proposé par Thomas Malone distingue trois dimensions principales qu'un système de jeu doit optimiser pour être le plus attirant possible : le *challenge*¹, la *curiosité* et la *fantaisie* [Malone 80b, Malone 82]. Le challenge concerne la probabilité de réussite : « *Pour être stimulant, un jeu doit proposer un but que le joueur n'est pas certain d'atteindre* », la curiosité concerne la complexité du jeu « *Un jeu vidéo peut susciter la curiosité du joueur en fournissant un environnement avec une complexité informationnelle optimale.* » et la fantaisie concerne les représentations qu'utilise l'univers du jeu : « *Un jeu sans fantaisie n'utilise que des symboles abstraits.* ». Malone distingue plus précisément la curiosité sensitive et la curiosité cognitive. La curiosité sensitive est satisfaite par la richesse en effets visuels et sonores d'un jeu vidéo. La curiosité cognitive suppose que le joueur a pour objectif d'obtenir une connaissance toujours plus complète, consistante et parcimonieuse du jeu auquel il est confronté. La curiosité sensitive est à rapprocher de la fantaisie : la première implique une variété et une originalité de représentations, la seconde juge de leur niveau d'abstraction. Parmi toutes ces dimensions, le challenge et la curiosité cognitive sont liés à notre sujet d'étude, la difficulté.

1. Nous avons gardé ici le mot challenge, qui a de nombreuses traductions en français : quelque chose de difficile, défi, stimulation. Mais par la suite nous traduirons « *challenging* » par stimulant.

Thomas Malone explique que le niveau de challenge est correct si un jeu permet au joueur de se confronter au bon niveau de difficulté, c'est à dire si le jeu fournit suffisamment de buts pour que chaque joueur trouve un challenge à sa hauteur. L'auteur propose plusieurs aspect d'un jeu à optimiser pour proposer un challenge adapté, comme par exemple utiliser des variables aléatoires ou cacher de l'information, ce qui nous ramène directement à nos dimensions de difficulté (section 3).

La dimension *curiosité cognitive* est issue de la complexité informationnelle, (i.e. pour nous des complexités sensorielles et logiques) du jeu. Malone précise que « *Le challenge correspond à ce qu'un joueur peut faire, la curiosité à ce qu'un joueur peut comprendre.* ». La curiosité cognitive offre au joueur un plaisir apparemment déconnecté de sa progression vers l'objectif : le simple fait de comprendre, de découvrir un schéma récurrent et maîtrisable du comportement de l'univers de jeu est gratifiant. Ce plaisir sera d'autant plus présent et renouvelé que l'univers de jeu sera riche et offrira de nombreuses voies à explorer.

La curiosité cognitive peut donc être vue comme le plaisir de la découverte d'un nouveau comportement pourvu de sens de l'univers du jeu. Le challenge, quant à lui, correspond au plaisir lié à la confirmation de la valeur des connaissances du joueur et de sa maîtrise du jeu, vis à vis d'un objectif donné. Un joueur attiré par le challenge cherchera avant tout à gagner et utilisera la meilleure stratégie, le joueur curieux sera attiré par l'originalité de la solution, par les informations qu'elle apporte.

Il est primordial de noter ici que le challenge et la curiosité cognitive sont des plaisirs du joueur qui se nourrissent de traits structurels commun : la recherche d'une connaissance toujours plus complète de l'univers du jeu est liée aux complexités logiques et sensorielles. Pour pouvoir faire, le joueur est obligé de comprendre.

Dans l'ensemble des études qui vont suivre, il sera donc nécessaire de ne pas faire l'amalgame immédiat entre complexité et challenge, mais de considérer que la complexité de l'univers jeu détermine à la fois le niveau de challenge et la curiosité cognitive du joueur.

Thomas Malone a étudié l'importance relative de ces trois dimensions grâce à plusieurs expériences conduites sur des élèves d'école élémentaire et du secondaire. Une première étude a consisté à recueillir les préférences de 65 élèves au sujet de différents jeux vidéo [Malone 80a]. Cette étude montre que la clarté du but du joueur est un élément fondamental, suivi par la présence d'un score, d'effets audio et de variables aléatoires. La clarté des objectifs est en effet une condition primordiale de l'existence même du jeu, comme nous l'avons vu précédemment (section 2.1). Le plaisir lié aux effets audio correspond à première vue aux dimensions de curiosité sensitive ainsi qu'à la fantaisie, et ne nous renseigne pas sur le lien entre difficulté et plaisir. On peut toute fois supposer que l'importance des effets audio tient également dans leur capacité à appuyer l'évolution du gameplay, en permettant par exemple de rendre plus évidents les objectifs du joueur. Le score et les variables aléatoires sont par contre directement révélateurs de l'importance de la difficulté, dans son impact sur le challenge et la curiosité.

Le score évalue plus finement la performance du joueur qu'une simple condition d'échec ou de réussite. Il permet, comme le note Malone, de créer des méta-buts (atteindre un score particulier décidé par le joueur) d'une difficulté choisie et concerne le plaisir lié au challenge. Cependant, le score offre également la possibilité d'évaluer plus précisément l'impact des actions du joueur et joue ainsi directement sur la complexité logique en guidant le joueur.

Les variables aléatoires, obscurcissent les conséquences des actions et augmentent directement la difficulté du jeu. Elles augmentent à la fois la difficulté et la curiosité : il devient plus difficile de mettre au point un comportement efficace et donc le challenge augmente, et le jeu apparaît comme plus riche car un plus grand nombre de comportements sont envisageables, puisque plus difficilement évaluables.

Cette première étude montre que challenge et la curiosité cognitive, et donc la complexité, sont des critères primordiaux. Le challenge semble particulièrement important, puisque la présence d'un but clair et d'un score montre que ces joueurs cherchent un cadre d'évaluation et ne se satisfont pas de la libre exploration d'un univers.

On peut noter que cette étude a l'avantage d'être facilement généralisable. En effet, Malone travaille ici sur une liste de jeux proposant différents gameplay, et ses conclusions s'étendent facilement à divers jeux vidéo du paysage vidéo-ludique des années 1980 destinés à de jeunes enfants.

Une autre étude, menée sur 80 élèves de classe élémentaire, montre l'impact important que la fantaisie peut avoir sur les préférences des élèves et souligne le fait que cette fantaisie doit correspondre au joueur ciblé, ayant eu un effet opposé sur les filles et les garçons dans cette expérience [Malone 82]. L'ajout d'un feedback constructif, c'est à dire le fait d'avoir donné au joueur plus de précisions sur le résultat de ses actions et donc diminué la difficulté, n'a eu d'impact que pour les garçons. Les résultats de cette expérience sont plus difficilement généralisables car effectués pour un gameplay bien particulier et pour une durée de jeu assez courte.

Cet étude souligne par contre une piste intéressante et confirmée dans des études postérieures : **sur un durée de jeu courte, le challenge et la curiosité cognitive, et donc la difficulté ne semblent pas des critères de plaisir primordiaux.** La difficulté ne semble donc pas être un paramètre de motivation à court terme.

Globalement, les études menées par Thomas Malone montrent que la difficulté dans son impact sur le challenge et la curiosité cognitive semble jouer un rôle important dans l'appréciation de certains jeux vidéo, mais au côté d'autres variables comme par exemple la présence de représentations non abstraites. Son travail théorique place la difficulté en bonne place parmi les facteurs déterminant du plaisir du joueur.

4.1.2 Hard Fun et Easy Fun

D'autres auteurs ont proposé de comprendre le plaisir du joueur. Nicole Lazzaro présente une étude réalisée sur trente joueurs, dont elle extrait quatre composantes du plaisir de jouer² : *hard fun*, *easy fun*, *Etat Altéré* et *Aspect Social* [Lazzaro 04]. Le hard fun représente les émotions liées au plaisir d'avoir atteint un objectif, surmonté des difficultés. L'easy fun correspond au plaisir de comprendre un univers, d'en découvrir le contenu, indépendamment de la réussite du joueur et des critères d'évaluation propres au jeu. Un joueur attiré par l'easy fun pourra explorer un jeu en utilisant un guide lui fournissant certaines solutions, par exemple. La dimension Etat Altéré concerne les joueurs qui « *jouent pour passer d'un état mental à un autre, penser ou ressentir quelque chose de différent* ». Cette dimension ne semble pas, selon nous, décrire d'où proviennent les émotions du joueur mais décrit le fait qu'un joueur joue pour ressentir une émotion, ce qu'expriment par exemple Salen et Zimmerman en expliquant qu'un jeu doit être vu dans une perspective esthétique. La dimension Aspect Social correspond au fait que le jeu peut être un outil social, et qu'une partie des émotions ressenties par le joueur provient de sa relation directe avec les autres joueurs, en dehors du système de jeu, mais facilitées par ce dernier. Les relations sociales dans les jeux sont particulièrement intéressantes, mais sortent de notre champ d'analyse, restreint au rapport d'un joueur avec un système de jeu.

Les dimensions hard fun et easy fun de Nicole Lazzaro semblent donc identiques aux dimensions de challenge et de curiosité de Thomas Malone, et sont donc également conditionnées par la difficulté. Le fait de qualifier la seconde dimension de fun *facile* traduit bien le fait qu'il est détaché de la mesure immédiate de l'efficacité du joueur et de la pression, de l'effort qu'elle suppose. L'easy fun, ou la curiosité, est une recherche, un parcours du contenu du jeu, le hard fun consiste à se soumettre directement à l'évaluation du gameplay. Les deux autres dimensions que propose Nicole Lazzaro sont pertinentes mais détachées de la difficulté. Sa modélisation, basée sur un travail empirique, confirme la catégorisation proposée par Malone et souligne à nouveau l'importance de la difficulté tout en nous rappelant qu'elle n'est pas l'unique source de plaisir.

4.1.3 Autodétermination

Richard Ryan et al proposent d'expliquer la motivation du joueur à partir de leur théorie de l'autodétermination [Ryan 06]. Plus spécifiquement, ces auteurs considèrent que les jeux vidéo satisfont des besoins psychologiques et permettent le développement d'un sentiment d'*autonomie*, de *compétence* et de *connexion*³.

2. Les dimensions de Lazzaro ne sont pas toutes traduites de l'anglais, les traductions *Fun Difficile* et *Fun Facile* nuisant à la clarté d'un propos déjà centré sur la difficulté.

3. Nous avons traduit relatedness par connexion.

L'autonomie décrit à la fois le fait que l'investissement du joueur est volontaire, qu'il entreprend de jouer uniquement pour la valeur intrinsèque de la tâche, et non du fait d'une pression ou d'une récompense extrinsèque (son autonomie dans le monde réel) mais également qu'il jouit d'une autonomie au sein même du jeu. L'autonomie dans le monde réel est tout à fait cohérente avec les définitions du jeu étudiées dans la première section. La liberté de choisir de jouer ou non est une caractéristique même du jeu. Dans le monde virtuel, l'autonomie du joueur peut être rapprochée de la curiosité chez Malone, dans le sens où le joueur jouit d'autonomie, de liberté, lorsqu'il lui est possible d'adopter un comportement original. C'est lorsqu'il va explorer et évaluer la profondeur de l'univers que le joueur se sentira autonome et pourra satisfaire sa curiosité. La curiosité correspond au plaisir de découvrir des possibilités, l'autonomie au pouvoir de choisir librement parmi ces possibilités. Mais ces deux niveaux d'interprétations sont tous les deux conditionnés par la richesse de l'univers.

Le sentiment de compétence est ressenti lorsque le joueur est confronté à un niveau optimal de challenge, acquiert de nouvelles compétences et reçoit une confirmation positive de ses progrès dans le jeu. Ryan et al considèrent que cette dimension est une des plus importantes : « *En effet, nous supposons que la compétence perçue est une des plus grandes satisfactions que procurent les jeux, car ces derniers représentent des univers où l'on peut éprouver un sentiment de contrôle et d'accomplissement.* » Cette dimension correspond au challenge, à l'évaluation de l'efficacité du joueur.

Le sentiment de connexion correspond au plaisir social de joueur, et n'entretient pas pour nous un rapport direct avec la difficulté du jeu.

Ryan et al ont réalisé quatre études successives décrites dans cette même publication [Ryan 06], pour vérifier à quel point leur modélisation permet de prédire effectivement le plaisir des joueurs. Les trois premières études concernent des jeux de Nintendo 64 et étudient le plaisir et la motivation des joueurs au travers de questionnaire administrés avant après une période de jeu. La première étude analyse les réponses des joueurs face à un seul jeu, *Mario 64 (Nintendo)*. La seconde étude utilise deux jeux, choisis pour leurs évaluations, respectivement très bonne et plus mauvaise : *Zelda : Ocarina of time (Nintendo)* et *A bug's life (Sony)*. La troisième étude utilise quatre jeux tous bien notés, *Super Mario 64 (Nintendo)*, *Super Smash Brothers (Nintendo)*, *Star fox 64 (Nintendo)* et *San Francisco Rush (Midway Games)*. Le plus grand nombre de jeux de cette étude permet aux auteurs d'étudier autant les différences inter-individuelles qu'intra-individuelles. La dernière étude porte sur un jeu massivement multijoueurs.

Toutes ces études montrent que les variables d'autonomie et de compétence permettent de prédire le plaisir du joueur. La dernière étude est la seule à utiliser un jeu multijoueurs, et montre, en plus de l'autonomie et de la compétence, l'impact de la dimension connexion sur le plaisir de jouer. Ces études sont basées sur des questionnaires, et les variables mesurées correspondent aux ressentis des joueurs. Toutes ces études permettent de supposer que les

niveaux de compétence et d'autonomie ressentis sont des prédicteurs de leur plaisir, ce qui souligne encore une fois l'importance à attacher à la gestion de la difficulté.

4.1.4 Choix et compétition

Vorderer et al [Vorderer 03] analysent également le plaisir lié aux jeux vidéo, et identifient comme dimensions principales de ce plaisir le nombre de possibilités offertes au joueur ainsi que la présence d'éléments compétitif. Ils considèrent qu'un jeu qui offre de nombreuses possibilités d'action génère plus d'intérêt qu'avec un nombre d'actions plus limités, ce qui nous ramène à la dimension de curiosité de Malone par exemple, et l'élément compétitif correspond au challenge, au fait d'avoir à un objectif identifié à atteindre.

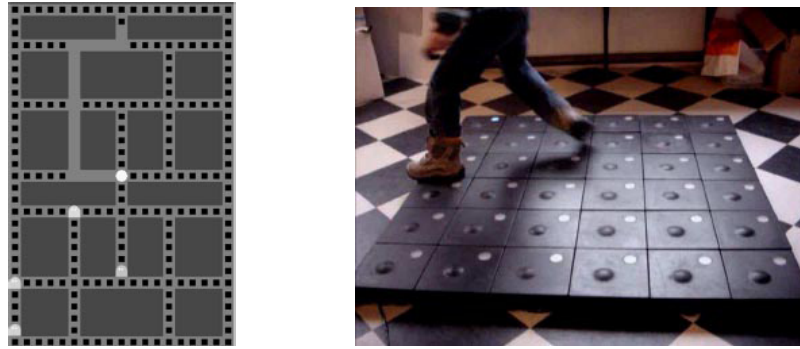
Leur première expérience consiste à proposer à des joueurs plusieurs situations imaginaires d'un jeu vidéo célèbre, *Tomb Raider (Eidos Interactive - Core Design)*. Chaque situation propose un nombre donné de possibilité d'action (le nombre d'armes du joueur) et présente ou non un élément compétitif (un monstre qui attaque le joueur). A partir d'un questionnaire auquel on répondit 349 acheteurs du jeu, Vorderer et al confirment que l'élément compétitif et le nombre d'actions possibles sont considérés par les joueurs comme sources de plaisir et confirment les hypothèses développées jusqu'ici.

La suite de l'étude concerne la prise en compte du profil social des joueurs. Certains joueurs sont par nature coopératifs, individualistes ou compétitifs, et leur manière d'aborder un jeu multijoueurs (ou contre d'autres joueurs incarnés par des IA, mais identifiables comme des adversaires) dépend de ce profil. Une seconde expérience, réalisée à partir d'un questionnaire en ligne, montre que le profil social des joueurs influence en partie le type de jeux auxquels ils jouent. Cette expérience rappelle que le plaisir du joueur dépend également de constructions psychologiques complexes issues de sa propre expérience, et que si la difficulté mérite d'être étudiée comme paramètre important du plaisir du joueur, c'est un paramètre bas niveau et en aucun cas un prédicteur total, parfait du plaisir du joueur.

4.1.5 Playground et Pacman

Le travail de Thomas Malone a servi de base aux expérimentations de Georgios Yannakakis et al, dont la direction de recherche consiste à explorer des techniques pour maximiser le plaisir du joueur. Yannakakis et al étudient l'impact des règles du jeu sur le plaisir du joueur, dans le cadre de deux univers ludiques différents : une version allégée du célèbre *pacman* de Namco, et un espace de jeu interactif, *Playground* (Figure 4.1).

Yannakakis et al ont réalisé de nombreuses expériences à partir de ces deux univers [Yannakakis 06], [Yannakakis 07c], [Yannakakis 07b], [Yannakakis 07a], [Yannakakis 09] et ont étudié à quel point la manipulation des dimensions proposées par Thomas Malone (challenge, curiosité et fantaisie) permettent d'influer sur le plaisir du joueur. Nous reprenons ici



Pacman par [Yannakakis 07c] Playground par [Yannakakis 09]

FIGURE 4.1 – Univers ludiques de Yannakakis et al

les expériences les plus récentes et les plus démonstratives.

Playground est un système de jeu composé de plaques de plastiques interactives, équipées de diodes électroluminescentes, de capteurs de pression et d'un micro contrôleur capable de contrôler l'affichage des LEDs en fonction d'un gameplay pré-programmé. Les joueurs visés par ce système sont des enfants d'une dizaine d'années. Pour réaliser leurs expériences, Yannakakis et al ont mis au point un jeu adapté au Playground, *Bugsmasher*. Dans *Bugsmasher*, les joueurs sont invités à écraser des insectes, symbolisés par l'allumage des LEDs d'une plaque de Playground. Les joueurs écrasent un insecte en marchant sur la plaque allumée. Les auteurs manipulent les dimensions de Malone au travers du temps d'apparition d'un insecte (S), de la variabilité de la position d'apparition d'un insecte (H), et de la variabilité des sons et couleurs de LED utilisées (F). Ces trois paramètres correspondent selon les auteurs aux trois dimensions de Malone. Le protocole expérimental de Yannakakis et al consiste à faire jouer différentes version de *Bugsmasher* à un groupe d'enfants, et à recueillir la préférence de chaque sujet pour l'un ou l'autre des jeux d'une paire donnée.

Plusieurs expériences, réalisées entre 2006 et 2009 fournissent des résultats intéressants vis à vis de l'impact de la difficulté sur le plaisir du joueur, au cours desquels les joueurs jouent consécutivement à deux gameplay, et doivent choisir celui qui leur plaît le plus. Une première expérience montre que parmi les trois dimensions de Malone, seule les variations de F (fantaisie) corrélient de manière significative avec les choix des joueurs [Yannakakis 06]. S et H, dimensions liées à la difficulté, ne suivent pas le choix des joueurs de manière significative. D'autres expériences confirment ces résultats. En 2007, Yannakakis et al construisent un réseau de neurone déterminant le choix d'un joueur à partir de S et H [Yannakakis 07b]. La meilleure fiabilité obtenue n'est que de 66.6%. Les auteurs concluent que les dimensions S et H ne prennent pas en suffisamment en compte les spécificités de chaque joueur ce qui diminue d'autant leur pouvoir de prédiction. En effet, S et H traduisent la difficulté absolue du jeu, par rapport à un joueur de référence, mais la difficulté ressentie par chaque joueur

dépend de son propre niveau. Dans une expérience ultérieure, Yannakakis et al déterminent que le plaisir des joueurs est corrélé avec leur temps de réponse, le nombre d'interactions et la pression exercée [Yannakakis 07a]. Ces variables sont bien plus significatives, car elles mesurent directement l'implication du joueur. Par contre, elles ne sont pas directement et uniquement images de la difficulté du jeu : le temps de réponse d'un joueur, par exemple, peut tout à fait être très élevé si le jeu lui plaît, indépendamment du temps de réponse minimal imposé par le game design (S). Le temps de réponse du joueur est une variable qui ne prend pas en compte l'objectif du jeu, mais mesure uniquement le comportement du joueur. Par contre, une autre variable mesurée par Yannakakis et al, le score du joueur, est un indicateur plus précis de la difficulté relative du jeu car il évalue le comportement du joueur en fonction de l'objectif du jeu. Cette dernière expérience montre que le score du joueur ne corréle pas linéairement de manière significative avec les préférences des joueurs, et est évalué comme la dimension la moins utile lors de la construction d'un réseau de neurones capable de prédire les préférences des joueurs.

Globalement, cette première série d'expériences montre que dans le cadre de bugsmasher, **une mesure moyenne de la difficulté absolue ne permet pas de prédire les préférences du joueur et qu'il n'existe pas de relation linéaire forte entre la difficulté relative moyenne et les préférences du joueur.**

Yannakakis et al réalisent parallèlement des expériences sur une version allégée de Pacman. Dans ce jeu, le joueur contrôle un avatar (Pacman) et doit ramasser des pastilles disséminées dans un labyrinthe. Quatre fantômes sont chargés de l'en empêcher, et le joueur perd une vie s'il se fait attraper. Les fantômes du Pacman de Yannakakis et al sont construits de manière à créer un intérêt de jeu I , calculé à d'une formule combinant : 1) l'écart entre la durée moyenne d'une partie et la partie la plus longue, 2) la variance de la durée des parties et 3) la diversité spatiale du comportement des fantômes [Yannakakis 07c]. Yannakakis et al attribuent le premier paramètre de leur formule au niveau de difficulté du jeu. Plus précisément, nous considérons que cette dimension, qui mesure l'écart entre la durée moyenne d'une partie et la partie la plus longue, correspond à la *variation* du niveau de difficulté. Le temps de jeu du joueur et en effet une bonne mesure de la difficulté du jeu, qui prend en compte son effort en fonction de l'objectif du jeu, et les deux premières dimensions de I sont une mesure de la variation de cette difficulté.

En 2007, Yannakakis et al montrent que le calcul de l'intérêt des joueurs corréle de manière significative avec leurs préférences, recueillies lors de l'expérimentation [Yannakakis 07c]. De plus, cette expérience montre que la performance moyenne des fantômes, calculée en fonction du temps de jeu moyen et du nombre de vies perdues, ne corréle pas avec les préférences recueillies. Au cours d'une autre expérience, Yannakakis et al construisent un réseau de neurones capable de prévoir les préférences des joueurs à 99.8%, à partir de la moyenne et de la variance de la durée des parties [Yannakakis 07b]. Ces résultats sont cohérents avec les expériences précédentes : la difficulté relative moyenne (performance des fantômes) ne per-

met pas de prédire les préférences du joueur. Par contre, ces expériences montrent que les variations de difficulté, dans le cadre de Pac-Man, ont un impact prévisible sur les préférences des joueurs.

Les expériences de Yannakakis et al sont révélatrices et pointent une piste intéressante concernant l'impact de la difficulté sur le plaisir des joueurs. Ce dernier ne semble pas entretenir de relation forte avec la mesure absolue de la difficulté, ni de relation linéaire avec une mesure relative de difficulté, mais **la variation de la difficulté dans le temps paraît avoir un impact déterminant sur le plaisir du joueur.**

4.1.6 Difficulté et premiers instants de jeu

Une étude menée par Klimmt et al [Klimmt 09] montre que la difficulté est un paramètre important de l'expérience du joueur mais que les joueurs ne recherchent pas forcément directement un challenge approprié. Dans leur étude, les sujets sont invités à jouer pendant dix minutes au First Person Shooter *Ureal Tournament 2 (Epic Games)*, et répartis sur trois niveaux de difficulté. Les résultats montrent une préférence des joueurs pour le niveau le plus facile, avec une difficulté quasiment inexistante. Cependant, et les auteurs en conviennent, la faible durée de l'expérience ne permet qu'une étude de la prise de contact des joueurs avec le jeu, et cette étude montre que si le niveau de challenge doit être adapté au cours de l'expérience ludique, **lors des toutes premières phases de jeu, les joueurs ont tendance à préférer un niveau de challenge très bas.** Cette interprétation est cohérente avec l'expérience réalisée par Thomas Malone et présentée en début de section.

4.1.7 Synthèse

L'ensemble des recherches présentées précédemment sont avant tout empiriques et permettent de constater un certain nombre de faits sur le rapport de la difficulté avec le plaisir du joueur. Ces expériences confirment en partie les modèles proposés par leurs auteurs, qui tentent d'identifier les caractéristiques primordiales d'une expérience de jeu. D'un point de vue synthétique, nous pouvons reprendre les caractéristiques pertinentes du point de vue de la difficulté, et récapituler leur relations avec les dimensions de complexité des premières sections (Figure 4.2).

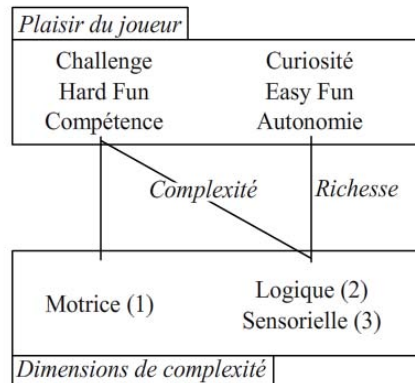


FIGURE 4.2 – Lien entre plaisir du joueur et dimensions de complexité.

La première dimension de complexité, les contraintes purement motrices d'exécution d'une action, n'influence que la difficulté du jeu et correspond uniquement au challenge de Malone et Yannakakis et al, hard fun de Nicole Lazzaro et à la compétence de Ryan et al. Les deux dernières dimensions de complexité permettent d'influer à la fois sur le challenge et sur le plaisir d'exploration : la curiosité de Malone et Yannakakis et al, l'easy fun de Lazzaro et l'autonomie de Ryan et al, selon qu'on les considère comme une source de richesse et de possibilités d'exploration ou comme un obstacle dans le parcours du joueur vers son objectif.

Ces différentes études font ressortir différentes particularités de l'impact de la difficulté sur la plaisir du joueur. Tout d'abord, la difficulté absolue moyenne ne semble pas être un bon indicateur du plaisir du joueur, mais les variations de difficulté semblent plus intéressantes. Ensuite, les plaisirs liés à la complexité du jeu, challenge et curiosité cognitive, semblent absents des toutes premières phases de jeu et demander un certain temps à s'installer.

Une fois en tête les résultats empiriques les plus récents et les théories de leurs auteurs, une interprétation théorique plus large est possible. Nous proposons, dans les sections suivantes, d'étudier plusieurs théories psychologiques plus générales qui permettent de fournir des explications plausibles du rôle de la difficulté dans le plaisir du joueur, et montrent que les résultats observés précédemment, dans le contexte précis de la difficulté, s'inscrivent dans un cadre théorique général.

4.2 Approche théorique

La section précédente présente une approche expérimentale du rapport entre plaisir du joueur et difficulté. Ces études peuvent être complétées d'une approche plus théorique, en s'intéressant notamment au lien entre les jeux vidéo et différents modèles en psychologie cognitive. Notre objectif principal concerne la difficulté, aussi cette section n'a pas pour but

de recenser de manière exhaustive l'ensemble des théories voués à expliquer le plaisir du joueur, mais se concentre sur les études et théories capables d'apporter des précisions sur le rôle de la difficulté vis à vis de ce plaisir.

4.2.1 Renforcement, regret et dissonance cognitive

En 1983, Geoffrey Loftus et Elizabeth Loftus réalisent une étude approfondie de l'attrait des joueurs pour les jeux vidéos [Loftus 83]. Tout d'abord, les auteurs tracent un parallèle entre le plaisir de joueur et le conditionnement ou renforcement, étudié par exemple au moyen des célèbres boîtes de Skinner (Figure 4.3). Dans une telle boîte, un rat peut agir sur le système au moyen d'un levier. S'il presse le levier après une certaine séquence visuelle et sonore (e.g. son aigu et lumière verte) alors le distributeur de nourriture fournit la récompense. Dans le cas contraire, la grille électrifiée se charge de punir le rat. L'expérimentateur peut alors étudier quel programme de renforcement permet d'obtenir le meilleur conditionnement du rat.

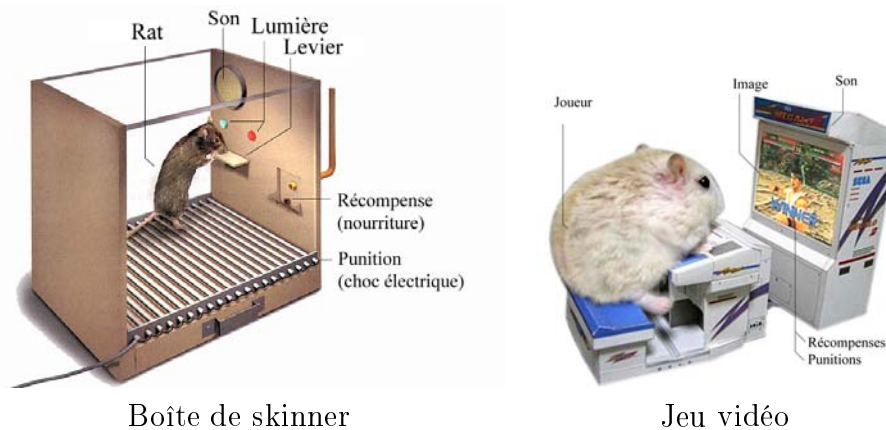


FIGURE 4.3 – Boîte de skinner et jeu vidéo

Les auteurs tracent un parallèle entre ce principe de renforcement et un joueur placé devant une borne d'arcade. Le joueur est récompensé par un score excellent ou des parties gratuites, et punit lorsqu'il perd et doit payer pour jouer à nouveau. Le parallèle devient intéressant à partir du moment où l'on insiste sur les différents programmes de renforcement. Les auteurs rappellent que si le renforcement est continu, c'est à dire si le rat gagne à tous les coups, alors il arrêtera très vite d'essayer si jamais il échoue. Bien sur, si le rat est punit en permanence, il s'arrêtera aussi rapidement ses essais. Par contre, si dès le départ, les récompenses sont distribuées par intermittence (renforcement partiel), alors le rat continuera à jouer malgré les échecs. Il n'a pas été habitué à gagner trop souvent, l'échec fait partie du jeu et il le supporte dans l'attente d'une récompense.

Le parallèle avec la difficulté d'un jeu vidéo est alors intéressant. Un très faible niveau de difficulté découragera le joueur au moindre échec, un niveau de difficulté trop important le découragera directement, mais un niveau de difficulté adapté, où gains et pertes s'enchaînent, aboutira au meilleur conditionnement. Cette idée appuie les résultats expérimentaux de la partie précédente : nous avons vu que le plaisir du joueur était corrélé au taux de variation de la difficulté. Une manière de distribuer les renforcements de manière intermittente est évidemment de faire varier la difficulté du jeu tout au long de l'expérience du joueur.

Observé sous le prisme du renforcement, le jeu montre aussi une autre particularité intéressante : les récompenses sont souvent immédiates. Lorsque Pacman mange une pastille, le score augmente immédiatement et un son approprié signale au joueur sa progression. Dans un First Person Shooter, chaque tir bien placé est souvent directement perçu par le joueur, le plus souvent par un cri de l'adversaire et une gerbe de sang appropriée. Les auteurs rappellent que ce type de renforcement, immédiat, est bien plus efficace qu'un renforcement à plus long terme. Bien sûr un renforcement à long terme nécessite un traitement cognitif de la part du sujet, qui doit attribuer la récompense à un comportement antérieur, et les rats sont sûrement moins doués que les joueurs humains à ce type de raisonnement. Néanmoins, le renforcement immédiat a la puissance de sa lisibilité et constitue sûrement un des caractères attrayants des jeux vidéo.

Loftus et al pointent également un autre aspect addictif du jeu vidéo : sa capacité à maximiser le phénomène de regret. En effet, le regret qu'on peut éprouver à la suite d'un choix dépend en partie de la distance entre *ce qui est* et *ce qui aurait pu être* [Gilovich 95]. Par exemple, imaginons deux personnes en route pour la gare, partageant le même taxi. En chemin, le taxi joue de malchance et se retrouve bloqué par un accident, des camions poubelle et une manifestation, et les deux voyageurs arrivent à la gare dépités et avec une heure de retard, persuadés d'avoir manqué leur trains respectifs. En se tournant vers le tableau de départ, l'un des deux voyageurs remarque que son train avait une heure de retard au départ et l'aperçoit qui prend de la vitesse en s'éloignant du quai. Bien que le résultat soit similaire pour les deux personnes (elles ont manqué leur train), celle qui n'a manqué son train que de quelques secondes éprouvera un regret plus poignant que celle en retard d'une heure. Cette différence viendrait du fait qu'il est bien plus facile d'imaginer la situation où le passager le moins en retard court un peu plus vite dans la gare et arrive à prendre son train, que la situation où, entre autres, le taxi évite la manifestation, les camion poubelles et prend des raccourcis. Plus la distance entre l'échec et la réussite est faible et plus le regret est fort.

Dans un jeu vidéo où le niveau de challenge est optimal, la réussite ou l'échec du joueur tiennent à très peu de choses. Si la difficulté est trop importante, le joueur aura du mal à imaginer une version du scénario où il gagne, mais s'il perd de peu, alors le regret sera le plus important, et l'envie de jouer à nouveau pour diminuer cette sensation de regret et la transformer en sentiment de réussite sera beaucoup plus importante. Un bon niveau de difficulté maximise le regret du joueur et l'amène à tenter à nouveau l'expérience.

Loftus et al s'appuient également sur la théorie de la dissonance cognitive pour expliquer l'attrait des jeux vidéo. Leur postulat suppose que le fait qu'un jeu soit payant⁴ le rend plus attrayant que s'il était gratuit. Les auteurs rappellent deux expériences, une concernant directement la dissonance cognitive [Festinger 59] et l'autre portant sur l'impact des renforcements extrinsèques sur la motivation intrinsèque [Lepper 73].

La dissonance cognitive est une théorie qui propose que l'esprit humain cherche à diminuer ses incohérences logiques. La première étude que citent Loftus et al montre que si l'on force quelqu'un à mentir sur son opinion, alors cette personne aura tendance à réviser son opinion [Festinger 59]. Cette étude montre également que plus le mensonge est justifié par une récompense extrinsèque (ici 20\$), moins la personne révisera son opinion, ce qui, selon la théorie de la dissonance cognitive, s'explique par le fait qu'elle pourra justifier son comportement sur la base du renforcement extrinsèque. Cette théorie explique par exemple la technique de manipulation dite du *pied-dans-la-porte* : si un vendeur à domicile oblige un client potentiel à lui consacrer du temps, le comportement auquel le client se voit contraint (par politesse par exemple) l'amène à réviser son jugement sur le vendeur et à perdre une part de son libre arbitre.

On retrouve cette idée d'un impact du comportement sur le ressenti sous une autre forme, dans un article de Nabi et al [Nabi 04] au sujet du plaisir associés aux médias en général, ainsi que dans un article de Fang et al [Fang 08] plus spécifiquement orientés vers le jeu vidéo. Ces écrits proposent que le plaisir associé aux jeux vidéo constitue une *attitude* et doit donc être étudié sous un angle cognitif (intellectuel), affectif et conatif (comportemental). Les auteurs rappellent ainsi que le comportement est source d'une évaluation et donc influence notre ressenti : le fait de rire à gorge déployée devant un film ou de ne pas s'y autoriser devant le même film vu au cinéma correspond à deux attitudes différentes et à deux perceptions différentes.

La théorie de la dissonance cognitive permet à Loftus et al de supposer que lorsque le joueur paie un faible droit d'entrée pour essayer un jeu, son opinion sur le jeu est automatiquement surévaluée pour justifier son geste. L'investissement est faible, donc facilement consenti par le joueur, mais le comportement a un impact sur l'opinion du joueur. En suivant Loftus et al un peu plus loin, on pourrait expliquer l'aspect addictif de jeux basés sur un système d'expérience acquise à partir de tâche courtes et répétitives comme de nombreux jeux de rôles. Dans ce type de jeu, le joueur doit typiquement vaincre des monstres à la chaîne pour emmagasiner des points d'expérience. Ces points permettent ensuite de modifier les habiletés de son personnage. Les monstres sont en général vaincus simplement et rapidement, et au départ, peu de points suffisent pour obtenir une nouvelle habileté : la courbe de difficulté d'un jeu comme Fly For Fun (Figure 3.11) est particulièrement faible au départ. Attiré par les aspects graphiques, la curiosité, le joueur s'autorise à courir après

4. En 1980, les bornes d'arcades payantes étaient un des modes principaux de consommation des jeux vidéo.

quelques monstres. Rapidement, la courbe de difficulté s'élève : les épreuves sont similaires mais la quantité de points nécessaire pour atteindre un niveau supplémentaire augmente. Mais plus le joueur joue, plus son comportement précédent justifie le faible investissement d'aller encore tuer un petit monstre de plus.

La première expérience citée par Loftus et al montre également qu'un renforcement extrinsèque diminue la dissonance cognitive. La seconde étude que rapportent Loftus et al teste l'*effet de surjustification* [Lepper 73]. Cette expérience montre que l'attente d'un renforcement extrinsèque diminue l'attrait d'une tâche auparavant considérée comme intrinsèquement attirante. De jeunes élèves, naturellement attirés par le dessin, ont vu leur motivation diminuer si une récompense extrinsèque était associée au fait de dessiner. La présence d'un renforcement extrinsèque est ainsi à la fois inhibitrice de la dissonance cognitive et de la motivation intrinsèque.

Dans un jeu vidéo, le renforcement extrinsèque est inexistant, voir même négatif. Il existe peu de joueurs professionnels de jeu vidéo et le joueur moyen s'entend plus souvent dire qu'en terme d'effets extrinsèques, le jeu est au mieux une perte de temps et au pire une source de comportement violents. Un jeu vidéo semble donc être un environnement parfait pour exploiter les effets motivationnels de la dissonance cognitive. Aucun renforcement extrinsèque ne vient inhiber l'effet supposé de la dissonance cognitive, le comportement du comportement du joueur justifie et renforce sur intérêt pour le jeu.

L'approche théorique de Loftus et al laisse donc supposer qu'une courbe de difficulté soigneusement construite a de nombreuses raisons de maximiser la motivation du joueur. Le parallèle entre jeu vidéo et dispositif de renforcement fournit une explication à l'impact des variations de difficulté sur le plaisir du joueur constaté dans la partie précédente. De nombreuses variations de difficulté supposent en effet des renforcements fréquents et variables, ce qui correspond à un programme de renforcement partiel, plus efficace en terme de conditionnement. Des résultats expérimentaux ont également montré que les joueurs préfèrent une faible difficulté en début de jeu, ce que l'hypothèse de dissonance cognitive pourrait en partie expliquer. De plus, le phénomène de regret décrit par Loftus et al permet de supposer que si le joueur échoue avec un niveau de difficulté relative très faible, le regret et donc la motivations seront maximisés.

4.2.2 L'attribution causale

La théorie de l'attribution, proposée par Bernard Weiner, est une théorie de la motivation [Weiner 85] [Weiner 05], à l'origine de l'étude empirique de Klimmt et al présentée précédemment [Klimmt 09]. Cette théorie propose qu'un échec ou une réussite importante ou inattendue, au delà des émotions immédiates et primaires de joie ou de tristesse qu'elle suscite, provoque une évaluation cognitive de ses causes, qui va conditionner la future motivation. Si l'attribution causale dépend de chaque individu et que de très nombreuses causes

peuvent être invoquées, du point de vue de leur impact sur la motivation, chacune de ces attributions peut être placée dans un espace à trois dimensions, en fonction de son *locus*, de sa *stabilité* et de sa *contrôlabilité* [Weiner 85].

Le locus correspond au fait qu'une cause peut être interne au sujet (je suis fatigué) ou externe (il pleut). La stabilité décrit à quel point la cause peut évoluer, elle peut être très stable (un trait de caractère) ou très variable (le temps qu'il fait). La contrôlabilité décrit à quel point le sujet peut modifier cette cause : on peut contrôler son temps de travail, mais pas le temps qu'il fait. La position d'une cause dans cet espace à trois dimensions permet de prédire en partie le comportement du sujet et de son entourage. Si le sujet échoue à un examen parce qu'il n'a pas assez travaillé, alors la cause est interne, variable et contrôlable. Il sera plus motivé par une nouvelle tentative que s'il considère qu'il n'est pas doué pour les études, une cause interne, stable et non contrôlable. A l'inverse, la première cause suscitera bien plus de sentiments négatifs (colère) dans l'entourage du sujet que la seconde (pitié).

Plus précisément, Weiner lie attribution causale et motivation au travers des concepts d'*espérance* et de *valeur*⁵. L'espoir d'atteindre un but, et donc en partie la motivation à entreprendre une action, dépend principalement de la stabilité de la cause. Pour maximiser la motivation, il est préférable qu'un succès soit attribué à une cause stable et un échec à une cause instable. La valeur du but recherché dépend également d'une évaluation de sa cause : recevoir un cadeau n'a pas le même impact si sa cause est accidentelle ou volontaire par exemple. Ainsi, selon le modèle d'attribution causale de Weiner, le locus d'une cause conditionne le sentiment de fierté et d'estime de soi, la contrôlabilité détermine les sentiments de colère, gratitude, honte et pitié, et la stabilité conditionne l'espoir.

La théorie de l'attribution donne des pistes sur la manière de manipuler la difficulté. Tout d'abord, si la réussite ou l'échec du joueur reposent sur des causes identifiables alors le joueur pourra effectuer une attribution causale. Jesper Juul cite ainsi Weiner et explique que l'utilisation d'une barre d'énergie, dans un jeu, permet d'accumuler et de synthétiser de nombreuses causes avant que le joueur ne perde, et limite ainsi l'attribution causale en cas d'échec [Juul 03]. Juul suppose ainsi qu'en empêchant l'attribution causale, le joueur subit plus facilement un échec. On peut noter également que si chaque incrément de difficulté se traduit par un événement observable, comme des ennemis d'une couleur différente par exemple, alors le joueur pourra plus facilement imputer son échec à une augmentation visible de la difficulté, plus évidente, qu'à un problème lié à ses propres capacités, ce qui devrait limiter de trop malmener son ego.

Mais l'attribution causale peut aussi être exploitée pour motiver le joueur. Parmi les liens entre causalité et motivation proposé par Weiner [Weiner 85], celui entre stabilité et espérance semble être également manipulable.

La difficulté sensorielle, par exemple, permet à la fois d'imputer un échec à la complexité

5. expectancy-value

du jeu (locus) ainsi qu'à une faible stabilité. Si le joueur doit découvrir une information cachée dans l'univers pour progresser, et que cette information ne peut pas être déduite, alors il sait 1) que ses capacités logiques et motrices ne sont pas mises en causes car la solution est arbitraire 2) que si le jeu est bien conçu, l'information sera découverte s'il fouille et observe l'univers du jeu.

Dans un jeu comme *Zelda*, le joueur se confronte à différents ennemis et doit découvrir comment les vaincre. La solution est souvent arbitraire : assommer avec un boomerang avant d'utiliser l'épée par exemple. Confronté à un nouvel ennemi, le joueur sait qu'il doit découvrir la règle qui permet de s'en débarrasser, et qu'il trouvera cette règle à la suite de plusieurs essais. Ces capacités logiques et motrices ne sont pas le principal obstacle, mais une information qu'il doit découvrir (cause instable). Par contre, une fois découverte, cette information lui permettra de vaincre tous les ennemis du même type, ce qui lui promet de nombreux succès, basés sur une cause stable car il ne va pas oublier cette information. Ce type de progression, basé sur la découverte d'informations arbitraires mais d'un impact durable sur les capacités du joueur permettent de garantir sa motivation, si l'on s'appuie le modèle d'attribution causal de Weiner.

4.2.3 Sentiment d'efficacité personnelle

La théorie du sentiment d'efficacité personnelle, proposée par Albert Bandura de l'université de Stanford, précise la théorie générale d'espérance - valeur [Bandura 77]. La notion d'espérance correspond à la probabilité qu'un événement se produise, et celle de valeur à l'utilité de cet événement.

Plus précisément, dans le cadre d'un jeu vidéo, l'évènement correspond à la réussite d'un challenge. La probabilité de réussir un challenge est conditionnée par la difficulté relative de ce challenge, c'est à dire à la fois par la difficulté absolue de ce challenge et par la capacité du sujet à réussir ce challenge. Lorsque que le sujet évalue ses chances de réussite, il se base à la fois sur une vision subjective de la difficulté absolue de la tâche et sur une vision subjective de ses propres capacités, ce Bandura nomme le sentiment d'efficacité personnelle.

Au delà du lien évident entre efficacité et succès, un des intérêts de cette théorie repose sur sa prise en compte l'aspect subjectif de l'espérance. Comme tout sentiment, celui d'efficacité personnelle peut être manipulé pour maximiser la motivation et la performance. Comme l'explique Albert Bandura :

« Il est supposé que l'espérance d'efficacité personnelle détermine si un comportement d'adaptation sera mis en oeuvre, la quantité d'effort produite, et la durée de cet effort faces aux obstacles et expériences aversives. La persistance dans des activités subjectivement menaçantes mais en fait relativement sûres produit, grâce à l'espoir de maîtrise, un renforcement du sentiment d'efficacité personnelle et

la réduction résultante de comportement défensif.⁶ »

Bandura rapporte diverses expériences où l'efficacité des sujets a pu être manipulée en leur fournissant une fausse mesure d'efficacité, de manière à maximiser ou minimiser leur sentiment d'efficacité personnelle [Bandura 03].

Les jeux vidéos emploient couramment ce type de manipulation : dans un jeu comme *Fallout 3* (Figure 2.4) par exemple, le joueur est survivant d'une apocalypse atomique. Il s'échappe d'un abri antiatomique où il a vécu toute son enfance et ne connaît donc rien du monde extérieur. Chaque rencontre avec un ennemi est subjectivement particulièrement menaçante : le joueur rencontre toute sorte d'animaux ayant muté suite à leur exposition aux radiations, des scorpions géants aux mutants humains surarmés. Ses rencontres se font en général dans des lieux inconnus, hostiles et isolés, où le joueur ne peut compter que sur lui-même. Les armes qu'il découvre sont vieilles, usées et s'abiment un peu plus à chaque utilisation. Néanmoins, si la difficulté perçue semble clairement titanique, la difficulté réelle est correctement définie par les game designers et le joueur a en général peu de chances de perdre. Les situations rencontrées sont subjectivement menaçantes mais objectivement relativement sûres. De plus, la victoire du joueur est en général représentée de manière à maximiser son sentiment de puissance, à grands renforts d'effets plus ou moins gores et de mouvements de caméra. Ce type de challenge, grâce à la disparité entre mode de représentation et risque réel, maximise le sentiment d'efficacité personnelle et donc la motivation du joueur.

Cette théorie peut également nous permettre de raisonner sur la progression de la difficulté. Albert Bandura explique que :

« Le succès augmente l'espoir de maîtrise ; les échecs répétés le diminuent, particulièrement si ces échecs se produisent tôt dans la course des événements. Une fois qu'une solide espérance d'efficacité s'est construite grâce à des succès répétés, l'impact négatif d'échecs occasionnels sera certainement réduit. En effet, les échecs occasionnels qui seront plus tard dépassés au prix d'un effort déterminé pourront renforcer la persistance auto-motivée, si l'on découvre par l'expérience que même les obstacles les plus difficiles peuvent être surmontés grâce à un effort soutenu. L'effet des échecs sur le sentiment d'efficacité personnelle dépendent ainsi en partie du timing et du schéma global d'expérience dans lequel s'inscrit cet échec.⁷ »

La théorie du sentiment d'efficacité personnelle confirme que la difficulté doit être plus faible dans les premières phases de jeu. Si le système de représentation de l'univers du jeu met en scène des challenges subjectivement menaçants, une difficulté faible permet de s'assurer que le joueur réussira et maximise son sentiment d'efficacité personnelle. La difficulté

6. Traduit de l'anglais : [Bandura 77] - p 191

7. Traduit de l'anglais : [Bandura 77] - p 195

pourra ensuite être augmentée de manière à maintenir le niveau de challenge. Au départ, le joueur ne possède pas de référence sur la difficulté du jeu et ne peut s'appuyer que sur les représentations de l'univers pour inférer la difficulté d'une tâche. Mais si la difficulté reste faible trop longtemps, le joueur découvrira qu'il a très peu de chances d'échouer, le sentiment de challenge diminuera, ainsi que celui d'efficacité et donc la motivation du joueur s'atténuera. Les variations de difficulté permettent ainsi de maintenir le sentiment de challenge tout en s'assurant que le joueur réussit suffisamment pour maintenir son sentiment d'efficacité personnelle et sa motivation. L'augmentation progressive de cette difficulté permet au joueur de construire ce sentiment lors des premières phases de jeu, puis de l'utiliser comme source de motivation à chaque pic de difficulté, qui une fois dépassés, viendront à leur tour le renforcer.

4.2.4 L'expérience optimale

La théorie de l'expérience optimale, ou du *flow*, proposée par Mihaly Csikszentmihalyi, est une des théories les plus citées pour expliquer l'attrait du challenge et l'importance de la difficulté dans les jeux vidéo [Csikszentmihalyi 91]. Cette théorie est issue d'une expérience menée par Mihaly Csikszentmihalyi, au cours de laquelle de nombreux sujets furent équipés, au cours de plusieurs semaines, d'un pager sonnant à intervalles aléatoires. Lorsqu'il sonnait, le sujet devait noter l'activité en cours et décrire son état d'esprit, ainsi que le niveau de challenge et de compétence que son occupation demandait. Si le niveau de challenge et de compétence dépassaient tous les deux un niveau moyen, les sujets étaient considérés impliqués dans une expérience de flow. Au contraire pour des niveaux de challenge et de compétence nécessaire bas, l'état du sujet était qualifié d'apathique. Plus un sujet rapportait être dans une expérience de flow, et plus son état d'esprit s'approchait de qualificatifs comme *fort, actif, créatif, concentré*. À l'inverse, les sujets en état d'apathie se qualifiaient souvent de *passifs, faibles, maussades, insatisfaits*.

L'expérience montre par exemple que lorsque le pager sonnait au travail, 54% des réponses fournies plaçaient les sujets dans un état de flow, contre 18% pendant le reste de la journée. À l'inverse pendant le reste de la journée, 16% des réponses correspondaient au flow et 52% à un état d'apathie⁸. Ces observations ont amené l'auteur à formuler la théorie de l'expérience optimale.

Cette théorie stipule que certaines expériences sont particulièrement agréables car elles nous permettent de canaliser toute notre attention vers des buts précis, d'une manière constructive : les expériences optimales nous enrichissent physiquement et intellectuellement. Lorsqu'un aventurier escalade un sommet, il ressort de cette expérience grandi, plus fort et plus expérimenté. Le plaisir apporté par ces expériences est à distinguer du plaisir

8. [Csikszentmihalyi 91] - p 159

sensuel, ressentit lorsque l'on comble un besoin physique. Se nourrir est agréable mais cyclique, permet de satisfaire un besoin pour un temps donné, mais la faim revient toujours et nous ramène au point de départ. Les expériences optimales ont la particularité de nous faire progresser, de « *complexifier notre moi* »⁹.

Les expériences optimales se distinguent par huit caractéristiques. Il faut 1) avoir une chance de réussir, 2) pouvoir se concentrer, la tâche doit fournir 3) des buts clairs et 4) un retour immédiat, 5) on s'y engage profondément mais sans effort et en oubliant les problèmes de la vie quotidienne, 6) on contrôle ses actions, 7) la conscience de soi disparaît mais réapparaît ensuite plus forte une fois la tâche terminée, 8) la notion du temps est altérée¹⁰.

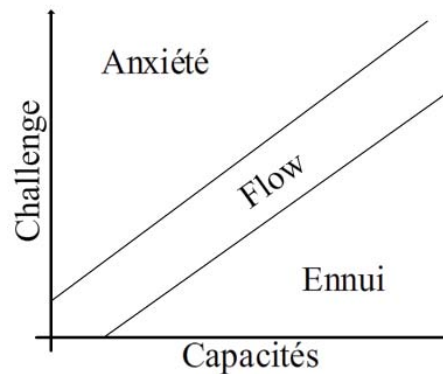


FIGURE 4.4 – Le flow : équilibre entre challenge et capacités [Csikszentmihalyi 91].

Plus précisément, l'état de flow est atteint lorsque le niveau de difficulté de la tâche est adapté. Le challenge doit être à la hauteur de nos capacités, ce que résume la figure 4.4. Si challenge et capacités sont adaptés, l'expérience optimale est possible et on peut atteindre l'état de flow. Par contre, si le niveau de challenge est inférieur à nos capacités, nous sommes dans un état d'ennui et à l'inverse, dans un état d'anxiété si le niveau de challenge est supérieur à nos capacités.

La théorie proposée par Mihaly Csikszentmihalyi s'applique parfaitement au jeu vidéo. Pour permettre au joueur de vivre une expérience optimale, un jeu vidéo doit proposer un niveau de challenge adapté aux capacités du joueur. Cette vision de l'équilibre entre difficulté et challenge fournit également une explication au fait que dans certaines études présentées dans la partie précédente, une faible corrélation linéaire ait été observée entre

9. « *Following a flow experience, the self is more complex than it had been before.* » [Csikszentmihalyi 91] - p 41

10. [Csikszentmihalyi 91] - p 49

difficulté relative et plaisir du joueur (section 4.1.5). La relation entre ces deux variables n'est pas linéaire mais correspondrait par exemple plus à une loi normale (Figure 4.5).

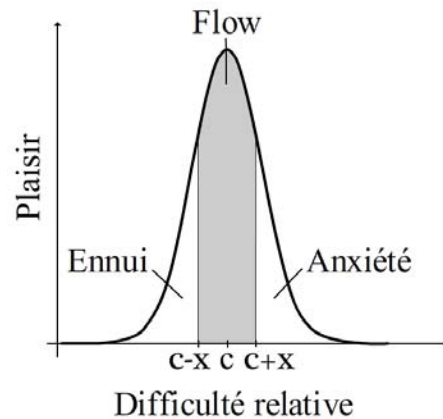


FIGURE 4.5 – Le flow et la difficulté relative.

c correspond à l'effort maximal que le joueur peut fournir et x à l'écart d'effort que la tâche peut demander sans que le joueur sorte de l'état de flow.

On remarque que la loi normale n'est pas centrée sur 0 mais sur c , effort maximal que peut fournir le joueur. En effet, selon Mihaly Csikszentmihalyi, une expérience de flow absorbe la totalité de notre attention. Si l'effort n'est pas ressenti, il est bien réel, l'organisme tout entier se consacre pleinement à une seule et même tâche. Nous avons défini la difficulté relative comme une mesure de l'effort du joueur. Cet effort est nul si la tâche ne lui demande aucune ressource, se traite par automatisme. Au cours d'une expérience optimale, le joueur est totalement absorbé, toutes ses capacités sont mobilisées et donc son effort est maximal.

La théorie du flow a fait l'objet de plusieurs études, plus spécifiquement centrées sur le jeu vidéo au domaine du jeu vidéo. Penelope Sweetser et Peta Wyeth proposent par exemple une grille d'analyse des jeux vidéo basée sur le flow [Sweetser 05]. Cowley et al s'inspirent du flow et de l'incongruité [Rauterberg 95] pour représenter le jeu sous forme d'un système de traitement d'information [Cowley 08]. John L. Sherry applique le flow au plaisir que procurent les média en général puis spécifiquement au jeux vidéo, et l'utilise pour expliquer les différences inter-genres de préférence : les choix des hommes et des femmes en terme de jeux vidéo s'expliqueraient par leurs différences cognitives et donc par le type de tâches pouvant leur apporter une expérience optimale [Sherry 04]. Beume et al tentent de mesurer le flow dans leur propre version de Pacman à partir d'une mesure objective de difficulté basée sur la distance entre pacman et les fantômes, ainsi qu'un questionnaire sur la difficulté

ressentie, une évaluation du fun et du temps de jeu supposé [Beume 08]¹¹. Jesper Juul a mené une étude sur la difficulté qui confirme la nécessité de faire gagner mais aussi perdre les joueurs, et note que ces résultats appuient la théorie du flow [Juul 09]. Noah Falstein s'appuie sur le flow pour étudier le plaisir de jouer, et remarque que la progression de difficulté, tout en restant dans le canal du flow, doit être variable [Falstein 05]

D'un point de vue général, la théorie du flow cadre parfaitement avec l'idée d'une courbe de difficulté adaptée à l'expérience du joueur. Cette théorie souligne que si la difficulté relative dépasse certains seuils positifs ou négatifs, alors le plaisir du joueur sera compromis.

4.2.5 Difficulté et émotions

Comme nous l'avons dit précédemment, le jeu vidéo est tout d'abord à considérer dans sa valeur esthétique, dans l'ensemble d'émotions qu'il procure au joueur [Salen 03]. L'étude des émotions est un domaine complexe de la psychologie, sujet à de nombreuses controverses, dont Gilles Kirouac propose par exemple une revue détaillée [Kirouac 94]. Une description des différents modèles serait ici à la fois hors de propos et hors de nos compétences. Cependant, certains travaux suggèrent l'importance de la difficulté dans l'esthétique des jeux vidéo et méritent d'être soulignés dans cette section.

Torben Grodal analyse l'impact émotionnel des jeux vidéo à partir de deux principes : les situations dangereuses, menaçant notre survie, sont propices à générer des émotions fortes, et ces émotions dépendent de l'évaluation cognitive de ces situations [Grodal 00]. Tomber dans le vide est par exemple une situation propice à l'expérience d'émotions fortes, dont la valeur dépendra par exemple du fait qu'on y tombe par accident (évaluation cognitive plutôt négative) ou par loisir, muni d'un parachute (évaluation cognitive plutôt positive).

Les jeux vidéo, ainsi que de nombreux autres médias, exploitent ce type de situations pour susciter diverses émotions chez le spectateur. La difficulté joue alors un rôle particulier dans ce processus : elle contrôle en grande partie le niveau de danger auquel le joueur est confronté. Bien sûr, le système de représentations joue aussi un rôle principal, et confronter le joueur à un orc géant et hurlant provoquera sûrement au départ de fortes réactions. Mais si les émotions sont le résultat d'un processus cognitif, ainsi que d'un apprentissage [Damasio 94], l'orc ne continuera à susciter un stress chez le joueur que s'il représente un danger potentiel, si le vaincre est associé à un certain niveau de difficulté. Si l'orc ne représente pas de réel danger, le joueur le comprendra et le retiendra, et son évaluation cognitive et donc son émotion en seront affectées.

D'un point de vue plus général, plus abstrait, on peut également considérer la difficulté dans une perspective de traitement de l'information, comme le niveau d'incongruité d'un

11. Le résultat le plus intéressant de cette étude montrent qu'une faible corrélation (0.24 $p=0.01$) existe entre les mesures objectives et subjectives de difficulté.

message par exemple. Le modèle mis au point par Rauterberg propose qu'un niveau d'incongruité optimal est source d'émotions positives, alors qu'une trop grande ou trop faible incongruité sera source d'émotions négatives [Rauterberg 95]¹². Cette vision est partagée dans des domaines artistiques comme la musique [Meyer 56] ou l'étude plus général des oeuvres ouvertes [Eco 65]. Dans ces deux cas, le spectateur est face à une oeuvre qu'il interprète sur la base de ses connaissances. Il s'attend à retrouver des formes, un style qu'il a appris à distinguer mais que l'oeuvre d'art manipule, remet en question. Ces variations entre l'objet présenté et l'ensemble de représentations du spectateur créent un suspense, une émotion, jusqu'à ce que la disparité qui au départ éveillait les sens finisse par être intégrée, habituelle.

Plusieurs auteurs considèrent que l'apprentissage, la recherche et l'intégration des disparités entre le matériel présenté et le modèle mental qu'en a construit le joueur sont l'essence même du jeu [Koster 05b] [Crawford 84] [Crawford 03]. Certaines études suggèrent en effet qu'il existe un plaisir inhérent à l'apprentissage, et pointent même les structures neurologiques responsables de ce plaisir [Biederman 06].

La complexité du jeu semble donc avoir un rôle fondamental à jouer vis à vis du plaisir du joueur. A chaque fois que la difficulté relative augmente, le joueur est soumis à un stress. Ce stress suscitera une émotion positive si le joueur parvient à le surmonter et réussit l'épreuve. Ce faisant, le joueur devra traiter des informations inhabituelles, inattendues, activité qui constituerait une source de plaisir en elle même. Ces deux mécanismes s'appuient sur une variation de la difficulté, et tendent à confirmer son statut fondamental vis à vis du plaisir du joueur.

4.3 Synthèse

Tout d'abord, ce chapitre a permis de souligner l'importance de la difficulté sur le plaisir et la motivation du joueur à la fois grâce aux études empiriques mais également à partir des différentes propositions théoriques relevées dans la littérature. La théorie du flow et les liens proposés entre difficulté et émotions montrent, d'une manière générale, que la complexité d'un jeu, correctement manipulée, est source de plaisir pour le joueur.

Les différentes études empiriques présentées dans la première section montrent tout d'abord que, globalement, la complexité du jeu a un impact certain sur le plaisir du joueur. Plusieurs dimensions proposées pour décrire le plaisir du joueur, telles que challenge et curiosité, hard fun et easy fun ou compétence et autonomie, dépendent toutes directement de la complexité du jeu.

Plus précisément ces expérimentations ont permis de proposer des propriétés de bonne formation d'une courbe de difficulté. Tout d'abord, il semble que la difficulté absolue n'ap-

12. Modèle que Lankveld propose effectivement d'appliquer au jeu vidéo [van Lankveld 08]

porte pas beaucoup d'informations sur le plaisir du joueur, ce à quoi nous pouvons proposer deux explications. Premièrement, à un même niveau de difficulté absolue vont correspondre des efforts différents pour chaque joueur, en fonction de leurs propres capacités. Mais plus important encore, une valeur de difficulté absolue calculée sur l'ensemble de l'expérience ne prend absolument pas en compte la forme de la courbe de difficulté, qu'elle réduit à une valeur unique, et son pouvoir descriptif est donc trop faible. La mesure de la difficulté relative permet de contourner le premier écueil puisqu'elle mesure l'effort fourni par chaque joueur, néanmoins, cette mesure est également d'un faible pouvoir descriptif. De plus la recherche d'une corrélation linéaire contredit la forme générale d'une relation entre difficulté et plaisir, que présente particulièrement bien la théorie du flow.

Par contre, ces études montrent que le taux de variation de la difficulté permet de prédire plus efficacement l'intérêt du joueur. Les modèles présentés par Yannakakis et al tendent à montrer que les joueurs apprécient un niveau de variabilité important ce qui reste à étudier plus précisément et dans d'autres contextes.

Finalement, l'expérience de Klimmt montre que pendant les premières minutes de découverte d'un jeu vidéo, le joueur semble apprécier un niveau de difficulté faible.

Ces résultats semblent confirmés par une analyse plus théorique du lien entre difficulté et plaisir du joueur. Tout d'abord, le phénomène de dissonance cognitive semble confirmer le bien fondé d'une courbe de difficulté faible en début de jeu. La théorie du sentiment d'efficacité personnelle appuie également cette idée. C'est en tout début de jeu qu'un échec serait le plus dévastateur pour le joueur et également à ce moment qu'il est le plus facile de créer une distance entre difficulté subjective et difficulté objective.

Ensuite, le parallèle entre programme de renforcement et courbe de difficulté confirme l'intérêt de proposer des variations fréquentes de difficulté, de manière à s'appuyer sur un programme de renforcement partiel.

Finalement, la notion de regret propose que l'échec du joueur survienne à un niveau de difficulté relative le plus faible possible. Si le joueur échoue à cause d'une simple petite erreur, son regret et sa motivation seront maximisés. La théorie de l'attribution le suggère également, si la difficulté relative est faible, le joueur pourra attribuer son échec à un relâchement d'attention, une cause personnelle, instable et contrôlable donc source de motivation.

Ainsi, observée à une grande échelle, une courbe de difficulté absolue en plateaux successifs paraît très intéressante. Si chaque incrément de difficulté permet de rejoindre ou de dépasser légèrement le niveau du joueur, sa motivation devrait être maximisée. En effet, si un incrément de la courbe de difficulté ne dépasse pas de beaucoup le niveau du joueur, celui-ci aura plus de chances d'attribuer son échec à une cause surmontable que si l'incrément est très important et le joueur dépassé. Ensuite, la phase de plateau lui permettra de sécuriser son apprentissage et de confirmer sa réussite, qu'il pourra attribuer à une cause stable.

Observée de manière plus précise, chaque plateau ne doit pas forcément proposer un niveau de difficulté constant. Un niveau de difficulté oscillant entre le niveau du joueur et

un niveau plus négatif serait plus approprié. De cette manière, si le joueur joue toujours à son meilleur niveau, il réussit, mais s'il relâche un peu son attention, la moindre erreur correspondra à un échec. Cet échec sera du à une erreur que le joueur aurait pu éviter avec plus d'investissement, donc à une cause instable. On sait que maintenir un niveau d'attention élevé est fatigant, et qu'il faut donc forcément laisser des plages de respiration dans la courbe de difficulté de manière à ce que le joueur se repose. Une courbe de difficulté optimale placerait donc des plages de respiration entre des plages de difficulté égales au niveau du joueur.

Le jeu Pacman fonctionne par exemple de cette façon. D'un point de vue macroscopique, la difficulté progresse en plateaux, sur des très nombreux niveaux. Pendant chaque niveau, la difficulté n'est pas constante mais varie. Les fantômes suivent en effet des cycles réguliers, chassant Pacman pendant une période puis retournant chacun dans leur coin la période suivante [Pittman 09]. De cette manière, le joueur dispose de périodes où il peut facilement confirmer sa supériorité, et de périodes plus difficiles, où il doit maintenir son attention au maximum pour échapper aux fantômes.

Etant donné les conclusions que nous tirons des différentes études rapportées dans ce chapitre, nous pouvons formuler dès maintenant les caractéristiques que devrait respecter une courbe de difficulté optimale :

- une première phase de progression graduelle,
- une seconde phase de forte variabilité,
- les challenges les plus difficiles doivent à peine dépasser le niveau du joueur.

Nous étudierons la validité dans la partie expérimentale de cette thèse. Mais avant d'être en mesure d'étudier expérimentalement la forme d'une courbe de difficulté, il convient d'en proposer une méthode de mesure. Dans le prochain chapitre, nous présentons nos toutes premières tentatives de mesure de difficulté.

Chapitre 5

Premières mesures de difficulté

Sommaire

5.1	Analyse de la difficulté et joueur synthétique	89
5.2	Premières expériences	90
5.2.1	Le cube-serpent	90
5.2.2	Skywar	92
5.2.3	Pacman	97
5.3	Limitations d'un joueur synthétique	99

Les premières expériences réalisées au cours de cette thèse ont consisté à analyser la difficulté de différents gameplays à l'aide d'un joueur synthétique, c'est à dire une intelligence artificielle chargée de remplacer le joueur. Cette première approche, empirique, a pour but de confirmer et d'étendre les recherches présentées dans la section 2.3.2, en nous concentrant plus particulièrement sur une analyse de la difficulté. Ce chapitre présente ces expériences et discute de leurs apports.

5.1 Analyse de la difficulté et joueur synthétique

Durant ces premières expériences, nous avons tenté d'analyser la difficulté de différents jeux, au moyen d'un joueur synthétique. Il est en effet possible de construire un agent logiciel qui soit en mesure de percevoir l'univers du jeu et d'y effectuer des actions, avec l'objectif de maximiser un résultat. Les différents objets de l'univers sont des structures de données à priori accessibles à n'importe quel agent logiciel, et les actions disponibles sont toutes connues et définies par le game designer, étant constitutives des règles du jeu. De plus, de nombreux jeux utilisent déjà des agents logiciels chargés de simuler un adversaire, et l'interface entre l'univers du jeu et ces agents est donc déjà définie.

L'utilisation d'un joueur synthétique présente de nombreux attraits, dont celui de supprimer tout facteur humain, ce qui permet d'organiser des expériences sans avoir à recruter de vrais joueurs. Les itérations entre différents tests sont alors beaucoup plus rapides et moins risquées, ce qui offre la souplesse nécessaire à des expériences réalisées en début de thèse. Cette approche permet également d'évaluer l'intérêt de la conception de joueurs synthétiques pour un studio de jeu. Développer un joueur synthétique demande un investissement, qui peut être rentabilisé par la suite en automatisant une partie des tests effectués sur le gameplay. Ces premières expériences visent à montrer, du point de vue de la difficulté, le type d'informations qu'est capable de fournir un joueur synthétique, ainsi que ses limitations.

5.2 Premières expériences

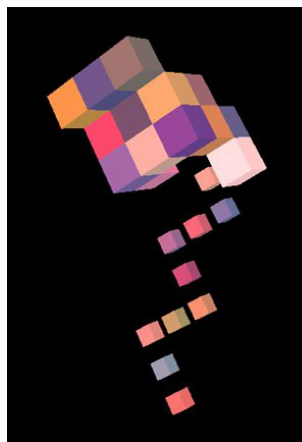
Nous présentons dans ces sections trois expériences au cours desquelles nous avons analysé le gameplay d'un jeu vidéo au moyen d'un joueur synthétique. Chaque expérience présente un type d'analyse et un gameplay particuliers, puis décrit les résultats obtenus.

5.2.1 Le cube-serpent

Cette première expérience étudie la difficulté d'un jeu aux règles simple mais dont l'objectif est particulièrement difficile à atteindre. Ce jeu n'est pas au départ un jeu vidéo, mais sa simplicité le rend particulièrement facile à simuler dans un univers virtuel. Ce jeu en bois s'appelle le *cube-serpent*, présenté avec son équivalent virtuel figure 5.1.



Cube serpent en bois



Cube serpent virtuel

FIGURE 5.1 – Cube-serpent

Le cube serpent se compose de petits cubes de bois attachés les uns aux autres. Chaque cube a au moins une face plaquée à celle d'un autre cube. Ces deux faces peuvent pivoter l'une contre l'autre, en effectuant ainsi une rotation autour de leur normales. L'objectif consiste à replier le serpent formé par cette suite de cubes en un hyper cube d'une taille donnée.

Il existe deux versions de ce jeu. La plus populaire propose un serpent à replier dans un hypercube de trois petits cubes de côté. Dans une version plus complexe, le serpent doit être replié dans un hypercube taille quatre. En effet, une bonne manière de complexifier un gameplay aux règles aussi basiques que celui du cube-serpent consiste à fournir un serpent plus long, et donc à augmenter le nombre de possibilités lorsqu'on tente de le replier.

Nous avons choisit de réaliser cette expérience après avoir constaté à quel point la difficulté différait entre les deux jeux. Si la version populaire du cube-serpent, de taille trois, permet d'atteindre une solution après quelques heures de recherche, celle du grand cube apparait totalement inatteignable, et ce même en possession de la petite fiche de solution. Nous avons cherché à vérifier si cet écart de difficulté apparaissait aussi important pour un joueur humain que pour un joueur synthétique.

Nous avons donc construit l'équivalent du cube-serpent sous forme virtuelle. Il s'agit d'une application qui exploite la librairie graphique Open-GL pour le rendu du serpent, et encode l'angle de rotation de chaque cube dans un tableau d'octets. Tous les cubes ne sont pas à manipuler par l'algorithme de résolution, car le serpent possède des sections *rigides*, décrites dans la figure 5.2. Dans ces sections, la rotation de l'ensemble des cubes produit le même effet sur le serpent, et de plus, il est impossible de replier ces régions.

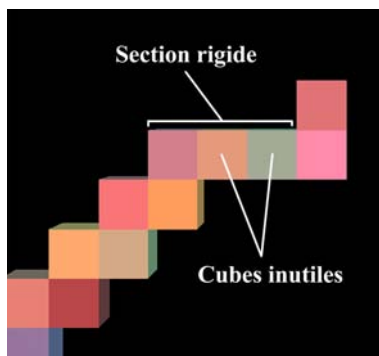


FIGURE 5.2 – Section rigide d'un cube-serpent.

La section rigide a deux propriétés : elle ne peut pas se replier sur elle même, et tous les cubes qui la composent ont un impact similaire sur le reste du serpent quand on les pivote.

Nous avons ensuite écrit un algorithme de recherche brutal, qui progresse section rigide par section rigide, en cherchant à chaque fois un angle de rotation qui maintienne la prochaine

section rigide à l'intérieur de l'hypercube. S'il n'existe aucune position valide pour la section rigide actuelle, c'est qu'un des choix précédents est mauvais, et l'algorithme revient une section en arrière. Lorsque la dernière section rigide est placée dans l'hypercube, l'algorithme a trouvé une solution. Nous avons conçu cet algorithme en cherchant à reproduire notre propre comportement, lorsque nous essayons de trouver une solution au cube-serpent en bois.

Nous avons exécuté l'algorithme plusieurs fois pour chaque type de cube-serpent, sur la même machine. Nous avons ensuite comparé le temps mis par l'algorithme pour parvenir à une solution dans chaque configuration. Ces résultats sont présentés figure 5.3.

Condition	Moyenne	Ecart type	Nombre d'essais
Serpent 3*3	125 ms	48 ms	20
Serpent 4*4	448 heures	244 heures	6

FIGURE 5.3 – Temps d'exécution pour découvrir la solution de cube-serpents.

On remarque que la différence de temps de calcul, et donc l'effort fourni par l'algorithme, passe de quelques millièmes de secondes à plusieurs jours selon qu'il s'agit du serpent 3*3 ou du serpent 4*4. L'algorithme synthétique semble donc confirmer d'une manière écrasante la différence de difficulté que nous avons ressentie entre les deux conditions. Agrandir l'hypercube d'un petit cube de côté a considérablement augmenté la difficulté du casse tête. Bien sûr, il faut noter que l'algorithme de résolution est aussi idiot que son concepteur, et qu'il existe peut être des heuristiques, au delà des simples sections rigides, qui permettent de percer le mystère du cube 4*4. L'algorithme a permis de valuer la différence de difficulté pour une méthode de résolutions donnée, qui n'est peut être tout simplement pas appropriée. Seul un test sur des joueurs humains aurait pu permettre de découvrir si un joueur moyen peut trouver la solution dans un temps raisonnable.

5.2.2 Skywar

Notre seconde expérience a consisté à étudier le gameplay d'un jeu vidéo pour tenter d'y déceler une erreur d'équilibrage de la difficulté. Ce genre d'expérience est en général particulièrement complexe à réaliser sur un gameplay conçu par un studio de jeu vidéo : il faut être en mesure de simuler le gameplay ou d'accéder au moteur du jeu, ce qui demande des informations ou des outils logiciels que les studios, en perpétuelle compétition, se gardent bien d'offrir. Il existe cependant certaines exceptions, comme par exemple le First Person Shooter Unreal Tournament, dont le moteur est commercialisé et dont l'interface logicielle est ainsi accessible. Mais dans leur grande majorité, les jeux mis sur le marché sont particulièrement protégés.



FIGURE 5.4 – Skywar (MotionTwin)

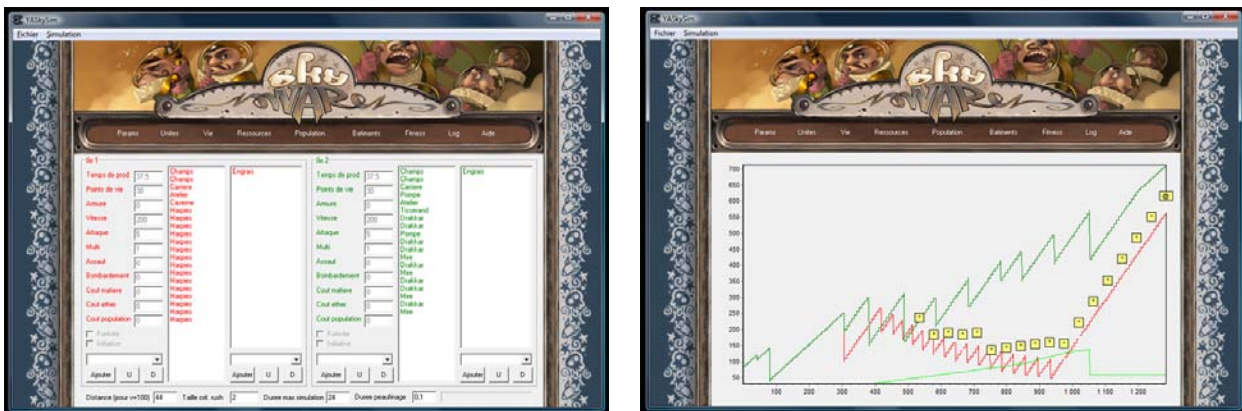
Chaque joueur possède un certain nombre d'îlots, sur lesquels il construit divers bâtiments. Ces bâtiments produisent des ressources et des unités de combats qu'il utilise pour attaquer les îles adverses. Une partie dure plusieurs jours, chaque action pouvant prendre plusieurs heures pour s'exécuter.

Certains jeux échappent cependant à cette règle, ce qui est le cas du jeu que nous avons souhaité étudier : Skywar (MotionTwin - fig. 5.4). Tout d'abord, Skywar est un jeu de stratégie temps réel, ce qui oblige ses concepteurs à fournir de nombreux détails concernant son gameplay. Les joueurs doivent en effet connaître l'ensemble des caractéristiques de chaque unité pour pouvoir élaborer leur stratégie, le jeu serait sinon sûrement bien trop complexe : ce serait comme jouer aux échecs tout en essayant de découvrir les règles de manipulation des pièces. Le premier avantage du jeu Skywar est donc tout d'abord de disposer d'une documentation précise.

Ensuite, Skywar a été conçu pour être joué sur le web, et réalisé en flash. La logique du jeu est calculée côté serveur, et transmise au client flash pour un simple rendu. Cette logique est donc particulièrement simple, elle ne demande qu'un calcul léger au serveur. Elle est de plus précisément décrite dans la documentation et les forums du jeu. Certains jeux plus complexes, comme Starcraft (Blizzard) par exemple, fournissent également une description des unités, mais la logique du jeu est bien plus sophistiquée et n'est précisément décrite nulle part.

Toutes ces caractéristiques rendent Skywar particulièrement intéressant pour cette expérience. Les RTS sont des jeux au gameplay émergent, particulièrement sensibles aux problèmes d'équilibrage. La simplicité du gameplay de Skywar ne diminue en rien sa qualité de jeu de stratégie, mais nous permet d'en construire facilement un simulateur, et d'en étudier l'équilibre.

Skywar est un jeu de stratégie temps réel, qui se joue en ligne, contre trois autres joueurs. Le terrain de jeu est un chapelet d'îles volantes. Chaque joueur choisit une île de départ, et peut ensuite coloniser d'autres îles. L'objectif de chaque joueur est de détruire tous les bâtiments et unités des autres joueurs, ou de les forcer à abandonner. Comme dans la plupart des RTS, le principe de Skywar consiste à construire des bâtiments, qui offrent au joueur différentes capacités : défendre l'île sur laquelle ils sont construits, produire des unités de combat ou récolter les ressources de l'île. En parallèle, les joueurs choisissent une suite d'améliorations à développer. Les améliorations sont activées les unes après les autres, en suivant un ordre défini par le joueur, et lui apportent différents bonus. Skywar propose un gameplay lent, c'est à dire que les actions du joueur demandent plusieurs dizaines de minutes avant d'être totalement exécutées, et donc les parties durent plusieurs jours.



Ecran principal

Construction des unités dans le temps

FIGURE 5.5 – Simulateur de SkyWar

Notre objectif, pour cette expérience, n'est pas de mesurer la difficulté de skywar dans un contexte précis, mais d'en vérifier l'équilibre. Skywar est un jeu multijoueurs, et donc la difficulté d'une partie repose sur l'habileté des opposants, si tant est que le gameplay permette toujours aux opposants de s'affronter. S'il existe une stratégie simple qui garantit la victoire au joueur qui l'exécute, alors la difficulté deviendra nulle pour ce joueur, dès la stratégie découverte. Notre objectif est donc d'explorer le gameplay de Skywar, à la recherche d'une telle stratégie.

Nous avons programmé un simulateur capable de reproduire la logique de Skywar. Ce logiciel simule une partie en quelques millisecondes, ce qui permet de tester le gameplay beaucoup plus rapidement qu'en jouant sur le serveur. Bien sûr, nous ne sommes pas en mesure de reproduire l'aspect stratégique de ce jeu, car nous ne pouvons pas simuler le comportement des autres joueurs. Néanmoins, nous sommes capables de simuler un scénario particulier, qui nous permet de comparer les capacités des bâtiments et unités, étant donné

un comportement basique des joueurs.

En effet, une partie de l'équilibrage d'un RTS peut être exprimé de la façon suivante :

« Etant donné deux joueurs s'affrontant de manière très basique, en envoyant par exemple toutes leurs unités attaquer l'autre joueur lorsque le nombre total d'unités en leur possession atteint un certain seuil, existe t'il un type de bâtiment et un type d'unité qui permette d'obtenir le plus souvent la victoire ? »

Les joueurs appellent **build**¹ la liste de bâtiments et d'unités construites au cours d'une partie. Les premiers instants de jeu sont toujours consacrés à la mise en place du build car les joueurs ne disposent d'aucune unité et doivent tout d'abord construire leur base puis un stock d'unités. Dans un RTS bien équilibré, il ne doit pas exister de build qui garantisse la victoire du joueur, il doit toujours exister une faiblesse qu'un autre build pourra exploiter, à la manière d'un jeu pierre-feuille-ciseaux.

Nous avons donc codé un scénario ne comportant que deux îles et deux joueurs. Les paramètres modulant le comportement de ces deux joueurs sont les suivants :

- La liste des bâtiments et unités construits par le joueur
- La liste des améliorations activées par le joueur
- Le nombre d'unités total nécessaire avant de les envoyer attaquer l'ennemi

Le simulateur permet ainsi de comparer deux builds très rapidement. De manière à accélérer encore l'étude du gameplay, et pour permettre de faire évoluer un build automatiquement, nous avons programmé un algorithme d'optimisation de type génétique. Il est en effet possible d'encoder un build sous la forme d'un gène, et d'opérer une mutation, c'est à dire une modification aléatoire d'un élément de ce build. Notre algorithme est très basique et n'utilise pas d'opérateur de croisement ni ne renouvelle la population. Il initialise simplement une population avec la stratégie fournie, puis fait muter l'ensemble de la population. Pour chaque individu, si la mutation fournit un build plus performant, celui-ci est sauvegardé, sinon, la mutation est annulée et une autre mutation est réalisée. La performance d'un build correspond au temps nécessaire pour vaincre l'ennemi lorsqu'il est utilisé.

Une alternance de recherche manuelle et d'optimisation automatique nous a permis d'obtenir un build particulièrement performant, auquel nous n'avons pu trouver aucun équivalent. Ce build est plutôt inattendu car il exploite à outrance une unité en apparence plutôt faible, la *harpie*, qui dispose de peu de points de vie et n'est réellement efficace que contre d'autres unités volantes. Néanmoins, cette unité peut être produite très rapidement, et pour un faible coût, ce qui lui permet d'établir rapidement sa suprématie.

Nous avons cherché à confirmer cette analyse en utilisant ce build contre de vrais joueurs, sur le serveur de jeu. Nous avons obtenu les résultats suivants, récapitulés figure 5.6.

Historique			
Bataille	Pos	Frag	Pts
Assaut n° 63359 du 2009-12-10 au 2009-12-13	2	1	0
Challenge n° 62680 du 2009-12-07 au 2009-12-10	6	0	0
Assaut n° 61079 du 2009-12-01 au 2009-12-03	1	3	0
Challenge n° 60488 du 2009-11-29 au 2009-12-01	4	0	2
Assaut n° 60054 du 2009-11-27 au 2009-11-30	2	1	0
Assaut n° 58133 du 2009-11-20 au 2009-11-23	1	3	0
Assaut n° 54055 du 2009-11-06 au 2009-11-09	1	2	0
Assaut n° 52909 du 2009-11-02 au 2009-11-05	1	3	0
Assaut n° 52009 du 2009-10-30 au 2009-11-01	1	3	0
Assaut n° 51142 du 2009-10-27 au 2009-10-30	1	1	0
Assaut n° 49641 du 2009-10-23 au 2009-10-27	1	2	0
Assaut n° 49168 du 2009-10-21 au 2009-10-23	1	3	0
Assaut n° 48670 du 2009-10-19 au 2009-10-21	1	3	0
Assaut n° 47968 du 2009-10-17 au 2009-10-19	1	3	0
Assaut n° 46555 du 2009-10-13 au 2009-10-17	1	2	0
Assaut n° 44919 du 2009-10-08 au 2009-10-13	1	3	0

FIGURE 5.6 – Résultats obtenus grâce au *build harpies*.

Seules les parties de type assaut correspondent à des tests du *build harpies*. Les parties challenges ont été jouées contre un autre type de joueurs, avec des unités différentes que nous n'avons pas étudiées. La couleur or correspond à la première place, argent à la seconde.

Sur les 14 parties effectuées en mode assaut, c'est à dire contre trois nouveaux joueurs humains pour chaque partie, nous en avons gagné 12, et terminé deux fois en seconde position, ce qui tend à confirmer l'efficacité du *build harpies*. Bien sûr, d'autres facteurs entrent en considération : les joueurs peuvent communiquer et décider de s'allier contre un seul joueur, ou la configuration de départ des îles, aléatoire, peut favoriser un joueur plus que les autres. Mais le gameplay de Skywar pourrait être équilibré en rallongeant par exemple légèrement la durée de fabrication des harpies. Notre application a bel et bien permis de mettre à jour une erreur d'équilibrage du gameplay.

Cette expérience démontre qu'avec un faible investissement, il est possible de découvrir certaines faiblesses d'équilibrage du gameplay développé par un studio reconnu. Ce type de déséquilibre peut nuire profondément au gameplay du jeu, en éliminant toute forme de challenge pour le joueur qui a découvert la super stratégie.

5.2.3 Pacman

Nous présentons ici une troisième expérience, au cours de laquelle nous avons étudié le gameplay du célèbre jeu Pacman (Namco) en développant un joueur synthétique chargé de diriger Pacman. Dans cette expérience, nous avons analysé une version simplifiée du gameplay d'origine, de manière à pouvoir interpréter plus facilement les résultats. Pacman n'est pourchassé que par un seul fantôme : *Blinky*, dont l'algorithme consiste à trouver le plus court chemin pour intercepter Pacman. Dans la version originale, quatre fantômes aux comportements différents peuvent intercepter Pacman, qui peut utiliser des jetons de puissance pour renverser la situation et être capable de manger les fantômes. Le gameplay original alterne également des phases de poursuites avec des phases de *dispersion* au cours desquelles les fantômes regagnent chacun un point précis du niveau et laissent Pacman évoluer assez librement. Dans notre version simplifiée, l'objectif du joueur reste de manger le plus possible de pastilles, dispersées dans tout le niveau.

Pacman et Blinky utilisent tous les deux une recherche de chemin basée sur l'algorithme A^* . Le graphe des chemins les plus courts est construit et calculé au départ, de manière à limiter les phases de calcul à l'exécution. Le joueur synthétique a une vision partielle de l'univers du jeu, construite à partir de cinq paramètres. Le premier paramètre donne la direction de la pastille la plus proche, et puisque Pacman ne peut aller que dans quatre directions, ce paramètre n'a que quatre valeurs possibles. Les quatre autres paramètres décrivent les quatre directions que peut emprunter Pacman. Pour chaque direction, Pacman sait s'il est face à un mur (0), face à un fantôme se trouvant à une (1) ou deux (2) cases de distance, face à un chemin libre de fantômes de moins de 18 cases de long (3), ou face à un chemin libre de fantômes de plus de 18 cases de long (4). Nous pensons que ces informations fournissent une bonne abstraction de la situation telle que perçue par le joueur.

L'**Intelligence Artificielle**² du joueur synthétique a été implantée sous la forme d'un Processus de Decision Markovien³, construit grâce à l'algorithme d'apprentissage $Q(\lambda)$, avec traces d'éligibilité [Sutton 98]. Les paramètres de $Q(\lambda)$ employés sont les suivants : $\alpha = 0.1$, $\gamma = 0.95$, $\lambda = 0.90$. Nous avons équilibré les phases d'exploration et d'exploitation à l'aide d'une sélection d'action ϵ -greedy, avec $\epsilon = 0.05$.

Nous avons évalué la difficulté de ce gameplay vis à vis d'un paramètre donné : la vitesse du joueur. Nous considérons que le niveau de difficulté peut être évalué à partir du nombre de pastilles mangées par Pacman. Nous avons choisi d'étudier l'impact de la vitesse de Pacman sur la difficulté du jeu car la vitesse relative de Pacman et des fantômes est un paramètre utilisé dans la version originale du jeu pour manipuler sa difficulté [Pittman 09]. Tous les 14 cycles, Blinky change sa position, en se déplaçant d'une case dans la direction choisie.

Nous avons évalué la performance du joueur synthétique, et donc la difficulté du gameplay, sous différentes configurations. Nous avons réalisé six expériences, avec une vitesse de Pacman

3. MDP, Markov Decision Process

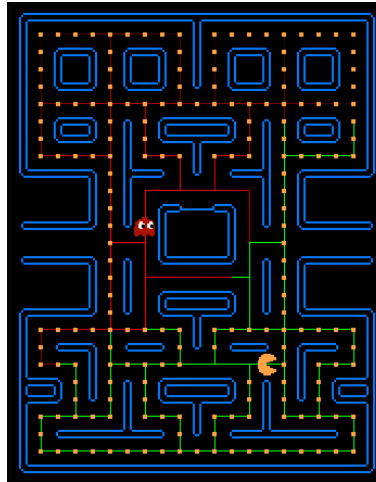


FIGURE 5.7 – Pacman chassé par Blinky.

variant de 0 (Pacman et Blinky se déplacent tous les 14 cycles) à 7 (Blinky se déplace tous les 14 cycles mais Pacman se déplace tous les 7 cycles). Nous avons laissé le joueur synthétique développer sa politique pendant 50 000 jeux, Pacman ayant trois vies par jeu.

La figure 5.8 présente une vue synthétique des résultats obtenus. Nous pouvons déduire de ces résultats que lorsque nous modifions la vitesse de Pacman, la difficulté du jeu n'évolue pas de manière linéaire. Il y'a beaucoup moins de différence de score entre les vitesses 0 et 5 qu'entre les vitesses 5 et 7. Cette analyse pourrait être utile pour un game designer chargé d'équilibrer la difficulté d'un tel jeu. Ces résultats lui révèlent que lorsque la vitesse de Pacman approche de deux fois celle de Blinky, le jeu devient facile beaucoup plus rapidement. Entre 0 et 5, la difficulté évolue de manière linéaire.

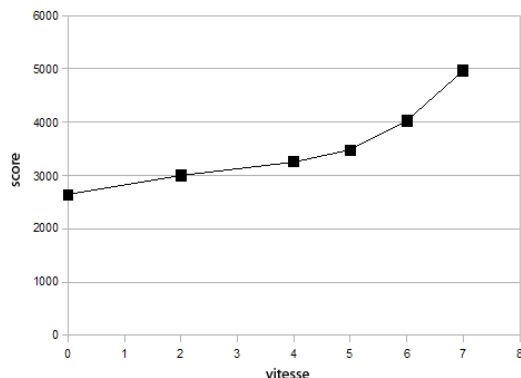


FIGURE 5.8 – Score de Pacman à différentes vitesses

Le score représenté et le score moyen obtenu par le joueur synthétique sur les 5000 dernières parties jouées à une vitesse donnée. A chaque pastille récupérée, le joueur marque 10 points.

5.3 Limitations d'un joueur synthétique

Les trois expériences précédentes montrent qu'il peut être utile de remplacer le joueur par une intelligence artificielle pour analyser un gameplay. Ces trois expériences ont confirmé ou mis en évidence des propriétés de l'équilibrage de gameplays, qui peuvent servir de base à la réflexion d'un game designer. Le développement d'un joueur synthétique demande certes un investissement, mais celui-ci peut être minimisé si le jeu dispose déjà d'un système d'agents intelligents, et donc de l'interface logicielle nécessaire au développement d'un joueur synthétique.

Néanmoins, dans l'optique d'analyser la difficulté d'un gameplay, le joueur synthétique montre rapidement ses limites. En effet, la difficulté du gameplay doit être évaluée pour un joueur humain, qu'une intelligence artificielle n'est pas en mesure de simuler. Les trois phases qui décrivent l'activité du joueur, c'est à dire la *perception*, la *décision* et l'*action*, permettent de mettre en évidence les failles du joueur synthétique.

La phase de perception est par exemple particulièrement complexe à simuler. Un joueur synthétique perçoit l'univers du jeu à partir d'une interface logicielle qui lui permet d'accéder aux structures de données qui décrivent cet univers. A priori, un agent peut donc obtenir une connaissance totale de l'état du jeu, à laquelle un joueur n'a pas forcément accès. Mais même si l'on limite l'accès de l'agent aux données qu'on suppose perceptibles pour le joueur humain, les deux perceptions restent fondamentalement différentes. Le joueur humain perçoit l'univers à partir d'une analyse complexe des signaux visuels et sonores, que nous sommes incapables de reproduire algorithmiquement. Les effets graphiques, le choix des textures, ont

par exemple un impact direct sur la perception de l'univers par le joueur humain, auquel est totalement insensible le joueur synthétique.

La phase de décision d'un humain est elle aussi très difficile à reproduire. Il existe plusieurs théories qui cherchent à décrire le raisonnement humain [Noveck 07]. Les théories qui s'appuient sur une logique de raisonnement innée, comme la théorie de la logique mentale [Braine 90], semblent les plus évidentes à simuler, à partir de l'implantation sous forme algorithmique des différents axiomes qui la composent. Mais les théories basées sur l'expérience, comme celles des schémas pragmatiques, sous-tendent que chaque individu construit ses propres méthodes de raisonnement par l'expérience, ce qui semble bien plus difficile à simuler. De même, le développement d'un agent raisonnant à partir de la construction de modèles mentaux [Johnson-Laird 02] semble particulièrement complexe.

De plus, outre le fait que les modèles fondamentaux du raisonnement humain soient toujours débattus, les nombreux biais de raisonnement aujourd'hui identifiés complexifient encore la modélisation d'un joueur synthétique [Noveck 07]. Les notions de fixité fonctionnelle, d'effet de halo, les biais de confirmation, de disponibilité, d'appariement, sont autant de particularités du raisonnement humain qu'un joueur synthétique devrait savoir simuler.

Encore à titre d'exemple, la théorie des marqueurs somatiques, proposée par Antonio Damasio [Damasio 94], cherche à expliquer la place des émotions dans notre système de raisonnement. Selon cette théorie, les émotions nous servent d'heuristiques, et nous permettent de simplifier nos raisonnements à partir du ressenti lié à certaines solutions envisagées. Ce type de théorie semble également très complexe à simuler pour un agent logiciel.

Pour finir, la phase d'action est elle aussi très différente chez l'agent et le joueur humain. L'agent exécute ses actions au moyen d'un appel logiciel, d'une précision parfaite. Le joueur humain doit interagir au moyen d'un contrôleur, spécifique à chaque plateforme de jeu. Il est par exemple admis qu'un même First Person Shooter réalisé pour console nécessite un équilibrage particulier pour pouvoir être joué sur un PC. Sur un ordinateur, les joueurs utilisent une souris, un dispositif de pointage absolu bien plus précis qu'une manette, qui ne fournit qu'un déplacement relatif du système de visée. Pour être capable de remplacer le joueur par un agent, il faudrait être en mesure de simuler les propriétés des différentes interfaces.

Un joueur synthétique reste donc un outil intéressant dans l'étude d'un gameplay, mais ne remplace que très partiellement la mise en place de véritables playtests. Le développement d'un joueur synthétique capable de simuler le raisonnement humain est aujourd'hui une tâche aussi titanesque qu'inaccessible, tant il nous reste à apprendre le raisonnement humain avant d'espérer parvenir à le simuler. Dans la suite de ce document, nous partirons du principe que nos mesures sont réalisées grâce à des tests réalisés avec de vrais joueurs, notre objectif étant plus de décrire la difficulté d'un gameplay que d'en rechercher les éventuelles failles.

Finalement, ces premières expériences mettent en évidence un autre problème fondamental. Dans chacune des expériences réalisées, nous avons mesuré la difficulté à partir d'une

fonction heuristique, propre à chaque jeu : le nombre de pastilles mangées par Pacman, la vitesse de destruction de la base ennemie pour Skywar ou le temps de résolution du casse tête. Rien ne nous permet de valider ces mesures, elles correspondent à une vision intuitive de la difficulté mais sont aussi recevables à priori que n'importe quelle autre fonction de l'état du jeu. Si nous voulons fournir une mesure correcte de la difficulté, il est avant tout nécessaire d'en fournir un modèle le plus générique et acceptable possible, ainsi que d'établir une méthode de mesure valide sous des hypothèses identifiées. Dans le prochain chapitre, nous proposons donc notre modèle de la difficulté d'un jeu vidéo.

Chapitre 6

Mesure de la difficulté

Sommaire

6.1	Objectif	104
6.2	Les différentes variables	104
6.2.1	L'effort nécessaire	105
6.2.2	Perception	109
6.2.3	Décision	111
6.2.4	Action	114
6.3	Modèle simplifié	115
6.4	Les challenges	117
6.5	Mesure de la performance	118
6.6	Mesure du niveau du joueur	119
6.7	Equation générale de la difficulté	121
6.8	Trace d'évènements	123
6.9	Calcul du niveau du joueur	124
6.10	Calcul de la difficulté	126
6.11	Apprentissage du joueur	128
6.11.1	Correspondance entre capacités locales	128
6.12	Temps et difficulté	129
6.13	Durée d'un challenge et théorie de la fiabilité	130
6.14	Tracer la courbe de difficulté	131
6.15	Synthèse	132

6.1 Objectif

Dans ce chapitre, nous décrivons une méthode de mesure de la courbe de difficulté d'un jeu vidéo. Nous avons mis au point cette mesure en nous imposant la contrainte de n'exploiter qu'un enregistrement du comportement du joueur dans l'univers du jeu. Des méthodes plus invasives, comme l'utilisation de questionnaires directs ou d'un système de mesure de variables physiologiques permettraient d'éclairer plus précisément de nombreuses facettes de la difficulté d'un jeu vidéo. Mais en nous limitant à une mesure indirecte et transparente, c'est à dire sans questionner directement le joueur ni l'équiper d'un appareil de captation particulier, nous proposons un système de mesure plus léger. Un tel système sera exploitable dans le contexte de tests à grande échelle, et en particulier dans le cadre de jeux disponibles sur internet et permettant l'enregistrement du comportement des joueurs sur un serveur de jeu. Les ordinateurs et consoles de salons permettent en effet aujourd'hui d'accéder à de nombreux systèmes de jeu en ligne, ce qui facilite particulièrement la réalisation de ce type de **playtest**¹. En contrepartie, ces tests se déroulent dans un environnement non contrôlé et seul le comportement du joueur dans l'univers virtuel nous est accessible.

Le fait de limiter ainsi notre mesure implique qu'une partie des variables utiles à l'évaluation de la courbe de difficulté ne seront pas observables. Nous pouvons néanmoins définir un certain nombre d'hypothèses qui, si elles sont respectées, permettent de contraindre ces variables non observables, et d'obtenir une mesure correcte de la courbe de difficulté d'un gameplay.

6.2 Les différentes variables

Dans la section 2.1, nous avons défini la difficulté d'un jeu comme la quantité d'effort que le joueur doit investir pour atteindre son objectif. Cette définition donne une bonne intuition de la notion de difficulté. Cependant, afin d'établir une mesure correcte de la difficulté, il est important de détailler précisément les différentes variables qui conditionnent cet effort. Il est ensuite nécessaire d'identifier précisément les différentes relations qu'entretiennent ces variables, ainsi que les variables que nous pouvons effectivement mesurer. Si certaines variables ne nous sont pas accessibles, nous devons fournir un ensemble de règles permettant de les contraindre et d'être ainsi tout de même capable d'effectuer notre mesure. La figure 6.1 présente un modèle complet des différentes variables impliquées dans une mesure de la difficulté, ainsi que leurs relations.

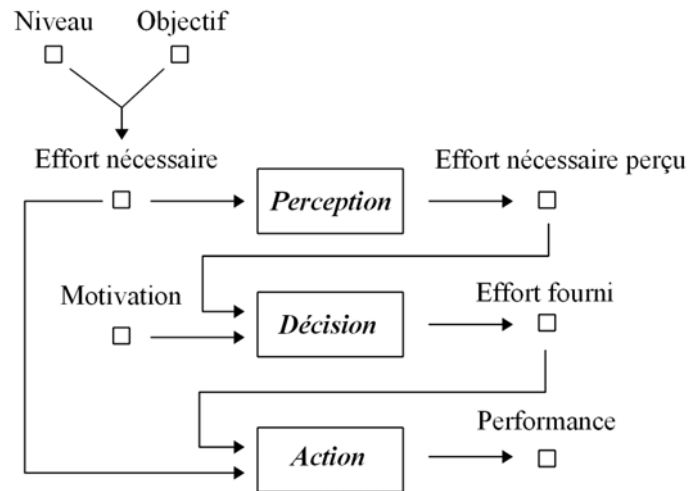


FIGURE 6.1 – Relations entre les variables du modèle de difficulté.

Les rectangles au texte en italique sont des processus, les carrés sont des variables. Cette figure décrit les liens entre les variables de notre modèle de la difficulté d'un jeu vidéo. L'objectif et le niveau du joueur déterminent l'effort nécessaire, c'est à dire la difficulté de la tâche. Dans un premier temps, le joueur *perçoit* l'effort nécessaire. Le joueur va ensuite *décider*, à partir de l'effort nécessaire perçu et de sa motivation, de l'effort qu'il va effectivement investir. Finalement, lorsque le joueur *agit*, le rapport entre l'effort fourni et l'effort nécessaire détermine la performance qu'il va atteindre.

Dans les sections suivantes, nous allons décrire l'ensemble des processus et des variables présentées dans la figure 6.1. Notre objectif consiste à mesurer la variable *effort nécessaire*, c'est à dire la difficulté de la tâche. Cette variable n'est malheureusement pas directement mesurable à partir du comportement du joueur, mais peut être déduite d'autres variables. Nous allons donc identifier les variables que nous sommes capables de mesurer, et les hypothèses sous lesquelles leur mesure nous permet de déduire la difficulté du jeu.

6.2.1 L'effort nécessaire

La première partie du modèle présenté dans la figure 6.1 montre que l'*effort nécessaire* est la résultante d'un *objectif* et du *niveau du joueur*. Dans cette section, nous détaillons cette partie du modèle, isolée dans la figure 6.2.

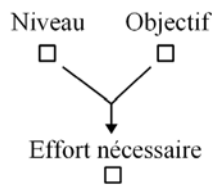


FIGURE 6.2 – Relation entre niveau du joueur et effort nécessaire pour réussir le challenge.

6.2.1.1 Variables impliquées

La difficulté est un effort déployé pour atteindre un objectif. Nous définissons d'abord ce que la variable *objectif* du modèle représente exactement :

- **Objectif** : un objectif est une fonction binaire de l'état du jeu. Pour qu'un objectif soit *atteint*, le joueur doit parvenir à modifier l'état pour satisfaire cette fonction.

L'objectif dont nous souhaitons calculer la difficulté est l'objectif du *challenge* en cours. Un challenge décrit le rapport qu'entretient un joueur avec un objectif particulier du jeu, à un instant donné². L'état d'un challenge permet par exemple de savoir si l'objectif a été atteint ou si le joueur est toujours en train d'essayer de l'atteindre, ou a échoué. Comme le définissent Rollings et Adams, le gameplay d'un jeu est une suite de challenges [Rollings 03]. Chaque challenge est considéré comme *atomique*, c'est à dire que le joueur n'est pas en mesure d'interrompre un challenge, ni d'effectuer plusieurs challenges en parallèle. De cette manière, nous savons que lorsqu'un challenge est actif, le joueur se consacre à l'objectif correspondant, et que son comportement peut être interpréter en fonction de ce contexte précis.

- **Challenge** : un challenge correspond à un objectif que le joueur doit atteindre. L'état d'un challenge décrit l'avancée du joueur vis à vis de cet objectif. Un challenge *actif* est en cours d'exécution par le joueur. Un challenge ne peut ni être interrompu, ni exécuté en parallèle avec un autre challenge.

Nous définissons deux types d'objectifs : l'*objectif du challenge* et les *sous-objectifs*. L'objectif du challenge est l'objectif général que poursuit le joueur, et correspond à la variable objectif de notre modèle. C'est l'effort nécessaire pour atteindre cet objectif que nous voulons évaluer. Mais pour atteindre l'objectif principal, le joueur va poursuivre des sous objectifs, qui lui permettent de s'en rapprocher. Dans un jeu, chaque challenge propose un objectif général, mais les sous objectifs que le joueur va poursuivre sont quant à eux souvent redondants entre challenges, se sont les briques de gameplay que le game designer utilise pour construire les challenges.

2. Voir sections 6.4 et 7.4.2 pour une description plus complète des challenges

- **Objectif du challenge** : objectif que le joueur doit atteindre pour réussir le challenge.
- **Sous-objectif** : objectif intermédiaire et poursuivi de manière répétitive, qui permet au joueur de progresser vers l'objectif du challenge.

Nous pouvons illustrer la différence entre les sous objectifs et l'objectif du challenge par l'exemple d'un jeu de plateforme. Chaque parcours que doit traverser le joueur, c'est à dire chaque challenge, est unique. Mais pour traverser le parcours, le joueur saute de plateformes en plateformes. Il existe un nombre limité de types de sauts que le joueur va effectuer : différentes combinaisons de plateformes stables ou mouvantes, plus ou moins éloignées ou plus ou moins larges. Chaque saut correspond à un sous objectif. Il va poursuivre ces sous objectifs de manière répétitive, et progresser petit à petit vers la fin du niveau, c'est à dire vers l'objectif du challenge.

L'existence de sous objectifs communs à différents challenges est nécessaire à la construction d'un gameplay : si chaque challenge reposait sur des sous objectif entièrement nouveaux, la manière de jouer chaque challenge serait également entièrement nouvelle, et en quelque sorte, le joueur jouerait à chaque fois à un nouveau jeu. Un jeu vidéo repose sur un gameplay principal, exploité au cours de différents challenges. Grâce à la redondance des sous objectifs, le joueur peut répéter des séquences d'actions de nombreuses fois et progresser dans sa maîtrise d'un gameplay.

Pour atteindre l'objectif du challenge, le joueur doit fournir un effort. Il est nécessaire de définir la notion d'effort, et avec elle la variable *effort nécessaire*.

- **Effort nécessaire** : effort que le joueur doit fournir pour atteindre l'objectif du challenge. Cette variable correspond à la difficulté que nous souhaitons mesurer.

Le joueur dispose d'une certaine quantité de ressources. Il dispose par exemple de ressources attentionnelles, qu'il consacre aux tâches de son choix. Il peut également choisir d'être plus ou moins actif cérébralement pour résoudre un problème, et consommer une quantité d'énergie plus ou moins importante. Sans rentrer dans le détail des différentes ressources exploitées pour jouer à un jeu vidéo, on peut simplement poser comme principe que le joueur dispose d'une certaine quantité de ressources, et que cette quantité est bornée.

L'effort peut être ensuite défini comme l'exploitation de ces ressources. Le dictionnaire Larousse définit l'effort de la façon suivante « *Mobilisation volontaire de forces physiques, intellectuelles, morales en vue de résister ou pour vaincre une résistance* ». Plus précisément, nous définirons l'effort du joueur comme le rapport entre la quantité de ressources investie et la quantité de ressources disponibles. Si le joueur investit la totalité des ressources disponibles dans une tâche, son effort est considéré comme maximal. On définit donc ici l'effort comme une valeur relative à chaque sujet : si deux joueurs possèdent par exemple une mémoire de travail de taille différente, mais l'exploitent au maximum, on considère que leur effort est similaire.

De la même manière, l'*effort nécessaire* est le rapport entre les ressources que le joueur doit investir et les ressources dont il dispose. L'effort nécessaire est donc dans l'intervalle $[0, +\infty]$ car un challenge peut demander au joueur d'investir une quantité de ressources bien supérieure à celle dont il dispose.

Nous distinguons l'objectif du challenge des sous objectifs car le premier décrit un objectif général, valide le temps d'un challenge, et les derniers permettent de décrire plus précisément le comportement du joueur. Les mesures liées aux sous objectifs nous permettent ainsi d'établir un modèle du joueur, de calculer son niveau :

- **Niveau du joueur** : effort que doit fournir le joueur pour valider un sous-objectif du challenge. Plus le niveau du joueur est élevé, moins l'effort investi est important.

D'un point de vue général, le niveau du joueur regroupe les savoirs et compétences qui lui permettent de convertir un effort en une certaine performance. Si deux joueurs investissent un effort similaire pour réussir un challenge, ils n'obtiendront pas forcément le même résultat car leur savoir sont différents, leur stratégies de résolutions sont différentes, plus ou moins efficaces, et plus ou moins bien exécutées. Ces savoirs et compétences sont acquis challenge après challenge, lorsque le joueur tente d'atteindre les sous-objectifs, et représentent la maîtrise qu'à le joueur du gameplay.

L'effort nécessaire, c'est à dire la difficulté du challenge, est conditionné à la fois par le niveau du joueur et par l'objectif à atteindre. En effet, la capacité du joueur à atteindre les sous objectifs du challenge détermine l'effort qu'il devra fournir pour atteindre l'objectif général du challenge.

6.2.1.2 Mesurabilité des variables

Le *niveau du joueur* correspond à l'ensemble des savoirs et compétences du joueur, au sujet du jeu que nous analysons. Comme nous l'avons décrit précédemment, certains jeux demandent aux joueurs de découvrir des informations enfouies dans l'univers du jeu : par exemple la position d'un objet caché, ou un chemin dans un labyrinthe. Ces informations peuvent être communiquées au joueur de manière extrinsèque, sous la forme d'un « *walk-through* » par exemple, c'est-à-dire une description des différentes informations nécessaires à la réussite des challenges. Les compétences, quant à elles, sont acquises par la pratique.

Dans un jeu de combat par exemple, le joueur doit à la fois découvrir les coups spéciaux de son personnage et être également capable de les exécuter. La séquence de touches est une information, d'ailleurs bien souvent disponible dans le manuel du jeu ou communiqué au travers des différents tutoriaux. Le fait d'être capable, en jouant, d'exécuter la séquence de touches au bon moment est une compétence, et ne peut être acquise que par l'entraînement.

Cependant, dans notre modèle, nous proposons de ne pas faire de distinction entre savoirs et compétences. Tout comportement du joueur dans le jeu vidéo est soumis à ces deux

aspects de l'apprentissage : le joueur doit d'abord découvrir que l'univers du jeu est manipulable d'une certaine manière, que certaines séquences d'actions sont autorisées et permettent d'obtenir un résultat utile. Ensuite, ce savoir n'est pas forcément suffisant pour garantir le succès du joueur : les contraintes psycho-motrices de manipulation de l'univers du jeu demandent au joueur de pratiquer suffisamment ces comportements pour acquérir le temps de réflexe et la précision nécessaires. Mais dans tous les cas, la somme des savoir et compétences du joueur sert une seule et même utilité : sa capacité à atteindre les sous objectif du challenge, et sera pour nous résumée à ça.

Que ce soit pour mesurer le niveau du joueur ou la difficulté du jeu, il s'agit d'évaluer l'effort nécessaire pour atteindre un objectif. En quelque sorte, le modèle de la difficulté d'un challenge est récursif, c'est à dire que le niveau du joueur pourrait également être décrit par le même schéma, à ceci près qu'il s'agirait d'atteindre un sous objectif au lieu de l'objectif d'un challenge. La figure 6.2.1 montre simplement le lien de causalité entre sous objectif et objectif général : la difficulté à atteindre un objectif général dépend de la difficulté à atteindre les sous objectif dont il dépend. Malgré leur apparente similitude, nous souhaitons cependant distinguer la difficulté à réussir un challenge de celle à atteindre un sous objectif : la généralité et la redondance des sous-objectifs font qu'ils correspondent à des savoir et compétences accumulés par le joueur, la ou la difficulté de l'objectif du challenge est bien plus spécifique et correspond à l'exploitation de ces savoirs et compétences dans un contexte particulier.

Le niveau du joueur ou la difficulté du challenge ne sont pas directement mesurable : seul l'effort effectivement fourni par le joueur pourrait être mesuré, ou déduit de la performance du joueur, c'est à dire du fait qu'il atteigne ou non l'objectif. Cependant ces déductions ne sont valides que sous un certain nombre d'hypothèses. Les sections suivantes présentent les hypothèses nécessaires à la mesure d'un effort nécessaire à partir de la performance obtenue, valables autant pour la mesure du niveau du joueur que pour celle de la difficulté du challenge.

6.2.2 Perception

Le premier processus intervenant dans notre modèle de la difficulté d'un challenge (Figure 6.1) est un processus perceptif. La figure 6.3 isole ce processus et les variables impliquées.

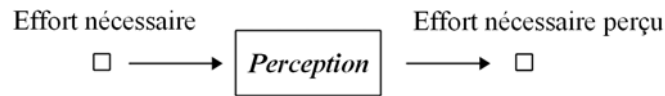


FIGURE 6.3 – Perception de l'effort.

6.2.2.1 Variables impliquées

Nous avons défini l'effort nécessaire dans la section précédente. La variable que fournit le processus perceptif est la suivante :

- **Effort nécessaire perçu** : l'effort que le joueur suppose devoir investir pour atteindre son objectif.

Au travers de ce processus perceptif, le joueur s'appuie sur son expérience du jeu pour déterminer l'effort qu'il doit investir pour atteindre son objectif. S'il a déjà tenté d'atteindre un objectif similaire, il peut se souvenir de l'effort qu'il avait investi et du résultat qu'il avait obtenu, et de cette manière, utiliser son expérience pour évaluer l'effort que la tâche demande.

Cette étape est importante car elle montre que le comportement du joueur n'est pas forcément directement lié à l'effort que demande la tâche, c'est à dire à sa difficulté. Le joueur doit d'abord percevoir cette difficulté pour ajuster son comportement, ce dont nous devons tenir compte dans notre mesure de la difficulté.

6.2.2.2 Mesurabilité des variables

Nous ne sommes pas capables de mesurer directement l'effort perçu par le joueur. Nous pourrions utiliser des questionnaires mais nous sortirions du cadre de notre mesure, qui doit n'être réalisée qu'à partir du comportement du joueur dans le monde du jeu, sans autre source d'information.

Néanmoins, nous pouvons simplifier notre modèle en formulant certaines hypothèses sur la perception de l'effort par le joueur.

6.2.2.3 Hypothèses

Nous supposons que dans un jeu vidéo, et sous certaines hypothèses, le joueur percevra toujours l'effort nécessaire comme maximal. Pour que le joueur puisse supposer qu'il peut atteindre son objectif sans être totalement investi, les deux conditions suivantes doivent être réunies :

- Le joueur doit avoir déjà atteint un objectif équivalent
- Le joueur doit avoir réussi à atteindre cet objectif similaire sans avoir investi un effort maximal

Nous supposons en effet que pour que le joueur prenne le risque de considérer qu'il n'est pas nécessaire d'investir un effort maximal pour réussir, il faut qu'il ait l'expérience, sûrement répétée, d'avoir atteint un objectif similaire en investissant effectivement un effort limité. Nous supposons donc que l'effort perçu par le joueur sera maximal si une des deux conditions suivantes est vérifiée :

- Les objectifs que le joueur doit atteindre sont en apparence suffisamment variés
- Le joueur ne parvient pas à atteindre ses objectifs de manière répétée

Nous formons l'hypothèse que plus le joueur sera soumis à l'échec et plus les objectifs seront variés, moins le joueur sera capable d'inférer l'effort nécessaire. Il n'aura alors d'autre choix que de considérer l'effort nécessaire comme maximal, afin d'assurer sa victoire.

Sous cette hypothèse, nous pouvons considérer que l'effort perçu par le joueur sera considéré comme maximal, et n'est donc plus une variable du modèle, mais une constante.

6.2.3 Décision

Le processus suivant de notre modèle de la difficulté d'un challenge (Figure 6.1) est une prise de décision. A partir de l'effort perçu et de sa motivation, le joueur va déterminer quel effort fournir. Nous avons isolé ce processus et ses variables dans la figure 6.4.

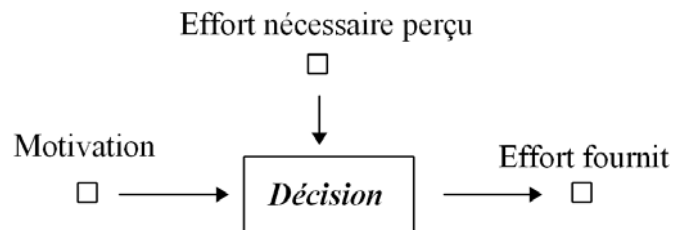


FIGURE 6.4 – Détermination de l'effort fourni.

6.2.3.1 Variables impliquées

Le processus de décision traite tout d'abord l'effort nécessaire perçu. Mais comme nous l'avons défini dans la section suivante, sous certaines hypothèses, cette variable est une constante. L'autre variable traitée par ce processus est la suivante :

- **Motivation** : La motivation du joueur correspond à sa volonté d'atteindre l'objectif qui lui est proposé.

La décision du joueur concerne l'effort qu'il va effectivement fournir :

- **Effort fourni** : L'effort fourni correspond à l'effort que le joueur investit effectivement pour atteindre l'objectif.

Le joueur n'est en effet pas nécessairement attiré par l'objectif qu'on lui propose. En effet, le comportement des joueurs varie en fonction de leurs préférences. Comme l'a par exemple étudié Richard Bartle [Bartle 96], on peut distinguer plusieurs profils motivationnels chez les joueurs de MMORPG. Certains joueurs s'attachent à réussir l'ensemble des challenges que propose le gameplay, d'autres recherchent la compagnie des joueurs, ou veulent avant tout découvrir l'ensemble du monde virtuel. Les motivations des joueurs peuvent influencer l'effort fourni pour deux raisons différentes.

Tout d'abord, le profil motivationnel du joueur peut l'amener à limiter l'effort investi dans la réalisation de la tâche proposée par le gameplay. Si la tâche proposée ne correspond pas aux motivations du joueur, son investissement pourra être moindre. D'une manière encore plus subtile, par exemple, le fait que le profil général du joueur soit basé sur une recherche de la réussite ou l'évitement de l'échec influencera l'effort investi pour atteindre l'objectif [Capa 08].

Ensuite, les préférences du joueur peuvent l'amener à investir une partie de son effort dans d'autres objectifs, à priori non proposés par le gameplay. Dans un First Person Shooter comme Crysis (Crytek - EA) par exemple, le joueur peut emprunter un chemin plus dangereux, plus long, simplement pour pouvoir, par exemple, utiliser un véhicule qu'il affectionne particulièrement. Le monde virtuel de Crysis est très ouvert, et le parcours du joueur y est faiblement contraint, ce qui laisse beaucoup de place aux objectifs personnalisés, non prévisibles par le game designer. Le joueur peut ainsi agrémenter le gameplay de meta-objectifs personnels, qui reflètent plus précisément ses préférences. En conséquence, une partie de l'effort fourni par le joueur sera attribué à ces meta-objectifs ce qui, en contrepartie, limitera l'effort investi dans l'objectif principal du gameplay.

6.2.3.2 Mesurabilité des variables

Nous ne sommes pas en mesure de mesurer l'effort fourni par le joueur. Nous pourrions par exemple tenter de le déduire au moyen d'un dispositif de captation des battements cardiaques et de l'activité électrique des muscles de la face [Capa 08], mais ces techniques sont particulièrement complexes à mettre en place et sortent du cadre de notre modèle, nous souhaitons effectuer nos mesures sans avoir recours à un dispositif de captation particulier.

La motivation du joueur peut quant à elle être évaluée à partir de questionnaires, mais nous souhaitons également nous passer de ce type de mesures, particulièrement difficiles

à mettre en place lorsque les tests sont réalisés hors laboratoire, comme par exemple lors d'une partie jouée sur internet. Les joueurs viennent alors principalement pour jouer et leur motivation à remplir des questionnaires hors environnement contrôlé n'est pas du tout assurée.

Nous devons donc définir sous quelles hypothèses les différentes variables du processus de décision peuvent être négligées.

6.2.3.3 Hypothèses

Nous supposons que si les deux règles suivantes sont valides simultanément, l'impact de la motivation du joueur sur son comportement sera minimisé :

- En recrutant pour le playtest des joueurs non rémunérés, issus d'une communauté de joueurs attirés par le type de gameplay proposé, la motivation des joueurs sera maximisée.
- Plus le gameplay du jeu sera fermé, c'est à dire limitera la capacité du joueur à créer des meta objectifs, plus on pourra considérer que le joueurs investissent leur effort pour atteindre l'objectif principal.

En effet, les joueurs peuvent être recrutés en fonction de leur appartenance à une communauté de joueurs attirés par le type de gameplay proposé, et rémunérés au minimum, de manière à ce que leur motivation intrinsèque soit la plus importante possible. De nombreux jeux proposent par exemple des versions betas dites *fermées*, c'est à dire uniquement accessibles après l'obtention d'une clé de déverrouillage, distribuées en nombre limité sur des sites spécialisés. La seule rétribution consiste à pouvoir jouer gratuitement et avant la sortie officielle du jeu, et garantit donc en partie le niveau de motivation général des joueurs.

Dans le présent modèle, nous ne chercherons également pas à prévoir les préférences du joueur et partons du principe que son objectif est l'objectif principal fourni par le gameplay, et qu'il n'investit pas une partie de son effort pour atteindre d'autres objectifs. Cette hypothèse est une des plus importantes à souligner. Pour reprendre la classification de Richard Bartle, notre modèle s'attache principalement à modéliser le comportement des *achievers*, qui cherchent à atteindre les objectifs principaux du gameplay, le plus efficacement possible. Cette hypothèse implique qu'en l'état, notre modèle aura plus de mal à fournir des résultats précis et pertinents pour des jeux à gameplay très ouvert, dotés d'univers très riches et contraignant faiblement le joueur. Lors de l'interprétation des résultats fournis par notre mesure, le game designer devra tenir compte du fait qu'un gameplay très ouvert peut amener les joueurs à développer des comportements éloignés de ceux d'un *achiever*, compromettant alors la validité des résultats.

Sous cet ensemble d'hypothèses, on peut considérer que l'effort investi par le joueur est similaire à l'effort qu'il suppose devoir investir, et nous pouvons donc négliger la variable de

motivation du joueur. En respectant également les hypothèses de la section précédente, nous pouvons considérer que l'effort nécessaire perçu est maximal et donc que l'effort investi par le joueur est également maximal. L'effort fourni devient donc une constante.

6.2.4 Action

Le dernier processus intervenant dans notre modèle de la difficulté d'un challenge (Figure 6.1) est le passage à l'action du joueur. La figure 6.5 isole ce processus et les variables impliquées.

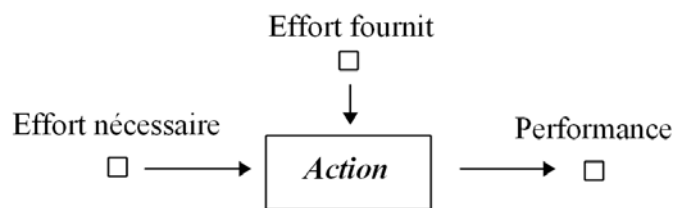


FIGURE 6.5 – Action du joueur.

6.2.4.1 Variables impliquées

L'effort fourni et l'effort nécessaire ont été décrits respectivement section 6.2.3 et section 6.2.1. Lorsque le joueur passe à l'action, ces deux variables conditionnent la performance qu'il va obtenir.

- **Performance** : résultat obtenu par le joueur à la fin du challenge.

Le dictionnaire Larousse donne plusieurs définitions de la performance : « *Résultat obtenu dans l'exécution d'une tâche.* », « *Résultat chiffré (en temps ou en distance) d'un athlète ou d'un cheval à l'issue d'une épreuve.* ». Ces définitions montrent les deux aspects primordiaux d'une mesure de performance : le fait qu'il s'agisse d'un résultat, et qu'il se mesure par rapport à une épreuve, ou une tâche, c'est à dire que la performance se mesure à l'issue d'un intervalle temporel identifié.

Dans le contexte du jeu vidéo, l'intervalle temporel, équivalent d'une tâche ou d'une épreuve, sera le *challenge*. Un challenge se définit par un début et une fin identifiables, et un objectif que le joueur doit atteindre : détruire un ennemi ou trouver un objet par exemple. Il est toujours possible de déterminer, une fois le challenge terminé, si le joueur a réussi ou non à atteindre l'objectif qui lui est associé.

En tenant compte des hypothèses précédentes, l'effort fourni par le joueur sera toujours maximal, c'est à dire qu'il investira toutes les ressources dont il dispose. Mais l'effort nécessaire peut être supérieur à l'effort fourni, auquel cas le joueur n'atteindra pas l'objectif et

sa performance diminuera. Si par contre, il fournit un effort supérieur à l'effort nécessaire, il parviendra à atteindre son objectif.

6.2.4.2 Mesurabilité des variables

La mesure de performance la plus simple, évidente et non discutable est une mesure binaire du résultat du joueur : si le joueur a atteint l'objectif, le résultat est maximal, sinon il est nul. De nombreux jeux proposent des mesures plus précises de la performance du joueur, un score, qui évalue le comportement du joueur durant le challenge. Cette mesure est heuristique, définie à priori par le game designer, comme une formule sur différentes variables du jeu, comme par exemple le nombre de points de vie perdus, le nombre d'ennemis abattus ou de blocs empilés. L'inconvénient d'une telle mesure est la difficulté à prouver sa validité : les choix des variables et de la formule qui les combine sont réalisés à priori et ne reflètent pas forcément la performance du joueur de manière précise, mais fournissent une certaine image de cette performance. Face au même challenge, deux game designer peuvent proposer des heuristiques de score différentes, sans qu'il soit possible de déterminer laquelle est la plus précise. La seule mesure sans équivoque est la mesure binaire, car la réussite ou l'échec du joueur fait partie des règles du jeu, est n'est donc pas discutable.

Le résultat du joueur fait partie intégrante de la notion de jeu, et un jeu vidéo détermine forcément si le joueur a réussi à atteindre l'objectif du challenge. Il est donc tout à fait possible d'introduire une portion de code spécialisée chargée d'enregistrer le résultat du joueur, et donc une mesure de la performance du joueur est tout à fait réalisable.

A partir de l'ensemble des hypothèses que nous avons formulé, nous sommes en mesure de simplifier notre modèle de la difficulté d'un jeu vidéo. Nous présentons ce modèle simplifié dans la section suivante.

6.3 Modèle simplifié

Le modèle simplifié de la difficulté, valide si l'on respecte l'ensemble des hypothèses définies précédemment, est présenté figure 6.6.

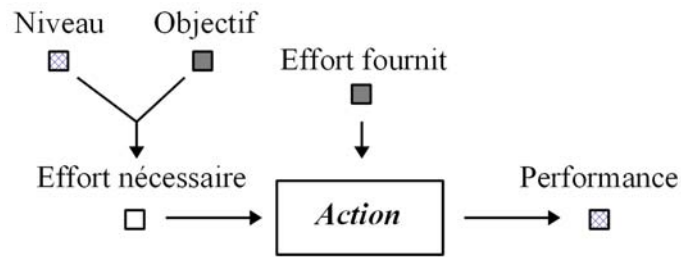


FIGURE 6.6 – Modèle simplifié de difficulté.

Les carrés noirs sont des constantes, les carrés hachurés sont des variables mesurables, le carré blanc représente la variables inférée.

En posant un certain nombre d’hypothèses, nous avons pu ramener le modèle complet de la difficulté à un modèle plus simple. Ce modèle nous permet d’établir les propositions suivantes :

Tout d’abord, on remarque que la performance est une image directe de l’effort nécessaire, c’est à dire de la difficulté du challenge. Les hypothèses de la section précédente nous permettent de considérer l’effort fourni par le joueur comme une constante, et donc la performance du joueur dépend uniquement de l’effort nécessaire. En conséquence, la mesure de la performance du joueur va nous permettre d’avoir une image de la difficulté du challenge, qui n’est pas directement mesurable³. De la même manière, il donc également possible de mesurer le niveau du joueur à partir de la performance du joueur vis à vis des sous objectifs.

La mesure de la performance et du niveau du joueur seront effectuées de manière statistique, à partir de la fréquence de réussite. Ce type de mesure soulève un problème particulier : ni le niveau du joueur ni la difficulté du challenge ne sont des valeurs constantes. Elles évoluent à chaque fois que le joueur joue, c’est à dire pratique et apprend. Cependant, le niveau du joueur sera plus facile à évaluer car les sous objectifs peuvent être mesurés de manière répétitive sur un court intervalle de temps, pendant lequel on peut faire l’hypothèse que le joueur a peu progressé. Le fait de pouvoir évaluer le niveau du joueur nous permet ensuite de construire la statistique de la difficulté du challenge.

En effet, la difficulté du challenge est directement conditionnée par le niveau du joueur. Nous sommes donc capables de sélectionner, parmi toutes les mesures de performance d’un challenge, celles pour lesquelles la difficulté était similaire : il suffit d’avoir au préalable

3. On peut supposer que la relation entre la performance mesurée et l’effort nécessaire est décroissante et monotone : plus l’effort nécessaire va augmenter, plus la performance du joueur va diminuer. Cette relation n’est valable que si le joueur maintient un effort constant, c’est à dire si l’ensemble des hypothèses de la section précédente sont respectées.

mesuré le niveau du joueur et de sélectionner les mesures pour lesquelles ce niveau est équivalent. De cette manière, nous pourrions mesurer la difficulté de manière statistique, en observant la performance de joueurs pour lesquels la difficulté était similaire.

Nous reprenons et détaillons ce principe de mesure dans la section suivante. Nous présentons notre méthode de mesure du niveau du joueur et de sa performance, ainsi que le calcul de la courbe de difficulté d'un jeu vidéo.

6.4 Les challenges

Comme nous l'avons défini précédemment, la performance du joueur se mesure dans un contexte donné : le challenge. Un challenge correspond, dans un jeu vidéo, à un objectif que le joueur peut chercher à atteindre. Un challenge peut ainsi avoir plusieurs états, selon la relation que le joueur entretient avec cet objectif. L'automate qui définit les états et transitions d'un challenge est représenté figure 6.7.

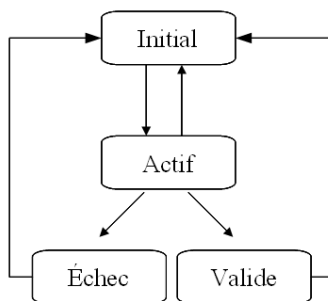


FIGURE 6.7 – Automate d'un challenge.

L'état par défaut d'un challenge est l'état *initial*. Dans cet état, le joueur ne cherche pas encore à atteindre l'objectif du challenge, il est par exemple occupé à atteindre les objectifs des challenges précédents. Si l'objectif du challenge devient un des objectifs du joueur, le challenge passe à l'état *actif*. Lorsque le challenge est à l'état actif, le comportement du joueur est considéré comme lié à ce challenge et ses actions sont donc interprétées en relation avec l'objectif courant. Les deux états *valide* et *échec* correspondent aux deux issues possibles du challenge, c'est à dire lorsque le challenge est *terminé*. Si le joueur arrive à atteindre son objectif, alors le challenge est dans l'état *valide*. Par contre, si le joueur ne parvient pas à atteindre l'objectif alors que le challenge est considéré comme terminé, ce dernier est à l'état *échec*.

Dans un jeu d'aventure par exemple, lorsque le joueur rencontre un groupe d'ennemis, son objectif consiste à les éliminer, et le challenge correspondant à ce groupe d'ennemis devient actif. Toutes les actions du joueur sont alors interprétées en fonction de l'objectif

courant. Chaque coup d'épée en l'air est une action manquée qui nous renseigne par exemple sur ses compétences au combat. Si le joueur perd tous ses points de vie, alors le challenge passe à l'état échec. Si par contre, il parvient à tuer tous ses ennemis sans avoir perdu tous ses points de vie, le challenge passe à l'état valide, le joueur a réussi. On définit donc E l'ensemble des états d'un challenge, ainsi que l'ensemble $E_T \subset E$ des états terminaux d'un challenge :

$$\begin{aligned} E &= \{initial, actif, valide, echec\} \\ E_T &= \{valide, echec\} \end{aligned} \quad (6.1)$$

Etant donné un ensemble de challenges C , on suppose donc qu'il existe une fonction qui donne l'état actuel d'un challenge du jeu :

$$Etat(c) : C \rightarrow E \quad (6.2)$$

Depuis le moment où le joueur va découvrir un jeu vidéo jusqu'à sa dernière partie, celui-ci va se confronter de manière répétitive aux différents challenges. Dans le cadre de la mesure difficulté, nous pouvons conserver l'état terminal de chaque tentative de validation du challenge, numérotée $i \in \mathbb{N}$ pour un joueur noté $j \in J$. Nous définissons ensuite la fonction $EtatHisto(c, i, j) : C \times \mathbb{N} \times J \rightarrow E_T$ qui fournit l'état terminal d'un challenge lors de la i^{eme} tentative de validation du joueur j :

$$EtatHisto(c, i, j) : C \times \mathbb{N} \times J \rightarrow E_T \quad (6.3)$$

Pour simplifier les notations suivantes, on considèrera l'ensemble H de tous les tuples (c, i, j) pour lesquels nous avons enregistré le résultat d'une tentative de validation.

- H est l'ensemble des tuples (c, i, j) pour lesquels $EtatHisto(c, i, j)$ est définie.

Notre objectif consiste à calculer la difficulté d'un challenge. Cette mesure ne doit s'appuyer que sur les seules variables mesurables à partir d'un enregistrement du comportement du joueur dans l'univers virtuel du jeu vidéo : sa performance et son niveau. Les sections suivantes présentent de quelle manière calculer la performance du joueur et son niveau, pour aboutir finalement à l'expression de la difficulté d'un challenge.

6.5 Mesure de la performance

La performance du joueur peut être mesurée sans équivoque si l'on s'appuie sur l'automate de transitions du challenge, présenté à la figure 6.7. En effet, il doit forcément être décidé, pour tout challenge terminé, si le joueur a réussi ou échoué. La notion de résultat valué fait partie de la définition même d'un jeu (voir section 2.1), la version la plus simple d'un tel résultat étant une simple valeur binaire, d'échec ou de réussite.

Afin d'obtenir la mesure la plus générale possible, nous avons choisit de nous appuyer uniquement sur la valeur binaire du résultat du challenge. Bien sûr, de nombreux jeux fournissent des résultats valués plus précis au joueur. Mais ces résultats ne traduisent pas forcément avec précision la performance du joueur. Leur valeur est plus précise mais le rapport entre cette valeur et la performance réelle du joueur est toujours discutable. Pourquoi attribuer 100 points à la destruction de l'ennemi A et 200 points pour celle de l'ennemi B ? Y'a t'il forcément une performance deux fois plus importante pour B que pour A ? En nous appuyant uniquement sur le résultat binaire du challenge, nous obtenons une mesure de performance certes en apparence moins précise mais qui repose sur un fondement stable et applicable à n'importe quel jeu.

La performance du joueur $j \in J$ lors de sa i^{eme} tentative de validation du challenge $c \in C$ peut donc être calculée à partir de l'historique des états terminaux de c , par la fonction $Performance(c, i, j)$:

$$Performance(c, i, j) = \begin{cases} 0 & \text{ssi } EtatHisto(c, i, j) = Echee \\ 1 & \text{ssi } EtatHisto(c, i, j) = Valide \end{cases} \quad (6.4)$$

Nous définissons donc la difficulté d'un challenge comme l'opposé de la performance moyenne qu'obtient ce joueur quand il essaie de valider ce challenge. La performance moyenne du joueur correspond à une estimation de la probabilité qu'a ce joueur de gagner, et l'opposé de la probabilité de gagner est la probabilité de perdre. On peut également définir $Performance(c, j)$ comme une variable aléatoire, dont $Performance(c, i, j)$ est une observation pour une tentative i . Notre définition de la difficulté d'un challenge c pour un joueur j est donc la suivante :

$$D(c, j) = Probabilite\{Performance(c, j) = 0\} \quad (6.5)$$

Dans un jeu de plateformes comme Super Mario, par exemple (Figure 2.2), la performance moyenne d'un joueur pour un parcours de jeu donné, qu'il doit traverser sans tomber dans les pièges qui lui sont tendus, correspond au rapport du nombre d'échecs sur le nombre total de tentatives.

6.6 Mesure du niveau du joueur

Telle que définie dans la section précédente (eq. 6.5), la difficulté n'est pas estimable de manière statistique, car la probabilité que le joueur échoue varie à chaque fois qu'il essaie. En effet, la probabilité qu'a le joueur de gagner augmente au fur et à mesure qu'il essaie de valider le challenge car ses savoirs et compétences évoluent. Afin de pouvoir calculer une statistique de la difficulté, nous devons calculer cette difficulté non pas pour un joueur donné, mais pour un niveau de joueur donné. Si nous effectuons nos mesures sur un échantillon de

joueurs du même niveau, alors la difficulté sera constante, et nous pourrons l'évaluer de manière statistique. Nous devons donc mesurer le niveau du joueur, et l'intégrer à notre mesure de difficulté.

Dans la section 6.2.1, nous avons défini le niveau du joueur de la façon suivante :

- **Niveau du joueur** : effort que doit fournir le joueur pour valider un sous-objectif du challenge. Plus le niveau du joueur est élevé, moins l'effort investi est important.

Nous proposons de modéliser le niveau du joueur comme un ensemble de *capacités*. Chaque capacité est une évaluation niveau du joueur, vis à vis d'un sous-objectif particulier.

- **Capacité** : effort que doit fournir le joueur pour atteindre un sous-objectif particulier.

On remarque la définition d'une capacité est semblable à celle de la difficulté d'un challenge : la difficulté d'un challenge est l'effort nécessaire pour l'objectif du challenge. Evaluer le niveau du joueur revient à calculer la difficulté qu'a le joueur à atteindre certains sous-objectifs. Nous nous heurtons donc au même problème que celui rencontré pour une mesure de la difficulté : le niveau du joueur évolue constamment et empêche a priori d'en calculer une statistique.

Cependant, le niveau du joueur est calculé sur des objectifs à très court terme, sur lesquels la variation du niveau du joueur est minime. Une capacité doit donc être définie sur un but à très court terme, mesurable plusieurs fois pendant la durée du challenge de manière à calculer le niveau du joueur comme une statistique sur ces différentes mesures. Nous supposons donc qu'une mesure du niveau du joueur pendant la durée d'un challenge est stable, et mesurable à partir de la fréquence relative avec laquelle il atteint l'objectif à court terme.

- **Mesure de capacité** : probabilité que le joueur atteigne un objectif à court terme donné au cours d'un challenge donné.

A chaque capacité correspond donc un but que le joueur cherche à atteindre. Si, à partir d'un enregistrement du comportement du joueur dans le monde virtuel, nous sommes capables d'identifier lorsque le joueur *tente* d'atteindre le but relatif à une capacité, et lorsqu'il *réussit* effectivement à l'atteindre, alors nous pouvons calculer la fréquence à laquelle il parvient à atteindre ce but, et donc évaluer le niveau du joueur, pour cette capacité.

Afin de simplifier encore le modèle, nous proposons de mesurer les capacités du joueur de manière discrète. La précision de discrétisation peut être modifiée sans compromettre la validité du modèle. Dans la suite de ce document, nous fixerons trois niveaux pour chaque capacité : $L = \{\text{Mauvais}, \text{Moyen}, \text{Bon}\}$.

Il est également important de noter qu'une mesure de capacité est relative à un challenge donné. En effet, le challenge détermine le contexte dans lequel le joueur évolue, et conditionne donc l'expression de ses capacités. Dans un jeu de combat, le joueur ne parviendra pas à

effectuer certaines suites d'actions face à un ennemi rapide, alors qu'il y parviendra tout à fait face à un ennemi plus lent. La mesure d'une capacité est donc relative à un challenge particulier, même si elle reflète une caractéristique du joueur.

Finalement, même si le niveau du joueur est considéré stable sur un challenge, nous prenons en compte le fait qu'il varie entre chaque challenge. Le niveau du joueur est donc défini pour une capacité, un challenge, mais également pour une tentative de validation de ce challenge $i \in \mathbb{N}$.

Nous proposons une fonction $Niveau(a, c, i, j)$ qui permet de calculer le niveau $l \in L$ d'un joueur $j \in J$ lors de sa i^{eme} tentative de validation du challenge $c \in C$, pour une capacité $a \in A$:

$$Niveau(a, c, i, j) : A \times C \times \mathbb{N} \times J \rightarrow L \quad (6.6)$$

Toujours dans Super Mario, nous pouvons définir la capacité du joueur à sauter sur certain types de plateformes : saut court, saut long, synchronisation avec une plateforme mouvante, bref, identifier les capacités fondamentales dont le joueur fait preuve tout au long du jeu et qui sont testées de manière répétitive. Ensuite, nous étudions le comportement du joueur lorsqu'il tente de franchir un niveau donné. A chaque fois qu'il est confronté à un saut long par exemple, on comptabilise une tentative pour cette capacité. S'il réussit le saut on comptabilise également une réussite, et s'il se blesse, un échec. La fréquence de réussite nous permet ensuite d'estimer le niveau du joueur, et ce au cours d'une tentative de résolution du challenge général qui consiste à terminer le niveau.

6.7 Equation générale de la difficulté

Le modèle simplifié de la difficulté, présenté figure 6.6, peut être ramené au réseau bayésien de la figure 6.8, si nous ne tenons compte que des variables mesurables, c'est à dire la performance du joueur et son niveau. Les variables du réseau correspondent aux capacités du joueur $\{a_0, a_1, \dots, a_n\}$ et ont valeur dans L , et la performance du joueur correspond à la variable *performance*, ayant valeur dans $\{0, 1\}$.

La formule générale de la difficulté peut être déduite de ce réseau bayésien. Nous souhaitons connaître la probabilité que le joueur perde un challenge, étant donné son niveau pour chaque capacité. Pour réaliser ce réseau bayésien, il faudrait connaître la probabilité de perdre en fonction de toutes les combinaisons de valeurs possibles des capacités du joueur. Cependant, nous considérons qu'une première mesure simple et utile serait de calculer la difficulté d'un challenge vis à vis de chacune des capacités du joueur, prise séparément. En récoltant suffisamment de données il serait cependant tout à fait possible de spécifier la table de probabilité conditionnelle complète de la variable *performance*, en prenant en compte l'ensemble de ses parents.

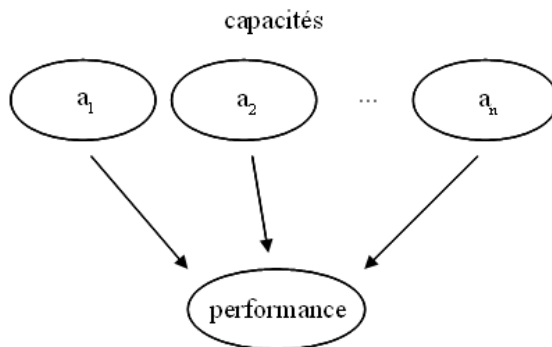


FIGURE 6.8 – Réseau bayésien : causalité entre capacités et performance.

Dans ce cadre, l'équation générale de la difficulté, pour une capacité donnée, est la probabilité de d'échouer, sachant le niveau du joueur pour cette capacité. On définit $Performance(c)$ comme une variable aléatoire dont l'observation pour une tentative de résolution i par un joueur j est donnée par la fonction $Performance(c, i, j)$, et $Niveau(a, c)$ une variable dont $Niveau(a, c, i, j)$ est la valeur pour une tentative de résolution i par un joueur j . L'équation générale de la difficulté est alors la suivante :

$$D(a, c) = Probability\{Performance(c) = 0 | Niveau(a, c)\} \quad (6.7)$$

On remarque que dans cette équation, la probabilité est calculée pour tout joueur $j \in J$ et tentative $i \in \mathbb{N}$ de l'historique. En effet, cette équation présente une statistique générale, calculée sur l'ensemble des joueurs. Nous ne souhaitons pas calculer la difficulté pour un joueur particulier ou un nombre d'essais donnés, mais uniquement par rapport à la maîtrise d'une capacité. La statistique décrite par l'équation précédente est donc calculée pour toutes les tentatives de tous les joueurs de notre échantillon, pour le challenge et la capacité concernés.

Toujours dans Super Mario, par exemple, calculer la difficulté d'un niveau particulier selon l'équation 6.7 revient à prendre toutes les tentatives à terminer un niveau particulier, pour tous les joueurs ayant participé au test. On trie ensuite toutes ces tentatives en fonction du niveau de jeu dont on fait preuve ces joueurs, pour une capacité donnée. Par exemple, tous les joueurs qui ont réussi plus de quatre vingt pourcents des sauts sur une plateforme mouvante. On obtient donc un sous-échantillon de tentatives de résolutions du challenge ou le niveau du joueur est équivalent, au moins pour la capacité de sauter sur une plateforme mouvante. En calculant la fréquence relative d'échec, on obtient la difficulté de ce niveau de Super Mario, pour les joueurs qui maîtrisent le saut sur plateformes mouvantes.

Le fait de formuler la difficulté d'un challenge en fonction d'une capacité a plusieurs

avantages :

- Définir la difficulté d'un challenge pour un niveau de joueur donné nous permet de mettre en évidence et de mesurer la causalité entre le niveau de maîtrise du gameplay par le joueur et la difficulté d'un challenge particulier
- Si notre mesure était basée uniquement sur un échantillon de joueurs, sans tenir compte de leur niveau, nous devrions être en mesure de construire un échantillon de joueur représentatif de la population de joueurs concernée, aux niveaux variés et correctement répartis. Le fait de calculer le niveau de chaque joueur permet de connaître le type de joueur pour lesquels la difficulté a été calculée, et donc le domaine de validité de notre mesure.

Cette formulation a également un inconvénient :

- Définir la difficulté d'un challenge pour un niveau de joueur défini par une seule capacité ne permet pas d'obtenir une véritable indépendance des observations. En effet, plus le nombre de capacités profondément impliquées dans la résolution du challenge augmente, plus ce modèle unidimensionnel doit montrer ses limites. Cependant, ce modèle peut en pratique être tout à fait étendu à plusieurs dimensions, il suffit de spécifier la partie conditionnelle de la probabilité d'échouer comme la conjonction de plusieurs niveaux. Cependant, il faudra également augmenter le nombre d'observations pour pouvoir calculer ce modèle plus juste. Notre modèle à une capacité est donc d'autant plus valable que la capacité choisie est une capacité fondamentale, représentative du niveau du joueur.

Les sections suivantes décrivent le calcul du niveau d'un joueur j pour une capacité a , un challenge c et une tentative i donnés, $Niveau(a, c, i, j)$, ainsi que le calcul de la difficulté d'un challenge en fonction de cette capacité $D(a, c)$. Nous souhaitons calculer la difficulté d'un challenge de manière statistique, à partir d'un enregistrement de son comportement dans l'univers du jeu. La première étape du calcul, décrite dans la section suivante, consiste à construire cet enregistrement.

6.8 Trace d'évènements

Notre objectif consiste à mesurer le niveau du joueur et sa performance à partir d'un enregistrement de son comportement dans le monde virtuel. Il nous est impossible d'enregistrer tous les changements d'états de l'univers virtuel : le nombre de variables est bien trop important. Il est cependant possible de définir une trace des évènements significatifs pour notre mesure, par exemple tous ceux liés aux changements d'états des challenges ainsi que ceux décrivant le comportement du joueur vis-à-vis des capacités que nous souhaitons étudier. Nous introduisons donc les notations suivantes :

- un évènement de jeu, noté $ge_{a,c} \in Ge_{a,c}$, est un changement important dans l'état du jeu, en relation avec la capacité a que nous souhaitons mesurer ou avec le changement d'état du challenge c pour lequel nous effectuons la mesure.
- une trace $t_{a,c} \in T_{a,c}$ est une suite d'évènements de jeu $ge_{a,c}$ enregistrés entre l'instant où le challenge c devient actif et l'instant où il atteint un l'état échec ou valide, pour une capacité a donnée.
- la fonction $resultat(t_{a,c}) : T_{a,c} \rightarrow \{valide, echec\}$ détermine si, à la fin d'une trace, le joueur a réussi ou échoué à atteindre l'objectif lié au challenge c .
- au cours de tests, un ensemble de traces d'évènements de jeu sera enregistré pour chaque challenge c . Il existe une fonction qui fournit, pour un challenge c , une capacité a , un joueur j , la i^{me} trace $t_{a,c}$ enregistrée : $trace(a, c, i, j) : A \times C \times \mathbb{N} \times J \rightarrow T_{a,c}$

6.9 Calcul du niveau du joueur

La trace d'évènements de jeu, définie dans la section précédente, va nous permettre d'évaluer les capacités du joueur. Comme expliqué dans la section 6.6, une capacité du joueur correspond à la facilité avec laquelle il est capable d'atteindre un but à court terme, avec l'objectif global de valider le challenge. Pour atteindre cet objectif global, le joueur va exploiter sa capacité, c'est à dire chercher à atteindre de manière répétitive le but à court terme qui lui correspond.

Nous avons défini dans la section 6.6 l'ensemble des niveaux du joueur pour une capacité, $L = \{Mauvais, Moyen, Bon\}$, et le niveau du joueur pour une capacité, $Niveau(a, c, i, j) : A \times C \times \mathbb{N} \times J \rightarrow L$. Le reste de cette section explique le calcul de $Niveau(a, c, i, j)$ à partir d'une trace d'évènements du jeu $t_{a,c} \in T_{a,c}$.

Nous définissons les ensembles et fonctions suivantes, nécessaire au calcul de $Niveau(a, c, i, j)$:

- chaque capacité $a \in A$ que le joueur peut utiliser dans un challenge $c \in C$ doit correspondre à un but à court terme $b \in B$.
- un but $b \in B : Ge_{a,c} \rightarrow \{true, false\}$ est une fonction qui détermine, à partir d'un évènement de la trace, si le sous objectif est atteint.
- il existe une fonction $but(a, c) : A \times C \rightarrow B$, définie de manière statique avec l'assistance du game designer, qui fournit le but associé à une capacité $a \in A$ et à un challenge $c \in C$.
- nous devons être capables de compter, dans une trace d'évènements de jeu $t_{a,c}$, le nombre de fois que le joueur a tenté d'atteindre le but $but(a, c)$, grâce à la fonction $nbEssais(t_{a,c}, a, c)$.
- nous devons être capables de compter, dans une trace d'évènements de jeu $t_{a,c}$, le

nombre de fois que le joueur a *réussi* à atteindre le but $but(a, c)$, grâce à la fonction $nbReussit(t_{a,c}, a, c)$.

Si nous savons combien de fois le joueur a tenté et réussi à atteindre le but à court terme associé à une capacité, alors nous pouvons estimer la probabilité qu'il a eu d'atteindre ce but :

$$ProbaBut(t_{a,c}, a, c) = \frac{nbReussit(t_{a,c}, a, c)}{nbEssais(t_{a,c}, a, c)} \quad (6.8)$$

Cette probabilité, $ProbaBut(t, a, c)$, permet d'évaluer le niveau du joueur pour cette capacité, dans le contexte d'un challenge particulier. Comme précédemment, nous ne souhaitons pas conserver une valeur continue du niveau du joueur mais attribuer au joueur un niveau parmi $L = \{Mauvais, Moyen, Bon\}$. La correspondance entre valeurs discrètes et continues peut par exemple être calculée de la façon suivante :

$$Niveau(a, c, i, j) = \begin{cases} mauvais & \text{iff } 0 \leq probaBut(trace(a, c, i, j), a) < 0.3 \\ moyen & \text{iff } 0.3 \leq probaBut(trace(a, c, i, j), a) < 0.6 \\ bon & \text{iff } 0.6 \leq probaBut(trace(a, c, i, j), a) \leq 1 \end{cases} \quad (6.9)$$

L'algorithme suivant précise l'ensemble du calcul de $Niveau(a, c, i, j)$.

```

1 fonction Niveau(Ability a, Challenge c, Index i, Joueur j)
2     return Level
3 {
4     fonction b = but(a, c); //Le but à court terme
5     var nbreussit = 0; //Compteur
6     var nbttotal = 0; //Compteur
7     var t = trace(a, c, i, j); //La trace correspondante
8
9     pour chaque sous-trace ti de t où le joueur tente d atteindre b
10    {
11        //une tentative de plus
12        nbttotal++;
13
14        pour chaque evenement ej de la trace ti
15        {
16            //Cette condition dépend de chaque couple (a, c)
17            si b(ej) = true
18                nbreussit++;
19        }
20    }

```

```

21
22 // Probabilité d'atteindre le but à court terme b
23 var p = nbreussit / nbttotal;
24
25 // Discrétisation
26 var level = "mauvais";
27 if(p > 0.3)
28     level = "moyen";
29 if(p > 0.6)
30     level = "bon";
31
32 return level;
33 }

```

Il existe plusieurs méthodes pour savoir si le joueur a réussi à atteindre un but à court terme donné (ligne 17 de l'algorithme précédent). Le plus souvent, un événement de jeu vient confirmer la réussite du joueur. Si on mesure la capacité du joueur à viser correctement, la collision entre le projectile et l'ennemi est un événement qui valide le fait que le joueur a atteint le sous but.

Nous avons décrit le calcul du niveau du joueur à partir d'une trace des événements du jeu. Nous pouvons désormais calculer l'équation générale de difficulté $D(a, c)$ (eq. 6.7). Nous présentons ce calcul dans la section suivante.

6.10 Calcul de la difficulté

Nous allons estimer statistiquement la fonction $D(a, c)$, à partir de l'ensemble de traces $T_{a,c}$ enregistrées. L'ensemble $T_{a,c}$ est partitionné en sous ensembles $T_{a,c}^{mauvais}$, $T_{a,c}^{moyen}$, $T_{a,c}^{bon}$, à l'aide de la fonction $Niveau(a, c, i, j)$:

- $T_{a,c}^{mauvais}$ est l'ensemble des traces $t_{a,c} \in T_{a,c}$ d'index i et de joueur j telles que $Niveau(a, c, i, j) = mauvais$
- $T_{a,c}^{moyen}$ est l'ensemble des traces $t_{a,c} \in T_{a,c}$ d'index i et de joueur j telles que $Niveau(a, c, i, j) = moyen$
- $T_{a,c}^{bon}$ est l'ensemble des traces $t_{a,c} \in T_{a,c}$ d'index i et de joueur j telles que $Niveau(a, c, i, j) = bon$.

Cette partition regroupe les traces en sous ensembles selon le niveau du joueur pour une capacité et un challenge donnés. Chaque niveau correspond à une valeur possible de la partie conditionnelle de la fonction $D(a, c)$. L'étape suivante consiste à calculer chacune des probabilités d'échec, pour chaque niveau du joueur.

À ce stade, nous pouvons déterminer, pour chaque trace d'événements, si le joueur a réussi à valider le challenge. L'événement correspondant fait partie de la trace et la fonction

$Resultat(t_{a,c})$ se charge de rechercher cet évènement et de nous indiquer si le joueur a gagné ou perdu. Il nous est donc possible de compter, dans chaque sous ensemble de la partition de $T_{a,c}$ définie précédemment, le nombre de traces où le joueur a perdu. La fonction chargée de compter ces traces dans un sous ensemble $T' \in P(T_{a,c})$ sera nommée $nbPerdu(T')$: $P(T_{a,c}) \rightarrow \mathbb{N}$.

A partir du nombre de traces ayant aboutit à un échec du joueur, nous sommes en mesure d'estimer la probabilité qu'a le joueur de perdre pour chaque niveau, étant donné une compétence et un challenge, c'est à dire pour chaque sous ensemble $T' \in \{T_{a,c}^{mauvais}, T_{a,c}^{moyen}, T_{a,c}^{bon}\}$.

$$ProbaEchec(T') = \frac{nbPerdu(T')}{card(T')} \quad (6.10)$$

Nous sommes donc capables de construire la table de probabilité conditionnelle de l'équation générale de la difficulté (eq. 6.7), pour une capacité a et un challenge c . Nous calculons, pour tout $T' \in \{T_{a,c}^{mauvais}, T_{a,c}^{moyen}, T_{a,c}^{bon}\}$ la probabilité d'échec $ProbaEchec(T')$. Ces valeurs correspondent à la probabilité qu'a un joueur de perdre étant donné son niveau pour une capacité donnée, probabilité conditionnelle exprimée par l'équation générale de difficulté, donnée pour rappel ci-après :

$$D(a, c) = Probability\{Performance(c) = 0 | Niveau(a, c)\} \quad (6.11)$$

L'algorithme suivant montre le calcul de l'équation générale de difficulté.

```

1 fonction Difficulte(Challenge c, Ability a, Level l)
2     return float
3 {
4     var nbEchec = 0; //Compte les echecs
5     var nbTotal = 0; //Total counter
6
7     //On récupère les traces de ce challenge et cette capacité
8     //pour tous les joueurs et indices temporels
9     pour tout joueur j de J
10    {
11        pour toute trace i enregistrée pour j
12        {
13            var t = traces(a,c,i,j);
14            si Niveau(a,c,i,j) == l
15            {
16                si Resultat(t) == echec
17                    nbEchec++;
18
19                nbTotal++;

```



```

20     }
21   }
22 }
23
24   return nbEchec/nbTotal;
25 }
```

6.11 Apprentissage du joueur

Les sections précédentes ont permis de décrire le calcul de la difficulté d'un challenge, étant donné le niveau du joueur. Mais comme nous l'avons suggéré précédemment, le niveau du joueur évolue en fonction du temps : ses savoirs et compétences augmentent, au fur et à mesure qu'il explore l'univers du jeu. Une mesure importante et particulièrement utile pour tracer la courbe de difficulté consiste à être capable d'estimer la progression du joueur, c'est à dire de quelle manière, pour une capacité donnée, un joueur va pouvoir changer de niveau. Les sections suivantes décrivent ce calcul.

6.11.1 Correspondance entre capacités locales

Les capacités définies dans les sections précédentes sont des capacités locales, dans le sens où les scores mesurés ne sont valables que dans le contexte d'un challenge donné. Par ailleurs, si un jeu propose de nombreux challenges au joueur, le nombre de capacités impliquées est lui bien plus limité. La précision avec laquelle un joueur vise est par exemple commune à l'ensemble des challenges de la plupart des First Person Shooter.

On peut ainsi considérer qu'un sous-ensemble des challenges proposés dans un jeu vidéo exploite une même capacité a donnée. Pour ce sous-ensemble particulier de challenges, il serait intéressant d'être capable de déterminer une correspondance entre les différents niveaux mesurés pour la capacité a . En effet, même si le contexte varie, on peut supposer qu'un joueur capable d'une grande précision au cours d'un challenge aura beaucoup plus de chances d'être également parmi les joueur les plus précis dans un autre contexte : le joueur transfère ses savoirs et compétences d'un contexte à l'autre et donc les niveaux observés dans deux challenges différents doivent être liés.

Nous définissons une fonction qui, pour deux challenges successifs c et c' et un niveau l d'une capacité a dans le contexte du challenge c , fournit le niveau correspondant dans le challenge c' :

$$\text{CorrespNiveau}(l, a, c, c') : L \times A \times C \times C \rightarrow L. \quad (6.12)$$

Nous pouvons définir un modèle probabiliste qui saisisse la dynamique des transitions entre niveau pour une capacité et un couple de challenges donnés. Un tel modèle peut être établi en observant, parmi toutes les traces enregistrées, la fréquence relative de changement de niveau d'une capacité a pour tous les couples de challenges (c, c') joués de manière successive.

La fonction *CorrespNiveau* est une fonction partielle, car tous les challenges d'un jeu ne seront pas joués de manière successive. Comme nous le verrons lorsque nous établirons un modèle du scénario d'un jeu vidéo, les challenges sont partiellement ordonnés, et parmi tous les couples de $C \times C$, seuls quelques uns sont des couples valides, que le joueur pourra effectivement jouer de manière successive. Cependant, les couples pour lesquels *CorrespNiveau* n'est pas défini sont des couples pour lesquels nous n'effectuerons pas de requête, car si aucun joueur ne joue jamais le couple (c, c') alors nous ne chercherons pas non plus à prévoir le niveau d'un joueur pour ce couple.

L'intérêt d'une telle fonction consiste à pouvoir prédire un parcours possible du joueur, à partir d'un challenge de départ et d'un niveau donnés. L'équation générale de difficulté (eq. 6.7) nous permet de connaître la probabilité qu'un joueur a d'échouer à un challenge donné, sachant son niveau. Si nous sommes en mesure de connaître le challenge suivant auquel le joueur sera soumis, à partir d'un modèle du scénario du jeu par exemple, la fonction que nous venons de présenter permettra de prévoir son niveau pour ce challenge et d'estimer ensuite sa probabilité d'échec. Nous serons donc en mesure d'estimer un parcours probable de ce joueur.

6.12 Temps et difficulté

Afin de tracer la courbe de difficulté d'un jeu, nous devons être en mesure de prévoir la durée d'un challenge. En effet, la courbe de difficulté se décrit dans le temps : le game designer doit être en mesure de connaître la difficulté d'un challenge, autant que de prévoir la durée pendant laquelle le joueur sera soumis à ce niveau de difficulté. Certains challenges peuvent être particulièrement longs, comme un combat contre l'ennemi le plus puissant du niveau, ou très courts, comme lorsque le joueur doit venir à bout d'un petit groupe d'ennemis. La courbe de difficulté permettra de visualiser le rythme de la difficulté du jeu, et pour être en mesure de la tracer, nous devons pouvoir évaluer la durée des différents challenges.

Dans l'optique d'une mesure générale de la difficulté, nous ne pouvons pas déterminer l'évolution de la difficulté *pendant* la durée du challenge. En effet, nous mesurons uniquement le résultat que le joueur obtient à la fin du challenge, qui représente la concrétisation de l'effort fourni pendant toute sa durée. Cet effort a pu varier au cours du challenge, le joueur ayant pu alterner entre différentes tâches : trouver tout d'abord le bon chemin pour se débarrasser ensuite d'un ennemi par exemple. Une manière de comprendre plus finement la

difficulté d'un tel challenge consisterait à le découper en deux challenges successifs et à en évaluer la difficulté séparément.

Nous représentons donc la difficulté d'un challenge comme une courbe plate, d'équation $y = D(a, c)$, sur un segment de droite d'une longueur égale à la durée moyenne observée pour ce challenge. Nous définissons pour cela la fonction $Duree(c, i, j)$, qui fournit la durée d'un challenge pour la i^{eme} tentative d'un joueur i , et $DureeMoyenne(c)$ qui fournit la durée moyenne d'un challenge, calculée sur l'ensemble des traces enregistrées pour ce challenge.

$$Duree(c, i, j) : C \times \mathbb{N} \times J \rightarrow \mathbb{R}. \quad (6.13)$$

$$DureeMoyenne(c) : C \rightarrow \mathbb{R} = \sum_{i,j} \frac{Duree(c, i, j)}{i * j}. \quad (6.14)$$

Néanmoins, il peut être particulièrement intéressant, pour un game designer, d'étudier plus finement la durée d'un challenge qu'à partir d'une simple moyenne. La durée d'un challenge est un élément très important du game design, car la difficulté rythme la progression du joueur. Nous pouvons donc lui fournir une vision complémentaire et plus précise de cette durée, en fournissant sa distribution de probabilité.

6.13 Durée d'un challenge et théorie de la fiabilité

Nous nous inspirons pour cela des théories de la fiabilité, qui décrivent, dans le temps, la probabilité qu'une panne survienne sur un dispositif donné. Nous pouvons tracer le même type de courbe pour un challenge, en remplaçant l'évènement *une panne est survenue* par l'évènement *réussite du joueur* ou *échec du joueur*. Nous sommes en effet capables, en considérant la variable aléatoire $Duree(c)$ dont $Duree(c, i, j)$ est une observation pour une trace donnée, d'évaluer la fonction suivante :

$$D(a, c, t) : Probability\{Performance(c) = 0 \wedge Duree(c) < t | Niveau(a, c)\}. \quad (6.15)$$

$D(a, c, t)$ fournit la probabilité, étant donné le niveau du joueur pour une capacité a , qu'il échoue au challenge c avant une durée t . Tracer cette courbe permet au game designer de comprendre si les joueurs échouent plus ou moins rapidement au challenge, et lui fournit une vision plus précise de la durée du challenge. On remarque que notre définition de la difficulté d'un challenge, $D(a, c)$ correspond à la limite, lorsque t tends vers l'infini, de $D(a, c, t)$.

6.14 Tracer la courbe de difficulté

Etant donné les différentes fonction définies dans les sections précédentes, nous sommes en mesure de tracer la courbe de difficulté d'un jeu, à partir d'une requête formulée par le game designer et en partant du principe que nous disposons d'un modèle du scénario du jeu, c'est à dire que nous sommes en mesure de connaître, à partir d'un challenge et de son état terminal, le challenge suivant que le jeu va proposer au joueur. Ce modèle fera l'objet d'une définition complète au chapitre 7. Nous pouvons noter cette fonction de la manière suivante :

$$\text{ChallengeSuivant}(c, e) : C \times E_T \rightarrow C. \quad (6.16)$$

Pour tracer la courbe de difficulté, nous utilisons les différents modèles probabilistes définis précédemment pour simuler un parcours possible du joueur et mesurer sa difficulté. Etant donné un challenge de départ et le niveau du joueur défini pour chaque capacité, nous pouvons déterminer la probabilité que le joueur a d'échouer, c'est à dire la difficulté du challenge, ainsi que sa durée probable (eq. 6.7 et eq. 6.13). Nous pouvons ainsi tracer une première portion de la courbe de difficulté, pour le niveau de difficulté et la durée que nous venons de calculer.

Il suffit ensuite d'effectuer un tirage en respectant la probabilité d'échec du joueur pour déterminer le résultat du challenge. En fonction de ce résultat, le modèle de scénario doit nous fournir le challenge suivant. Nous sommes également en mesure de déduire le niveau du joueur pour ce nouveau challenge (eq 6.12). A partir de ce nouveau modèle du joueur et du nouveau challenge, nous pouvons répéter la procédure et tracer la portion suivante de la courbe de difficulté.

Le game designer peut répéter sa requête plusieurs fois, et obtenir à chaque fois une courbe de difficulté respectant les probabilités observées lors du playtest, et visualiser ainsi les courbes de difficulté proposées par la portion de gameplay étudiée.

L'algorithme suivant décrit le traçage de la courbe de difficulté :

```

1 fonction TraceCourbe(Challenge c, Ability a, Level l)
2 {
3   var tailleCourbe = 10; //Taille maximale de la suite de challenge
4     mesurée
5   if(c != null et tailleCourbe > 0)
6   {
7     d = Difficulte(c, a, l); //Difficulté du challenge pour ce niveau de
8       joueur
9     t = Duree(c); //Durée moyenne du challenge
10    //Trace une ligne horizontale de longueur t et de hauteur d.

```

```
11     Tracer(d,t);
12
13     //Simule le résultat
14     var tirage = rand();
15     var e = 'valide';
16     if(tirage < d)
17         e = 'echec';
18
19     //Demande le challenge suivant
20     c2 = ChallengeSuivant(c,e);
21
22     //Si il existe un challenge suivant, réévalue le niveau
23     if(c2 != null)
24         l2 = CorrespNiveau(l,a,c,c2);
25
26     c = c2
27     l = l2
28     tailleCourbe--;
29 }
30 }
```

6.15 Synthèse

Dans ce chapitre, nous avons tout d'abord défini le système de variables qui déterminent la difficulté d'un challenge. Nous avons montré que de nombreux facteurs avaient leur importance, comme la perception de l'effort demandé au joueur, ou l'effort réellement fourni par le joueur. Nous avons proposé un ensemble d'hypothèses, sous lesquelles nous pouvons réduire le nombre de variables impliquées et arriver ainsi à mesurer l'effort que demande le challenge, c'est à dire sa difficulté. Ces hypothèses sont, pour rappel, les suivantes :

- Les objectifs que le joueur doit atteindre sont en apparence suffisamment variés
- Le joueur ne parvient pas à atteindre ses objectifs de manière répétée
- En recrutant pour le playtest des joueurs non rémunérés, issus d'une communauté de joueurs attirés par le type de gameplay proposé, la motivation des joueurs sera maximisée.
- Plus le gameplay du jeu sera fermé, c'est à dire limitera la capacité du joueur à créer des meta objectifs, plus on pourra considérer que le joueurs investissent leur effort pour atteindre l'objectif principal.

Nous avons ensuite proposé un modèle qui, étant données les hypothèses précédentes, permet d'évaluer statistiquement la difficulté d'un challenge. Nous estimons la difficulté d'un challenge à partir de la seule mesure sans équivoque de la performance du joueur, son succès ou son échec. Un nombre suffisant d'observations indépendantes nous permet de déterminer une probabilité d'échec, c'est à dire la difficulté du challenge.

Nous avons ensuite proposé d'exprimer la difficulté d'un challenge sous la forme d'une probabilité conditionnée par une mesure des capacités du joueur. Cette formulation a tout d'abord l'avantage d'être plus descriptive, en permettant au game designer de comprendre de quelle manière certaines capacités influent sur la réussite du joueur pour un challenge donné. Ensuite, elle permet d'obtenir une mesure correcte de la difficulté avec un échantillon de joueurs moins importants, mais qui fournissent chacun plusieurs tentatives de résolution du challenge. Ces tentatives ne sont alors plus indépendantes, puisque chaque joueur évolue, apprend à mieux jouer et donc augmente ses chances de réussites. Mais en mesurant le niveau du joueur, la mesure peut être faite pour un ensemble de tentatives réalisées par des joueurs de même niveau, tentatives qui respectent l'indépendance conditionnelle.

Ce modèle présente cependant diverses faiblesses. Tout d'abord, l'indépendance conditionnelle n'est atteinte que si la capacité choisie est la capacité fondamentale. Plus le niveau du joueur est complexe et s'appuie sur une variété de capacités, plus les liens de causalité entre capacité et performance sont nombreux et donc plus l'indépendance conditionnelle ne peut être atteinte qu'en prenant en compte un nombre important de capacités. Le modèle peut être facilement étendu pour tenir compte de plusieurs capacités, mais le nombre de tentatives doit lui aussi augmenter considérablement.

Dans le chapitre suivant, nous présentons une formalisation du parcours du joueur, c'est à dire de la suite de challenges que le joueur va affronter. En effet, notre mesure de difficulté s'appuie sur un découpage du gameplay en une suite de challenges. Le prochain chapitre propose une logique qui permette de spécifier et manipuler graphiquement cette suite de challenges de manière efficace.

Chapitre 7

Modéliser le parcours du joueur

Sommaire

7.1	Objectif	135
7.2	Challenges et construction de la difficulté	137
7.3	Formalisations existantes	138
7.4	Modèle de scénario	141
7.4.1	Interface avec le moteur de jeu	141
7.4.2	Les challenges	143
7.4.3	Hypergraphe de scénario	145
7.4.4	Calcul de l'état du graphe	147
7.5	Synthèse	148

7.1 Objectif

Dans ce chapitre, notre objectif consiste à fournir un langage graphique de qui permette de définir la suite de challenges que propose un gameplay, ainsi que d'en calculer l'état lorsqu'un joueur progresse dans ce gameplay. Notre mesure de difficulté repose en effet sur le découpage d'un gameplay en challenges, que doit effectuer le game designer. Un tel langage nous permet tout d'abord de définir une interface graphique qui permette au game designer de définir plus aisément la suite de challenges et de la manipuler pour étudier les différents calculs de difficulté réalisés. Ensuite, le calcul de l'état des challenges nous permet de connaître, étant donné les challenges déjà terminés par le joueur, les prochains challenges qui lui seront proposés, et d'alimenter ainsi l'algorithme chargé de tracer la courbe de difficulté du jeu présentée section 6.14.

Nous avons défini dans le chapitre 6 la notion de *challenge*, qui correspond à un objectif que le joueur doit atteindre. Dans un jeu vidéo, les challenges sont présentés au joueur selon un enchaînement pré-établi par le game designer. La logique de cet enchaînement permet de proposer des challenges adaptés, qui tiennent compte de l'apprentissage du joueur et du rythme que le game designer veut donner au gameplay. La courbe de difficulté d'un jeu vidéo dépend donc de cet enchaînement : la suite d'objectifs que le joueur essaie d'atteindre détermine les niveaux de difficulté successifs auxquels il est confronté. Nous appellerons *scénario* la suite de challenges auxquels le joueur est confronté.

Nous employons ici le mot scénario avec une intention bien précise. Nous considérons que la plupart des jeux sont construits comme un enchaînement de challenges, d'objectifs que le joueur essaie d'atteindre. Durant chaque challenge, le joueur tente d'atteindre l'objectif en interagissant avec le monde virtuel, grâce aux différentes actions autorisées par les règles du jeu. A l'issue du challenge, le scénario du jeu permet de déterminer l'ensemble de challenges auquel le joueur pourra être ensuite confronté. Le scénario du jeu est donc pour nous la logique d'enchaînement des challenges, particulièrement pertinente dans le cadre d'une mesure de la difficulté. Cette logique est présente dans le jeu sous différents aspects. La construction de l'univers du jeu, c'est-à-dire l'agencement des différents objets de l'univers est une part de cette logique : de nombreux jeux rythment la suite de challenges par un système de porte et de clefs par exemple. Cette logique est également présente dans les scripts du jeu, qui permettent de définir le comportement des objets du jeu en fonction de la progression du joueur par exemple. Nous souhaitons représenter cette logique de manière synthétique, dans un modèle graphique.

Il apparaît en effet particulièrement pertinent d'intégrer une représentation graphique du scénario d'un jeu vidéo à un outil de mesure de la courbe de difficulté. A l'issue des différents calculs réalisés par le modèle de difficulté (chap. 6), chaque challenge sera annoté de diverses mesures de difficulté, et le designer pourra naviguer dans son scénario pour exploiter ces informations. Une représentation graphique du scénario facilite cette navigation, en offrant une vue d'ensemble plus facilement manipulable que sous forme purement textuelle. De plus, le scénario d'un jeu structure son déroulement temporel, et un tel déroulement se satisfait particulièrement d'une représentation graphique en deux dimensions.

Il est important de noter que notre modèle de scénario constitue une représentation *supplémentaire* de la logique du jeu. Il ne s'agit pas de remplacer les outils de scripting visuels déjà fournis par les moteurs de jeu, comme Kismet pour le moteur Unreal (Epic) ou le Flow Graph du CryENGINE (Crytek), mais de compléter ces systèmes en offrant une vue particulière de la logique de progression du joueur. Une vue synthétique de l'enchaînement des challenges permet de considérer le rythme du gameplay, et en particulier sa difficulté, au niveau de détail le plus adapté, et sans ajouter à la complexité visuelle déjà importante des scripts visuels utilisés pour tout le reste du level design.

La première version de ce modèle a d'abord été mis au point au cours du projet DEEP¹ [Bossier 07], et constitue une extension de travaux précédemment réalisés au sein de CEDRIC pour modéliser le scénario d'un jeu vidéo [Vega 04]. L'objectif de ce projet a consisté à permettre aux personnages non joueurs d'un jeu vidéo, les **PNJ**², de bénéficier d'une liberté d'improvisation. Cette liberté doit cependant rester contrainte au déroulement du jeu tel que prévu par les designers, ce dont notre modèle de scénario avait la charge. Nous avons adapté ce modèle à la problématique particulière du découpage d'un gameplay en une suite de challenges.

Ce chapitre est organisé de la façon suivante. Tout d'abord, nous reviendrons sur le principe de construction de la difficulté d'un jeu vidéo, et sur l'intérêt d'une modélisation des différents challenges et du scénario d'un jeu. Ensuite, nous définirons plus formellement le modèle de scénario, ainsi que le calcul de son état.

7.2 Challenges et construction de la difficulté

Comme nous l'avons montré dans le chapitre 6, dans un jeu vidéo, le joueur poursuit en permanence un objectif. Nous avons défini la notion de challenge, qui encode la relation entre le joueur et un objectif, au travers de ses différents états. Pour construire la courbe de difficulté d'un jeu vidéo, c'est à dire les différents niveaux de difficulté auxquels le joueur va être confronté, le game designer va définir un ensemble de challenges ainsi que la logique avec laquelle ces challenges seront présentés au joueur, c'est à dire le scénario du jeu.

Le scénario d'un jeu permet de tenir compte de l'évolution du niveau du joueur. En effet, lorsque le joueur se confronte à un challenge, il développe ses capacités. Il découvre de nouvelles suites d'actions qui lui permettent de se rapprocher d'un objectif particulier, et sa pratique lui permet de réaliser ces suites d'actions plus rapidement, plus précisément. Lorsqu'un challenge est terminé par le joueur, ses capacités sont modifiées, et le challenge suivant doit être choisi de manière à tenir compte de cet apprentissage pour obtenir la difficulté souhaitée. C'est le scénario du jeu qui permet de tenir compte de l'évolution du niveau du joueur, afin d'obtenir la courbe de difficulté désirée.

D'une manière plus précise, nous pourrions définir la construction de la difficulté d'un jeu vidéo à partir de deux ensembles de challenges, les challenges *atomiques* et les challenges *composites*. Un challenge atomique introduit un contexte de jeu entièrement nouveau pour le joueur. Sa seule expérience est celle de jeux différents mais au gameplay similaire auxquels il a pu jouer. Les challenges composites, quant à eux, sont construits en tenant compte des autres challenges. Les challenges composites exploitent, combinent dans un nouveau contexte les savoir et compétences acquises dans les challenges précédents.

Un challenge atomique d'un jeu comme ceux de la série Zelda (Nintendo) (Figure 2.3),

1. Dialogue basé sur les émotions, l'expérience et la personnalité

par exemple, permet au joueur de découvrir comment se déplacer, ou comment manipuler un objet donné, comme l'épée ou le boomerang. Un challenge composite va placer le joueur face à un ennemi qui ne peut être blessé que dans le dos, en tenant compte du fait que le joueur a appris à utiliser le boomerang, très utile pour prendre un ennemi à revers. Ce challenge composite est construit en tenant compte du challenge atomique au cours duquel le joueur a appris à utiliser le boomerang, et lui permet d'exploiter et de combiner ses connaissances dans un nouveau contexte.

La construction des challenges est donc finement ciselée par le game designer, et il apparaît donc tout à fait approprié de lui fournir un outil capable d'en offrir une vue synthétique. La difficulté de chaque challenge doit être perçue dans le contexte du parcours du joueur dans le scénario du jeu, et la mesure de difficulté que nous avons définie dans la section précédente sera d'autant plus pertinente qu'elle pourra être représentée dans le contexte scénaristique global du jeu.

Dans la section suivante, nous présentons les différents travaux de recherche à partir desquels nous avons élaboré un modèle du scénario d'un jeu vidéo.

7.3 Formalisations existantes

La formalisation du scénario pour les applications ludiques interactives est particulièrement étudiée dans le domaine de la narration interactive. Les recherches dans ce domaine ont pour objectif d'offrir un scénario adaptatif, capable de prendre en compte les envies du joueur tout en conservant les propriétés d'un *bon* scénario. Notre objectif est plus modeste, et consiste à fournir aux designers un outil pour encoder le scénario d'un jeu vidéo, du point de vue des challenges proposés par le gameplay. Nous allons donc passer brièvement en revue certains des modèles proposés dans le domaine de la narration interactive, et présenter ce que nous leur empruntons pour atteindre notre propre objectif.

Avant toute chose, nous tenons à préciser clairement que notre objectif n'est en aucun cas de réaliser une application de narration interactive, telles que la recherche dans ce domaine les conçoit actuellement. Étudiée sous le prisme de la narration interactive, notre démarche peut même apparaître comme rétrograde, s'inspirant de modèles aujourd'hui abandonnés par ces chercheurs, car ne permettant pas d'offrir la plasticité nécessaire à un algorithme générateur d'histoires. Mais notre objectif est tout autre, nous cherchons à fournir au designer de jeu vidéos la capacité d'encoder l'histoire de leur jeu, c'est à dire l'enchaînement des challenges, pour pouvoir l'étudier. La narration interactive promet certes de grandes avancées sur le plan des histoires interactives, et peut sûrement permettre de structurer et d'enrichir les applications souhaitant offrir une *narration émergente*. Mais dans un jeu vidéo, les phases de narration émergentes sont en réalité des phases de gameplay, qui doivent satisfaire de nombreuses contraintes propres à leur aspect ludique, comme par exemple le

fait de proposer un niveau de difficulté adapté. Ces contraintes particulières font qu'il est aujourd'hui extrêmement complexe d'avoir recours à un algorithme génératif pour organiser la progression du joueur dans l'univers d'un jeu. La très grande majorité des jeux conçus aujourd'hui s'appuient donc sur le travail de scénaristes et de game designers, qui garantissent l'expérience ludique du joueur en limitant fortement ses possibilités, en ponctuant un scénario global faiblement interactif avec des phases de gameplay pur, ou challenges, finement calibrées. C'est cette structure, cette suite d'étapes créées par les designers, que nous voulons représenter pour permettre d'étudier de la courbe de difficulté du jeu. Nous allons donc nous inspirer des travaux réalisés en narration interactive, tout en tenant compte du fait que nos objectifs divergent.

Certaines solutions de narration interactive reposent sur la création d'une base de règles logiques générales, permettant de générer ensuite le scénario à la volée, comme le proposent IDTension de Nicolas Szilas [Szilas 05, Szilas 07] ou DEFACTO de Nikitas Sgouros [Sgouros 99]. Ces travaux ne s'appuient pas sur une représentation graphique des différentes étapes du scénario mais sur la formalisation de règles de construction d'un scénario, et proposent donc des modèles purement génératifs, les plus éloignés de notre objectif. Néanmoins, ces travaux posent certaines bases communes aux modèles de représentation d'un scénario. Szilas, reprenant Propp [Propp 28] et Bremond [Bremond 74] pose le principe selon lequel une histoire peut être divisée en une suite de processus, chaque processus ayant plusieurs états possibles : virtualité, passage à l'acte, achèvement ([Bremond 74] p.131) [Szilas 99].

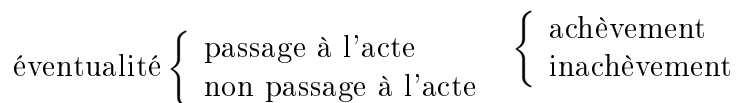


FIGURE 7.1 – Trois temps du développement d'un processus [Bremond 74]

La notion de processus, très générique, se retrouve dans plusieurs systèmes de Narration Interactive. Le système utilisé pour réaliser une des applications de narration interactive les plus connues, Façade, utilise une suite de *beats* narratifs, semblables aux processus de Bremond [Mateas 03]. Ils sont au départ virtuels, puis sélectionnés si leur impact dramatique correspond aux besoins de la narration, et ensuite exécutés ou interrompus.

D'autres systèmes s'appuient également sur une représentation explicite des différentes étapes du scénario et de leur ordre. Le groupe Liquid Narrative, de l'université de Caroline du Nord, a par exemple travaillé sur le système Mimesis, qui génère et encode la narration sous la forme d'un plan en ordre partiel [Saretto 01, Young 04, Young 99]. Chaque étape du plan correspond à une action narrative qui doit être réalisée dans l'univers du jeu. Les plans sont générés à la volée, de manière à satisfaire un objectif scénaristique tout en respectant toute contrainte qualitative pouvant être exprimée de manière formelle.

L'objectif de Mimesis n'est pas d'encoder une fois pour toutes la narration du jeu, mais de générer cette narration au fur et à mesure. Néanmoins, l'utilisation d'un plan comme image d'un scénario possible contribue à notre réflexion. Chaque étape d'un plan respecte le principe de processus narratif de Bremond. Chaque action du plan doit satisfaire des préconditions (virtualité du processus), et peut s'exécuter avec ou sans succès (achèvement ou inachèvement du processus). Par contre, la notation sous forme de plan n'est pas assez ouverte pour nous permettre de réaliser notre modèle. En effet, les plans sont générés par le système Mimesis pour fournir une histoire possible. Le plan est en ordre partiel et ne spécifie pas totalement l'ordre d'exécution ce qui offre une légère plasticité. Mais si une action échoue, il faut alors construire un plan différent, car chaque plan n'encode qu'une seule histoire possible.

D'autres auteurs se sont appuyés sur l'utilisation de plans pour les applications de narration interactive. Brian Magerko, de l'université du Michigan, utilise également un plan en ordre partiel, à ceci près qu'il n'annote pas causalement les liens entre les actions du plan [Magerko 05]. Marc Cavazza, de l'université de Teeside, utilise des réseaux hiérarchique de tâches pour permettre à chaque personnage de planifier son comportement en fonction des ses objectifs [Cavazza 02]. Cette notation permet à chaque personnage de planifier à nouveau son comportement en cas d'erreur, car le réseau hiérarchique de tâches peut encoder plusieurs solutions pour atteindre le même objectif. Néanmoins, l'objectif de cette architecture consiste à faire émerger la narration de l'interaction entre personnages et ne représente pas la structure globale de l'histoire, qui nous intéresse particulièrement.

Riedl et Young ont montré l'équivalence entre la génération de plan par Mimesis et les graphes de scénario [Riedl 06b]. Riedl et Young décrivent les graphes de scénario comme un ensemble de plans et de choix du joueur. Les noeuds du graphe correspondent aux parties linéaires de l'histoire, encodées sous forme de plan, et les arcs du graphe correspondent aux choix du joueur, donc d'une manière générale aux différentes éventualités qui vont guider le joueur entre les histoires possibles. Afin d'atteindre notre objectif, c'est à dire encoder toutes les histoires prévues par le game designer, nous pouvons donc choisir d'étendre la notation en plan, c'est à dire un ordre partiel entre actions, vers une notation sous forme de graphe, en attribuant une sémantique particulière à ses arcs.

Ces travaux de recherche en narration interactive fournissent les bases d'un modèle de scénario pour le jeu vidéo, tel que nous l'entendons, c'est à dire dans l'objectif de modéliser la suite de challenges proposés au joueur par un jeu vidéo :

- Tout d'abord, nos challenges peuvent s'appuyer sur la logique générale des processus décrits par Bremond, logique cohérente avec l'ensemble des représentations étudiées. Nous devons définir des conditions pour déterminer si le joueur a débuté un challenge, et des conditions permettant de savoir si le joueur a réussi le challenge.
- Ensuite, nous souhaitons représenter l'ensemble des parcours possibles du joueurs,

c'est à dire décrire au maximum la logique d'enchaînement des challenges. Nous avons choisit de nous appuyer sur une représentation en graphe, en attribuant une sémantique particulière aux arcs liant les challenges. Cette sémantique nous permet de définir visuellement des relations logiques et temporelles entre les différents challenges.

Les sections suivantes présentent notre modélisation du scénario d'un jeu vidéo. Nous situons tout d'abord le modèle en présentant son interface avec un moteur de jeu, puis décrivons plus précisément le graphe de scénario.

7.4 Modèle de scénario

Les sections suivantes présentent le modèle de scénario d'un jeu vidéo. Tout d'abord, la première section décrit l'interface entre le moteur de jeu et le modèle de scénario, de manière à expliquer plus précisément la place du modèle dans le cadre d'un jeu vidéo.

7.4.1 Interface avec le moteur de jeu

Le modèle de scénario permet d'encoder graphiquement la logique de présentation des challenges au joueur. Comme nous l'avons dit précédemment, cette logique est déjà encodée dans la topologie du jeu et dans les scripts qui gouvernent le fonctionnement des objets du jeu : la plupart des jeux vidéos proposent des suites de challenges au joueur sans avoir recours à un modèle explicite du scénario du jeu. A ce titre, le modèle de scénario est une vue particulière du code qui régit le fonctionnement de l'univers du jeu.

Nous avons donc conçu le modèle de scénario comme un module supplémentaire du moteur de jeu, qui ne vient pas remplacer des fonctionnalités déjà existantes. Dans un premier temps, notre modèle doit permettre au designer d'encoder cette logique, par ailleurs déjà présente à des multiples endroits du code du jeu, et d'observer son bon déroulement. Cette séparation permet de construire un modèle facilement adaptable à de nombreux moteurs de jeux, car son objectif n'est que d'observer le déroulement du jeu, sans intervenir spécifiquement. En effet l'objectif global de notre modèle est d'offrir un outil d'analyse de la difficulté d'un jeu vidéo, et non pas un outil de conception d'un jeu vidéo.

Mais bien évidemment, ce travail apparait redondant : le level designer va devoir encoder la logique du jeu deux fois. De manière exhaustive et effective dans le code et la topologie du jeu vidéo, et de manière rapide et synthétique dans le modèle de scénario pour pouvoir réaliser différentes analyses de difficulté. Il est donc évident que lors de l'intégration définitive d'un tel modèle dans un moteur de jeu, il sera particulièrement utile de limiter au maximum cette redondance. Un premier moyen consisterait par exemple à permettre au designer, à partir du graphe de scénario, d'accéder directement au code de chaque challenge, c'est à dire d'intégrer plus finement les différentes interfaces de création du graphe de scénario et

du code de level design. De cette manière les deux vues seraient développées de manières conjointes et participeraient l'une et l'autre à l'élaboration du jeu.

Cependant, la description d'une telle interface est spécifique au moteur choisit et sort du cadre précis d'analyse de la difficulté que nous nous sommes fixés. La figure 7.2 présente l'interface minimale du moteur de jeu est du modèle de scénario, nécessaire à l'analyse de la difficulté d'un jeu vidéo.

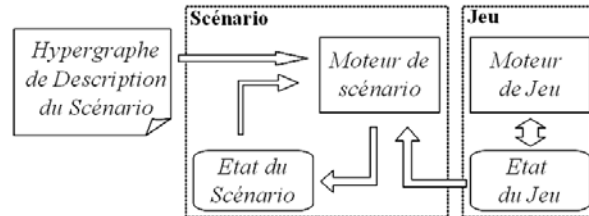


FIGURE 7.2 – Interface entre jeu et scénario.

La figure 7.2 met en évidence la séparation entre le modèle de scénario et le moteur de jeu. Le moteur de jeu calcule son propre état et suit sa propre logique mais communique au moteur de scénario tout événement important. Nous avons définis ces événements dans le chapitre précédent (sec. 6.8). Le moteur de scénario se charge d'interpréter ces événements pour calculer l'état du scénario, c'est à dire l'état des différents challenges qui le composent. Cette séparation a l'avantage de permettre au moteur de scénario de fonctionner sans faire appel au moteur de jeu, à partir d'une simple trace des événements importants survenus pendant une session de jeu. De cette manière, un outil d'analyse de la difficulté peut être utilisé sans le moteur de jeu.

Bien que le modèle de scénario ne soit qu'un simple observateur du moteur de jeu, celui-ci doit être capable de transmettre au préalable une description des événements particuliers qu'il souhaite observer. Le moteur de jeu n'est en effet pas capable de déterminer, a priori, les changements d'état du jeu susceptibles d'alimenter le calcul de l'état du modèle de scénario. Il doit donc exister une phase d'intégration du modèle de scénario au jeu en cours de développement, qui permette de décrire, au niveau du moteur de jeu, les événements importants que le moteur de scénario souhaite observer.

Nous définissons ainsi un ensemble de tests T . Ces tests ont valeur dans \mathbb{B} et sont définis par le game designer sous forme textuelle. Il peut s'agir de savoir si le joueur est vivant ou mort, ou s'il possède tel ou tel objet. Ces tests sont ensuite intégrés au moteur de jeu. Lorsqu'un de ces tests change d'état, par exemple si le joueur meurt, ou ramasse un objet, un événement correspondant est enregistré dans la trace et communiqué au moteur de scénario. Ce dernier conserve ainsi l'état courant de tous les tests utiles sur l'état du jeu, qu'il met à jour à chaque fois que le moteur de jeu lui communique un nouvel événement. Grâce à

l'ensemble T , le moteur de scénario dispose d'une abstraction de l'état du jeu, sous forme d'un ensemble de valeurs booléennes. Chaque état du jeu correspond à une configuration des valeurs prises par les différents tests de T . On notera U l'ensemble de toutes les combinaisons de valeurs possibles des éléments de T , c'est à dire l'ensemble des états du jeu possibles, tels que perçus par le moteur de scénario.

- T est un ensemble de tests sur l'état du jeu, réalisés par le moteur de jeu et dont la valeur est fournie au moteur de scénario sous forme d'évènements
- U correspond aux différentes valeurs attribuables simultanément aux éléments de T . U est l'ensemble des états possibles du jeu, tels que perçus par le moteur de scénario.

Nous avons donc conçu le modèle de scénario comme une vue particulière de la logique du jeu et de son exécution, parallèle à la spécification et à l'exécution de la logique globale du jeu. Le moteur de scénario est donc pour l'instant extérieur au moteur de jeu et permet simplement d'en étudier le fonctionnement. L'intégration de notre outil à un moteur particulier pourra permettre de limiter la redondance lié à l'encodage consécutif de la logique du jeu dans le moteur de jeu puis dans le moteur de scénario, mais ne fait pas partie des objectifs de cette thèse.

Dans cette section, nous avons décrit l'interface entre le moteur de jeu et le moteur de scénario. Dans les prochaines sections, nous décrivons plus précisément le modèle de scénario, en débutant par les challenges, dont la formalisation est présentée dans la section suivante.

7.4.2 Les challenges

Nous représentons un scénario sous forme d'un hypergraphe. Les sommets de cet hypergraphe sont les challenges auxquels le joueur pourra être confronté, et les arêtes de l'hypergraphe permettent de définir un ordre partiel et une partie de la logique d'enchaînement des challenges. Dans cette première section, nous décrivons plus précisément les challenges, et leur représentation dans le modèle de scénario.

Nous avons déjà introduit les challenges précédemment, dans la section 6.4. Les challenges représentent un objectif que le joueur doit atteindre, et leur état décrit la progression du joueur vers cet objectif. Un challenge peut être à l'état *initial*, auquel cas le joueur n'a encore aucun rapport avec lui. Lorsque le challenge est *actif*, le joueur est effectivement en train de résoudre le challenge. Finalement, si le challenge passe à l'état *valide*, c'est que le joueur a réussi à atteindre l'objectif, mais s'il passe à l'état *echec* c'est que le joueur a échoué. L'automate de la figure 7.3 présente les différents états du challenge ainsi que les transitions possibles entre ces états.

A tout instant, le moteur de scénario doit être capable de calculer l'état du modèle de scénario, c'est à dire, au niveau des challenges, être capable de décider d'une transition entre deux états. Ces transitions dépendent de l'évolution de l'état du jeu : pour savoir si le joueur

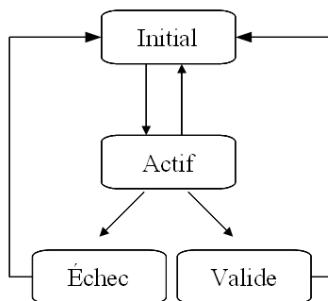


FIGURE 7.3 – Automate d'un challenge.

à validé un challenge, le moteur de scénario s'attend à observer un changement dans l'état du jeu traduisant cette victoire. Comme nous l'avons défini précédemment, toute modification de l'état du jeu susceptible d'aider le moteur de scénario à calculer l'état du scénario est communiqué sous forme d'évènement. Nous avons défini U comme l'ensemble des différents états du jeu perçus par le moteur de scénario.

Pour chaque challenge, le game designer va construire en ensemble de conditions, qui régissent le fonctionnement de son automate. Ces conditions définissent quels éléments de U , c'est à dire quels états du jeu, entraînent une transition de l'automate. A chaque challenge correspondent ainsi quatre conditions :

- La condition d'activation, $\omega_a : U \rightarrow \mathbb{B}$, permet de déterminer si un challenge peut passer de l'état initial à l'état actif³.
- La condition d'échec, $\omega_e : U \rightarrow \mathbb{B}$, permet de déterminer si un challenge peut passer de l'état actif à l'état échec.
- La condition de validation, $\omega_v : U \rightarrow \mathbb{B}$, permet de déterminer si un challenge peut passer de l'état actif à l'état valide.
- La condition de reset, $\omega_r : U \rightarrow \mathbb{B}$, permet de déterminer si un challenge peut passer de l'état valide ou échec à l'état initial.

Ces différentes conditions permettent de calculer l'état d'un challenge en fonction de l'état du jeu. Toutefois, le calcul de l'état d'un challenge ne dépend pas uniquement de ces conditions. En effet, nous souhaitons encoder explicitement les relations d'ordre entre challenges, au moyen d'un hypergraphe. Le calcul de l'état du challenge doit donc tenir compte de la place du challenge dans l'hypergraphe de scénario. Les sections suivantes décrivent l'hypergraphe de scénario ainsi que sa prise en compte dans le calcul de l'état des challenges.

3. Cette transition ne dépend pas que de ω_a mais également de l'état des liens entrants du challenge, que nous définissons par la suite, section 7.4.3.

7.4.3 Hypergraphe de scénario

Dans la section précédente, nous avons défini les challenges ainsi que le calcul des transitions entre leurs différents états. Dans cette section, nous nous préoccupons plus particulièrement de modéliser l'ordre de présentation des challenges au joueur. Nous allons construire un hypergraphe de scénario, dont les sommets sont les challenges, et dont les hyperarcs permettent de définir des conditions logiques qui contraignent l'activation des challenges. Par exemple, nous pourrions écrire qu'un challenge ne peut être débuté par le joueur que si deux autres challenges sont au préalable validés, et qu'un autre n'est pas encore débuté. Ce challenge ne pourra débuter que si l'hyperarc entrant qui définit cette condition est *satisfait*, c'est à dire si les challenges à l'origine de cet arc satisfont cet état particulier.

En effet, comme nous l'avons dit précédemment, les challenges sont présentés au joueur selon un ordre et une logique particulière. Les conditions de transition d'état définies dans la section précédente pourraient permettre de prendre en compte cet ordre : il suffirait de spécifier dans la condition d'activation des challenges ω_a , l'état que doivent respecter les challenges précédents. Néanmoins, nous pensons qu'il est beaucoup plus lisible et intuitif d'encoder cet ordre sous la forme d'un hypergraphe. De cette manière, la condition d'activation ω_a est beaucoup plus simple à écrire car elle ne doit pas tenir compte des challenges précédents, et surtout, la logique de déroulement du scénario peut être représentée sous forme graphique et gagner ainsi en lisibilité. Cette représentation graphique permet de faire coïncider le déroulement temporel et logique du scénario avec sa représentation dans l'espace, offrant une vision globale et intuitive du scénario qui ne pourrait être obtenue au moyen d'une suite de formules logiques textuelles. Nous décrivons donc, dans cette section, de quelle manière l'hypergraphe de scénario encode l'ordre des challenges, et son impact sur le calcul de l'activation d'un challenge.

Nous définissons un hypergraphe de scénario comme un tuple $(C, L, \phi_L, \lambda_L)$, avec :

- C : l'ensemble des challenges, sommets de l'hypergraphe, définis dans la section précédente.
- L : l'ensemble des *étiquettes de liens* (ou *étiquettes d'hyper-arcs*), caractérisés par les deux fonctions ϕ_L et λ_L suivantes :
- $\phi_L : L \rightarrow \mathbb{P}(C) \times \mathbb{P}^1(C)$ est une fonction de lien, qui à toute étiquette de lien fait correspondre un hyper-arc orienté, c'est à dire une paire ordonnée $(O, \{c\})$ où O est un ensemble non vide de challenges nommé *origine du lien*, et c est un challenge nommé *cible du lien*, et tel que $c \notin O$. Nous ne considérons que des liens dont le challenge cible est un singleton.
- $\lambda_L : C \times L \rightarrow \{=, \neq\} \times \{\text{actif, valide, echec}\}$ est la *précondition d'un lien*, vis à vis d'un challenge d'origine, qui fait correspondre, à tout couple (c, l) tels que c appartient à l'origine de l , un couple (op, e) où op est un opérateur d'égalité ou d'inégalité, et e une étiquette correspondant à l'un des trois états $\{\text{actif, valide, echec}\}$ d'un challenge.

Un *lien* peut être vu comme un triplet $(\{c_i\}, (op_i, e_i), r)$, qui relie l'*origine* du lien, un ensemble de challenges $\{c_i\}$, à la *cible* du lien, un challenge unique r . La *précondition* (op_i, e_i) spécifie l'état que chaque challenge c_i doit respecter pour *satisfaire* le lien. Nous notons un lien l , tel que $\phi_L(l) = (\{c_1, \dots, c_n\}, r)$ et $op_i, e_i = \lambda_L(c_i, l)$, de la façon suivante : $\{c_i \overset{op_i, e_i}{\rightsquigarrow} r\}$.

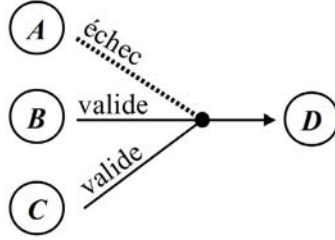


FIGURE 7.4 – Exemple de lien

Les pointillés indiquent une négation.

La figure 7.4 présente un exemple d'hypergraphe, avec un seul lien. On a donc, pour ce lien l , $\phi_L(l) = (\{A, B, C\}, D)$, c'est à dire que l'origine de l est l'ensemble de challenges $\{A, B, C\}$ et que le challenge D est sa cible. Le lien l décrit l'état que chaque challenge d'origine de l doit respecter. Par exemple, $\lambda_L(B, l) = \{=, valide\}$, ce qui signifie que pour que l soit satisfait, il faut le challenge B soit valide. Le lien l exprime aussi le fait que pour pouvoir activer D , il faut que A ne soit pas en échec, et que B et C soient valides.

Pour qu'un lien soit satisfait, sa précondition doit être satisfaite, c'est à dire que l'ensemble des challenges d'origine doivent respecter un état particulier. Si on note $\pi(c)$ l'état d'un challenge, alors l'ensemble des liens satisfait L_S^π peut être exprimé de la façon suivante :

$$L_S^\pi = \{\{p_i \overset{op_i, e_i}{\rightsquigarrow} r\} \in L \mid \bigwedge_i op_i(\pi(p_i), e_i)\} \quad (7.1)$$

Les liens permettent d'ordonner les challenges en spécifiant une condition supplémentaire à l'activation d'un challenge. En effet, pour qu'un challenge passe de l'état initial à l'état actif, il faut à la fois que ω_a soit satisfaite, mais également que *si le challenge est la cible de liens, au moins un de ces liens soit satisfait*. Si on appelle la condition générale de transition de l'état initial à l'état actif ω'_a , celle ci s'écrit :

$$\omega'_a(c) = \omega_a \wedge [(\exists l / \phi_L(l) = (X, c) \wedge l \in L_S^\pi) \vee \nexists l / \phi_L(l) = (X, c)] \quad (7.2)$$

En attribuant une sémantique particulière aux hyper-arcs de l'hypergraphe de scénario, nous sommes ainsi en mesure de spécifier un pré-ordre sur les challenges, assorti de conditions

logiques. Dans la section suivante, nous présentons le calcul complet de l'état du graphe, à partir des conditions de transition et de la satisfaction des liens.

7.4.4 Calcul de l'état du graphe

L'algorithme suivant est appelé à chaque fois que le moteur de scénario reçoit un nouvel évènement, lu dans une trace ou reçu en temps réel, ou à l'initialisation du moteur. Cet algorithme montre le calcul de l'état du scénario, c'est à dire de l'état des liens et des challenges de l'hypergraphe de scénario. On considère qu'à chaque fois que le moteur reçoit un évènement, il met à jour son abstraction de l'état du jeu, et la fournit à cet algorithme.

```
1 fonction MajScenario(GameState u, Graphe g)
2 {
3
4     var etatModifie:Booleen = true;
5
6     tant que(etatModifie)
7     {
8         var c:Challenge;
9         var l:Lien;
10
11         //Initialisation
12         etatModifie = false;
13
14         //On met a jour tous les challenges
15         pour chaque challenge c de g
16         {
17             //On applique les conditions de transition
18             si(MajChallenge(c,u))
19                 etatModifie = true;
20         }
21
22         //On met à jour tous les liens
23         pour chaque lien l de g
24         {
25             //On vérifie la satisfaction du lien
26             si(MajLien(l,u))
27                 etatModifie = true;
28         }
29     }
30 }
```

L'algorithme `MajScenario` montre les différentes étapes de mise à jour du graphe de scénario. Tout d'abord, les conditions de transition des challenges sont évaluées de manière à calculer leur nouvel état. Les challenges sont uniquement autorisés à effectuer une transition dans leur automate d'état, si les conditions de transitions le permettent. L'état des challenges obtenu n'est pas forcément stable, c'est à dire qu'une nouvelle exécution de la boucle pourrait déclencher d'autres transitions. Mais il est nécessaire de répercuter tout changement d'état d'un challenge sur ses liens sortants, aussi les challenges ne sont ils mis à jour qu'une seule fois, avant de mettre à jour les liens.

Le nouvel état des challenges est ensuite utilisé pour calculer le nouvel état des liens. Tant qu'il existe un challenge ou un lien dont l'état change, ces deux mises à jour sont répétées, jusqu'à obtention d'un graphe de scénario stable. Bien sûr, on remarque que dans le cas d'un hypergraphe cyclique, ce calcul peut être infini. Nous n'avons toutefois pas limité le graphe de scénario à un modèle acyclique, car les scénarios de jeux s'appuient par essence sur des cycles, lorsqu'ils demandent par au joueur d'effectuer plusieurs fois un même challenge. Mais si le scénario est bien construit, le cycle demandera forcément une intervention du joueur, et donc une variation de l'état du jeu. Dans un scénario bien construit, la boucle de l'algorithme `MajScenario` ne sera donc jamais infinie. L'implémentation de cet algorithme dans notre application limite néanmoins le nombre de mises à jour successives de manière à détecter une boucle infinie, signe d'une erreur dans la modélisation du scénario.

7.5 Synthèse

Dans ce chapitre, nous avons présenté comment encoder le scénario d'un jeu vidéo, au sens ou nous l'entendons, c'est à dire la logique d'enchaînement des challenges d'un jeu vidéo. Cette suite de challenges est primordiale pour un outil d'analyse de la difficulté d'un jeu vidéo. Elle permet de faire apparaître la structure sur laquelle s'appuie le game designer pour construire la difficulté du jeu. Nous nous sommes inspirés des recherches effectuées en Narration Interactive pour construire notre propre formalisme. De ces recherches, nous avons principalement retenu la notion de processus et celle d'ordre, dont une représentation graphique est particulièrement pertinente.

Nous avons ensuite décrit plus précisément l'interface entre le moteur de scénario et un moteur de jeu. Notre modèle propose une vue particulière de la logique du jeu, et nous l'avons conçu comme un module supplémentaire, parallèle au moteur de jeu. Nous sommes conscients que lors de l'intégration définitive de notre moteur de scénario à un moteur de jeu particulier, il sera pertinent de permettre une plus forte interaction entre la logique décrite par le scénario et celle décrite au sein des scripts de level design. Néanmoins, notre objectif est de fournir ici un modèle adaptable à n'importe quel moteur de jeu, et son intégration au sein d'une plateforme particulière est une étape supplémentaire qui sort du champ de cette

thèse.

Nous avons ensuite décrit le calcul de l'état du scénario, c'est à dire de l'état des challenges et des différents liens qui composent l'hypergraphe de scénario. Ce calcul permet de décrire précisément la sémantique du graphe de scénario, et plus particulièrement la manière dont l'état des liens interagit avec l'état des challenges, en ajoutant une précondition d'activation. Ce calcul, réalisé par le moteur de scénario, permet au designer de visualiser, à partir d'une trace d'évènements ou pendant une session de jeu, l'évolution de l'état du scénario.

Le fait d'utiliser une représentation graphique pour le modèle de scénario permet de fournir une interface particulièrement pertinente pour la description des différents calculs liés à l'évaluation de la difficulté. Nous avons développé une application exploitant cette interface, que nous présentons dans le chapitre suivant.

Chapitre 8

Logiciel de mesure de la difficulté

Sommaire

8.1	Objectif	151
8.2	Algorithme général de fonctionnement du logiciel	152
8.3	Interface	154
8.3.1	Edition du scénario	155
8.3.2	Edition des algorithmes de calcul des capacités	158
8.3.3	Courbe de difficulté	160
8.4	Synthèse	161

Dans ce chapitre, nous présentons un logiciel développé pour étudier la difficulté d'un gameplay. Nous y avons implanté le modèle de difficulté ainsi que le modèle de scénario décrits dans les chapitres 6 et 7. Nous allons tout d'abord présenter l'objectif de ce logiciel, puis son interface graphique. Pour finir, nous décrirons son architecture logicielle.

8.1 Objectif

Nous avons développé ce logiciel afin de pouvoir exploiter les modèles de difficulté et de scénario dans le cadre de différents gameplays. Nous souhaitons à la fois évaluer la faisabilité d'un outil de mesure de la difficulté, et définir plus précisément l'intégration logicielle du modèle de scénario et de calcul de la difficulté. En implantant les différents modèles et algorithmes définis dans les sections précédentes, nous sommes en mesure de les manipuler directement à partir d'une interface graphique, ainsi que de les confronter à des données réelles.

Le logiciel de mesure de la difficulté doit fournir l'interface nécessaire pour :

- spécifier le scénario d'un jeu vidéo, c'est à dire permettre de construire et de manipuler l'hypergraphe de scénario,
- spécifier une méthode de reconnaissance des capacités, de manière à pouvoir calculer l'algorithme `Difficulte 6.10`,
- calculer et afficher l'état du modèle de scénario en fonction de la lecture d'une trace,
- calculer et afficher la courbe de difficulté d'un gameplay.

Dans la prochaine section nous décrivons tout d'abord l'algorithme général de fonctionnement du logiciel. Par la suite, nous décrirons l'interface graphique du logiciel de mesure de la difficulté. Dans la section suivante, nous en présentons l'architecture logicielle.

8.2 Algorithme général de fonctionnement du logiciel

Nous décrivons à la figure 8.1, l'enchaînement des différentes étapes qui permettent de calculer la courbe de difficulté à laquelle chaque joueur s'est trouvé confronté.

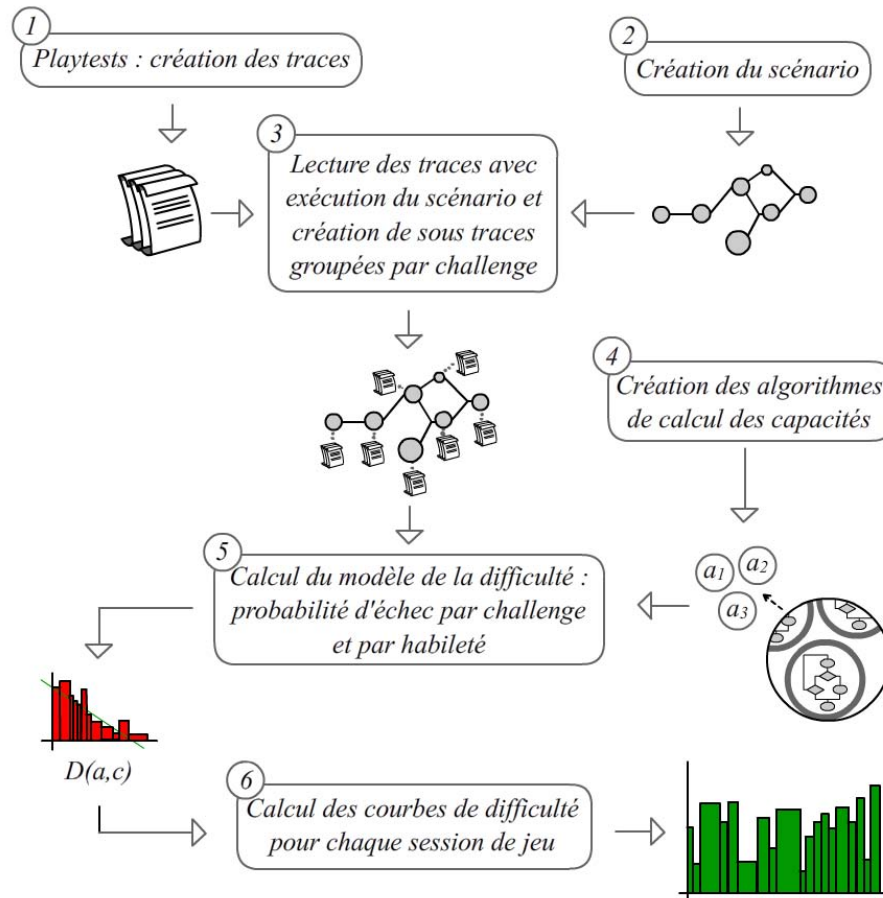


FIGURE 8.1 – Fonctionnement du logiciel

Les étapes 1,2 et 4 de la figure 8.1 sont réalisées par l'expérimentateur. L'étape 1 consiste à recueillir les traces d'évènements qui décrivent les sessions de jeu de plusieurs joueurs. Le moteur de jeu doit donc être pourvu d'un code capable d'enregistrer les évènements utiles et de générer ainsi une trace par session de jeu, c'est à dire un ensemble de fichiers XML que le designer fournit au logiciel d'analyse de la difficulté. L'étape 2 consiste à décrire le scénario du jeu, c'est à dire l'enchaînement des différents challenges, à l'aide de l'éditeur d'hypergraphe de scénario du logiciel. L'étape 4 consiste à décrire, pour chaque capacité que le game designer souhaite mesurer, un algorithme capable de déterminer si le joueur tente d'atteindre le but à court terme correspondant, et s'il y parvient. Les étapes 2 et 4 seront plus amplement décrites dans les sections suivantes.

Au cours de l'étape 3, le logiciel va rejouer chaque trace, en recalculant l'état du scénario à chaque nouvel évènement. De cette manière, le logiciel segmente la trace de chaque joueur

en sous-traces, chaque sous-trace correspondant à l'exécution d'un challenge particulier. A l'issue de l'étape 3, chaque challenge est associé à un ensemble de sous traces, qui décrivent les multiples tentatives de validation pour le groupe de joueurs étudié.

L'étape 5 permet au logiciel de calculer le modèle de difficulté. Pour chaque challenge, le logiciel va fournir l'ensemble des sous traces aux algorithmes de calcul des capacités¹. Chaque sous trace est ainsi notée en fonction de l'habileté montrée par le joueur. Les sous traces sont ensuite regroupées par classe de niveau. A partir de la fréquence de sous-traces aboutissant à un échec, le logiciel détermine la probabilité d'échec pour chaque niveau.

L'étape 6 utilise le modèle de difficulté pour calculer la courbe de difficulté de chaque session de jeu. L'étape 5 a permis de noter les capacités de chaque joueur pour chaque sous trace. Nous pouvons donc connaître, pour chaque session de jeu, la suite de challenges qu'a tenté le joueur et les capacités dont il a fait preuve. Nous pouvons donc obtenir une mesure plus précise de sa performance, non pas une valeur binaire d'échec ou de réussite, mais la probabilité d'échec en fonction de son comportement. Le logiciel trace donc une courbe de difficulté à partir des probabilités d'échec successives².

Dans la prochaine section, nous décrivons de quelle manière le designer peut accomplir chacune de ces étapes à l'aide du logiciel, en présentant plus précisément son interface.

8.3 Interface

La section suivante présente l'interface du logiciel de calcul de la difficulté d'un gameplay.

1. Voir algorithme *Niveau* section 6.9. Attention, ce que nous appelons sous-traces ici sont les traces fournies par la fonction *trace*, que l'algorithme *Niveau* sub-divise encore.

2. Voir algorithme *TraceCourbe* section 6.14

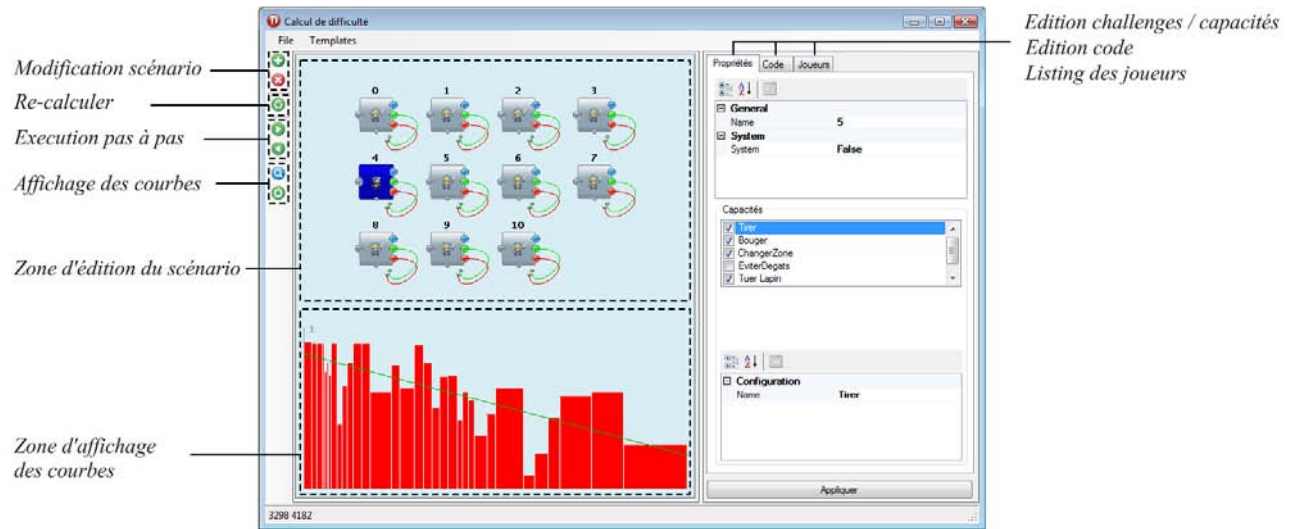


FIGURE 8.2 – Logiciel de calcul de la difficulté

La figure 8.2 présente la fenêtre principale du logiciel. Seule la page d'édition des propriétés des capacités et des challenges est visible, et masque de fait les pages d'édition du code et la liste de joueurs.

8.3.1 Edition du scénario

Le logiciel de calcul de la difficulté permet de créer et d'éditer le scénario d'un jeu vidéo, sous forme d'un hypergraphe. Les challenges sont représentés sous forme de boites rectangulaires (Figure 8.3). En fonction de son état, la couleur du challenge varie : bleu si actif, rouge pour l'échec et vert une fois validé. Les zones circulaires font office de point d'attaches pour les liens entrants et sortants. En particulier, la couleur des zones d'attache des liens sortants permet de connaître l'état testé par le lien, c'est à dire l'étiquette fournie par la fonction λ_L (sec. 7.4.3), à savoir $\{\text{actif}, \text{valide}, \text{echec}\}$.

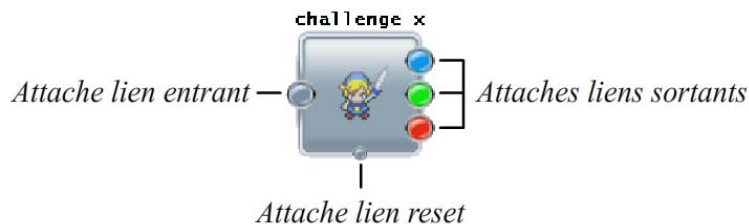


FIGURE 8.3 – Représentation d'un challenge

La couleur de la zone d'attache est reportée sur le lien, et un lien en pointillé indique une négation. La figure 8.4 présente un lien qui n'est satisfait que si le challenge *A* est valide, et le challenge *B* n'est pas en échec.

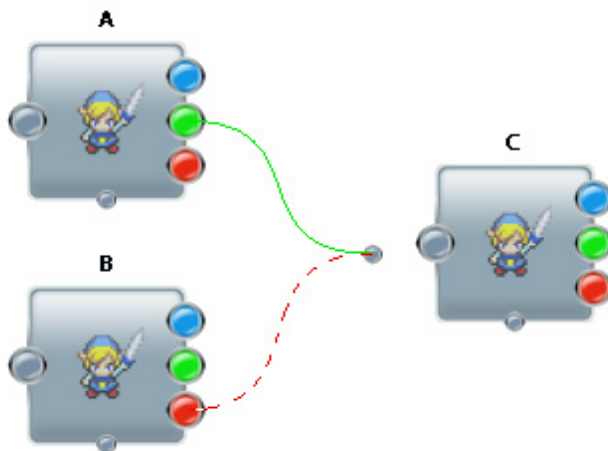


FIGURE 8.4 – Exemple de lien entre challenges

C ne s'activera que si A est valide et B n'est pas en échec.

Les différents challenges peuvent être déplacés et supprimés de même que l'ensemble des liens. L'utilisateur a la possibilité de sélectionner la trace d'un joueur particulier et de l'exécuter en mode pas à pas, de manière à visualiser son parcours.

8.3.1.1 Mise à jour de l'état du jeu

La programmation de la mise à jour de l'état du jeu en fonction des événements reçus par le moteur de jeu, comme présenté section 7.4.1, est une tâche que doit accomplir le designer et qui dépend de chaque gameplay. Le code de mise à jour des variables de jeu est un code dynamique, qui doit être chargé par l'application à son exécution. Nous avons choisi d'utiliser le langage *lua*, et intégré un petit éditeur de texte dans l'application de calcul de la difficulté. Le listing suivant présente un exemple de code de mise à jour des variables d'état du jeu.

```

1 GArmeJoueur = "";
2
3 // MISE A JOUR ETAT DU JEU
4 // Appel a chaque nouvel evenement

```

```
5 // code : code de l evenement
6 // nbValues : nombre de parametres lies a l evt
7 // values : tableau de parametres lies a l evt
8 fonction UpdateEtatJeu(code, time, nbValues, values )
9 if code == 11 then
10   GARmeJoueur = values [0];
11 end
12 return
13 end
```

La variable *GARmeJoueur* décrit l'arme qu'utilise actuellement le joueur (préfixée G pour rappeler sa portée globale à l'ensemble du code Lua). Le seul moyen pour le moteur de scénario de connaître ce choix consiste à recevoir l'évènement numéro 11. Le code du listing se charge de récupérer la bonne valeur dans la trame pour l'assigner à la variable *GARmeJoueur*. Ce code de mise à jour est appelé par le moteur de scénario à chaque lecture d'un nouvel évènement de la trace, ce qui permet de maintenir à jour l'état du jeu sous formes de variables Lua, exploitables dans n'importe quelle autre partie du code.

8.3.1.2 Préconditions des challenges

Les transitions entre les différents états d'un challenges sont soumises à diverses préconditions, comme présenté section 7.4.2. Ces précondition sont également codées en lua. Le listing suivant présente par exemple une précondition d'activation :

```
1 // Test d activation
2 // Retourner si le plotpoint peut s activer
3 fonction IsActif23( )
4
5 if GARmeJoueur == "Rocket_Launcher" then
6   return 1
7 end
8
9 return 0
10 end
```

Le challenge ne pourra s'activer que si les conditions propres à ses lien entrants sont satisfaites, mais également si la fonction *IsActif23* retourne la valeur 1, c'est à dire si le joueur est armé du *Rocket Laucher*. Lorsque le designer clique sur un challenge, le code des conditions est chargé dans la fenêtre d'édition pour que le designer puisse le modifier à loisir.

8.3.2 Edition des algorithmes de calcul des capacités

L'étape 4 de l'algorithme 8.1 correspond à l'édition des algorithmes de calcul des capacités. Le designer peut mesurer autant de capacités qu'il le souhaite, qui seront listées sur l'écran principal de l'application, figure 8.2. Pour chaque capacité, le designer doit écrire un algorithme, chargé de comptabiliser le nombre de fois où le joueur tente d'atteindre le but à court terme associé à la capacité, ainsi que le nombre de fois où il y arrive effectivement. Le listing suivant présente un exemple de code de calcul d'une capacité, ici la capacité de viser précisément.

```

1 //-----
2 //--- VARIABLES
3 //-----
4 local nbReu = 0;
5 local nbTry = 0;
6 local nbTryPending = 0;
7 local lastTime = 0;
8
9 //-----
10 //--- RECONNAISSANCE
11 //-----
12 //--- Nouvel event
13 //--- Donne un nouvel event de la trace.
14 //--- Retourne nombre de nouvelles tentatives puis de nouvelles
    reussites
15 //--- code : code de l evenement
16 //--- nbValues : nombre de parametres lies a l evt
17 //--- values : tableau de parametres lies a l evt
18 function NewEvent1(code, time, nbValues, values )
19 if code == 32 and (values[0] == "True" or values[1] == "True") then
20     nbTry = nbTry+1;
21     nbTryPending = nbTryPending +1;
22     lastTime = time;
23 end
24
25 if code == 02 and nbTryPending > 0 then
26     nbTryPending = nbTryPending -1
27     nbReu = nbReu+1;
28 end
29
30 if time - lastTime > 5 then
31     nbTryPending = 0;

```

```
32 end
33
34 return nbReu ,nbTry ;
35 end
36
37 //— Restart
38 //— Raz de la detection de capacites
39 function ResetStrat1( )
40 nbReu = 0;
41 nbTry = 0;
42 nbTryPending =0;
43 lastTime = 0;
44 end
```

L'évènement 32 correspond à un tir du joueur, et ses deux valeurs correspondent au fait que lorsque le joueur tire, il peut soit voir un ennemi, soit être pris en chasse par un ennemi. Ceci nous permet de ne pas tenir compte des tirs que le joueur effectue au hasard, pour le simple plaisir d'utiliser son arme, mais tient uniquement compte des tirs utiles. L'évènement 2 correspond à un dégât subit par un joueur ennemi. L'algorithme évalue ainsi le nombre de tirs utiles du joueur, ainsi que le nombre d'impacts correspondants à ces tirs.

Le designer doit écrire, pour chaque capacité qu'il souhaite mesurer, l'algorithme permettant de compter réussites et tentatives, à partir de la suite d'évènements d'une trace. Lua est un langage de script simple et efficace pour cette tâche, directement accessible à partir de l'éditeur intégré à l'application. Le designer doit uniquement déclarer ses variables et remplir le corps des trois fonctions : leur prototype et les commentaires sont automatiquement générés à la création du challenge, afin de minimiser et de cadrer au maximum cette partie du design.

Une fois l'étape 5 de la figure 8.1 atteinte, le designer est en mesure de visualiser les probabilités d'échec en fonction des capacités du joueur. La figure 8.5 montre un exemple calculé par le logiciel à partir de données d'expérience, pour la capacité spécifiée précédemment, c'est à dire la précision de tir du joueur. On remarque que plus le joueur est précis, plus sa probabilité d'échec diminue.

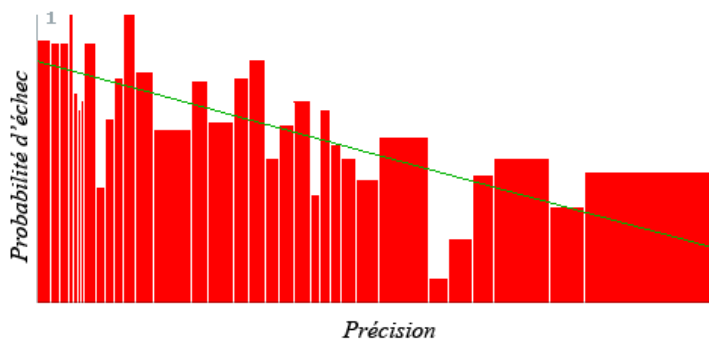


FIGURE 8.5 – Probabilité d’échec en fonction de la précision au tir.

En abscisse, la précision du joueur (probabilité de toucher la cible), et en ordonnée, la probabilité d’échec. La droite verte est la régression linéaire ($r = -0.72$, $p = 0.000002$)

Dans notre logiciel, nous modélisons la probabilité d’échec en fonction des capacités par la droite de régression linéaire calculée sur les données. Le coefficient de corrélation linéaire nous permet ainsi d’évaluer la qualité de cette régression, et de mesurer la force du lien qui relie la mesure de capacité à la probabilité d’échec du joueur. Le coefficient de corrélation nous permet de classer une capacité : plus sa valeur absolue est élevée et plus la mesure de capacité permet de prédire la difficulté du jeu. De même le coefficient de corrélation ainsi que sa significativité nous permettent de déterminer si le modèle réussit à prédire la difficulté du jeu : nous rejetons systématiquement les corrélations inférieures à 0.3 ou dont la significativité dépasse 0.1³.

8.3.3 Courbe de difficulté

Une fois l’étape 6 de la figure 8.1 atteinte, le designer est en mesure d’étudier les courbes de difficulté de chaque session de jeu. A partir de la liste des joueurs, le designer peut consulter l’ensemble des courbes tracées et des mesures de capacités de chaque joueur. Le calcul de la difficulté est réalisé à partir de la moyenne des difficultés prédites pour chaque capacité, pondérée par la valeur absolue de leur coefficient de corrélation. Toute capacité dont le coefficient de corrélation est inférieure à 0.3 ou la significativité supérieure à 0.1 sont écartées du calcul. Si aucune capacité n’est suffisamment significative, cette partie de

3. Nous partons du principe, dans ce prototype, qu’un modèle linéaire suffit à décrire la relation entre capacité et difficulté. Mais d’autres modèles peuvent être envisagés, comme par exemple un réseau de neurone entraîné sur la moitié des données et évalué sur l’autre.

la courbe n'est pas tracée et elle est remplacée par un fond rouge, comme le montre la figure 8.6.

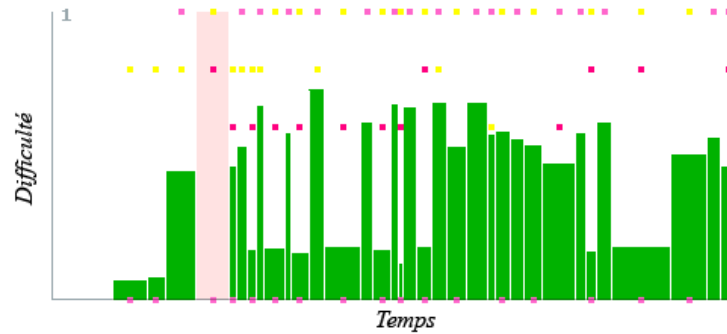


FIGURE 8.6 – Courbe de difficulté.

Les parties en rouges correspondent à des challenges pour lesquels le moteur ne peut prévoir la difficulté, par manque de valeurs ou parce qu'aucune capacité ne décrit précisément la difficulté du jeu.

Sur la courbe de difficulté sont reportées diverses informations supplémentaires. Nous avons incorporé, dans nos traces, différentes valeurs issues de questionnaires soumis au joueur à la fin de chaque challenge. Les points jaunes indiquent ainsi le niveau de plaisir rapporté par le joueur, les points magentas la difficulté rapportée par le joueur. Les points roses indiquent le résultat de chaque challenge.

8.4 Synthèse

Dans ce chapitre, nous avons présenté le prototype d'un logiciel de mesure de la difficulté. Ce logiciel permet de réaliser diverses tâches, représentées graphiquement figure 8.1. Le designer dispose d'un éditeur de scénario, qui lui permet de modéliser la suite de challenges que le joueur peut rencontrer, ainsi que la logique qui décrit leur écoulement. Une fois cet outil intégré à un moteur de jeu, cette description pourrait être intégrée plus finement au reste du code du moteur afin de faciliter le travail du designer. Le game designer dispose ainsi d'une représentation graphique de l'enchaînement des challenges du jeu, qui lui permet de visualiser le parcours du joueur et d'accéder rapidement et intuitivement aux différentes mesures réalisées.

L'intégration du langage de script Lua permet au designer de décrire à la fois le fonctionnement de l'hypergraphe de scénario ainsi que les algorithmes de calcul des capacités du

joueur. A partir de ces algorithmes, le logiciel est en mesure de mesurer la difficulté, c'est à dire la probabilité d'échec en fonction des capacités du joueur. Notre modèle de difficulté permet, dans une dernière étape, de construire la courbe de difficulté de chaque session de jeu.

Ce logiciel pourrait être amélioré de diverses manières. Par exemples, la logique du scénario décrite par le designer est ici sous exploitée. La logique de déroulement du scénario dépend en effet de deux types de variables : les variables du modèle de scénario, entièrement calculées par ce dernier, et les variables décrivant l'état du jeu, dont la valeur n'est pas prévisible. L'état des conditions de validation et d'échec des challenges peuvent être prédites à l'aide du modèle de difficulté, et l'état des liens est entièrement calculé par le moteur. Seules les conditions d'activation des challenges peuvent s'appuyer sur des variables d'état du jeu dont la valeur reste imprévisible. En conséquence, si le designer construit un scénario sans utiliser de conditions d'activation, *le déroulement du scénario devient entièrement prévisible à partir du calcul de satisfaction des liens et du modèle de difficulté*. Un tel scénario permettrait de prévoir les parcours les plus probables de certains joueurs, par exemple en fonction de leur capacités, et par conséquent la courbe de difficulté la plus probable dans ce cas. Ce type de fonctionnalité exploite pleinement la logique de scénario afin de fournir au designer des informations particulièrement utiles.

En l'état, ce prototype offre néanmoins une première vision instructive de la difficulté d'un gameplay et de son lien avec le comportement du joueur. Ce logiciel nous a permis de réaliser diverses expérimentations, que nous présentons dans le chapitre suivant.

Chapitre 9

Expérimentations

Sommaire

9.1 Objectifs de l'expérience	164
9.2 Dispositif expérimental	165
9.2.1 Choix matériels et logiciels	165
9.2.2 Gameplay et courbes de difficulté	167
9.2.3 Déroulement de l'expérience	171
9.2.4 Données recueillies	172
9.3 Résultats	173
9.3.1 Evaluation des capacités du joueur et difficulté	173
9.3.2 Difficulté calculée et difficulté théorique	182
9.3.3 Difficulté calculée et difficulté subjective	184
9.3.4 Plaisir de jouer et niveau de difficulté	185
9.3.5 Plaisir de jouer et courbe de difficulté	189
9.4 Discussion et Conclusion	190

Dans les chapitres précédents de cette thèse (chap. 6, 7 et 8), nous avons proposé un modèle de mesure de la difficulté d'un jeu vidéo. Nous avons également soulevé certaines hypothèses quant aux caractéristiques optimales de la courbe de difficulté d'un jeu vidéo (chap. 4). Dans ce chapitre, notre objectif consiste à mettre à l'épreuve nos hypothèses ainsi qu'à tester l'efficacité et la validité de notre modèle sur un cas réel. Nous présentons donc une étude expérimentale de la difficulté d'un jeu vidéo, à partir d'un gameplay réalisé par nos soins, implanté dans un moteur de jeu et soumis à un panel de joueurs.

Dans la première partie de ce chapitre, nous présentons plus en détail les objectifs de cette partie expérimentale. Nous décrivons ensuite nos choix matériels, nos choix de gameplay, puis analysons les résultats obtenus. Nous terminons ce chapitre par une discussion des résultats obtenus.

9.1 Objectifs de l'expérience

L'objectif principal de cette thèse consiste à fournir aux game designers un modèle de mesure de la difficulté d'un jeu vidéo. Ce chapitre de la thèse se consacre tout d'abord à l'étude de la validité et de l'efficacité de notre modèle en l'employant effectivement dans un cas concret. Ensuite, notre travail a consisté à proposer un certain nombre d'hypothèses quant à l'impact de la forme d'une courbe de difficulté sur le plaisir du joueur (chap.6). Nous souhaitons donc également éprouver ces hypothèses.

Nous avons ainsi mis au point un gameplay, de manière à soumettre un échantillon de joueurs à différentes courbes de difficulté. Nous avons étudié le comportement des joueurs et recueilli leur ressenti. Nous avons ainsi pu éprouver notre modèle en tentant de mesurer la difficulté de ce gameplay, tout en vérifiant également la validité de nos hypothèses, par l'étude des résultats subjectifs suscités par chaque type de courbe de difficulté.

Nous souhaitons tout d'abord montrer qu'il est effectivement possible de définir et de mesurer certaines des capacités du joueur. Nous avons proposé d'encoder la détection des capacités sous la forme d'un algorithme de traitement des traces d'évènements. Cette expérience doit démontrer que sur un gameplay suffisamment représentatif des jeux commerciaux actuellement développés, cette technique est effectivement applicable.

Nous désirons ensuite montrer que notre modèle permet de mettre en évidence le lien qui unit les capacités du joueur à la difficulté du jeu. Un tel lien confirmerait, dans le cadre de cette expérience, que notre mesure de capacité fournit bel et bien la valeur d'une variable explicative de la difficulté du gameplay.

Nous souhaitons également montrer qu'il existe un lien entre la difficulté empirique, calculée par notre modèle à partir des capacités du joueur, et la difficulté théorique que nous avons évalué lors de la création du jeu. Nous avons en effet construit notre gameplay dans l'objectif de soumettre le joueur à une certaine difficulté, que nous sommes plus ou moins capables d'évaluer a priori. Nous allons étudier si la difficulté calculée par le modèle correspond à notre vision intuitive, prévisionnelle, de la difficulté du jeu.

Ensuite, nous souhaitons comprendre si notre mesure de difficulté permet d'évaluer la difficulté subjective, ressentie et rapportée par le joueur. Nous avons intégré à notre dispositif expérimental un questionnaire permettant de recueillir la perception qu'a le joueur de la difficulté du jeu. Cette mesure doit nous permettre d'étudier le lien entre la difficulté telle que nous la calculons et la difficulté telle que le joueur la perçoit.

Pour finir, nous souhaitons vérifier l'hypothèse formulée au chapitre 6. Nous postulons dans ce chapitre qu'une courbe de difficulté optimale respecte les caractéristiques suivantes :

- une première phase de progression graduelle,
- une seconde phase de forte variabilité,
- les challenges les plus difficiles doivent à peine dépasser le niveau du joueur.

Notre expérience se déroule donc en deux phases consécutives. Une de ces phases propose une courbe de difficulté basique de suivi du niveau du joueur, tandis que l'autre propose une courbe respectant l'ensemble des caractéristiques pré-citées¹. L'ordre de ces phases est tiré au hasard en début de partie pour éviter tout effet d'ordre. Nous avons également intégré un questionnaire portant sur le plaisir ressenti par le joueur. L'étude du lien entre le type de courbe de difficulté et le plaisir rapporté par le joueur devrait nous éclairer sur la validité de notre hypothèse.

9.2 Dispositif expérimental

9.2.1 Choix matériels et logiciels

Pour réaliser cette expérience, nous devons avoir accès à un logiciel d'édition de jeu vidéo, de manière à construire notre univers et les lois qui le gouvernent. Mais au delà de l'édition des comportements des objets du jeu, nous devons également avoir accès au coeur du moteur de jeu. Nous devons en effet être en mesure d'implanter le code d'enregistrement des événements retraçant le parcours du joueur. En tant qu'académiques, l'accès aux moteurs de jeu nous est relativement restreint. Les collaborations avec les studios de jeu sont complexes à mettre au point pour ce type d'expérience. Il est en effet nécessaire de nous laisser accéder au code source du jeu, ce que les studios autorisent très rarement. Nous avons également besoin d'accéder à un jeu suffisamment abouti. Or, plus le jeu est abouti, plus le studio approche de la date de mise sur le marché et donc moins celui-ci est en mesure de consacrer du temps à des activités de recherche. Une fois le jeu terminé, l'accès au code reste problématique dans la mesure où le studio passe très rapidement à un nouveau projet. Le cycle de production d'un jeu est donc particulièrement complexe à synchroniser avec une activité de recherche.

Cependant, certains studios ont pour objectif de commercialiser leur moteur de jeu, comme c'est le cas de Valve ou d'Epic Games. Leurs moteurs de jeu sont parmi les plus performants, et gratuitement accessibles au grand public pour une utilisation non commerciale. Ces moteurs jouissent donc d'une grande popularité chez les universitaires. Nous avons choisi d'utiliser le moteur commercialisé par Epic Games, l'Unreal Engine. Ce moteur permet avant tout la réalisation de jeu de type First Person Shooter. Ce type de jeu nous convient parfaitement puisqu'il est particulièrement populaire et correspond au type d'une grande partie des jeux les plus récents. Nous avons donc réalisé un First Person Shooter à l'aide de l'Unreal Development Kit (UDK).

L'Unreal Development Kit permet de compiler des jeux exécutables par plusieurs types de plateformes. Nous avons choisi de développer notre jeu pour PC, de manière à simplifier les développements annexes nécessaires. En effet, UDK ne permet pas, à l'origine, d'enregistrer

1. Ces courbes sont décrites dans les sections suivantes

facilement une séquence d'évènements. Le seul accès possible au système de fichier est fourni par la fonction `'log()` qui permet d'ajouter un message au journal d'erreurs du moteur. Ce fichier est géré de manière automatique par le moteur, et par exemple effacé à chaque démarrage, ce qui ne nous permet pas de réaliser notre expérience. Nous avons donc développé une librairie dynamique qui nous permet d'enregistrer n'importe quel évènement dans un fichier XML unique, créé automatiquement à chaque début d'expérience. Cette librairie enrichit le langage de script d'UDK d'une fonction permettant d'accéder à ce log particulier. Cette fonction nous a permis d'éditer de nombreux scripts régissant le comportement du moteur de jeu pour rediriger les évènements utiles vers notre propre fichier de log.

Les ressources sonores et graphiques nécessaires au développement sont également très largement fournies par la version gratuite d'UDK. Ils proviennent du jeu *Unreal Tournament*, First Person Shooter développé par Epic Games. Le moteur dispose de suffisamment d'objets graphiques pour permettre de créer un environnement de jeu complet. Afin d'apporter une touche originale à l'univers et de s'éloigner de l'univers plutôt noir et violent d'Unreal Tournament, nous avons toutefois choisit d'utiliser un personnage populaire et décalé comme ennemi du joueur : le lapin crétin (Figure 9.1)². L'univers d'Unreal Tournament cible particulièrement les joueurs dits *hardcore*, les plus passionnés, et nous souhaitons élargir ce public, ce que permet en partie l'utilisation du lapin crétin.

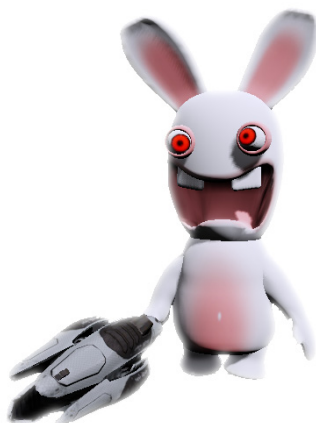


FIGURE 9.1 – L'ennemi du joueur

2. Le lapin crétin est une création originale d'Ubisoft, que nous avons importé ici uniquement à titre expérimental.

9.2.2 Gameplay et courbes de difficulté

Pour cette expérience nous avons conçu un gameplay proche du mode *Deathmatch* d'Unreal Tournament. Dans une partie de type *Deathmatch*, le joueur est face à un groupe d'ennemis. Chaque personnage du jeu, joueur ou non joueur, doit remplir les mêmes objectifs et peut s'attaquer à n'importe quel autre personnage. Une partie de ces objectifs concerne l'exploration et la collecte : se déplacer dans le **Niveau de jeu**³ pour découvrir et ramasser armes, munitions et boucliers. Tous les joueurs explorent le même terrain de jeu, et l'objectif principal consiste à tirer sur ses adversaires pour les détruire, tout en évitant de subir le même sort. A chaque fois qu'un personnage est détruit, il réapparaît à un endroit choisis aléatoirement dans le niveau de jeu, et le personnage qui l'a détruit marque un point. Le match se termine lorsqu'un score ou une durée particulière sont atteints. Le personnage avec le plus de points à la fin du match a gagné.

Nous avons apporté quelques modifications au mode *Deathmatch* d'Unreal Tournament. Nous avons modifié l'Intelligence Artificielle des personnages non joueurs, de manière à pouvoir maîtriser leur comportement. L'Intelligence Artificielle originale dispose de comportements tactiques complexes que nous souhaitons simplifier, ainsi que d'une adaptation automatique de niveau qu'il nous fallait bien sur désactiver. Le niveau d'un ennemi correspond à la précision de ses tirs, à son temps de réaction ainsi qu'au type d'actions qu'il peut réaliser. Nous avons également choisis d'assujettir la vitesse d'un ennemi à son niveau, en plus des autres paramètres déjà gérés par les scripts originaux de l'UDK, de manière à pouvoir offrir un niveau de difficulté le plus bas possible.

Dans le gameplay original d'Unreal Tournament, le niveau général des ennemis est fixé en début de partie, et chaque ennemi modifie ensuite son niveau automatiquement : à chaque fois qu'un personnage non joueur est détruit, son niveau augmente, et à chaque fois qu'il tue le joueur, son niveau diminue. Nous avons supprimé cette adaptation répartie de la difficulté pour la remplacer par un dispositif centralisé, qui permet d'attribuer un niveau à l'ensemble des ennemis du joueur en fonction de la difficulté que nous souhaitons donner au jeu.

Nous avons ensuite construit un mode de jeu plus segmenté dans le temps que ne l'est le mode *Deathmatch* original. Les personnages non joueurs sont tous des lapins crétins et sont tous ennemis du joueur, mais ne s'attaquent pas entre eux. Le joueur doit venir à bout de l'ensemble des ennemis du niveau pour marquer un point, et un ennemi détruit ne réapparaît plus automatiquement. De cette manière, le joueur affronte des vagues successives d'ennemis, ce qui nous permet d'avoir un contrôle plus précis de la courbe de difficulté. Au départ de chaque vague d'ennemis, le joueur dispose de cent points de vie, et aucun objet du niveau ne lui permet d'en récupérer. Chaque vague d'ennemis correspond ainsi à un challenge, dont la liste est fournie figure 9.2.

Ces challenges, classés figure 9.2 par ordre de difficulté que nous supposons croissante, nous permettent de composer notre courbe de difficulté. Nous soumettons le joueur à deux

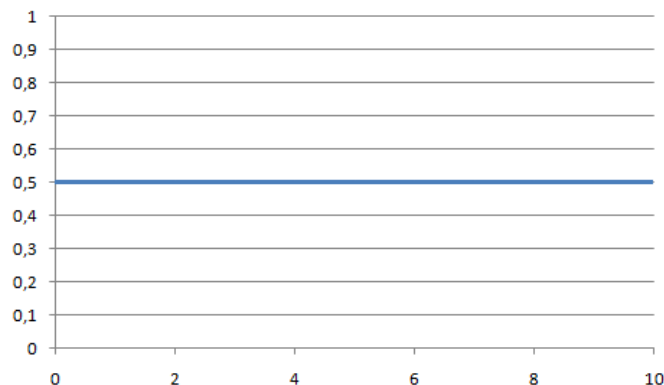
Challenge	Nombre d'ennemis	Niveau des ennemis
0	1	0
1	1	1
2	1	2
3	2	2
4	2	3
5	3	3
6	3	4
7	4	4
8	4	5
9	5	5
10	6	5

FIGURE 9.2 – Liste des challenges.

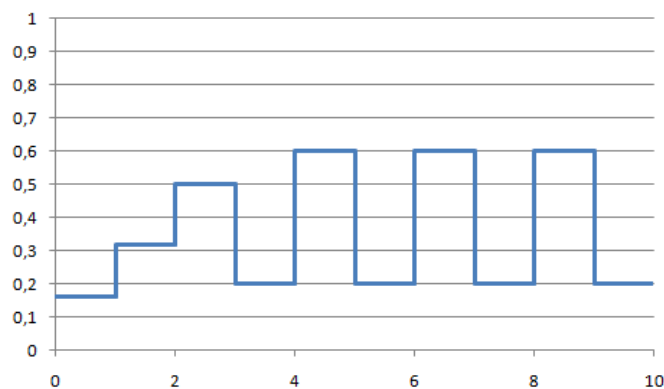
courbes différentes, pendant deux phases de jeu successives, l'ordre des phases étant aléatoire.

La première courbe de difficulté est une courbe de *suivi*, qui adapte en permanence la difficulté du jeu au niveau du joueur. Elle correspond à l'interprétation la plus directe des travaux de Mihaly Csikszentmihalyi sur le *flow* étudiée chapitre 4 [Csikszentmihalyi 91]. La courbe de difficulté relative théorique de *suivi* est représentée figure 9.3. On note que cette courbe décrit la difficulté comme nous la définissons, c'est à dire sous la forme d'une probabilité d'échec. Si le niveau du joueur et le niveau du jeu sont strictement équivalents, alors la probabilité que le joueur gagne est aussi forte que celle qu'il perde, ce qui correspond à une difficulté relative de 0.5.

Une courbe de difficulté relative qui correspond au suivi parfait du niveau du joueur est en permanence à 0.5, telle que tracée figure 9.3. Dans les faits, cette courbe est plus souvent oscillante, car la difficulté absolue du jeu ne peut évoluer pour suivre le niveau du joueur que lorsque ce changement de niveau a été constaté, et donc lorsque la difficulté relative constatée d'un challenge est différente de 0.5. Une courbe de suivi parfaite ne serait réalisable qu'en étant capable d'anticiper précisément les progrès du joueur.

FIGURE 9.3 – Courbe de difficulté relative et théorique de type *suivi*

Le second type de courbe de difficulté auquel un joueur est confronté dans notre expérience reprend les caractéristiques listées dans la section 9.1. C'est une courbe supposée optimale, avec une phase de progression graduelle suivie d'une oscillation autour du niveau du joueur. Les phases hautes d'oscillation ne dépassent que très légèrement le niveau du joueur. Cette courbe est représentée figure 9.4.

FIGURE 9.4 – Courbe de difficulté de type *progression puis oscillation*

Pour construire ces deux courbes de difficulté théoriques, nous avons évalué le niveau du joueur comme le numéro du challenge qui lui correspond le mieux. Nous mettons à jour le niveau du joueur, à chaque fois qu'il termine un challenge, selon l'algorithme présenté ci-après.

```
1 int evalNiveauJoueur(int niveauActuel, int numChallenge, string
   resultat)
```

```

2 {
3     int niveauJoueur = niveauActuel;
4
5     //Si le joueur perd alors que le challenge est à son niveau ou
6     //plus facile
7     if(resultat == "Echec")
8     {
9         if(numChallenge <= niveauActuel)
10            niveauJoueur = numChallenge-1
11    }
12
13    //Si le joueur gagne alors que le challenge est à son niveau ou
14    //plus difficile
15    if(resultat == "Valide")
16    {
17        if(numChallenge >= niveauActuel)
18            niveauJoueur = numChallenge+1
19    }
20 }

```

Nous avons construit nos courbes de difficulté en évaluant à priori les probabilités d'échec du joueur. Cette probabilité dépend de la différence entre le niveau du joueur et le numéro de challenge qu'il doit relever. Ces valeurs constituent la difficulté théorique de notre gameplay, définie à priori. Elles sont présentées figure 9.5.

Différence de niveau	Difficulté théorique
<-2	0.95
-2	0.8
-1	0.6
0	0.5
1	0.4
2	0.2
>2	0.05

FIGURE 9.5 – Difficulté théorique d'un challenge.

La différence de niveau correspond à la soustraction du niveau du challenge au niveau du joueur.

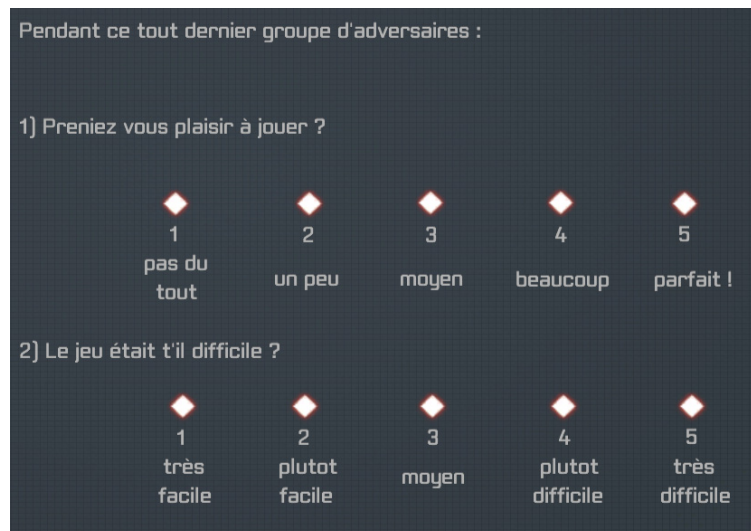
9.2.3 Déroutement de l'expérience

L'ensemble de l'expérience est réalisée dans l'univers du jeu, en limitant ainsi au maximum les interventions de l'expérimentateur. Un premier écran invite le joueur à fournir ses informations personnelles, son nom, prénom, âge et genre. Suit une fenêtre d'explication de l'interface de contrôle de l'avatar. Le joueur est ensuite invité à évoluer durant deux minutes dans l'environnement de jeu, limité pour l'instant à une petite arène centrale. Pendant toute cette phase, l'expérimentateur est à côté du joueur et lui apporte tout éclaircissement ou rappel nécessaire, si le joueur en fait la demande.

Vient ensuite la première phase de l'expérience, pendant laquelle nous évaluons une première fois le niveau du joueur. L'objectif du joueur consiste à éliminer le plus de lapin possibles en un temps limité. Les lapins apparaissent dans l'arène l'un après l'autre, et sont de niveau croissant. Leur Intelligence Artificielle est un peu particulière car ils ne choisissent jamais d'attaquer le joueur. Ils se contentent de fuir avec une agilité croissante. De cette manière, nous évaluons grossièrement la capacité du joueur, sans jamais le frustrer puisqu'il ne peut pas perdre.

À l'issue de cette phase de découverte et d'évaluation, le joueur se voit offert l'accès à l'ensemble du Niveau de jeu. Le premier challenge débute alors, avec une difficulté correspondant à une des deux courbes définies dans la section précédente. Au bout de 12 minutes, le système change de courbe. Au bout de 24 minutes, l'expérience est terminée.

Tout au long de l'expérience, à l'issue de chaque challenge, le joueur remplit un rapide questionnaire sur son plaisir et la difficulté qu'il perçoit. Ce questionnaire se compose de deux échelles de Lickert à 5 niveaux, présentées figure 9.6.



Pendant ce tout dernier groupe d'adversaires :

1) Prenez vous plaisir à jouer ?

1	2	3	4	5
pas du tout	un peu	moyen	beaucoup	parfait !

2) Le jeu était t'il difficile ?

1	2	3	4	5
très facile	plutot facile	moyen	plutot difficile	très difficile

FIGURE 9.6 – Questionnaire sur la difficulté et le plaisir du joueur

9.2.4 Données recueillies

Le système enregistre de nombreux événements qui retracent le parcours du joueur et son comportement, listés figure (Figure 9.7) page 172.

Numéro	Titre	Données
03	Player Dead	
04	Bot Dead	Name
05	Heal	Action.HealAmount
06	Player Pickup	Inv.ItemName
07	Touch	Identification
08	UnTouch	Identification
09	BornIn	TrigVolume.Identification
10	Name	Name
11	Change Weapon	NewWeapon.ItemName
12	Reset Skill	Skill :Pawn.GroundSpeed :Pawn.Name
13	Set specific level	Skill :Pawn.GroundSpeed :Pawn.Name
14	Bot Heal	Action.HealAmount
15	Damage Player	Damage :Health :by bot by player
16	Damage Bot	Damage :Health : by bot by player
17	Health Status Bot	Health
18	Health Status Player	Health
19	Damage Scaling Bot	DamageScaling
20	Damage Scaling Player	DamageScaling
21	Start of experience	
22	Possess	sBotName
23	UnPossess	sBotName :LogDead
24	Destroyed	sBotName :LogDead
25	Bot Dead	sBotName :LogDead
26	Identite	Nom :Annee :Homme :Femme
27	Report	Difficulte :Fun
28	Niveau	NivJoueur :NivJeu :NbBots :NivBots
29	Difficulte	SlowStart Oscillation Suivi
30	Start Challenge	NivJeu :NivJoueur
31	End Challenge	1 0
32	Fire	danger :cible
33	HasMoved	X :Y :Z :danger :cible
34	Danger	danger :cible

FIGURE 9.7 – Liste des événements.

Nous évaluons trois capacités du joueur à partir de la trace des différents événements : sa capacité à viser, sa capacité à rester en mouvement, et sa capacité à parcourir le niveau. La capacité du joueur à viser, sa précision, correspond au rapport entre le nombre de tirs supposés utiles qu'il effectue, c'est à dire lorsqu'il voit un ennemi ou lorsqu'un ennemi le pourchasse, et le nombre de fois ou un de ses tirs aboutit à blesser un ennemi, calculé grâce

aux trames 32 et 02. Sa capacité à rester un mouvement correspond au nombre d'évènements 33 reçus par seconde, lorsque le joueur est en danger ce dont nous informe l'évènement 34. Sa capacité à parcourir le niveau correspond au nombre de fois où le joueur change de zone par seconde, ce dont nous informe l'évènement 07, lorsqu'il est en danger, ce dont nous informe l'évènement 34.

L'évènements 27 nous fournit les résultat du questionnaire de fin de challenge, à savoir le niveau de difficulté et le plaisir perçus par le joueur.

9.3 Résultats

Nous avons réalisé trois sessions d'expérimentation. La première s'est déroulée dans notre laboratoire, au CEDRIC à Paris, et a regroupé 30 sujets. Une seconde s'est déroulée à dans les locaux de l'ENJMIN, à Angoulême, sur 35 sujets. La dernière a été réalisée au sein d'une association de joueurs⁴ et a regroupé 7 sujets. Les deux dernières expériences ont été encadrées par deux élèves ergonomes de l'ENJMIN⁵. Nous avons donc, en tout, pu étudier le comportement de 72 sujets, 56 hommes et 16 femmes. La moyenne d'âge des joueurs est de 26 ans, l'ensemble des âges étant répartis comme le présente la figure 9.8.

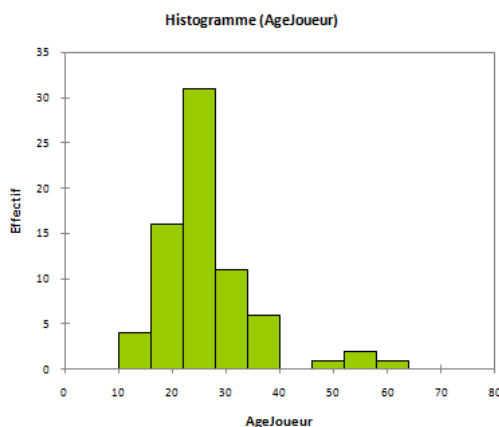


FIGURE 9.8 – Age des participants

9.3.1 Evaluation des capacités du joueur et difficulté

Chaque sujet a pu jouer à une suite de challenges, pendant les 24 minutes du test. Nous avons ainsi pu enregistrer 2401 tentatives de résolution de challenge, couronnées ou non

4. Association BLASS de Brioux sur Boutonne.

5. Merci à Joel Gueulin et Cédric Mondamert.

de succès. La figure 9.9 montre la distribution du nombre de challenges joués par joueur. Nous avons choisit de ne pas tenir compte des deux joueurs ayant enregistré un nombre de challenges anormalement bas. En effet, ces deux joueurs n'ont pas réalisé l'expérience jusqu'au bout mais ont demandé à arrêter au bout d'une dizaine de minutes. Leur abandon peut dénoter un manque de motivation et fausse donc nos analyses de performance, comme nous l'avons détaillé chapitre 6. Nous avons donc conservé 70 joueurs, c'est à dire 2368 tentatives de résolution d'un challenge.

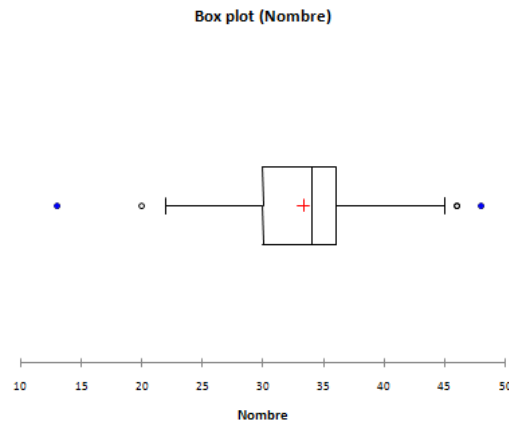


FIGURE 9.9 – Nombre de challenges joués par joueur

Pour chacune de ces tentatives, nous avons analysé, à l'aide des traces et du logiciel de calcul de la difficulté, les capacités démontrées par le joueur. Nous avons ensuite étudié le lien entre les capacités mesurées et l'échec des joueurs. Les résultats sont présentés par challenge. En effet, la réussite de chaque challenge ne dépend pas forcément des mêmes capacités. Certains challenges sollicitent plus les capacités de déplacement du joueur que d'autres, par exemple. La figure 9.10 montre la distribution du nombre de traces enregistrées pour chaque challenge.

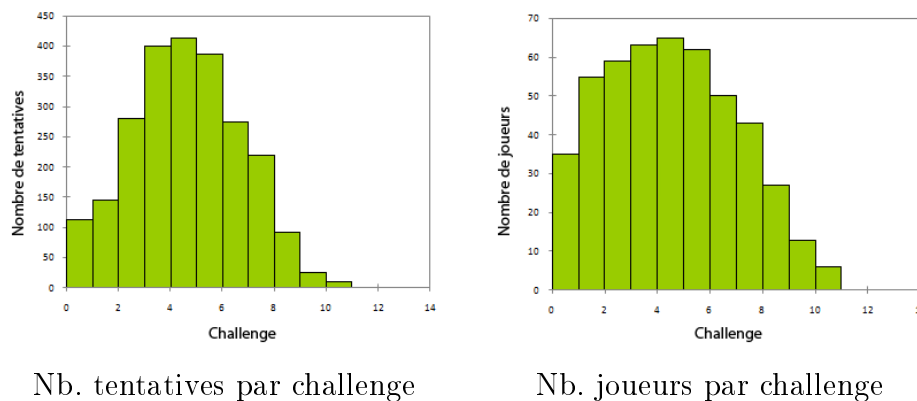


FIGURE 9.10 – Quantité de données par challenge.

Nous ne détaillons complètement ici que l'analyse du challenge le plus joué, c'est à dire le challenge 4. Nous présentons par la suite une synthèse des résultats obtenus pour l'ensemble des challenges.

9.3.1.1 Analyse détaillée du challenge 4

Nous avons analysé les traces du challenge 4 à l'aide du logiciel d'analyse de la difficulté présenté chapitre 8. Les algorithmes de mesure des capacités ont pu extraire des valeurs correspondant aux trois capacités que nous souhaitons mesurer, c'est à dire la précision des tirs, la fréquence de mouvement et la fréquence de déplacement du joueur. Pour rappel, le mouvement correspond pour nous à la capacité du joueur à ne pas rester statique, et sa capacité de déplacement correspond au fait que non seulement il bouge mais en plus se déplace de manière significative dans le niveau de jeu, et donc change de zone. Une fois ces valeurs extraites, nous avons analysé leur relation avec le résultat obtenu par le joueur.

Nous considérons les variables de capacité comme des variables explicatives du résultat, et recherchons des corrélations linéaires entre ces variables. La figure 9.11 montre les distributions des variables de capacités et de la variable de résultat. On remarque que ces distributions ne sont pas normales, ce qui proscrie l'usage des tests statistiques paramétriques. Nous avons choisi de tester nos hypothèses à l'aide d'un test d'indépendance χ^2 .

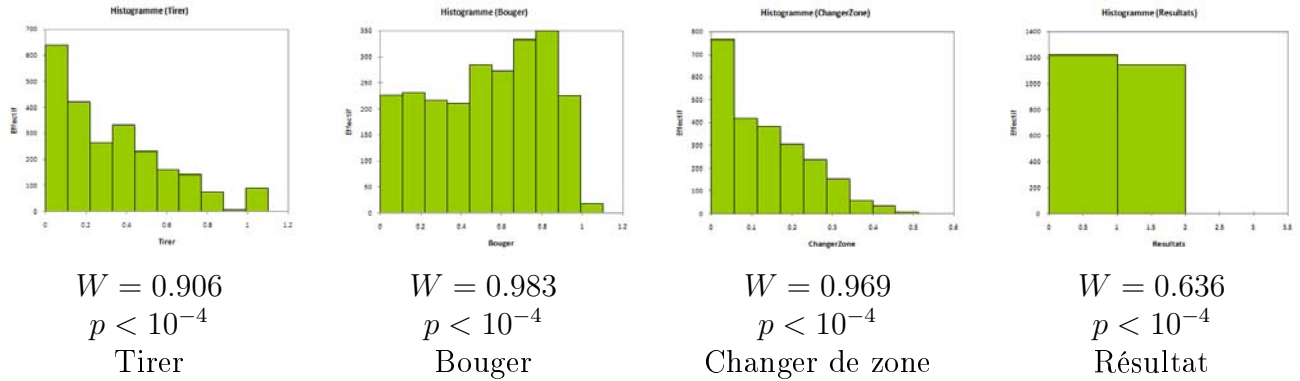


FIGURE 9.11 – Distributions des variables et tests de normalité (Shapiro/Wilks).

La figure 9.12 montre le coefficient de corrélation linéaire calculé pour les différents couples de variables. La capacité du joueur à tirer semble être la plus liée au résultat, avec une corrélation de -0.22 . Cette corrélation est faible mais la variable résultat est une variable binaire, qui reflète moins la difficulté d'un challenge que la probabilité d'échec par niveau, que nous avons établi comme notre mesure de la difficulté.

	Tirer	Bouger	Changer de zone	Résultat
Tirer	1	0.229	0.141	-0.215
Bouger	0.229	1	0.736	-0.111
Changer de zone	0.141	0.736	1	-0.068
Résultat	-0.215	-0.111	-0.068	1

FIGURE 9.12 – Tableau de corrélation capacités et résultat (Coefficient de Pearson).

Nous avons cherché à vérifier si'il existait un lien significatif entre la capacité du joueur à tirer et le résultat qu'il obtient à la fin du challenge. Cette hypothèse peut être formulée comme l'alternative de l'hypothèse nulle H_0 suivante : *La distribution du nombre de joueurs ayant gagné ou perdu est indépendante de leur précision*. Nous avons construit le tableau de contingence présenté figure 9.13 et réalisé un test χ^2 d'indépendance avec un seuil $\alpha = 10^{-2}$. Nous avons construit les classes du tableau de contingence à l'aide de l'algorithme de classification des k-moyennes.

	Gagne	Perdu	Total
Precision 1	29	32	61
Precision 2	35	60	95
Precision 3	13	18	31
Precision 4	25	10	35
Precision 5	15	26	28
Precision 6	18	10	61
Precision 7	22	11	33
Precision 8	25	10	35
Precision 9	11	12	23
Precision 10	20	12	32
Total	213	201	414

FIGURE 9.13 – Tableau de contingence entre précision et résultat.

Le résultat du test montre que nous pouvons rejeter l'hypothèse H_0 avec un risque inférieur à $3 * 10^{-4}$. Nous acceptons donc l'hypothèse alternative : *Il existe un lien entre le résultat obtenu par le joueur et sa précision au tir*. Nous avons réalisé le même test pour les autres capacités et obtenu le résultat inverse : nous ne pouvons pas rejeter H_0 pour les capacités de mouvement ($p = 0.26$) et de déplacement ($p = 0.085$) et considérer qu'il existe un lien entre ces capacités et le résultat. Seule la précision semble être liée significativement au résultat.

Il existe donc bien un lien significatif entre la capacité au tir du joueur et les résultats qu'il obtient au challenge 4. Ce lien peut par exemple être calculé comme une corrélation linéaire de -0.22 entre ces deux variables. Mais comme nous l'avons dit précédemment, la variable *résultat* reflète mal la performance du joueur, car elle la binarise. Par contre, à partir de multiples essais, nous pouvons calculer la probabilité d'échec du joueur, qui reflète plus précisément sa performance.

On notera qu'une partie des résultats a été obtenue *par le même joueur*, et qu'à ce titre, les différentes observations de la variable *résultat* ne sont pas totalement indépendantes, propriété nécessaire pour permettre dévaluer une probabilité à partir d'observations. En effet, les résultats successifs d'un même joueur sont dépendants puisque ce joueur apprend à chaque essai et donc modifie sa probabilité de réussite. Mais nous cherchons à calculer la probabilité de réussite des joueurs d'un même niveau, et satisfaisons donc l'hypothèse d'indépendance conditionnelle. En d'autres termes, nous supposons que la seule dépendance existant entre les différentes observations de la variables *résultat* vient de l'évolution du niveau du joueur, facteur que nous maîtrisons en regroupant les observations par classes de niveau similaires. Cependant, l'indépendance conditionnelle n'est totalement vérifiée que si nous tenons compte de l'ensemble des dimensions du niveau du joueur. Notre indépendance est donc approximative car nous nous contentons ici d'exprimer le niveau du joueur par sa

capacité principale, c'est à dire sa capacité à tirer pour le challenge 4. Nous contenterons de cette approximation dans la suite de l'expérience.

Nous évaluons la corrélation entre une capacité du joueur et la difficulté du challenge comme nous l'avons définie au chapitre 6, c'est à dire comme la probabilité d'échec par classe de niveau, pour cette capacité donnée. La capacité *Tirer* semble la seule liée au résultat, nous calculons donc la difficulté en fonction de cette capacité et considérons qu'elle est représentative du niveau du joueur. Pour établir les différents niveaux, nous utilisons l'algorithme de classification des k-moyennes, de manière à obtenir 10 classes de niveau. Ces classes sont représentées figure 9.14.

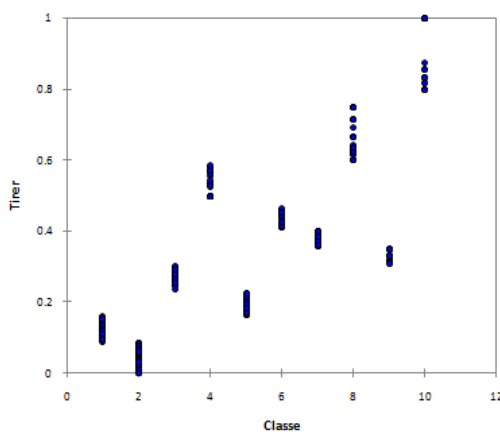


FIGURE 9.14 – Classes de niveau pour la capacité Tirer

Nous avons ensuite évalué les probabilités de réussite par classes, en calculant la moyenne des valeurs de *Résultats* au sein de chaque classe. Nous avons également représenté chaque classe de niveau de précision par la moyenne des valeurs de niveau qui la composent. Nous obtenons une corrélation linéaire de -0.76 entre la probabilité d'échec du joueur et sa capacité à tirer. La régression linéaire correspondante est tracée figure 9.15.

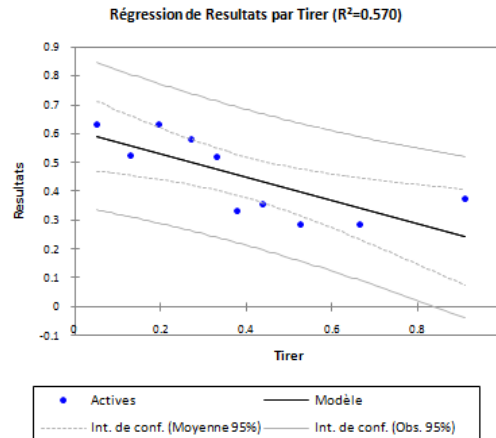


FIGURE 9.15 – Régression linéaire entre précision et difficulté

Notre méthode permet donc, dans le cadre de cette expérience, de mesurer certaines des capacités du joueur et de mettre en évidence une partie du lien qu'elles entretiennent avec la difficulté du gameplay, telle que nous la définissons. Pour le challenge 4, la capacité du joueur à tirer, sa précision, semble la plus déterminante. Dans la section suivante, nous mettons en évidence ce type de lien pour l'ensemble des challenges.

9.3.1.2 Analyse de l'ensemble des challenges

Les dix challenges qu'offrent notre gameplay ne peuvent pas, à la suite de notre expérience, être analysés avec la même précision. La figure 9.10 nous a montré que les challenges 9 et 10, par exemple, n'ont été que très peu joués, et seront donc écartés de cette analyse. En effet, notre échantillon de joueurs est centré autour d'un niveau donné. Notre gameplay s'adapte à ce niveau, en proposant des challenges dont la difficulté oscille plus ou moins fortement autour du niveau du joueur. Les challenges les plus proches du niveau moyen du groupe de joueurs sont donc les plus joués, et certains challenges ne sont quasiment jamais proposés.

La figure 9.16 montre l'ensemble des coefficients de corrélation linéaire calculés entre les variables décrivant les capacités du joueur et la performance obtenue. Les nombres en gras sont ceux pour lesquels le χ^2 d'indépendance montre que résultats et capacités ne sont pas indépendants (** $p < 0.01$, * $p < 0.05$).

Challenge	Tirer(ρ)	Bouger (ρ)	Changer de zone (ρ)
0	-0.276**	-0.057	0.008
1	-0,184	-0,068	0,019
2	-0,229**	-0,084	-0,007
3	-0.245**	-0.277**	-0.162**
4	-0,215**	-0,111	-0,068
5	-0,307**	-0,058**	0,012**
6	-0,148*	0,080**	0,139**
7	-0,195	0,121*	0,095
8	-0,071	-0,112	0,125

FIGURE 9.16 – Corrélacion capacités et résultat par challenge (Coefficient de Pearson).

Les nombres en gras sont ceux pour lesquels le test du χ^2 d'indépendance montre que résultats et capacités ne sont pas indépendants (** $p < 0.01$, * $p < 0.05$).

Les résultats obtenus semblent montrer que pour 6 challenges sur 9, dont les challenges les plus joués et de difficulté la plus centrale par rapport aux niveaux des joueurs, la précision du joueur telle que nous la calculons est liée au résultat binaire qu'il obtient. A chaque fois, cette relation peut être calculée comme une corrélation linéaire comprise entre -0.148 et -0.307 . Pour les challenges 3, 5 et 6, on observe un lien entre les capacités de mouvement, de déplacement et le résultat binaire obtenu. Ce lien n'est exprimable sous forme d'une corrélation linéaire que pour le challenge 3. Les challenges 1, 7 et 8 ne semblent mettre en évidence quasiment aucun lien entre les capacités calculées et le résultat obtenu, si ce n'est la capacité de mouvement pour le challenge 7.

Les résultats précédents montrent en synthèse qu'il est possible, pour certaines parties du gameplay seulement, de mettre en évidence un lien entre capacités du joueur et résultat obtenu. Nous avons choisit de calculer ce lien comme une corrélation linéaire entre les variables décrivant le niveau du joueur et la variable de résultat. Parmi les liens mis en évidence par le test de χ^2 , certains ne semblent cependant pas modélisables sous cette forme.

Le modèle paraît ainsi incomplet, ne détectant pas de lien significatif entre capacités et résultat pour tout les challenges, et en partie naïf, puisque la corrélation linéaire ne semble pas suffisante pour exprimer les liens mis en évidence par le test de χ^2 . Nous avons donc choisit d'évaluer la pertinence du modèle de manière plus globale, en vérifiant dans quelle mesure celui ci permet de prévoir de *nouveaux* résultats, c'est à dire dans quelle mesure les corrélations établies à partir d'une partie des données permettent de prédire la valeur des résultats de l'autre partie des données.

De manière à évaluer la robustesse du lien entre résultat et comportement du joueur, nous avons construit un modèle de prédiction de la difficulté à partir de régressions linéaires

calculées sur 75% des données récoltées. Nous avons ensuite évalué la correspondance entre le résultat obtenu par le joueur et celui prédit par le modèle, sur les 25% de données restants, c'est à dire 587 traces. Notre algorithme de prédiction utilise de manière naïve les régressions linéaires calculées pour chaque capacité du joueur : chaque capacité donc le coefficient de corrélation est supérieur à 0.1 est utilisée pour prédire la difficulté du challenge. Nous réalisons donc une prédiction par capacité, qu'il nous faut combiner pour réaliser une prédiction globale. Ces prédictions sont combinées sous la forme d'une moyenne, pondérée par leur qualité relative, c'est à dire pour nous par la valeur du coefficient de corrélation de chaque capacité. Ainsi, la prédiction réalisée à partir d'une capacité qui corrèle faiblement avec le résultat du joueur sera moins prise en compte dans la prédiction globale que celle réalisée à partir d'une capacité avec un meilleur coefficient de corrélation.

Nous obtenons une corrélation de 0.48 entre difficulté prédite et résultat obtenus. La figure 9.17 présente les distributions de difficulté prédite pour chaque résultat obtenu par le joueur, pour les 25% des données utilisées pour valider le modèle. Si nous binarisons la prédiction en considérant que toute valeur supérieure à 0.5 est égale à 1, et 0 sinon, alors sur les 25% des données utilisées pour valider le modèle, notre modèle prédit le résultat du challenge avec une précision de 70% (411 prédictions correctes sur 587 résultats). Le modèle semble plutôt pessimiste, la plupart des erreurs étant des prédictions d'échec alors que le joueur a effectivement réussi.

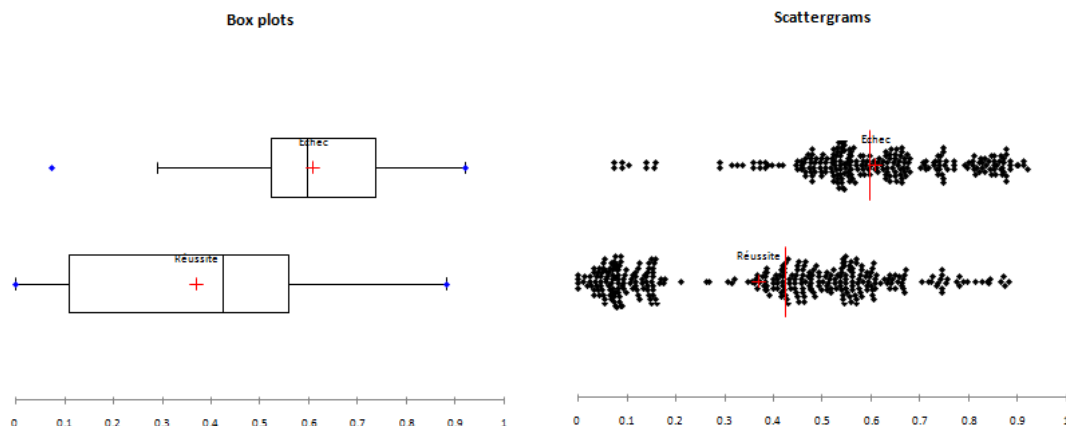


FIGURE 9.17 – Predictions de difficulté pour chaque résultat obtenu sur données de validation.

Ces premiers résultats montrent qu'il est possible de mesurer certaines des capacités du joueur ainsi que de mettre en évidence un lien entre ces capacités et le résultat obtenu. Pour une partie des challenges de cette expérience, une ou plusieurs des capacités du joueur est liée au résultat qu'il obtient lorsqu'il tente de le valider. Si nous exprimons ce lien sous forme

d'un coefficient de corrélation linéaire, alors le modèle est suffisamment robuste pour prédire le résultat du challenge avec une précision de 70%.

9.3.2 Difficulté calculée et difficulté théorique

Nous avons mis au point cette expérience à partir d'une évaluation à priori de la difficulté de chaque challenge. L'objectif principal de notre modèle consiste à permettre au game designer de confronter son évaluation de la difficulté à priori avec le résultat d'une étude expérimentale. Nous avons donc comparé la difficulté calculée par notre modèle à partir des capacités du joueur à la difficulté théorique, que nous avons évaluée comme présenté figure 9.5. La figure 9.18 montre les distributions des différentes valeurs de difficulté calculées, pour chaque niveau de difficulté théorique. Le test d'indépendance χ^2 met en évidence un lien entre difficulté théorique et difficulté calculée ($p < 10^{-4}$). Il existe également une corrélation linéaire de 0.51 entre ces deux difficultés.

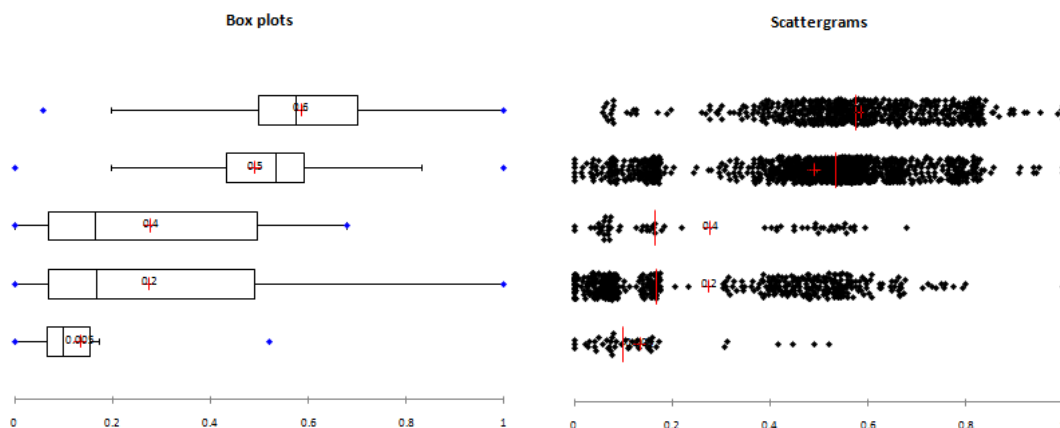


FIGURE 9.18 – Distributions des prédictions de difficulté pour chaque difficulté théorique.

En tant que designer, ces résultats nous permettent de pointer les parties de notre gameplay qui respectent le moins bien la difficulté théorique que nous souhaitons imposer au joueur. Très peu de challenges ont été joués avec une difficulté de 0.05 ou de 0.4, difficulté qui n'est proposée que lors des phases de progression graduelle de la difficulté. En effet, la courbe de suivi propose continuellement une difficulté de 0.5 et la partie oscillante de la courbe optimale propose une difficulté variant entre 0.2 et 0.6. Pour les valeurs les plus hautes, c'est à dire les difficultés de 0.5 et 0.6, la distribution des valeurs observées reste principalement groupée autour de la valeur théorique. Cependant pour la valeur de difficulté la plus basse, 0.2, on remarque que la distribution est plus étalée, avec un premier quartile

à 0.07 et un troisième quartile à 0.49, même si la médiane à 0.16 est proche de la difficulté théorique voulue.

Plusieurs erreurs dans la conception du gameplay peuvent expliquer ce résultat. Parmi ces erreurs, il en est une que nous pouvons étudier sans modifier le gameplay et réaliser une nouvelle expérience. En effet, nous pourrions sûrement améliorer notre gameplay en calculant le niveau du joueur d'une manière un peu plus stable et précise que celle utilisée pendant l'expérience. Comme le montre l'algorithme présenté section 9.2.2, nous calculons le niveau du joueur en fonction de son dernier résultats, sans tenir compte des résultats précédents. Un bon joueur victime de malchance ou à l'inverse, un joueur moins bon qui gagne par coup de chance vont voir leur niveau instantanément mis à jour et être soumis à un niveau de difficulté qui ne correspond pas à celui désiré. Nous avons donc recalculé les niveaux de difficulté théorique, toujours selon la table construite a priori (fig. 9.5) mais en se basant sur un niveau du joueur calculé comme une moyenne des 5 derniers niveaux qui lui ont été attribués. Ce calcul plus *juste* du niveau du joueur devrait permettre d'obtenir une meilleure évaluation de la difficulté théorique à laquelle il a été soumis.

Les résultats obtenus sont les suivants : le test de χ^2 montre qu'il existe un lien entre difficulté théorique et difficulté calculée ($p < 10^{-4}$). Il existe également une corrélation linéaire de 0.61 entre ces deux difficultés, supérieure à celle obtenue précédemment. La figure 9.19 montre les distributions obtenues avec la réévaluation du niveau du joueur.

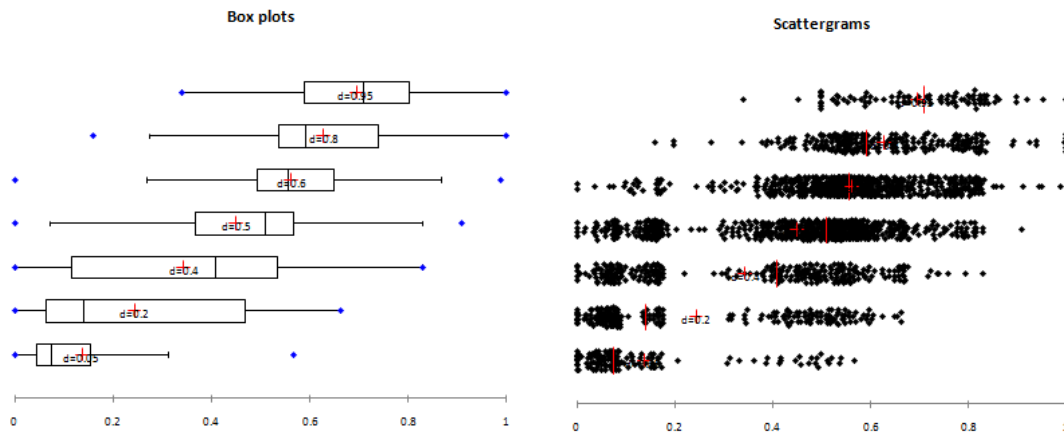


FIGURE 9.19 – Distributions des prédictions de difficulté pour chaque difficulté théorique (niveau du joueur réévalué).

Un calcul plus juste du niveau du joueur rapproche donc la difficulté théorique de la difficulté calculée par notre modèle. Ce résultat nous conforte en partie sur la validité du modèle de calcul de la difficulté : une amélioration de la précision du système de régulation de la difficulté permet d'obtenir des résultats plus proches de la difficulté calculée par notre

modèle. Réciproquement, ces résultats mettent en évidence le type de faiblesse que notre modèle est capable de mettre à jour au sein d'un gameplay.

9.3.3 Difficulté calculée et difficulté subjective

Une autre manière d'évaluer la pertinence de notre modèle consiste à vérifier à quel point la difficulté rapportée par les joueurs correspond à celle calculée par le modèle. La figure 9.20 montre les corrélations linéaires calculées entre la difficulté subjective d'une part, et le résultat obtenu par le joueur, la difficulté calculée par le modèle et la difficulté théorique⁶ d'autre part.

	Diff Subjective
Résultat	0.335**
Diff Calculée	0.318**
Diff Théorique	0.235**

FIGURE 9.20 – Tableau de corrélation de la difficulté subjective.

Le test d'indépendance du χ^2 montre un lien significatif entre les variables dépendantes et explicatives. Exprimé sous forme d'une corrélation linéaire, ce lien est le plus fort pour le résultat obtenu. En effet, on comprend facilement que le résultat final ait un impact sur la perception de la difficulté par le joueur. Ensuite, la difficulté calculée à partir du comportement du joueur corrèle au même niveau avec la difficulté subjective, de manière supérieure à la difficulté supposée à priori.

Ces résultats semblent montrer tout d'abord que la difficulté calculée par le modèle entretient un lien au moins aussi important que le résultat obtenu par le joueur sur son évaluation subjective de la difficulté, ce qui confirme en partie sa validité. Ensuite, ces résultats montrent que ce lien est plus important que celui entretenu par notre évaluation a priori de la difficulté, et que notre modèle nous permet donc d'évaluer une difficulté plus proche du sentiment du joueur que celle que nous avons évalué au départ.

Plus généralement, ces résultats montrent également que notre modèle ne permet pas d'expliquer totalement l'évaluation de la difficulté faite par le joueur. La figure 9.21 montre les distributions de difficulté calculées pour chaque niveau de difficulté subjective rapporté par le joueur. Ces résultats semblent montrer que l'évaluation de la difficulté par le joueur suit un processus plus complexe que la simple observation du résultat obtenu, ou du résultat qu'il aurait statistiquement du obtenir étant donné son comportement.

6. Cette difficulté théorique prend en compte les améliorations de la section 9.3.2 pour le calcul du niveau du joueur.

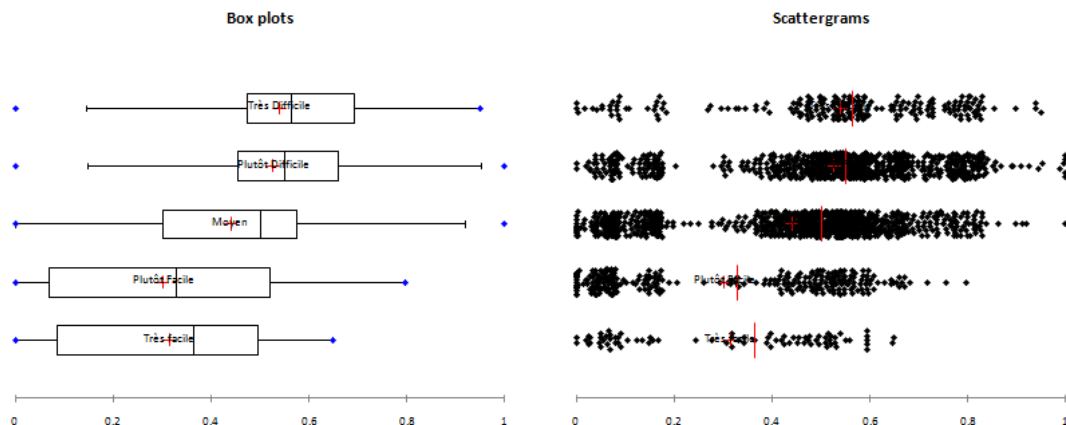


FIGURE 9.21 – Distributions des prédictions de difficulté pour chaque niveau difficulté subjective.

9.3.4 Plaisir de jouer et niveau de difficulté

Au delà de l'évaluation de notre modèle, cette expérience a pour but d'étudier la validité des hypothèses formulées au chapitre 6 et rappelées dans la section 9.1 de ce chapitre quant à la forme optimale d'une courbe de difficulté. Tout d'abord, il est intéressant d'étudier le lien qu'il existe entre le plaisir rapporté par les joueurs et la difficulté du jeu. Selon la théorie du *flow*, plus la difficulté s'écarte du niveau du joueur, c'est à dire plus le jeu devient trop dur ou trop difficile, moins le joueur est satisfait. Nous avons donc calculé le coefficient de corrélation linéaire existant entre le plaisir du joueur d'une part, et la valeur absolue des écarts de difficulté subjective, prédite et théorique par rapport à une difficulté équilibrée, c'est à dire 0.5, d'autre part. On notera cette valeur δD pour une difficulté quelconque D avec $\delta D = \text{abs}(0.5 - D)$. Selon la théorie du *flow*, plus δD est réduit et plus le joueur doit être satisfait.

La figure 9.22 montre les corrélations entre difficulté et plaisir du joueur, entre δD et plaisir du joueur, ainsi que les cases pour lesquelles le test du χ^2 a indiqué une dépendance significative. Ces résultats nous montrent que seule la difficulté subjective semble entretenir une relation linéaire avec le plaisir rapporté par le joueur. La mesure objective de difficulté ainsi que la difficulté théorique semblent être liées au plaisir du joueur de manière significative, mais ce lien n'est pas mis en évidence par le calcul d'un coefficient de corrélation. On peut donc tout d'abord uniquement noter que lorsque le joueur ressent une difficulté plus importante, son plaisir augmente légèrement.

	Plaisir		Plaisir
D Subjective	0.233**	δD Subjective	0.238**
D Prédite	0.069*	δD Prédite	-0.060*
D Théorique	0.066**	δD Théorique	-0.071**

FIGURE 9.22 – ρ entre plaisir de joueur et difficulté.

Les résultats en gras sont ceux pour lesquels le test du χ^2 d'indépendance montre qu'écart de difficulté et plaisir ne sont pas indépendants (** $p < 0.01$, * $p < 0.05$).

Les distributions de la figure 9.23 permettent de visualiser plus aisément la relation entre difficulté subjective et plaisir du joueur. On note que plus le plaisir du joueur est important, plus la difficulté est fréquemment évaluée comme importante ce qui est cohérent avec le coefficient de corrélation de 0.23 obtenu précédemment.

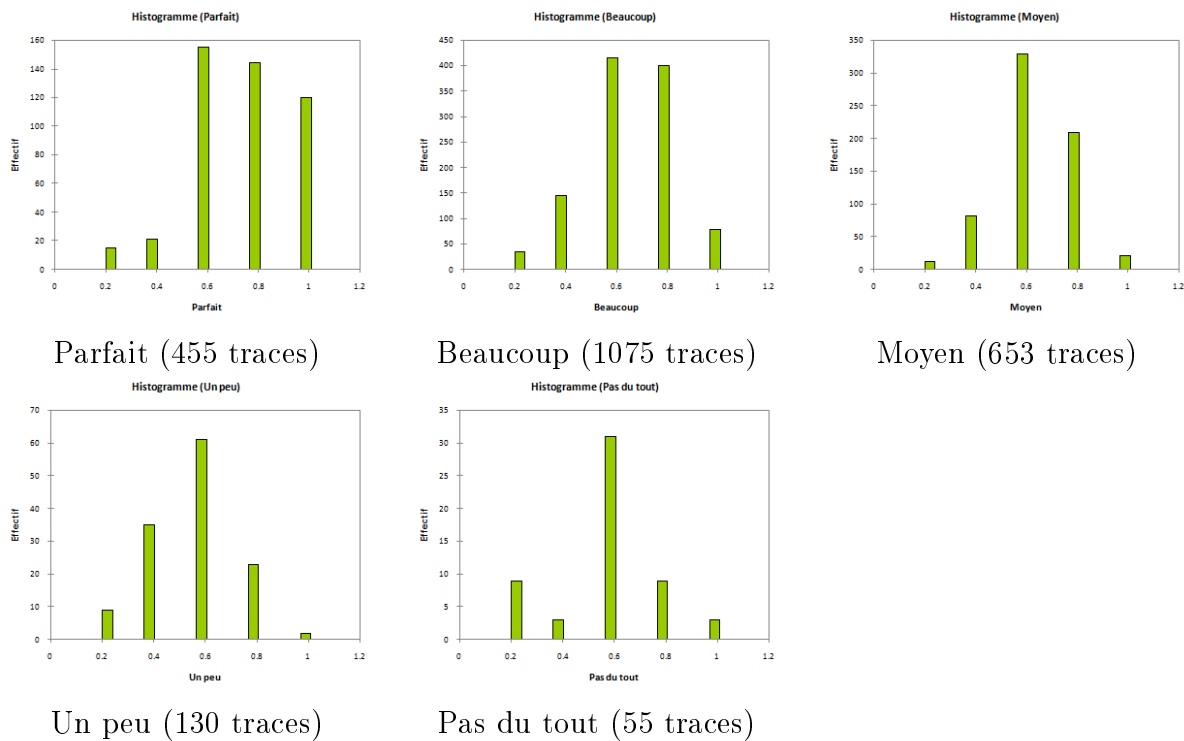


FIGURE 9.23 – Distributions des difficultés subjectives pour chaque niveau de plaisir.

Lorsqu'on étudie les distributions des difficultés objectives pour chacun des niveaux de plaisir (fig 9.24), on constate cependant le phénomène suivant : lorsque le joueur n'est pas

insatisfait, ce qui arrive dans la très grande majorité des cas⁷, on remarque qu'une partie importante de la distribution est centrée sur la difficulté équilibrée de 0.5. On remarque également que plus on s'éloigne de la note *Parfait*, plus la distribution présente un deuxième pic, centré sur les faibles difficultés. Puis lorsque le joueur s'avoue légèrement voire très insatisfait, le premier pic est plus important et le second se déplace légèrement vers la droite.

A la différence de la difficulté subjective, la distribution ne semble réellement évoluer que lorsque le joueur est insatisfait : pour les notes *Parfait*, *Beaucoup* et *Moyen*, le gros de la distribution reste centré sur 0.5. Par contre, on constate que la distribution évolue plus franchement pour les satisfactions *Un peu* et *Pas du tout*. Une corrélation linéaire calculée sur l'ensemble des données capture difficilement cette relation car les joueurs ont très rarement rapporté être insatisfaits.

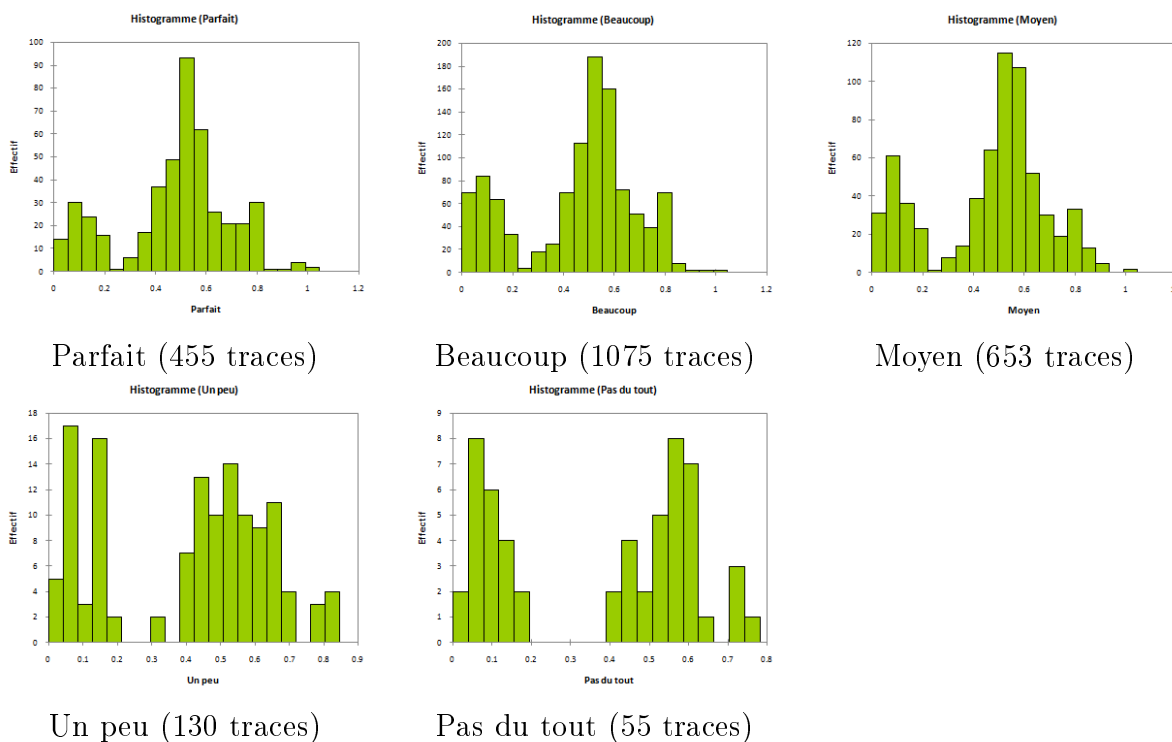


FIGURE 9.24 – Distributions des difficultés calculées pour chaque niveau de plaisir.

Si on calcule, pour chaque niveau de plaisir du joueur, la moyenne de la valeur absolue des écarts de la difficulté par rapport à 0.5, c'est à dire à quel point la difficulté s'écarte,

7. 92% des cas si l'on considère que lorsque le joueur répond *Moyen*, il n'est pas insatisfait.

en moyenne, d'une valeur équilibrée, alors la corrélation linéaire avec le plaisir est de -0.96 , mais sur seulement cinq points. La régression linéaire est présentée figure 9.25.

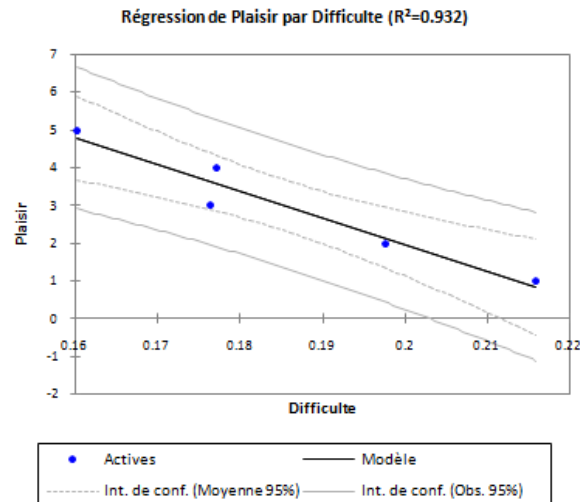


FIGURE 9.25 – Régression linéaire entre écarts de difficulté objective et plaisir.

On peut donc, en définitive, interpréter ces résultats de la façon suivante :

- On ne peut pas dire que plus la difficulté objective s'écarte de la valeur d'équilibre, moins le joueur apprécie le jeu. En l'occurrence, dans de très nombreux cas, le joueur a dit apprécier le jeu alors que la difficulté était effectivement déséquilibrée.
- On peut par contre dire que plus le joueur est insatisfait, plus les valeurs de difficultés semblent s'écarter, en moyenne, de l'équilibre. Les joueurs ont été rarement insatisfait, mais les valeurs de difficulté étaient plus extrêmes dans ce cas. De plus, il semble que ce soient les difficultés les plus basses qui évoluent le plus. Si le pic des difficultés fortes s'est légèrement déporté vers les hautes valeurs, celui des difficultés les plus basses a bien plus évolué.

On observe également que la difficulté subjective semble plus directement liée au plaisir du joueur que la difficulté objective. En effet, on observe directement une corrélation de 0.24 entre l'écart de difficulté subjective et le plaisir du joueur. De plus la corrélation positive de 0.23 entre difficulté subjective et plaisir indique que lorsque le joueur est satisfait, il indique plus souvent une difficulté importante.

9.3.5 Plaisir de jouer et courbe de difficulté

Tout au long de ce chapitre, nous avons rappelé que cette expérience pouvait nous permettre de tester nos hypothèses quand à la forme optimale d'une courbe de difficulté. Nous supposons donc en effet qu'une courbe respectant les caractéristiques suivantes est optimale :

- une première phase de progression graduelle,
- une seconde phase de forte variabilité,
- les challenges les plus difficiles doivent à peine dépasser le niveau du joueur.

Nous avons donc soumis les joueurs à deux phases de jeu de 12 minutes, chacune suivant une courbe de difficulté particulière, et ce dans un ordre aléatoire. La courbe de *Suivi* est une courbe standard, qui cherche à suivre le plus précisément possible le niveau du joueur. La courbe d'*Oscillation* respecte quand à elle les caractéristiques que nous considérons comme celles d'une courbe optimale. Durant chacune de ces deux périodes, les joueurs ont régulièrement reporté leur niveau de plaisir. La seule modification du gameplay entre ces deux phases est le changement de courbe de difficulté, ce qui nous permet donc d'évaluer l'impact de chaque type de courbe sur le plaisir du joueur.

Un test de χ^2 d'indépendance montre qu'à $p = 10^{-3}$, on peut accepter l'hypothèse selon laquelle le type de courbe influence le plaisir du joueur. La corrélation linéaire calculée entre le type de courbe et le plaisir du joueur n'est que de 0.075. La figure 9.26 donne la moyenne et l'écart type du plaisir du joueur pour chaque type de courbe de difficulté. La figure 9.27 donne les distributions du plaisir du joueur, pour chaque type de courbe de difficulté.

	Oscillation	Suivi
Moyenne	3.67	3.81
Ecart-Type	0.90	0.91

FIGURE 9.26 – Plaisir du joueur pour chaque type de courbe de difficulté.

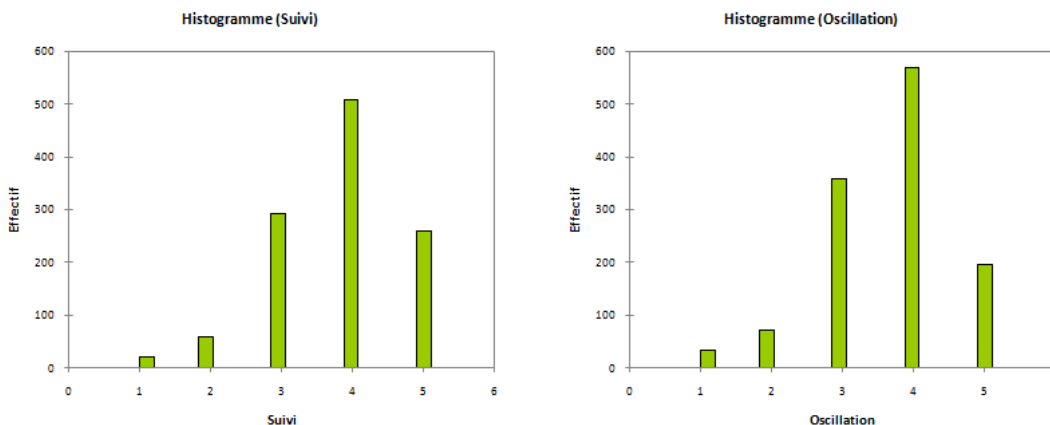


FIGURE 9.27 – Distributions du niveau de plaisir pour chaque type de courbe de difficulté.

Les différences entre les deux distributions sont assez minces, mais significatives selon le test du χ^2 . Elles vont à l’encontre de nos hypothèses : la moyenne du plaisir du joueur est plus faible dans la situation que nous considérons comme optimale. En observant les distributions, on remarque que pendant les phases d’oscillation, les joueurs ont moins répondu *Parfait* (5) et plus souvent *Beaucoup* (4) ou *Moyen* (3). En conséquence, ces données contredisent nos hypothèses et montrent que dans cette expérience, les joueurs ont préféré la courbe de suivi à la courbe d’oscillation.

9.4 Discussion et Conclusion

Nous avons tout d’abord évalué si notre modèle était capable, pour un gameplay représentatif d’une grande partie des jeux commerciaux produits aujourd’hui, d’évaluer les capacités du joueur. A partir d’un enregistrement d’événements survenus au cours de la session du jeu, nous avons été en mesure de mesurer ces capacités. Nous avons ensuite pu constater que pour 7 challenges sur 9, dont les challenges les plus joués, il existait un lien entre les capacités observées et le résultat obtenu par le joueur. Nous ne sommes par contre pas capable d’expliquer pourquoi certains challenges ne semblent pas exploiter certaines capacités. Ces résultats montrent que le joueur influe effectivement sur le résultat des challenges, mais surtout que nous pouvons mesurer en partie sa capacité à le faire.

Nous avons ensuite choisit de représenter ce lien sous forme d’un modèle linéaire. Ces régressions linéaires nous permettent de prédire une probabilité d’échec du joueur en fonction de son niveau, c’est à dire de la mesure de ses capacités. En combinant les prédictions réalisées pour chaque capacité en les pondérant par leur qualité supposée, nous parvenons à prédire le résultat du challenge avec une fiabilité de 70%, sur 25% des données conservées uniquement

pour la validation. Ce test montre que notre modèle linéaire naïf mesure bien une propriété du gameplay que nous avons construit, et non un artefact particulier des données recueillies. Il serait maintenant particulièrement intéressant d'étudier la performance de modèles non linéaires plus complexes, comme par exemple les réseaux de neurones, à prédire le résultat du joueur. Il est de plus intéressant de noter que le modèle semble avoir plus de mal à prédire le résultat des challenges les plus faciles. Nous pouvons supposer qu'il est effectivement plus difficile de prédire le résultat des challenges les plus simples, et donc soumis au joueurs les moins expérimentés, au cours desquels le comportement du joueur est désorganisé, incohérent et dont le résultat repose peut être sur une part de chance plus importante que pour les challenges de niveau plus élevé.

Nous avons pu vérifier que la difficulté théorique, c'est à dire évaluée à priori, correspondait en partie à celle calculée par le modèle. Les deux difficultés sont liées selon le test de χ^2 d'indépendance, et ont une corrélation linéaire de 0.51. De plus, l'étude de la relation entre difficulté calculée et difficulté théorique nous permet d'envisager diverses améliorations. Par exemple, les valeurs théoriques de difficulté les plus hautes (fig. 9.19) semblent correspondre à des difficultés calculées plus faibles. Nous devrions surement augmenter la différence de niveau entre joueurs et ennemis pour atteindre effectivement une difficulté de 0.95. Nous avons également pu étudier une amélioration apportée au calcul du niveau du joueur, que nous avons choisit de moyennner sur une fenêtre glissante. Cette amélioration du calcul du niveau du joueur a rapproché les valeurs de difficulté théoriques des valeurs calculées par le modèle, ce qui souligne la qualité des résultats qu'il fournit.

L'étude de la difficulté perçue par le joueur confirme également la pertinence de notre modèle. Il existe une relation significative entre difficulté subjective et calculée, et leur corrélation linéaire est de 0.32. On remarque cependant que pour les valeurs hautes et basses, l'évaluation de la difficulté par le joueur semble plus complexe que celle que fournit notre modèle. Le joueur note comme *Très Difficile* une distribution de traces qui ne semble pas très différente de celle notée comme *Plutôt Difficile*. Les distributions des traces notées *Très Facile* et *Plutôt Facile* semblent plus décalées, mais dans le mauvais sens. Il serait donc particulièrement intéressant d'étudier plus précisément de quelle manière le joueur attribue des notes de difficulté très élevées ou très basses.

L'étude du plaisir rapporté par le joueur en fonction de la difficulté relative à laquelle il est soumis confirme la théorie du flow d'une manière intéressante. Les difficultés subjectives et calculées ne semblent pas être liées de la même manière au plaisir du joueur. Le joueur semble en effet plus satisfait lorsqu'il évalue une difficulté plus importante : la distribution des difficultés subjectives des tentatives correspondant à un plaisir *Parfait* est principalement répartie sur les valeurs *Moyenne*, *Plutôt Difficile* et *Très Difficile*. Par contre, lorsqu'on observe les niveaux de difficulté calculée, on remarque que lorsque le joueur est mécontent, la difficulté calculée se répartit sur les valeurs très faibles et moyennement hautes, alors qu'elle reste principalement centrée sur 0.5 lorsque le joueur est satisfait. En synthèse, dans notre

expérience, il semble que le joueur apprécie une difficulté objective adaptée et subjective haute, et apprécie moins les difficultés objectives et subjectives faibles. Il serait d'autant plus intéressant de comprendre comment le joueur évalue les valeurs importantes de difficulté, de manière à pouvoir exploiter ces résultats : fournir une difficulté réelle adaptée tout en simulant une difficulté importante. Il est possible que la désaffection du joueur pour les difficultés basses s'explique également par une baisse des rétributions intrinsèques : pour créer un challenge simple, nous avons limité le nombre d'ennemis. Cette méthode limite sûrement également le plaisir du joueur, car le faible nombre d'ennemis signifie clairement la baisse de difficulté et limite les occasions de signifier la réussite du joueur et donc de le récompenser, ce qui arrive à chaque fois qu'il tue un ennemi.

Finalement, l'étude du plaisir du joueur en fonction du type de courbe de difficulté semble contredire nos hypothèses. La courbe de *Suivi* semble être légèrement plus appréciée que celle d'*Oscillation*. On peut expliquer ces résultats à la lumière des résultats précédents. Tout d'abord, la courbe de *Suivi* n'est pas une courbe de suivi parfaite, mais oscille autour du niveau du joueur en s'y adaptant constamment. Ensuite, la relation entre difficulté relative et plaisir du joueur semble montrer que le joueur n'apprécie pas les niveaux de difficulté trop bas, ce que notre courbe d'oscillation propose régulièrement. De plus, notre modification de la difficulté pourrait être améliorée pour permettre une modification de la difficulté objective plus difficile à percevoir, en gardant un nombre d'ennemis constant mais en modifiant leur niveau. Globalement, la courbe de suivi respecte en grande partie nos hypothèses d'optimalité : la phase de tutorial préliminaire à l'expérience permet de toute façon une phase d'ascension de la difficulté, et le suivi offre une oscillation qui ne dépasse que de très peu le niveau du joueur. De plus la courbe de *Suivi* ne lui propose pas de challenges objectivement et subjectivement trop faciles. Il serait particulièrement intéressant de réaliser à nouveau cette expérience, mais avec la courbe d'*Oscillation* respectant les hypothèses suivantes :

- une première phase de progression graduelle,
- une seconde phase de variabilité moyenne,
- les challenges les plus difficiles doivent à peine dépasser le niveau du joueur mais paraître plus difficiles.
- les challenges les plus faciles ne doivent pas paraître trop simples ni limiter les occasions d'afficher la réussite du joueur.

Nous ne sommes donc pas réellement en mesure de tester l'hypothèse selon laquelle une courbe de suivi est moins efficace qu'une courbe d'oscillation, puisqu'une courbe de suivi est aujourd'hui impossible à mettre en place. Les recherches effectuées sur les algorithmes d'adaptation de la difficulté le permettront cependant peut être un jour. En attendant, nous pouvons néanmoins essayer de construire une oscillation optimale, à partir des modifications apportées à nos hypothèses.

Chapitre 10

Conclusion

Au cours de ce travail de thèse, nous avons étudié, d'une manière générale, la question de la difficulté dans les jeux vidéo. Plus précisément, notre problématique a consisté à mettre au point des outils théoriques et un logiciel de mesure de la difficulté d'un jeu vidéo. Il nous a fallu, dans un premier temps, définir l'intérêt d'une telle mesure, déterminer ce qui motive particulièrement notre travail de thèse.

Au cours des deux premiers chapitres, nous avons montré que la difficulté, tour à tour observée dans son rapport au design d'un jeu vidéo, puis plus généralement dans son impact sur la psychologie du joueur, est un composant essentiel de l'activité vidéo ludique. Elle transparait de la notion même de jeu vidéo, qui repose en partie sur la mise en scène d'un effort, et c'est à juste titre que la plupart des études consacrées au plaisir du joueur lui desservent une place importante. Il nous a donc semblé particulièrement pertinent d'y consacrer une étude théorique, ainsi que d'en définir une méthode de mesure.

Cette étude bibliographique de la difficulté dans les jeux vidéo nous a tout d'abord permis de proposer une catégorisation de son mode d'expression. Nous avons en effet décrit la difficulté en parallèle d'un modèle général de traitement de l'information, et considéré ainsi trois dimensions de difficulté : les difficultés *perceptive*, *logique* et *motrice*. Cette catégorisation a pour but d'aider le game designer à formaliser la difficulté du gameplay qu'il conçoit, en observant son jeu tour à tour selon chacun de ces trois axes. Notre méthode de mesure demande par exemple au game designer d'identifier les *capacités* dont le joueur doit faire preuve, et cette typologie peut l'aider à y parvenir. Nous avons montré que la plupart des gameplay recensés dans les ouvrages de game design pouvaient être analysés grâce à ces trois catégories.

Cette catégorisation de la difficulté souligne également la nécessité de faire tester le jeu *par des joueurs humains*. En effet, la difficulté naît de l'ensemble des phases de traitement de l'information soumise au joueur, à savoir tout autant de la perception que de la réflexion ou de l'action. Ces phases sont toutes particulièrement complexes et quasiment impossibles

à simuler correctement. Les algorithmes utilisés pour tester un jeu en simulant le joueur sont donc forcément limités et ne peuvent évaluer la difficulté réelle du jeu. Nous privilégions donc une mesure s'appuyant sur des données issues de playtests réalisés par de vrais joueurs. Il s'en suit qu'elle doit par exemple tenir compte du fait que chaque joueur a ses propres capacités, qui influent directement sur la performance observée.

Dans un second temps, l'étude de diverses recherches consacrées au plaisir du joueur nous a permis de déduire un certain nombre d'hypothèses relatives à l'évolution de la difficulté d'un jeu vidéo. Selon ces hypothèses, une courbe de difficulté optimale respecterait les caractéristiques suivantes :

- une première phase de progression graduelle,
- une seconde phase de forte variabilité,
- les challenges les plus difficiles doivent à peine dépasser le niveau de performance du joueur.

Nous avons tenté d'évaluer ces hypothèses lors d'une phase expérimentale, sur laquelle nous reviendrons par la suite.

Nous nous sommes ensuite consacrés à l'étude expérimentale de plusieurs gameplay. Notre objectif a consisté à montrer tout d'abord par l'exemple que diverses méthodes de mesure de difficulté étaient exploitables et apportaient des résultats pertinents pour un game designer. Mais ces expériences avaient également pour objectif de nous permettre de comprendre les faiblesses des techniques couramment utilisées.

En particulier, la difficulté d'un jeu est aujourd'hui souvent mesurée comme une fonction de diverses variables du jeu, fonction construite selon l'intuition du game designer. Ce type d'heuristique pose le problème de son interprétation : nous ne sommes jamais exactement sûrs de la pertinence de la mesure, et encore moins de sa généralité : la valeur d'une difficulté équilibrée, par exemple, nous est inconnue à priori et varie d'une heuristique à une autre. Nous avons donc eu pour objectif de respecter une seconde contrainte : *offrir une valeur bien fondée, facile d'interprétation et générique de la difficulté d'un jeu vidéo.*

Nous nous sommes donc attachés à mettre au point une mesure réalisée à partir de l'observation du comportement de vrais joueurs et qui fournisse une évaluation bien fondée, intelligible et générique de la difficulté d'un jeu. Tout d'abord, nous avons défini un certain nombre d'hypothèses qui permettent de prendre en compte le facteur humain et de limiter au maximum l'impact de variables pouvant troubler notre mesure :

- Les objectifs que le joueur doit atteindre sont en apparence suffisamment variés
- Le joueur ne parvient pas à atteindre ses objectifs de manière répétée
- Les joueurs intervenant lors des playtests sont motivés : par exemple, recruter des joueurs non rémunérés, issus d'une communauté de joueurs attirés par le type de

gameplay proposé.

- Plus le gameplay du jeu sera fermé, c'est à dire limitera la capacité du joueur à créer des meta-objectifs, plus on pourra considérer que le joueurs investissent leur effort pour atteindre l'objectif du challenge et que l'on mesure bien la difficulté du challenge.

Nous avons montré que sous ces hypothèses, la difficulté, c'est à dire l'effort nécessaire pour atteindre les objectifs fixés par le gameplay, pouvait être déduite de la performance obtenue par le joueur, et était conditionnée par les capacités du joueur.

Nous avons ensuite choisit de scinder le gameplay en plusieurs unités dont nous pouvions mesurer la difficulté, les *challenges*. Chaque challenge correspond à une phase de jeu durant laquelle le joueur poursuit des objectifs particuliers, qu'un certain nombre d'obstacles l'empêchent d'atteindre. La plupart des jeux définissent clairement les objectifs du joueur de manière segmentée, et s'appuient naturellement sur une organisation en suite de challenges. Il existe en effet toujours un instant où le joueur a gagné ou perdu, ce qui marque la fin du challenge en cours. La granularité et l'organisation de ces challenges sont du ressort du game designer, mais font partie de l'essence même de tout jeu vidéo. Ce découpage est donc réalisable sur la plupart des jeux vidéo, ce qui nous permet de respecter notre contrainte de généralité.

A partir du principe de challenge, nous avons défini la difficulté comme la probabilité d'échec du joueur à ce challenge. Cette mesure est à la fois bien fondée, intelligible et générique. Elle repose sur la fondation théorique de réussite ou d'échec du joueur, et peut donc être utilisée pour n'importe quel gameplay. Elle fournit une valeur qui ne dépend pas du type de jeu observé, et reste facile à interpréter. Par exemple, on pourra constater qu'un challenge équilibré a une difficulté de 0.5, c'est à dire lorsque le joueur a autant de chances de perdre que de gagner.

Nous prenons en compte les *capacités* du joueur car la probabilité d'échec dépend tout autant des caractéristiques du challenge que de celles du joueur. Chaque joueur dispose de sa propre expérience des jeux vidéo, et cette expérience évolue à chaque fois qu'il tente de résoudre un challenge que nous lui proposons. Nous avons proposé de définir ces capacités comme des buts à *court terme*, que le joueur cherche à atteindre *de manière récurrente* tout au long du challenge, et dont l'évolution est suffisamment lente pour être considérée comme *constante au cours du challenge*. Ainsi définies, les capacités du joueur sont les probabilités d'atteindre ses buts à court terme, que nous pouvons évaluer statistiquement au cours du challenge à partir d'observations répétées. La mesure des capacités permet également d'apporter une description plus précise de la difficulté, en montrant quelles capacités influencent plus ou moins fortement le résultat obtenu par le joueur.

Une fois ce modèle définit, nous nous sommes attachés à mettre au point un logiciel de mesure de la difficulté d'un jeu vidéo. Ce travail s'est réalisé en deux phases. Tout d'abord, nous avons pu permettre au game designer de segmenter l'expérience vidéoludique qu'il

construit en une suite claire de challenges, facilement manipulable. Nous avons donc mis au point un langage graphique, qui permet tout à la fois de définir formellement et de manipuler facilement la suite de challenges que compose le gameplay. Comme nous l'expliquons par la suite, ce langage est toutefois sous-exploité car utilisé actuellement presque exclusivement dans son rôle d'interface.

Nous avons ensuite défini et implanté une architecture logicielle capable d'exploiter des traces d'évènements enregistrées lors de sessions de jeu, et d'en réaliser une étude statistique. Nous avons intégré le langage *Lua* de manière à permettre au designer de spécifier les algorithmes de mesure des capacités, et mis au point l'ensemble des interfaces permettant de manipuler la suite de challenges et de visualiser les mesures de difficulté correspondantes. Le logiciel est fonctionnel et a pu être utilisé au cours d'une expérimentation.

Pour finir, nous avons exploité le logiciel de mesure de la difficulté au cours d'une expérience concrète, et testé ainsi la validité de notre modèle ainsi que des hypothèses d'optimalité de la courbe de difficulté. Nous avons ainsi pu montrer qu'il était effectivement possible de mesurer les capacités du joueur au moyen d'un algorithme et à partir d'une trace d'évènements. Nous avons également constaté qu'un simple modèle linéaire permettait de capturer en partie le lien existant entre la mesure des capacités du joueur et le résultat obtenu par ce dernier. Au cours de cette expérience, notre modèle a correctement prévu l'issue du challenge à partir du comportement du joueur, avec une précision de 70%.

Nous avons également pu évaluer l'intérêt de notre mesure dans la détection d'erreurs d'équilibrage du gameplay. Nous avons pu constater que la difficulté théorique, que nous avons évaluée au moment du design, correspondait mieux à la difficulté calculée par notre modèle si nous employions une méthode plus *précise* d'évaluation du niveau du joueur. L'étude des corrélations entre le plaisir du joueur et la difficulté nous a permis de constater que bien que les joueurs aient été peu souvent insatisfaits, ils l'étaient plus souvent lors des challenges de difficulté basse. Cette constatation semble souligner une autre faiblesse du gameplay utilisé pour le test : nous avons limité la difficulté de manière trop apparente pour le joueur, en limitant le nombre d'ennemis. Une meilleure approche aurait consisté à laisser le gameplay en apparence intact, en gardant le même nombre d'ennemis, mais à limiter l'efficacité de chaque ennemi.

Nos hypothèses quant à l'optimalité d'une courbe de difficulté semblent à priori réfutées par cette expérience. Les joueurs n'ont pas montré de préférence pour la courbe optimale, et même légèrement préféré la courbe de suivi. Nous expliquons ce résultat de deux manières : tout d'abord, la courbe optimale propose plus souvent une difficulté faible, qui semble légèrement moins appréciée des joueurs. Ensuite, la courbe de suivi n'est pas une courbe de suivi parfaite, puisqu'elle s'adapte en permanence aux capacités du joueur, et produit donc une oscillation. De plus les deux courbes sont précédées de la phase d'évaluation préliminaire, et donc d'une progression de difficulté graduelle. Il serait particulièrement instructif de réaliser à nouveau cette expérience, en évaluant plus précisément le niveau du joueur et

sans diminuer le nombre d'ennemis lorsque la difficulté diminue.

De nombreuses améliorations peuvent être apportées à notre modèle, et susciter des recherches ultérieures. Tout d'abord, le langage de description du scénario est particulièrement sous exploité. Nous l'utilisons comme interface, afin de permettre au designer de manipuler facilement la suite de challenges qui composent la gameplay. Mais nous pourrions nous appuyer sur cette structure pour étudier plus finement la difficulté des différents chemins proposés au joueur. Le game designer pourrait fournir des requêtes plus précises, étudier la difficulté d'une suite particulière de challenges au sein du graphe de scénario. Nous pourrions également simuler un parcours probable du joueur étant donné un modèle du joueur hypothétique fixé par le designer, c'est à dire un niveau supposé pour chaque capacité.

Il serait également très intéressant d'étudier les capacités prédictives d'un modèle non linéaire, qui devrait peut être permettre de dépasser les 70% obtenus précédemment. Les test du χ^2 ont mis en évidence des liens entre capacité et résultat qu'une relation linéaire ne savait pas représenter, et qu'un modèle plus complexe parviendrait peut être à capturer.

Il serait aussi pertinent d'étudier les liens entre les capacités d'un joueur dans le contexte de différents challenges. En effet, le joueur ne démontrera pas les même capacités selon le challenge auquel il est confronté, mais nous devrions trouver des relations entre les capacités démontrées dans chaque contexte, et être ainsi par exemple capables de prédire le niveau d'un joueur pour un challenge B , sachant son niveau pour un challenge précédent A .

Une autre piste de recherche pourrait consister à déterminer la validité d'une heuristique établie de manière intuitive par le game designer. Cette heuristique peut être envisagée comme un descripteur de challenge. A la suite des playtests, nous possédons des données qui lient chaque challenge à une valeur de difficulté, pour un type de joueur donné. Ces données pourraient être exploitées par un algorithme d'apprentissage, et permettre ensuite par exemple, par généralisation, de prédire la difficulté de versions légèrement modifiées des challenges testés. Un tel outil permettrait au designer d'ajuster la difficulté des challenges sans avoir systématiquement recours à de nouveaux tests.

Une suite naturelle de nos recherches consiste donc à étendre notre modèle pour permettre l'adaptation dynamique de la difficulté d'un jeu vidéo. Certains jeux suivent une suite de challenge en partie générée à partir de challenges types, comme c'est le cas des jeux sérieux, qui proposent d'étudier un problème donné sous la forme de divers exercices, renouvelés en permanence. Il serait particulièrement utile de prédire la difficulté de ces challenges, de manière à pouvoir proposer au joueur une difficulté toujours adaptée.

Glossary

build Suite de bâtiments, d'améliorations et d'unités construites par un joueur dans un jeu de stratégie temps réel. Un build retrace les premiers instants de jeu, fournissant une méthode pas à pas pour développer la base d'un joueur.. 89

challenge Un challenge correspond à un ensemble d'objectifs que le joueur doit atteindre. L'état d'un challenge décrit l'avancée du joueur vis à vis de ces objectifs.. 16

First Person Shooter First Person Shooter, jeu de tir en vue subjective.. 17, 25, 32-34, 37, 50, 67, 70, 86, 94, 105, 121, 155, 156

game designer Le game designer conçoit les règles du jeu, il a une vision générale du gameplay, qui sera instancié, contextualisé par le level designer. Une section spéciale lui est consacré dans la partie Vocabulaire.. 13, 127

gameplay Rollings et Adams définissent le gameplay de la façon suivante : *Les challenges, ainsi que les actions que le joueur peut entreprendre pour les réussir, constituent le gameplay. D'une manière plus large, on peut considérer le gameplay comme ce que fait le joueur lorsqu'il joue, c'est à dire le type de problèmes qui lui sont posés et la manière dont il les résout..* 9, 14, 17, 20, 50, 99, 101, 129

Intelligence Artificielle Intelligence Artificielle.. 91, 157, 160

level designer Le level designer se charge d'appliquer la vision du game designer sur l'ensemble des cas concrets auxquels le joueur sera confronté. Il construit l'espace du jeu, y dispose les objets pour mettre en scène les règles définies par le game designer. Une section spéciale lui est consacré dans la partie Vocabulaire.. 13

MMORPG Massively Multiplayer Online Role Playing Game : jeu de rôle massivement multijoueurs en ligne.. 56, 105

Niveau de jeu Le niveau de jeu désigne l'espace dans lequel évolue le joueur, plus particulièrement lorsque l'univers du jeu est présenté au joueur de manière clairement segmentée, sous espace après sous espace.. 156, 160

playtest Un playtest et un processus par lequel un game designer teste un nouveau jeu, pour en éliminer les erreurs logicielles ou pour l'améliorer, avant sa mise sur le marché.. 97, 106, 124, 126, 184

PNJ Personnage Non Joueur, personnage du jeu n'étant pas l'avatar d'un joueur humain, mais dirigé par une IA. Le plus souvent associé aux personnages peuplant l'univers des jeux de rôles ou d'aventures.. 128

point and click Jeu d'aventure dans lequel l'activité principale du joueur consiste à cliquer à l'écran pour interagir.. 42

RTS Real Time Strategy : jeux de stratégie temps réel. 27

Bibliographie

- [Adams 08] Ernest Adams. *The Designer's Notebook : Difficulty Modes and Dynamic Difficulty Adjustment*. Gamasutra : <http://www.gamasutra.com/> (last access 01/2009), 2008.
- [Agogino 00] A. Agogino, K. Stanley & R. Miikkulainen. *Online Interactive Neuroevolution*. Neural Process. Lett., vol. 11, no. 1, pages 29–38, 2000.
- [Aha 04] David W. Aha & Matthew Molineaux. *Integrating learning in interactive gaming simulators*. In Challenges of Game AI : AAAI'04 Workshop Proceedings, 2004.
- [Aha 05] David W. Aha, Matthew Molineaux & Marc J. V. Ponsen. *Learning to Win : Case-Based Plan Selection in a Real-Time Strategy Game*. In Case-Based Reasoning, Research and Development, 6th International Conference, on Case-Based Reasoning, ICCBR 2005, Chicago, IL, USA, August 23-26, 2005, Proceedings, pages 5–20, 2005.
- [Andrade 05] Gustavo Andrade, Geber Ramalho, Hugo Santana & Vincent Coruble. *Extending Reinforcement Learning to Provide Dynamic Game Balancing*. In IJCAI 2005 Workshop on Reasoning, Representation, and Learning in Computer Games, pages 7–12, 2005.
- [Bandura 77] Albert Bandura. *Self-efficacy : Toward a Unifying Theory of Behavioral Change*. Psychological Review, vol. 84, no. 2, pages 191–215, 1977.
- [Bandura 03] Albert Bandura & Edwin A. Locke. *Negative Self-Efficacy and Goal Effects Revisited*. The Journal of applied psychology, vol. 88, no. 1, pages 87–99, 2003.
- [Barber 07] Heather Barber & Daniel Kudenko. *Dynamic Generation of Dilemma-based Interactive Narratives*. In AIIDE : First Artificial Intelligence and Interactive Digital Entertainment Conference, Marina del Rey, California, USA, 2007.

- [Bartle 96] Richard Bartle. *Heart, Clubs, Diamonds, Spades : Players who suit muds*. Volume 1, Issue 1, Journal of MUD Research (<http://www.mud.co.uk/richard/hcde.htm>), 1996.
- [Bates 92] Joseph Bates. *Virtual reality, art, and entertainment*. Presence : Teleoper. Virtual Environ., vol. 1, no. 1, pages 133–138, 1992.
- [Beume 08] N. Beume, H. Danielsiek, C. Eichhorn, B. Naujoks, M. Preuss, K. Stiller & S. Wessing. *Measuring flow as concept for detecting game fun in the Pac-Man game*. In Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on, pages 3448–3455, June 2008.
- [Biederman 06] Irving Biederman & Edward A. Vessel. *Perceptual Pleasure and the Brain*. American Scientist, vol. 94, pages 247–253, 2006.
- [Bohm 05] Niko Bohm, Gabriella Kókai & Stefan Mandl. *An Evolutionary Approach to Tetris*. In 6th Metaheuristics International Conference, 2005.
- [Bonet 01] J. Bonet & C. Stauffer. *Learning to play Pac-man using incremental Reinforcement Learning*. <http://www.ai.mit.edu/people/stauffer/Projects/PacMan>, 2001. Accessed 5 december 2008.
- [Bossier 07] Anne-Gwenn Bossier, Guillaume Levieux, Karim Sehaba, Axel Buendia, Vincent Corruble, Guillaume de Fondaumière, Viviane Gal, Stéphane Natkin & Nicolas Sabouret. *Dialogs Taking into Account Experience, Emotions and Personality*. In ICEC, pages 356–362, 2007.
- [Boutros 08] Daniel Boutros. *Difficulty is Difficult : Designing for Hard Modes in Games*. Gamasutra : <http://www.gamasutra.com/> (last access 01/2009), 2008.
- [Braine 90] M.D.S. Braine. Reasoning, necessity and logic : Developmental perspectives, chapitre The "natural logic" approach to reasoning, pages 135–158. Hillsdale : Lawrence Erlbaum Associates, 1990.
- [Bremond 74] Claude Bremond. Logique du récit. Seuil, Paris, 1974.
- [Byrne 04] Ed Byrne. Game level design (game development series). Charles River Media, December 2004.
- [Capa 08] Rémi L. Capa, Michel Audiffren & Stéphanie Ragot. *The interactive effect of achievement motivation and task difficulty on mental effort*. International Journal of Psychophysiology, vol. 70, no. 2, pages 144 – 150, 2008.
- [Cavazza 02] Marc Cavazza, Fred Charles & Steven J. Mead. *Character-Based Interactive Storytelling*. IEEE Intelligent Systems, vol. 17, no. 4, pages 17–24, 2002.

- [Chan 04] B. Chan, J. Denzinger, D. Gates, K. Loose & J. Buchanan. *Evolutionary behavior testing of commercial computer games*. Evolutionary Computation, 2004. CEC2004. Congress on, vol. 1, pages 125–132 Vol.1, June 2004.
- [Cole 04] Nicholas Cole, Sushil J. Louis & Chris Miles. *Using a Genetic Algorithm to Tune First-Person Shooter Bots*. In Proceedings of the 2004 congress on evolutionary computation CEC 2004, Portland OR, USA, June 19-23 2004.
- [Cowley 08] Ben Cowley, Darryl Charles, Michaela Black & Ray Hickey. *Toward an understanding of flow in video games*. Comput. Entertain., vol. 6, no. 2, pages 1–27, 2008.
- [Crawford 84] Chris Crawford. The art of computer game design. Osborne/McGraw-Hill, Berkeley, CA, USA, 1984.
- [Crawford 03] Chris Crawford. Chris crawford on game design. New Riders Games, June 2003.
- [Csikszentmihalyi 91] Mihaly Csikszentmihalyi. Flow : The psychology of optimal experience. Harper Perennial, March 1991.
- [Damasio 94] Antonio Damasio. Desartes' error : Emotion, reason, and the human brain. Avon Books, 1994.
- [Davis 05] John P. Davis, Keith Steury & Randy Pagulayan. *A survey method for assessing perceptions of a game : The consumer playtest in game design*. Game Studies, vol. 5, 2005.
- [Demasi 03] Pedro Demasi & Adriano J. de O. Cruz. *Online Coevolution for Action Games*. In International Journal on Intelligent Games & Simulation, volume 2, pages 80–88, 2003.
- [Douville 09] Brett Douville. *Opinion : Ten Tips For Managing Difficulty In Games*. Gamasutra : <http://www.gamasutra.com/> (last access 03/2009), 2009.
- [Eco 65] Umberto Eco. L'oeuvre ouverte. Seuil, 1965.
- [Falstein 05] Noah Falstein. *Understanding Fun : The Theory of Natural Funativity*. In Steve Rabin, editeur, Introduction to Game Development, pages 71–98. Charles River Media, 2005.
- [Fang 08] Xiaowen Fang, Susy Chan, Jacek Brzezinski & Chitra Nair. *Measuring Enjoyment of Computer Game Play*. In Americas Conference on Information Systems, 2008.

- [Festinger 59] L. Festinger & J. M. Carlsmith. *Cognitive Consequences of Forced Compliance*. Journal of Abnormal and Social Psychology, vol. 58, pages 203–210, 1959.
- [Gallagher 03] Marcus Gallagher & Amanda Ryan. *Learning to play Pac-Man : an evolutionary, rule-based approach*. In Evolutionary Computation (CEC '03), volume 4, pages 2462– 2469, 2003.
- [Gallagher 07] Marcus Gallagher & Mark Ledwich. *Evolving Pac-Man Players : Can We Learn from Raw Input ?* In Computational Intelligence and Games (CIG), 2007.
- [Gilovich 95] Thomas Gilovich & Victoria Husted Medvec. *The Experience of Regret : What, When, and Why*. Psychological Review, vol. 102, no. 2, pages 379–395, 1995.
- [Gomez-Martin 06] Marco Antonio Gomez-Martin, Pedro Pablo Gómez-Martín, Pedro A. González-Calero & Belén Díaz-Agudo. *Adjusting game difficulty level through Formal Concept Analysis*. In Max Bramer, Frans Coenen & Tony Allen, editeurs, AI-2006, the XXVI SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, pages 217–230, Cambridge, UK, December 2006. Springer.
- [Graepel 04] Thore Graepel, Ralf Herbrich & Julian Gold. *Learning to Fight*. In Proceedings of the International Conference on Computer Games : Artificial Intelligence, Design and Education, 2004.
- [Grodal 00] Torben Grodal. *Video Games and the Pleasures of Control*. In Media Entertainment : The Psychology of its Appeal, chapitre 11, pages 197–212. D. Zillmann & P. Vorderer, 2000.
- [Huizinga 51] Johan Huizinga. *Essai sur la fonction sociale du jeu*. Gallimard, 1951.
- [Hunicke 04] Robin Hunicke, Marc LeBlanc & Robert Zubek. *MDA : A Formal Approach to Game Design and Game Research*. In AAAI Workshop on Challenges in Game Artificial Intelligence, 2004.
- [Hunicke 05] Robin Hunicke. *The case for dynamic difficulty adjustment in games*. In Advances in Computer Entertainment Technology, pages 429–433, 2005.
- [Jimenez 09] Eduardo Jimenez. *Race Script : An Alternative to Rubber Banding*. Game AI Conference, 2009.
- [Johnson-Laird 02] P. N. Johnson-Laird & R. M. Byrne. *Conditionals : a theory of meaning, pragmatics, and inference*. Psychol Rev, vol. 109, no. 4, pages 646–678, October 2002.

- [Juul 03] Jesper Juul. *The Game, the Player, the World : Looking for a Heart of Gameness*. In Marinka Copier & Joost Raessens, editeurs, *Level Up : Digital Games Research Conference Proceedings*, pages 30–45, 2003.
- [Juul 09] Jesper Juul. *Fear of Failing? The Many Meanings of Difficulty in Video Games*. In *The Video Game Theory Reader 2*. Wolf & Bernard Perron (eds.), 2009.
- [Kirby 08] Neil Kirby. *AI as Gameplay Analysis Tool*. In *Game Programming Wisdom 4*, chapitre 1, pages 39–49. Course Technology, Cengage Learning., 2008.
- [Kirouac 94] Gilles Kirouac. *Les Emotions*. In *Traité de Psychologie Experimentale*. Presses Universitaire de France, 1994.
- [Klimmt 09] Christoph Klimmt, Christopher Blake, Dorothee Hefner, Peter Vorderer & Christian Roth. *Player Performance, Satisfaction, and Video Game Enjoyment*. In ICEC, pages 1–12, 2009.
- [Koster 05a] Ralph Koster. *A Grammar of Gameplay - Game Atoms : can games be diagrammed?* Game Designer Conference talk : <http://www.theoryoffun.com/grammar/gdc2005.htm> (last access 01/2010), 2005.
- [Koster 05b] Raph Koster. *A theory of fun for game design*. Paraglyph Press, Scottsdale, Arizona, 2005.
- [Lazzaro 04] Nicole Lazzaro. *Why We Play Games : Four Keys to More Emotion Without Story*. In *Game Developers Conference*, March 2004.
- [Lee-Urban 08] Stephen Lee-Urban, Megan Smith & Hector Munoz-Avila. *Learning Winning Policies in Team-Based First-Person Shooter Games*. In *Game Programming Wisdom 4*, chapitre 7, pages 659–671. Course Technology, Cengage Learning., 2008.
- [Lepper 73] M. R. Lepper, D. Greene & R. E. Nisbett. *Undermining Children's Intrinsic Interest with Extrinsic Rewards : A Test of the "Overjustification" Hypothesis*. *Journal of Personality and Social Psychology*, vol. 28, pages 129–137, 1973.
- [Loftus 83] Geifrey R. Loftus & Elizabeth F. Loftus. *Mind at play, the psychology of video games*. Basic Books, 1983.
- [Lucas 05] Simon M. Lucas. *Evolving a Neural Network Location Evaluator to Play Ms. Pac-Man*. In *Proceedings of the 2005 IEEE Symposium on Computational Intelligence and Games (CIG05)*, 2005.

- [Ludwig 07] Jeremy Ludwig & Art Farley. *A Learning Infrastructure for Improving Agent Performance and Game Balance*. In AIIDE 07 Workshop on Optimizing Player Satisfaction, 2007.
- [Madeira 04] Charles Madeira, Vincent Corruble, Geber Ramalho & Bohdana Ratitch. *Bootstrapping the Learning Process for the Semi-automated Design of a Challenging Game AI*. In AAAI 2004 workshop on Challenges in Game AI, pages 72–76, 2004.
- [Madeira 06] Charles Madeira, Vincent Corruble & Geber Ramalho. *Designing a Reinforcement Learning-based Adaptive AI for Large-Scale Strategy Games*. In AAAI conference on Artificial Intelligence and Interactive Digital Entertainment, pages 121–123, 2006.
- [Magerko 04] Brian Magerko & John E. Laird. *Mediating the Tension between Plot and Interaction*. In Challenges in Game Artificial Intelligence, Papers from the 2004 AAAI Workshop, pages 108–112. AAAI Press, 2004.
- [Magerko 05] Brian Magerko. *Story Representation and Interactive Drama*. In AIIDE : First Artificial Intelligence and Interactive Digital Entertainment Conference, June 1-5, 2005, Marina del Rey, California, USA, pages 87–92, 2005.
- [Malone 80a] Thomas W. Malone. *What Makes Things Fun to Learn, A Study of Intrinsically Motivating Computer Games*. Rapport technique CIS-7 (SSL-80-11), Xerox Palo Alto Research Center, Palo Alto, California, August 1980.
- [Malone 80b] Thomas W. Malone. *What makes things fun to learn? heuristics for designing instructional computer games*. In SIGSMALL '80 : Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems, pages 162–169, New York, NY, USA, 1980. ACM.
- [Malone 82] Thomas W. Malone. *Heuristics for designing enjoyable user interfaces : Lessons from computer games*. In Proceedings of the 1982 conference on Human factors in computing systems, pages 63–68, New York, NY, USA, 1982. ACM.
- [Manslow 04] John Manslow. *Using Reinforcement Learning to Solve AI Control Problems*. In AI Game Programming Wisdom 2. Charles River Media, Inc., 2004.
- [Mateas 03] Michael Mateas & Andrew Stern. *Façade : An Experiment in Building a Fully-Realized Interactive Drama*. In Game Developers Conference (GDC'03), 2003.

- [McClelland 79] James L. McClelland. *On the Time Relations of Mental Processes : An Examination of Systems of Processes in Cascade*. Psychological Review, vol. 86, no. 4, pages 287–330, Jul 1979.
- [Meyer 56] Leonard B. Meyer. *Emotion and meaning in music*. The University of Chicago Press, Chicago, 1956.
- [Mott 06] Bradford W. Mott & James C. Lester. *U-director : a decision-theoretic narrative planning architecture for storytelling environments*. In AAMAS '06 : Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, pages 977–984, New York, NY, USA, 2006. ACM.
- [Nabi 04] Robin L. Nabi & Marina Krcmar. *Conceptualizing Media Enjoyment as Attitude : Implications for Mass Media Effects Research*. Communication Theory, vol. 14, no. 4, pages 288–310, 2004.
- [Nantes 08] Alfredo Nantes, Ross Brown & Frederic Maire. *A Framework for the Semi-Automatic Testing of Video Games*. In Artificial Intelligence and Interactive Digital Entertainment Conference. AAAI, October 2008.
- [Nelson 05] Mark J. Nelson & Michael Mateas. *Search-Based Drama Management in the Interactive Fiction Anchorhead*. In R. Michael Young & John E. Laird, éditeurs, AIIDE, pages 99–104. AAAI Press, 2005.
- [Nelson 06] Mark J. Nelson, David L. Roberts, Charles L. Isbell Jr. & Michael Mateas. *Reinforcement learning for declarative optimization-based drama management*. In AAMAS '06 : Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, pages 775–782, New York, NY, USA, 2006. ACM.
- [Nelson 08] Mark J. Nelson & Michael Mateas. *Another look at search-based drama management*. In AAAI'08 : Proceedings of the 23rd national conference on Artificial intelligence, pages 792–797. AAAI Press, 2008.
- [Noveck 07] Ira Noveck, Hugo Mercier, Sandrine Rossi & Jean Baptiste Van der Henst. *Psychologies du raisonnement*, chapitre Psychologie cognitive du raisonnement, pages 40–76. De Boeck, 2007.
- [Pittman 09] Jamey Pittman. *The Pac-Man Dossier*. Gamasutra : <http://www.gamasutra.com/> (last access 01/2009), 2009.
- [Ponsen 05] Marc J. V. Ponsen, Héctor Muñoz-Avila, Pieter Spronck & David W. Aha. *Automatically Acquiring Domain Knowledge For Adaptive Game AI Using Evolutionary Learning*. In AAAI, pages 1535–1540, 2005.
- [Ponsen 06] Marc Ponsen, Pieter Spronck & Karl Tuyls. *Towards Relational Hierarchical Reinforcement Learning in Computer Games*. In Proceedings

- of the 18th Benelux Conference on Artificial Intelligence (BNAIC 2006), October 5-6, Namur, Belgium, 2006.
- [Propp 28] Vladimir Propp. *Morphologie du conte*. Seuil, 1928.
- [Rauterberg 95] Matthias Rauterberg. *About a framework for information and information processing of learning systems*. In Proceedings of the IFIP international working conference on Information system concepts, pages 54–69, London, UK, UK, 1995. Chapman & Hall, Ltd.
- [Riedl 03] Mark Riedl, C. J. Saretto & R. Michael Young. *Managing interaction between users and agents in a multi-agent storytelling environment*. In AAMAS '03 : Proceedings of the second international joint conference on Autonomous agents and multiagent systems, pages 741–748, New York, NY, USA, 2003. ACM.
- [Riedl 06a] Mark O. Riedl & Andrew Stern. *Believable Agents and Intelligent Story Adaptation for Interactive Storytelling*. In TIDSE, pages 1–12, 2006.
- [Riedl 06b] Mark O. Riedl & R. Michael Young. *From Linear Story Generation to Branching Story Graphs*. IEEE Comput. Graph. Appl., vol. 26, no. 3, pages 23–31, 2006.
- [Riedl 09] Mark O. Riedl. *Incorporating Authorial Intent into Generative Narrative Systems*. In Proceedings of the AAAI Spring Symposium on Intelligent Narrative Technologies II, Palo Alto, California, 2009.
- [Roberts 06] David L. Roberts, Mark J. Nelson, Charles L. Isbell, Michael Mateas & Michael L. Littman. *Targeting specific distributions of trajectories in MDPs*. In AAAI'06 : proceedings of the 21st national conference on Artificial intelligence, pages 1213–1218. AAAI Press, 2006.
- [Roberts 07] David Roberts, Christina Strong & Charles Isbell. *Estimating Player Satisfaction Through the Author's Eyes*. In Workshop on Optimizing Player Satisfaction at AIIDE, 2007.
- [Rollings 03] Andrew Rollings & Ernest Adams. *Andrew rollings and ernest adams on game design*. New Riders Publishing, 2003.
- [Ryan 06] Richard M. Ryan, C. Scott Rigby & Andrew Przybylski. *The Motivational Pull of Video Games : A Self-Determination Theory Approach*. Motivation and Emotion, vol. 30, no. 4, pages 344–360, decembre 2006.
- [Salen 03] Katie Salen & Eric Zimmerman. *Rules of play : Game design fundamentals*. The MIT Press, October 2003.
- [Saretto 01] C.J. Saretto & Michael Young. *Mediation in mimesis liquid narrative*. In ACSME, 2001.

- [Sgouros 99] Nikitas M. Sgouros. *Dynamic generation, management and resolution of interactive plots*. Artif. Intell., vol. 107, no. 1, pages 29–62, 1999.
- [Sherry 04] John L. Sherry. *Flow and Media Enjoyment*. Communication Theory, vol. 14, no. 4, pages 328+, 2004.
- [Soedarmadji 06] Edwin Soedarmadji. *Decentralized Decision Making in the Game of Tic-tac-toe*. In IEEE Symposium on Computational Intelligence and Games, pages 34–38, 2006.
- [Southey 05] Finnegan Southey, Gang Xiao, Robert C. Holte, Mark Trommelen & John W. Buchanan. *Semi-Automated Gameplay Analysis by Machine Learning*. In AIIDE, pages 123–128, 2005.
- [Spronck 02] Pieter Spronck. *Evolving Improved Opponent Intelligence*. In GAME-ON 3rd International Conference on Intelligent Games and Simulation, pages 94–98, 2002.
- [Spronck 05] Pieter Spronck. *Adaptive Game AI*. PhD thesis, Maastricht University, 2005.
- [Spronck 06] Pieter Spronck, Marc Ponsen, Ida Sprinkhuizen-Kuyper & Eric Postma. *Adaptive game AI with dynamic scripting*. Mach. Learn., vol. 63, no. 3, pages 217–248, 2006.
- [Spronck 08] Pieter Spronck. *Automatic Generation of Strategies*. In Game Programming Wisdom 4, chapitre 7, pages 659–671. Course Technology, Cengage Learning., 2008.
- [Sternberg 69] Saul Sternberg. *The discovery of processing stages : Extensions of Donders' method*. Acta Psychologica, vol. 30, pages 276–315, 1969.
- [Sutton 98] Richard S. Sutton & Andrew G. Barto. Reinforcement learning : An introduction. The MIT Press, March 1998.
- [Sweetser 05] Penelope Sweetser & Peta Wyeth. *GameFlow : a model for evaluating player enjoyment in games*. Comput. Entertain., vol. 3, no. 3, page 3, July 2005.
- [Szilas 99] Nicolas Szilas. *Interactive drama on computer : beyond linear narrative*. In Narrative Intelligence, Association for the Advancement of Artificial Intelligence (AAAI), Fall Symposium. AAAI Press, 1999.
- [Szilas 05] Nicolas Szilas. *The future of interactive drama*. In IE2005 : Proceedings of the second Australasian conference on Interactive entertainment, pages 193–199, Sydney, Australia, Australia, 2005. Creativity & Cognition Studios Press.

- [Szilas 07] Nicolas Szilas, Jason Barles & Manolya Kavakli. *An implementation of real-time 3D interactive drama*. Computers In Entertainment (CIE), vol. 5, no. 1, page 5, 2007.
- [Szita 06] István Szita & András Lörincz. *Learning tetris using the noisy cross-entropy method*. Neural Comput., vol. 18, no. 12, pages 2936–2941, 2006.
- [Szita 07] István Szita & András Lorincz. *Learning to Play Using Low-Complexity Rule-Based Policies : Illustrations through Ms. Pac-Man*. Journal of Artificial Intelligence Research (JAIR), vol. 30, pages 659–684, 2007.
- [Temprado 93] Jean-Jacques Temprado & Jean-Pierre Famose. *Analyse de la difficulté informationnelle et description des tâches motrices*. In INSEP-Publications, editeur, Cognition et Performance. INSEP-Publications, 1993.
- [Thureau 03] Christian Thureau, Christian Bauckhage & Gerhard Sagerer. *Combining Self Organizing Maps and Multilayer Perceptrons to Learn Bot-Behaviour for a Commercial Game*. In GAME-ON, pages 119–123, 2003.
- [Thureau 04] C. Thureau, C. Bauckhage & G. Sagerer. *Learning Human-Like Movement Behavior for Computer Games*. In Proc. 8th Int. Conf. on the Simulation of Adaptive Behavior (SAB'04), 2004.
- [Timuri 07] Timor Timuri, Pieter Spronck & H. Jaap van den Herik. *Automatic Rule Ordering for Dynamic Scripting*. In AIIDE, pages 49–54, 2007.
- [Ulam 05] Patrick Ulam, Ashok Goel, Joshua Jones & William Murdoch. *Using Model-Based Reflection to Guide Reinforcement Learning*. In IJCAI Workshop on Reasoning, Representation and Learning in Computer Games, 2005.
- [van Lankveld 08] Giel van Lankveld, Pieter Spronck & Matthias Rauterberg. *Difficulty Scaling through Incongruity*. In Proceedings of the Fourth Artificial Intelligence and Interactive Digital Entertainment Conference, Stanford, California, USA, October 22-24 2008.
- [Vega 04] Liliana Vega. *Modélisation et analyse spatiale et temporelle des jeux vidéo basées sur les réseaux de Pétri*. PhD thesis, Conservatoire National des Arts et Métiers, 2004.
- [Vorderer 03] Peter Vorderer, Tilo Hartmann & Christoph Klimmt. *Explaining the enjoyment of playing video games : the role of competition*. In ICEC

- '03 : Proceedings of the second international conference on Entertainment computing, pages 1–9, Pittsburgh, PA, USA, 2003. Carnegie Mellon University.
- [Weiner 85] Bernard Weiner. *An Attributional Theory of Achievement Motivation and Emotion*. Psychological Review, vol. 92, no. 4, pages 548–573, 1985.
- [Weiner 05] Bernard Weiner. Motivation from an attribution perspective and the social psychology of perceived competence, chapitre 5, pages 73–84. A. J. Elliot and C. S. Deweck, 2005.
- [Yannakakis 06] G. N. Yannakakis, J. Hallam & H. H. Lund. *Comparative Fun Analysis in the Innovative Playware Game Platform*. In Proceedings of the 1st World Conference for Fun 'n Games, pages 64–70, Preston, England, June 26-28 2006.
- [Yannakakis 07a] Georgios N. Yannakakis & John Hallam. *Game and Player Feature Selection for Entertainment Capture*. In IEEE Symposium on Computational Intelligence and Games, pages 244–251, Hawaii, USA, 2007.
- [Yannakakis 07b] Georgios N. Yannakakis & John Hallam. *Modeling and Augmenting Game Entertainment through Challenge and Curiosity*. International Journal on Artificial Intelligence Tools, vol. 16, no. 6, pages 981–999, 2007.
- [Yannakakis 07c] Georgios N. Yannakakis & John Hallam. *Towards Optimizing Entertainment in Computer Games*. Applied Artificial Intelligence, vol. 21, no. 10, pages 933–971, 2007.
- [Yannakakis 09] Georgios N. Yannakakis & John Hallam. *Real-time Game Adaptation for Optimizing Player Satisfaction*. IEEE Transactions on Computational Intelligence and AI in Games, vol. 1, no. 2, pages 121–133, June 2009.
- [Young 99] R. Michael Young. *Notes on the Use of Plan Structures in the Creation of Interactive Plot*. In Narrative Intelligence, Association for the Advancement of Artificial Intelligence (AAAI), Fall Symposium. AAAI Press, 1999.
- [Young 01] R. Young. *An Overview of the Mimesis Architecture : Integrating Intelligent Narrative Control into an Existing Gaming Environment*. In Artificial Intelligence and Interactive Entertainment, Association for the Advancement of Artificial Intelligence, Spring Symposium, AAAI Press, 2001.

- [Young 04] R. Michael Young. *An architecture for integrating plan-based behavior generation with interactive game environments*. Journal of Game Development, vol. 1, no. 1, pages 51–70, 2004.

Résumé :

Cette thèse a pour objectif de donner une définition générale et mesurable de la difficulté du gameplay dans un jeu vidéo. Elle propose une méthode et un outil pour mesurer cette difficulté. La méthode de mesure couramment employée est en effet principalement heuristique et propre au contexte de chaque jeu. Nous proposons une approche générique d'analyse du gameplay qui prend en compte l'apprentissage du joueur et permet une évaluation statistique de la difficulté d'un gameplay. Dans un premier temps, la thèse explore les liens entre difficulté, game design, et plaisir de jouer. Nous étudions diverses formes de difficultés : sensorielles, logiques et motrices. Après diverses expérimentations d'analyse automatique de gameplay, nous détaillons notre modèle de mesure de la difficulté, et l'analyse en challenges et capacités d'un gameplay, ainsi que le logiciel associé. Finalement, nous présentons une expérience, dont l'objectif est de tester la faisabilité et la précision de notre modèle.

Mots clés :

game design, mesure de difficulté, analyse de gameplay, outil de game design

Abstract :

The goal of this thesis is to propose a general and measurable definition of the difficulty in video games. The current approach, widely adopted, is mainly heuristic, and depends on each game's context. We propose a generic way to analyse a gameplay, taking into account the player's apprenticeship, which allows to statistically evaluate the gameplay's difficulty. The thesis first explores the links between difficulty, game design and the player's enjoyment. Then, we study different types of difficulties, sensory, logical and motor. After a few experiments on automatic gameplay analysis, we detail our measurement model, based on the splitting of gameplay into challenges and capacities. We present the developed software, and report an experiment that we ran to test the feasibility and accuracy of our measuring technique.

Keywords :

game design theory, difficulty measuring, gameplay analysis, game design tool