



HAL
open science

Conception et développement d'interactions immersives pour jeux sérieux

Guillaume Loup

► **To cite this version:**

Guillaume Loup. Conception et développement d'interactions immersives pour jeux sérieux. Environnements Informatiques pour l'Apprentissage Humain. Le Mans Université, 2017. Français. NNT : 2017LEMA1041 . tel-01820652

HAL Id: tel-01820652

<https://theses.hal.science/tel-01820652v1>

Submitted on 22 Jun 2018

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.

Thèse de Doctorat

Guillaume LOUP

*Mémoire présenté en vue de l'obtention du
grade de Docteur de Le Mans Université
sous le sceau de l'Université Bretagne Loire*

École doctorale : *Mathématiques et Sciences et technologies de l'information et de la
communication (MathSTIC)*

Discipline : Informatique (Section CNU 27)

Spécialité :

Unité de recherche : Laboratoire d'Informatique de l'Université du Mans (LIUM)

Soutenue le 4 décembre 2017

Thèse N° : 2017LEMA1041

Conception et développement d'interactions immersives pour jeux sérieux

JURY

Rapporteurs : **Jean-Pierre JESSEL**, Professeur, Université Paul Sabatier-Toulouse 3
Domitile LOURDEAUX, Maître de conférences HDR, Université de Technologie de Compiègne

Examineurs : **Philippe FUCHS**, Professeur, École des Mines ParisTech,
Daniel MESTRE, Directeur de recherche, CNRS, Université de la Méditerranée

Directeur de Thèse : **Sébastien GEORGE**, Professeur, Université du Mans

Co-encadrante de Thèse : **Audrey SERNA**, Maître de conférences, INSA de Lyon

Remerciements

Je souhaite tout d'abord remercier mon directeur de thèse, Sébastien George, non seulement pour la confiance qu'il m'a accordée sans laquelle cette thèse n'aurait pas été possible mais surtout pour son soutien sans faille, ses relectures multiples et ses conseils avisés. Je remercie également Audrey Serna, actuelle responsable du projet ANR JEN.lab et co-encadrante de la thèse, pour avoir su m'impliquer dans les différentes phases du projet, ainsi que pour ses précieux conseils.

Je remercie vivement Jean-Pierre Jessel et Domitile Lourdeaux de m'avoir fait l'honneur de rapporter cette thèse, ainsi que pour le temps consacré et l'intérêt apporté à la lecture de ce manuscrit.

Je tiens à remercier Philippe Fuchs et Daniel Mestre de m'avoir fait l'honneur de faire partie de ce jury et de l'intérêt qu'ils ont voulu porter à mon travail.

Un grand merci aux doctorants pour tous leurs conseils utiles et leurs marques de soutien quotidiennes : Quentin, Inès, Zeyneb, Vincent, Aïcha, Esteban, Damien et Aous. Merci également à tous les membres de l'équipe IEIAH du LIUM m'ayant offert un cadre de travail particulièrement agréable, et à ceux de l'équipe SICAL du laboratoire du LIRIS.

J'exprime ma reconnaissance aux enseignants, Frank Sauret, Jérôme Canteloube et Matthieu Nué, du lycée Aristide Briand avec qui j'ai pris plaisir à collaborer dans le cadre de la conception et de l'expérimentation des JENP, objets de cette thèse. Je remercie l'ensemble des participants aux expérimentations, à savoir les élèves 1ère STI2D du lycée Aristide Briand, les étudiants de la licence professionnelle MMI de l'IUT de Laval et les étudiants du master MT3D des Arts et Métiers ParisTech, pour le temps qu'ils ont bien voulu accorder sans contrepartie.

Je remercie plus particulièrement ma femme Emilie pour ses encouragements et son aide précieuse, et mes enfants pour leur calme et leur patience en particulier dans les périodes de rédaction.

Glossaire

Apprentissage authentique : L'apprentissage authentique est une approche pédagogique qui permet aux apprenants d'explorer, de discuter et d'élaborer, de façon concrète, des concepts et des relations. Le contexte se doit d'être composé de problèmes réels et de projets pertinents pour l'apprenant. (Donovan *et al.*, 1999)

Jeux Épistémiques Numériques (JEN) : Jeux sérieux traitant des problèmes non-déterministes, complexes, pluridisciplinaires dans un contexte authentique et au travers d'une communauté (Loup *et al.*, 2015).

Jeux Épistémiques Numériques Pervasifs (JENP) : Jeux sérieux cumulant les caractéristiques des JEN et de JP (Loup *et al.*, 2016)

Jeux Pervasifs (JP) : Le jeu pervasif est un genre de jeu qui repousse systématiquement le périmètre spatial, temporel et social du jeu traditionnel (Montola, 2005).

Situation complexe : Un problème complexe requiert des ressources internes et externes à l'individu. Il ne peut être résolu en le divisant en sous-problèmes simples et indépendants. La situation complexe est susceptible de pouvoir évoluer durant le processus de résolution. Il est également possible que plusieurs objectifs soient fixés pour sa résolution et que certains de ces objectifs doivent être pondérés et soient même antagonistes (Funke, 2010).

Tables des matières

Chapitre 1 : Introduction	13
1.1 Contexte.....	13
1.1.1 RMV, EIAH et IHM.....	13
1.1.2 Projet JEN.lab.....	15
1.2 Problématique de recherche et objectifs	17
1.3 Approche	21
1.4 Plan de thèse	22
PARTIE 1 : Conception et Expérimentation d'un Jeu Épistémique Numérique Pervasif.....	23
Chapitre 2 : Vers des jeux épistémiques numériques pervasifs	25
2.1 État de l'art des JEN	25
2.1.1 Définitions des JEN.....	25
2.1.2 Caractéristiques des JEN	27
2.1.3 Apports et Limites	29
2.2 État de l'art des Jeux Pervasifs	29
2.2.1 Définitions	29
2.2.2 Caractéristiques des jeux pervasifs	30
2.2.3 Apports et limites	34
2.3 Proposition : Jeux Épistémiques Numériques Pervasifs.....	35
2.3.1 Définition des jeux épistémiques numériques pervasifs	36
2.3.2 Caractéristiques théoriques des JENP	37
2.3.3 Apports potentiels	40
2.4 Synthèse.....	41
Chapitre 3 : Prototypage et évaluation d'un JENP.....	43
3.1 Prototype RearthM3	43
3.1.1 Synopsis du JENP	43
3.1.2 Objectifs pédagogiques et scénario	44
3.1.3 Implémentation de l'immersion et la persistance.....	46
3.2 Évaluation de l'apport du JENP	48
3.2.1 Hypothèses de recherche	48

3.2.2	Protocole expérimental.....	49
3.2.3	Évaluation de la motivation.....	50
3.2.4	Evaluation de l'engagement	51
3.2.5	Conclusion.....	61
3.3	Réflexion méthodologique sur la co-conception de JENP	62
3.3.1	Méthodologie des jeux sérieux immersifs.....	62
3.3.2	Proposition pour l'aide à la conception d'un JENP	65
3.4	Synthèse.....	74
PARTIE 2 : Faciliter le développement d'une interaction immersive .		77
Chapitre 4 : État de l'art sur les outils de développement d'interaction		
.....		81
4.1	Besoins des développeurs.....	81
4.1.1	Évolution du développement d'interaction	81
4.1.2	Vers un besoin pour tous types de développeurs	83
4.2	Environnements de développement pour applications	
immersives	85
4.2.1	Plateforme de développement dédiée à la réalité virtuelle	
	85	
4.2.2	Moteurs de jeu équipés de bibliothèques de fabricants..	87
4.2.3	Moteurs de jeu équipés de <i>Middleware</i>	89
4.2.4	Moteurs de jeu équipés d'outils d'assistance	91
4.2.5	Solutions hybrides	93
4.3	Synthèse.....	95
Chapitre 5 : Modélisation d'une interaction 3D immersive.....		99
5.1	Interaction 3D	99
5.2	Modélisation de l'interaction.....	101
5.3	Proposition d'un métamodèle pour les interactions immersives	
	106	
5.3.1	Représentation visuelle de l'interaction	107
5.3.2	Représentation scriptée	108
5.4	Synthèse.....	110
Chapitre 6 : Conception et évaluation de MIREEDGE		111
6.1	Besoins identifiés et choix de conception.....	111
6.1.1	Facilité d'utilisation.....	112

6.1.2	Capitalisation du code	113
6.1.3	Partage de connaissances	114
6.1.4	Réutilisation des algorithmes	115
6.1.5	Capacités d'évolution	115
6.1.6	Synthèse	116
6.2	Caractéristiques de MIREEDGE	117
6.2.1	Programmation Visuelle.....	119
6.2.2	Outil communautaire pour la création de composants d'interactions	123
6.2.3	Génération de script	124
6.2.4	Implémentation de MIREEDGE.....	125
6.3	Évaluation de MIREEDGE.....	126
6.3.1	Objectif.....	126
6.3.2	Méthode.....	127
6.3.3	Résultats	130
6.3.4	Satisfaction et autres retours subjectifs	136
6.4	Limites et évolutions	137
6.5	Synthèse.....	139
	Bilan et perspectives.....	141
7.1	Contributions	141
7.1.1	Apports pour la recherche en environnements informatiques pour l'apprentissage humain	141
7.1.2	Apports pour la recherche en réalité mixte et virtuelle	142
7.2	Limites et Améliorations	144
7.2.1	JENP.....	144
7.2.2	MIREEDGE.....	146
7.3	Perspectives de recherche.....	149
7.3.1	Expérimentations des JENP	149
7.3.2	Métamodèle de MIREEDGE	149
7.3.3	Réingénierie par MIREEDGE	150
	Références	151
	Production personnelle	161
	Publications et communications scientifiques.....	161

Ateliers	162
Démonstrations.....	162
Salons	162
Annexes	163
Annexe 1 : Métamodèle détaillé de MIREEDGE.....	163
Annexe 2 : Extrait de l’algorithme de génération de code pour Unity3D v5 en C#	164
Annexe 3 : Questionnaire de satisfaction.....	167

Liste des figures

Figure 1 : Principale interface de Land Science.....	27
Figure 2 : Un prototype du projet SEGAREM permettant à l'apprenant d'utiliser un objet physique (un pistolet à colle) face à un écran qui représentera la colle virtuellement appliquée.	33
Figure 3 : JENP, une extension spatiale, temporelle et sociale des JEN	36
Figure 4 : Jeux sérieux, épistémiques et pervasifs	36
Figure 5 : Interface de programmation de l'astromobile dans RearthM3	45
Figure 6 : Interface d'observation pour voir l'environnement de l'astromobile.....	45
Figure 7 : Journal de bord des événements de l'astromobile	46
Figure 8 : Taux de terrain exploré par élève selon le groupe non-pervasif	53
Figure 9 : Taux de terrain exploré par élève selon le groupe pervasif	53
Figure 10 : Statistiques sur les simulations de chaque élève du groupe non-pervasif.....	55
Figure 11 : Statistiques sur les simulations de chaque élève du groupe pervasif	56
Figure 12 : Statistiques sur les transmissions de chaque élève du groupe non-pervasif.....	57
Figure 13 : Statistiques sur les transmissions de chaque élève du groupe pervasif	58
Figure 14 : Durée passée dans chaque module par étudiant du groupe non-pervasif.....	59
Figure 15 : Durée passée dans chaque module par étudiant du groupe pervasif	60
Figure 16 : Les 7 étapes pour concevoir les jeux sérieux selon Marfisi (2010)	63
Figure 17 : Utilisation de jen cards en séance de co-conception pour le projet JEN.lab.....	67
Figure 18 : Représentation de toutes les cartes technologiques	68
Figure 19 : Procédure des séances de créativité d'un JENP	70
Figure 20 : Phase 2 des séances de créativité d'un JENP	71
Figure 21 : Etape 3 de la Phase 2 pour le prototype RearthM3	72
Figure 22 : Phase 3 des séances de créativité d'un JENP	73
Figure 23 : Processus de développement d'un JENP sans outil dédié	78
Figure 24 : Cycle du Hype selon Gartner en 2017.....	79
Figure 25 : Programmation d'interactions en environnement de développement dédié à la réalité virtuelle.....	86

Figure 26 : Programmation d'interactions sous Virtools	87
Figure 27 : Programmation d'interactions dans un moteur de jeu avec des bibliothèques de fabricants de dispositifs RMV	89
Figure 28 : Programmation des interactions dans un moteur de jeu avec un middleware RMV	90
Figure 29 : Paramétrage des périphériques sous MiddleVR	91
Figure 30 : Programmation d'interaction dans un moteur de jeu muni d'outils d'assistance RMV	92
Figure 31 : Interface de paramétrage RUIS sous Unity3D	93
Figure 32 : Exemple de modèle d'interactions gestuelles avec le formalisme IRVO+ (Delomier, 2013)	103
Figure 33 : Exemple de modèle de comportement avec le formalisme VR-Wize (Pellens <i>et al.</i> , 2007)	104
Figure 34 : Exemple de modèle d'interaction avec le formalisme VR-Wize (Pellens <i>et al.</i> , 2007)	104
Figure 35 : Représentation simplifiée du métamodèle d'interaction	107
Figure 36 : Programmation visuelle et génération de script, une solution hybride avec MIREEDGE	118
Figure 37 : Représentation d'un bloc MIREEDGE	119
Figure 38 : Exemple d'algorithme sous MIREEDGE	121
Figure 39 : Processus de développement communautaire d'une interaction immersive sous MIREEDGE.....	124
Figure 40 : Interface utilisée pour réaliser la tâche 1	129
Figure 41 : Les principales attractions entre les groupes et les tâches réalisées par les participants	133

Liste des tableaux

Tableau 1 : Principales différences entre le prototype JEN et JENP ..	47
Tableau 2 : Résultats sur la motivation (n=effectifs, m=moyenne, et=écart type).....	50
Tableau 3 : Comparaison des outils de développement RMV selon les besoins des développeurs non-RMV	95
Tableau 4 : Nombre de participants par groupes expérimentaux en fonction du degré de complexité de la tâche	131
Tableau 5 : Temps de réalisation de la tâche simple (en secondes)...	135
Tableau 6 : Temps de réalisation de la tâche modérée (en secondes)	135
Tableau 7 : Moyenne, écart-type, min et max des scores obtenus aux échelles de Likert à 10 points (22 questions du <i>CSUQ</i>).....	136

Chapitre 1 : Introduction

Les jeux vidéo, initialement destinés au divertissement, ont été utilisés au fil de ces dernières décennies dans d'autres domaines. Ainsi pour le domaine de l'apprentissage, les ressorts ludiques de ces jeux vidéo ont été associés à des scénarios pédagogiques pour mener au concept de jeux sérieux. Une multitude d'études a été menée sur leurs conceptions et leurs apports pédagogiques. Cependant, ces jeux étant directement liés à des technologies, leurs évolutions sont continues. Les scénarios et les interactions peuvent ainsi devenir toujours plus riches mais leur conception et leur développement peuvent en contrepartie se complexifier. Or une nouvelle approche pédagogique ne prendra toute sa valeur qu'à partir du moment où elle sera efficace et accessible à l'enseignant et l'apprenant. Ainsi nos travaux portent sur une nouvelle forme de jeux sérieux tout en prenant en compte les problématiques liées à sa conception et son développement.

Ce premier chapitre a pour objectif de présenter le contexte de nos travaux de recherche ainsi que la problématique de la thèse et l'approche utilisée.

1.1 Contexte

1.1.1 RMV, EIAH et IHM

Cette thèse s'inscrit au carrefour du domaine de recherche de la Réalité Mixte et Virtuelle (RMV), des Environnements Informatiques pour l'Apprentissage Humain (EIAH) et de l'Interaction Homme-Machine (IHM).

Selon Fuchs *et al.* (2006a), la finalité de la réalité virtuelle est de permettre à une personne (ou à plusieurs) une activité sensori-motrice dans un monde artificiel, qui soit imaginaire, symbolique ou une simulation de certains aspects du monde réel. Le domaine de la RMV est quant à lui un champ scientifique qui implique des disciplines aussi variées que l'électronique, l'intelligence artificielle mais également l'ergonomie et les facteurs humains. Les travaux de recherches portent sur la conception matérielle et logicielle des dispositifs, jusqu'à leur implémentation et leurs évaluations. Même si la RMV est souvent présentée comme un outil de divertissement sur le marché du grand public, de nombreuses perspectives s'ouvrent pour la santé et l'éducation.

Selon Tchounikine (2009), les EIAH sont des artefacts conçus pour susciter ou accompagner un apprentissage. Le domaine des EIAH est quant à lui un champ scientifique qui implique des disciplines aussi variées que le génie logiciel et l'intelligence artificielle en informatique que la psychologie et la didactique en sciences humaines et sociales. Les recherches peuvent porter aussi bien sur la conception des EIAH que sur leur mise en œuvre et leur évaluation.

Par l'essor des technologies RMV et de nombreuses études, un sous ensemble des EIAH a fait son apparition, les Environnements Virtuels pour l'Apprentissage Humain (EVAH). Ce type d'application, en permettant aux apprenants d'interagir avec des objets virtuels, apporte de nombreux avantages par rapport aux environnements réels. Ainsi les avantages majeurs des EVAH sont la réalisation de tâches en toute sécurité, au travers d'un large panel de configurations scénaristiques et dans un espace limité (Lourdeaux, 2001). Les limites technologiques des EVAH telles que le manque de qualité visuelle et d'interaction, ont mené à de nouveaux travaux sur des méthodes spécifiques pour la conception,

le développement et l'évaluation en IHM. Ainsi des méthodes de conception centrée utilisateur (Loup-Escande *et al.*, 2013) sont proposées à des processus spécifiques à la RMV. Des évaluations de l'impact sur l'apprentissage dans un jeu sérieux RM peuvent être également menées aux travers d'études comparatives (George *et al.*, 2014).

1.1.2 Projet JEN.lab

Cette thèse a été financée par l'Agence Nationale pour la Recherche dans le cadre du projet JEN.lab (ANR-13-APPR-0001) démarré en 2014. Ce projet est consacré aux situations d'apprentissage ludiques qui conduisent l'apprenant à résoudre des problèmes complexes, pluridisciplinaires et non déterministes. Ce concept a été introduit par Shaffer (2006) sous le terme de Jeu Épistémique Numérique (JEN). L'objectif de ce projet est d'ouvrir de nouvelles perspectives aux JEN en élargissant l'authenticité du contexte et des interactions à l'aide de nouvelles technologies. Ainsi la coopération des laboratoires LIUM, LIRIS, ICAR, Institut Français de l'éducation et l'entreprise Symetrix doit permettre de formaliser de nouvelles méthodologies de conception et d'évaluation en collaboration avec les enseignants et formateurs. Le projet a également pour objectif initial de formaliser un modèle de JEN du point de vue des technologies à employer et des interactions permises par la situation. La méthodologie de recherche utilisée dans ce projet est celle de la recherche collaborative orientée par la conception ou *Design-Based Research* (Sanchez et Monod-Ansaldi, 2015). Des prototypes ont été conçus pour différents établissements de formation, en collaboration avec des enseignants de lycées.

Les établissements LPA La Martellière, LEG Germaine Tilion, LPA Bel Air et le LEGTA de la Côte Saint André ont permis la co-conception et

l'expérimentation du prototype *Insectophagia*. Ce jeu permet aux apprenants de mener un processus de création d'entreprise de vente d'insectes. À l'aide d'une suite d'outils en ligne, les joueurs doivent effectuer des recherches entomologiques sur les insectes ainsi que sur l'emplacement de l'usine et les énergies utilisées par leur entreprise. La dernière étape du jeu consiste à concevoir et présenter un spot publicitaire afin d'élire l'entreprise la plus prometteuse. La nature des problèmes traités incite l'apprenant à se positionner et à mener sa réflexion à la fois sur la vente et sur le respect du développement durable. Chacune des étapes du processus de ce jeu est animée par différents ressorts ludiques tels que l'utilisation de badges, de cartes et de jetons.

Le lycée Aristide Briand de Saint-Nazaire, quant à lui, permis la co-conception et l'expérimentation du prototype de jeu *Rearth* dédié à la conquête spatiale. Au travers de nombreuses missions, les apprenants sont invités à utiliser des concepts en mathématiques, physiques, développement durable et programmation pour la recherche, l'exploration et la colonisation d'une exoplanète. Là encore, les apprenants doivent associer plusieurs compétences pour la résolution de problèmes. Les apprenants sont également rattachés à différentes communautés ayant chacune des points de vue différents sur les solutions envisageables. Ils peuvent régulièrement se documenter afin d'effectuer des choix argumentés. Dans ce scénario proche de la science-fiction, les interactions jouent un rôle central pour l'authenticité du contexte.

Les expérimentations du projet *JEN.lab* ont déjà été menées en milieu écologique à 3 reprises sur chaque site.

1.2 Problématique de recherche et objectifs

Selon Shaffer (2006), l'épistémologie étant l'étude de la connaissance, elle évolue constamment et chaque époque possède des épistémologies très différentes. Shaffer estime que l'épistémologie de beaucoup de nos écoles semble encore structurée pour les épistémologies de l'ère industrielle, avec pour objectif de créer de la richesse grâce à la production en série. Il perçoit l'école comme un lieu dans lequel nous sommes encore invités à réfléchir comme des ouvriers d'usine. Selon lui, les réponses seraient encore majoritairement justes ou fausses, et les étudiants y seraient limités à suivre les instructions, qu'ils aient compris ou non.

A l'ère numérique, les apprenants sont voués à produire de l'information sous différents formats et doivent apprendre à résoudre des problèmes en construisant et interprétant des modèles dynamiques de processus du monde réel (Jenkins *et al.*, 2009). L'extension d'Internet et l'arrivée des ordinateurs dans les foyers et les écoles, a permis la conception d'applications pour de nombreuses méthodes pédagogiques (Kebritchi et Hirumi, 2008). Ainsi, à l'aide d'un scénario pédagogique, des applications telles que les jeux sérieux permettent de combiner la communication, l'information et l'enseignement avec des ressorts ludiques issus du jeu vidéo (Alvarez, 2007). Shaffer a expérimenté une catégorie de jeux sérieux, nommés jeux épistémiques numériques (JEN) permettant de confronter l'apprenant à des situations complexes en tant que membre d'une profession ou une communauté.

Dans ce cas, une situation est dite complexe lorsqu'elle requiert des ressources internes telles que les connaissances, le vécu, les capacités et

la culture mais également les ressources externes telles qu'une base de données, des outils ou des méthodes particulières. La situation complexe ne peut être résolue en la divisant en problèmes simples et indépendants. De plus, la situation complexe est susceptible de pouvoir évoluer durant le processus de résolution. Il est également possible que plusieurs objectifs soient fixés pour résoudre la situation complexe, et que certains de ces objectifs doivent être pondérés et soient même antagonistes (Funke, 2010).

L'objectif pédagogique des JEN, est que les informations acquises durant le jeu persistent au-delà du cadre de l'enseignement. Dans ces JEN, les apprenants collaborent, le plus souvent, au travers d'une plateforme en ligne, accessible à tout moment. Cette plateforme facilite la collaboration des apprenants mais se limite souvent à une interface WIMP¹. Il est cependant important de se demander si ce choix de technologie ne limite pas l'intérêt pédagogique du JEN.

En effet, nous basculons aujourd'hui dans ce qui est appelé la révolution industrielle 4.0. L'industrie 4.0 représente les nouvelles formes d'intelligence artificielle et l'internet des objets (Posada *et al.*, 2015). Dans ce contexte, des études montrent que les formateurs d'ingénieurs devront préparer des scénarios dans des mondes virtuels, d'une autre façon qu'ils rédigeaient des diaporamas numériques en « Industrie 3.0 » (Richert *et al.*, 2016). Pour cela, des mondes virtuels tel que Second Life ont prouvé au travers de nombreuses études leurs potentiels pédagogiques concernant la simulation de contextes particuliers, l'appartenance à une communauté, la découverte d'une culture ou même d'une identité (Warburton, 2009). Ainsi la persistance, l'immersion et les relations sociales possibles dans ce type d'environnement permettent

¹ WIMP : Windows, Icons, Menus and Pointing device

à l'utilisateur de procurer un sentiment de présence dans ce monde virtuel ainsi que de présence auprès des autres utilisateurs (Schultze et Michael Leahy, 2009). Des expérimentations ont démontré la possibilité de créer des situations d'apprentissage plus motivantes et engageantes à l'aide de ces mondes virtuels (Warburton, 2009). Cependant il semble que les effets positifs soient liés aux nouvelles pédagogies qui peuvent être mises en place plus qu'au seul concept de monde virtuel (Richards et Taylor, 2015). En effet sans une pédagogie adaptée, le monde virtuel peut également inciter à la distraction voire mener à une surcharge cognitive sans intérêt pédagogique.

Au-delà de ces mondes virtuels, Montola (2005) regroupe l'ensemble des jeux numériques aux dimensions spatiale, temporelle et sociale étendues sous le terme de jeux pervasifs. Nous étudierons le concept de jeux épistémiques numériques pervasifs (JENP) afin de déterminer si l'immersion, la persistance et de nouveaux moyens de communications pourraient influencer les apports pédagogiques des JEN.

Plusieurs sous problèmes sont directement liés aux enseignants et formateurs.

- Quel type d'apport pédagogique, l'usage de technologies immersives et persistantes peuvent-elles fournir aux JEN ?
- Comment permettre à des formateurs non-initiés aux technologies immersives d'effectuer des choix de conception favorables aux JENP ?

Même si la co-conception d'un JENP représente une expérience nouvelle aussi bien pour les enseignants, les concepteurs que les développeurs. Pour que les JENP deviennent accessibles à un grand nombre

d'apprenants, il semble indispensable que leur réalisation ne soit pas réservée exclusivement à des spécialistes de la RMV. Ainsi nous nous intéresserons plus particulièrement aux outils permettant à des développeurs de jeux sérieux non-initiés à la RMV (développeurs non-RMV) de réaliser des interactions immersives. Il est à noter que même si la communauté RMV et celle du jeu vidéo se rapprochent progressivement et s'influencent mutuellement (Zyda, 2005), elles n'ont pas fusionnées pour autant. Ainsi malgré la production importante des casques immersifs, les développeurs de jeux sérieux et les développeurs RMV produisent encore difficilement des jeux sérieux immersifs. Les développeurs de jeux sérieux, aujourd'hui nombreux, ont le plus souvent la capacité à programmer l'ensemble de ce nouveau type d'application à l'exception des interactions immersives qui réclament des compétences spécifiques.

Notre approche s'intéresse donc à déterminer si, à défaut de pouvoir leur fournir une interaction adaptée à chaque contexte, un outil pourrait leur permettre de programmer l'algorithme correspondant sans avoir recours à des compétences RMV.

Pour répondre aux besoins des développeurs non-RMV, de nombreuses solutions ont déjà été proposées. Beaucoup de ces outils simplifient la conception des interactions immersives en proposant une liste prédéfinie d'interactions à paramétrer. Ainsi cette simplification est souvent fondée sur le concept de réutilisation d'algorithmes ou d'abstraction de périphériques à l'aide d'un *framework* RMV (Steed, 2008). Pour aller au-delà du seul paramétrage, les interfaces de certains outils proposent la solution de programmation visuelle par bloc (Tramberend, 1999). Pour prendre en compte un plus grand nombre de périphériques RMV, une architecture spécifique du système de développement est recommandée

(Blach *et al.*, 1998). Pour cela, plusieurs solutions s'appuient sur le concept d'*entité-composant* (Fischbach *et al.*, 2017).

La solution InTml de Figueroa *et al.* (2008) propose un cumul de ces recommandations dans le but de faciliter la collaboration entre deux communautés : les concepteurs RV et les développeurs RV. S'en inspirer et l'adapter aux besoins de la communauté des développeurs de jeux sérieux non-RMV semble être une piste pertinente.

1.3 Approche

Le projet JEN.lab dans lequel la thèse s'inscrit, nous permet de réaliser trois cycles de co-conception, développement et évaluation, à l'aide d'une méthodologie dite « agile ». Ainsi la nature itérative permet de répondre progressivement aux besoins définis au cours des cycles précédents. La qualité est évaluée à chacune des itérations. Les cycles doivent également s'inscrire dans une temporalité permettant d'adapter l'objet au besoin.

Pour traiter les problématiques liées aux formateurs, nous avons organisé des ateliers de co-conception pour la création de nouveaux jeux. Nous leur avons demandé de choisir des interactions liées à des technologies émergentes en fonction de leurs besoins. Nous avons nous-même développé le prototype de JENP RearthM3 conçus lors des ateliers. Ce développement a permis d'identifier le niveau d'expertise RMV requis. Le jeu a ensuite été expérimenté en salle de classe afin d'évaluer auprès d'apprenants l'apport pédagogique de ce type de jeu. Durant le développement de nouvelles versions de jeux, nous avons pu définir plus précisément les verrous pour les développeurs. Le nouvel outil facilitant le développement d'interaction immersive a alors été expérimenté et validé auprès de développeurs aux profils variés.

1.4 Plan de thèse

Le chapitre suivant est dédié à l'état de l'art sur les JEN ainsi que sur les jeux dits pervasifs. De plus, le nouveau concept de jeu épistémique numérique pervasif y sera également présenté.

Le chapitre 3 aborde les problématiques de conception d'un JENP par l'enseignant. Ainsi une nouvelle méthode de co-conception, s'appuyant sur les concepts du chapitre précédent sera évoquée. De plus, un prototype co-conçu sera présenté ainsi que les résultats de son expérimentation en milieu écologique.

La seconde partie du manuscrit sera dédiée aux problématiques du développeur pour réaliser des JENP. Le chapitre 4 est consacré à l'état de l'art sur les outils permettant le développement de ce type de jeu, ainsi que les compétences requises pour leurs utilisateurs.

Le chapitre 5 est consacré à la proposition d'un métamodèle d'interactions immersives adapté au développement des jeux sérieux.

Dans le chapitre 6, nous proposons l'outil MIREEDGE, s'appuyant sur les concepts présentés dans les chapitres 4 et 5. L'expérimentation, permettant d'évaluer son apport, y est également détaillée.

Le dernier chapitre permet de synthétiser les apports de l'ensemble de nos travaux avant d'évoquer les perspectives ouvertes.

PARTIE 1 : Conception et Expérimentation d'un Jeu Épistémique Numérique Pervasif

Chapitre 2 : Vers des jeux épistémiques numériques pervasifs

2.1 État de l’art des JEN

2.1.1 Définitions des JEN

Depuis de nombreuses années, des outils pédagogiques connus sous le terme de « jeux épistémiques » ont été expérimentés. Selon Collins et Ferguson (1993), l’expression « jeux épistémiques » permet de décrire l’activité d’investigation des sciences. Le terme de jeu fait référence à l’ensemble des règles et des stratégies qui guide le questionnement de l’apprenant sur un domaine. Cette démarche s’appuie sur une structure et un modèle épistémique spécifique à une discipline. Perkins utilise, quant à lui, le jeu épistémique comme un modèle de raisonnement afin de définir la caractérisation, l’explication et la justification (Perkins, 1997). Ainsi parce qu’un historien, un avocat et un physicien peuvent décrire, expliquer ou justifier les mêmes faits sous une forme très différente, il est nécessaire que l’apprenant travaille des capacités et des méthodes de raisonnement bien différentes.

Pour parvenir à cela, Shaffer (2006) a proposé d’élargir le modèle « jeu épistémique » en ne le limitant plus qu’à une seule science ou une seule discipline, telle que la physique ou les mathématiques, mais à un ensemble de disciplines regroupées autour de communautés de pratique. Ainsi, le jeu épistémique selon Shaffer, suggère de mener la réflexion au

travers de la vision d'un ensemble d'experts et en prenant en compte les spécificités de chaque communauté. L'objectif de cette approche pédagogique est de bénéficier des apports des deux approches qui s'opposent depuis des dizaines d'années dans nos systèmes scolaires : l'une qui prône l'apprentissage par la pratique et l'autre qui souhaite que l'apprenant acquière un maximum de théories avant de pratiquer (Shaffer et Gee, 2005).

A l'ère du numérique, Shaffer a souhaité expérimenter ce concept aux travers de nouvelles applications nommées « Jeux Epistémiques Numériques » (JEN). Dans ces jeux de rôle éducatifs utilisant des supports numériques, les apprenants sont invités à penser comme des professionnels (Hatfield et Shaffer, 2006). Ainsi dans Land Science², l'apprenant incarne un stagiaire en cabinet de planification urbaine en charge de réaménager tout un quartier. L'objectif principal est de constituer un plan de la ville qui prend en compte les attentes de plusieurs communautés pour des besoins aussi divers que l'emploi, le logement, la pollution. Le plan de chaque communauté est ainsi très différent et pourtant, l'apprenant ne peut en proposer qu'un seul. L'apprenant devra comprendre la vision de chaque communauté pour proposer un des meilleurs compromis pertinent.

Malgré des évolutions technologiques permanentes, les JEN les plus récents se limitent à l'utilisation de techniques classiques telles que la messagerie, la navigation web, ou de la vidéoconférence (Boots et Strobel, 2014). Comme l'illustre la figure 1 ci-après, l'interface principale de Land Science ne propose qu'une représentation schématique du quartier dans lequel l'apprenant devra se projeter. De plus, les échanges avec les différents protagonistes s'effectuent par envoi

² <http://edgaps.org/gaps/projects/land-science/>

de courrier électronique ou messagerie instantanée. Ce type d'interface a pour première intention de faciliter le travail à distance des apprenants mais cela se fait au détriment d'une représentation plus réaliste.

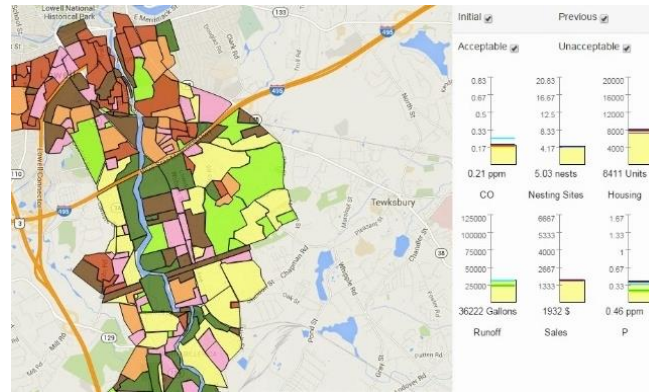


Figure 1 : Principale interface de Land Science

2.1.2 Caractéristiques des JEN

Les jeux utilisés en éducation sont qualifiés de « sérieux » (ou en anglais de « *serious games* », ou bien encore « *learning games* »). De fait, les JEN étant des jeux favorisant l'apprentissage (Perkins, 1997), ils appartiennent à cette catégorie. Si l'expression « jeux sérieux » est très souvent utilisée dans la littérature depuis ces dernières années, le terme de JEN, quant à lui, reste faiblement employé. En ce sens, on peut supposer que certains auteurs utilisent le terme de « jeu sérieux » plutôt que celui de JEN pour traiter de jeux offrant pourtant des caractéristiques « épistémiques ». Il paraît donc judicieux de définir les caractéristiques permettant d'identifier des JEN parmi ces jeux sérieux.

En s'appuyant sur les travaux de [3, 4, 11, 12, 13], nous pouvons considérer un jeu comme un JEN dans la mesure où il :

- propose la résolution de problèmes non-déterministes (Sanchez *et al.*, 2012), comme dans Clim@ction³,
- concerne la résolution de problèmes complexes (Shaffer et Gee, 2005), comme dans Digital Zoo⁴,
- s'appuie sur des activités pluridisciplinaires (Salmani Nodoushan, 2009), comme dans Urban Science⁵,
- suggère à l'apprenant, un contexte réaliste et authentique (Sanchez *et al.*, 2012), comme dans Clim@ction,
- repose sur un « cadre épistémique » (Shaffer, 2006), c'est-à-dire lorsque l'apprenant doit mener son activité avec le savoir-faire, les méthodes, les connaissances et les valeurs du professionnel qu'il incarne (Shaffer *et al.*, 2009), comme dans Science.net⁶.

Il est à noter que l'ensemble de ces caractéristiques sont présentes dans les jeux cités en exemple. Cependant certaines caractéristiques sont davantage mises en évidence dans certains JEN que dans d'autres.

Au-delà de ces caractéristiques principales, certaines plus « secondaires » semblent renforcer le concept de JEN. Parmi elles, nous avons référencé le fait que :

- l'apprenant joue le rôle d'un novice/débutant afin de renforcer le réalisme de la situation de jeu ainsi que la nécessité d'apprendre (Shaffer *et al.*, 2009), comme le rôle de stagiaire dans Science.net,

³ Clim@ction : <http://eductice.ens-lyon.fr/EducTice/recherche/jeux/jpael/climaction/2011-2012/>

⁴ Digital Zoo : <http://edgaps.org/gaps/projects/digital-zoo-2/>

⁵ Urban Science : <http://edgaps.org/gaps/projects/urban-science/>

⁶ Science.net : <http://edgaps.org/gaps/projects/science-net/>

- l'application apporte de meilleurs résultats lorsqu'elle s'appuie sur des communautés liées à des domaines à forte innovation (Shaffer et Gee, 2005).

2.1.3 Apports et Limites

Ces caractéristiques réunies ont pour ambition d'offrir à l'apprenant un apprentissage expérientiel et un apprentissage théorique afin qu'il construise un maximum de connaissances et de compétences qu'il sera capable de mobiliser et transférer dans plusieurs situations (Shaffer et Gee, 2005). Malgré la pertinence de cette approche qui fusionne les deux types d'apprentissage et la croissance de l'industrie du jeu dans le domaine éducatif, peu de jeux intègrent l'ensemble de ces caractéristiques principales et secondaires identifiées dans la littérature. Ce constat suggère que la conception et le développement d'un JEN est pertinent et nécessite des conditions très spécifiques par rapport aux autres jeux sérieux.

2.2 État de l'art des Jeux Pervasifs

Même si les JEN se limitent actuellement à des interactions classiques, de nouvelles formes de jeux dits « pervasifs » savent exploiter pleinement des technologies plus avancées.

2.2.1 Définitions

Même si le terme de « jeux pervasifs » peut se restreindre à l'utilisation de technologies mobiles (Montola *et al.*, 2009), nous lui donnons dans nos travaux, un sens plus large à l'image de Hinske *et al.* (2007) pour qui

ce terme regroupe : les jeux « augmentés », les jeux en « réalité mixte » ou encore les jeux « informatiques mobiles ».

2.2.2 Caractéristiques des jeux pervasifs

Un jeu pervasif est un jeu possédant une ou plusieurs caractéristiques élargissant le « cercle magique du jeu » de façon sociale, spatiale ou temporelle (Montola, 2005). À l'intérieur de ce cercle, l'individu se sent appartenir au jeu ; à l'extérieur, il se projette uniquement dans le réel. Au-delà de la volonté de participer, l'utilisateur juge ce qui appartient ou non au jeu en s'appuyant sur des repères tels que des méthodes et du matériel couramment utilisés d'autres jeux. Les extensions des jeux pervasifs consistent donc à élargir le périmètre de jeu en utilisant des moyens que l'utilisateur ne trouve que rarement dans les environnements ludiques.

2.2.2.1 Extension sociale grâce aux nouveaux moyens de communication

Les jeux en réalité alternée (JRA) sont dans une catégorie de jeux vidéo repoussant les usages courants de la communication pour offrir aux utilisateurs, la possibilité de résoudre collectivement divers problèmes. Pour que la confrontation à ces problèmes soit toujours plus crédible, l'utilisateur est invité à trouver les réponses au milieu d'informations réelles telles que les forums et les sites web (Kim *et al.*, 2009). Mais le jeu possède une véritable extension sociale dès lors qu'il utilise des moyens de communication tels que les sms et les appels téléphoniques ne faisant a priori pas parti de l'univers du jeu dans l'esprit du joueur. Un concept clé des JRA est également d'inviter ses joueurs à se déplacer physiquement sur un site et même se donner rendez-vous dans un lieu

public. Cela peut permettre de découvrir de nouveaux éléments toujours plus concrets mais également de rencontrer et créer des liens avec d'autres joueurs sous une autre forme que celles habituellement possibles par messagerie instantanée.

Contrairement à un jeu en réalité mixte dans lequel une interface virtuelle vient recouvrir la réalité (ou *vice versa*), un JRA permet d'alterner entre les sessions dans le monde numérique et les phases de jeu nécessitant des actions dans le monde réel, le tout dans un scénario cohérent et crédible. De plus, le jeu en réalité alternée est défini comme une forme de « Transmedia Storytelling ». En effet, les JRA peuvent utiliser divers médias tels que les journaux, la télévision, internet et même le cinéma pour impliquer personnellement et émotionnellement les participants dans une expérience narrative (Raybourn, 2014). Bien que leur utilisation soit habituellement dédiée au divertissement ou à la publicité, et que leur conception fasse encore l'objet de nombreuses réflexions, les JRA présentent des caractéristiques intéressantes sur le plan pédagogique. En effet, au-delà de la résolution de problèmes complexes, ces jeux poussent les utilisateurs à imaginer des alternatives à la réalité et donc les encouragent à raisonner avec des arguments d'innovation (Bonsignore *et al.*, 2012). De plus, comme tout apprentissage transmedia, les JRA fournissent un cadre à partir duquel les apprenants peuvent acquérir des compétences de synthèse d'information au travers de flux média (Raybourn, 2014).

2.2.2.2 Extension spatiale par immersion

La réalité mixte a été définie comme un *continuum* qui relie le monde réel et le monde virtuel (Milgram et Kishino, 1994). L'objectif est d'enrichir un environnement virtuel par des éléments réels, ou à l'inverse, d'ajouter des éléments virtuels à un environnement réel. Ce

mélange peut être réalisé à l'aide de nombreuses technologies telles que les écrans, les caméras, les lunettes « see-through », les interfaces mobiles, les interfaces tactiles ou tangibles.

Des études sur l'intégration d'éléments de réalité mixte dans des applications éducatives ont mis en évidence leur potentiel, principalement pour améliorer l'ancrage de l'apprentissage (Cook *et al.*, 2008) et le positionnement des apprenants dans des situations relativement authentiques (Egenfeldt-Nielsen, 2006). Cette authenticité est particulièrement soutenue par le fait que l'apprenant/joueur peut bénéficier d'un ensemble d'informations sur le monde réel et d'un ensemble d'informations issues de la simulation.

L'ajout d'information à des objets réels est très utile pour les situations d'apprentissage, surtout lorsque l'information ne peut pas être perçue spontanément. Des études, telles que celle de Stedmon et Stone (2001), montrent que les concepts techniques sont plus faciles à saisir lorsque les artefacts physiques sont améliorés avec des informations connexes. David *et al.* (2010) a étudié comment les artefacts physiques (machines industrielles et ordinateurs) peuvent être améliorés avec des données numériques pour favoriser l'apprentissage juste à temps. Plus récemment, le projet SEGAREM illustre comment la technologie de réalité mixte pourrait être intégrée dans ces jeux sérieux (Orliac *et al.*, 2012).



Figure 2 : Un prototype du projet SEGAREM permettant à l'apprenant d'utiliser un objet physique (un pistolet à colle) face à un écran qui représentera la colle virtuellement appliquée.

2.2.2.3 Extension temporelle par persistance

L'extension des aspects temporels d'un jeu s'appuie sur un concept de persistance, largement utilisé dans de célèbres MMORPG⁷ tels *World of Warcraft* et *Second Life*. Ces jeux offrent à des milliers de personnes, la possibilité d'interagir ensemble dans un monde virtuel commun. Selon Gonzalez *et al.* (2013), un monde virtuel est défini comme un environnement persistant et simulé par ordinateur permettant à un grand nombre d'utilisateurs, représentés par des avatars, d'interagir en temps réel les uns avec les autres dans un environnement simulé.

La persistance d'un monde virtuel signifie que tout élément appartenant à ce monde continue d'exister et d'évoluer même lorsque le joueur quitte le jeu. Cette persistance a un impact dans la façon dont les utilisateurs interagissent avec d'autres participants, élargissant ainsi la dimension sociale du jeu. Un joueur n'est ainsi plus le centre du monde mais un membre d'une communauté dynamique. Le joueur peut considérer le système qu'il intègre sous plusieurs dimensions (écologique, économique, politique, etc.) avec ou sans sa présence (Bell, 2008).

Dans les apprentissages s'appuyant sur les technologies, ce concept peut être utilisé pour améliorer l'engagement des apprenants. Par exemple, le monde virtuel d'apprentissage appelé MMOLE⁸ permet le développement d'activités de simulation. De plus, le MMOLE peut stimuler la participation active et minimiser les problèmes d'intervention

⁷ MMORPG: Massively Multiplayer Online Role-Playing Game

⁸ Massively Multilearner Online Learning Environments

stressante devant un public nombreux. Cependant, la communication verbale et l'interaction en face-à-face restent irremplaçables (González *et al.*, 2013).

2.2.3 Apports et limites

Ces jeux pervasifs ont notamment donné des perspectives intéressantes dans le domaine de l'éducation et leurs effets ont d'ailleurs été étudiés sur l'enseignement et l'apprentissage. En effet, en augmentant le nombre de *stimuli* aussi bien par une expérience physique qui consolide l'ancrage dans le réel, qu'avec une expérience sociale et immersive pour augmenter la motivation et l'appropriation, il semble possible de proposer de vivre une expérience plus riche qu'au travers d'une interface classique (Ardito *et al.*, 2011). On peut citer, par exemple, l'usage de la réalité augmentée embarquée dans les téléphones et tablettes pour donner des informations contextualisées à l'apprenant, l'usage d'objets physiques pour manipuler le monde virtuel ou encore l'utilisation de QRCode pour contextualiser un scénario d'apprentissage (Loiseau *et al.*, 2013).

Concernant la conception et le développement de ces extensions, chaque aspect possède des limites bien différentes. Une des extensions sociales les plus attendues seraient de pouvoir dialoguer avec une entité que l'on pense être humaine alors qu'elle ne serait qu'un agent virtuel. Même si les utilisateurs sont aujourd'hui habitués à établir des relations au travers de plateformes d'échanges, l'intelligence artificielle voire même la réalité virtuelle devront proposer des agents virtuels capables d'agir comme des humains (Heyselaar *et al.*, 2017). Au-delà du coût des serveurs informatiques, la dimension temporelle impose aux concepteurs de jeux, de coordonner un nombre d'activités aussi important que sera la persistance du jeu. Enfin l'extension de la dimension spatiale du jeu

oblige les concepteurs et les développeurs de jeux à devoir être capables d'intégrer de nouvelles technologies adaptées à leur contexte de jeu. Cela suppose que les développeurs devront, soit posséder des compétences RMV, soit disposer d'un outil permettant la mise en place d'interactions immersives sans expertise RMV.

2.3 Proposition : Jeux Épistémiques Numériques Pervasifs

Les avancées technologiques caractérisant les jeux pervasifs couplées aux apports des JEN permet d'aboutir à des « Jeux Épistémiques Numériques Pervasifs » (JENP), illustrés en figure 3. Ceci constitue un objet de recherche novateur et pertinent dans le domaine des environnements informatiques pour l'apprentissage humain. Les JENP soulèvent, en particulier, des questions liées 1) à leur conception, à savoir quels modèles, méthodes et outils utiliser pour les concevoir et les développer, et 2) à leur usage en milieu écologique quant à leur appropriation et leurs impacts sur l'apprentissage.

Pour contribuer à ces questionnements, nous proposons un cadre théorique pour définir et caractériser ces nouveaux JENP. Nous positionnons les JENP en mettant en évidence leur valeur ajoutée comparativement aux JEN.

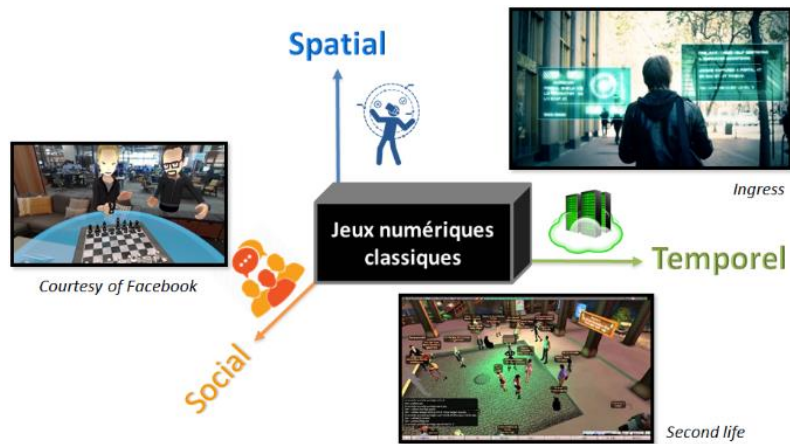


Figure 3 : JENP, une extension spatiale, temporelle et sociale des JEN

2.3.1 Définition des jeux épistémiques numériques pervasifs

Nous définissons un jeu épistémique numérique pervasif comme un jeu épistémique numérique dans lequel les concepts et technologies pervasives permettent d'étendre la dimension spatiale, temporelle et sociale afin d'enrichir l'authenticité de la situation d'apprentissage.

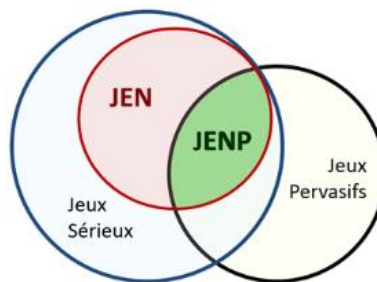


Figure 4 : Jeux sérieux, épistémiques et pervasifs

2.3.2 Caractéristiques théoriques des JENP

2.3.2.1 De nouvelles interactions pour la construction de connaissances

Nous considérons qu'il est essentiel lors de la conception d'un JENP de déterminer les interactions les plus adaptées pour enrichir le contexte ainsi que de les répartir rigoureusement dans les différentes phases de jeu.

Plus d'interactions pour une meilleure authenticité. Le cadre épistémique constitue la caractéristique majeure des JEN. Les apports propres aux jeux pervasifs peuvent enrichir ces JEN en offrant à l'utilisateur un plus grand nombre d'interactions susceptibles d'offrir à l'apprenant un contexte (social et professionnel) plus authentique.

Les canaux utilisés en entrée (pour interagir avec le dispositif) et en sortie (pour présenter l'information) peuvent être visuels (*Cave Automatic Virtual Environment*, lunettes stéréoscopiques, *Head-Mounted Display*, etc.), sonores (reconnaissance vocale, son 3D spatialisé, etc.) voire haptiques (gants de données, etc.) (Fuchs et Moreau, 2003).

L'apport des nouvelles technologies peut se situer à trois niveaux d'interaction :

- Les interactions avec le système (interactions tactiles, vocales, tangibles, etc.)
- Les interactions en contexte (informations contextualisées, mobilité de l'apprenant, intégration d'éléments de l'environnement dans le jeu etc.)
- Les interactions avec les autres apprenants (partage d'objets, utilisation d'espaces communs et d'espaces personnels connectés, édition collaborative de documents, etc.).

Plus d'interactions pour une meilleure construction des connaissances. Dans le cadre des JENP, les interfaces ont pour vocation de proposer un réalisme et une complexité en adéquation avec les interactions proposées et les capacités de l'apprenant. En outre, même si certaines interfaces peuvent être choisies pour offrir plus de ludicité au jeu, il n'en demeure pas moins que l'apport pédagogique doit rester l'aspect central du JENP (i.e. la complexité des tâches doit être en adéquation avec les objectifs pédagogiques). L'objectif du JENP est bien d'enrichir la situation et les actions pour améliorer la construction des connaissances et des compétences (Ardito *et al.*, 2011).

2.3.2.2 Univers persistant

Certaines technologies spécifiques aux jeux pervasifs peuvent offrir la possibilité aux apprenants de « rester » dans le jeu en dehors des heures de formation (univers persistant). Les effets de ces univers persistants ont été particulièrement étudiés pour les jeux de rôle en ligne massivement multijoueur. Ainsi l'utilisateur n'a pas la possibilité de sauvegarder ou reprendre une phase de jeu et n'a donc pas la possibilité de figer son contexte de jeu durant son absence. Son univers reste actif en permanence. Les apports pédagogiques et ludiques sont de maintenir la compétition entre joueurs, de soutenir la motivation apportée par un environnement plus écologique, ainsi que de valoriser la prise de décision (Papargyris et Poulymenakou, 2005). Cela permet aussi de faire évoluer le monde du jeu quand l'apprenant n'est pas connecté, ce qui oblige à anticiper et à faire preuve de réactivité. Cependant, il est important de trouver un équilibre lors de la conception du JENP pour éviter qu'une mauvaise décision prise en fin de séance ne frustre pas l'apprenant jusqu'à la prochaine séance sans qu'il puisse intervenir. Ainsi dans un MMORPG, durant son absence, le joueur ne peut pas être

en situation d'échec mais ne peut pas non plus bénéficier d'opportunités pour faire progresser son personnage. Les JENP peuvent ainsi envoyer des messages directement aux apprenants en dehors des séances de cours afin de les informer de l'évolution de la situation. Comme les ARG l'ont déjà démontré, cette mécanique ludique est appréciée par les utilisateurs. Cependant elle ne doit pas être trop invasive, et ne pas nuire au repos de l'apprenant. D'un point de vue pédagogique, le rappel de l'information est largement favorable à sa mémorisation.

2.3.2.3 Suivi pédagogique du tuteur

Les JENP conçus doivent être compatibles avec le futur contexte d'utilisation. En effet, ces jeux éducatifs doivent être intégrés par l'enseignant dans les activités pédagogiques qu'il met en place. Certaines interactions comme celles liées à la géolocalisation ou la réalité augmentée peuvent demander des sorties en milieu extérieur. Par ailleurs, un jeu persistant, qui se déroule en continu, peut demander un suivi en dehors des heures d'enseignement.

La méthode *Epistemic Network Analysis* (ENA), a été proposée par Shaffer dans le but d'évaluer la pertinence d'un cadre épistémique mais également afin de mesurer la progression utilisant un JEN. En multipliant les dispositifs, les espaces et les temps d'apprentissage, les traces sont multiples et variées (Shaffer *et al.*, 2009). Pour que l'ensemble de ces informations puisse être interprété par le tuteur, il est indispensable que le JENP propose de nouveaux indicateurs pédagogiques à l'enseignant. La scénarisation et la configuration de chaque activité du JENP prennent également une dimension plus large. En offrant plus de possibilités, une simplification devra également être mise en place pour ne pas imposer une surcharge aux enseignants, tout en leur permettant d'adapter le JENP à leurs situations.

2.3.2.4 Utilisation de périphériques déjà intégrés au contexte d'usage

Une des premières barrières de l'utilisation de la RMV reste son coût financier. La croissance du marché du jeu vidéo profite à une industrialisation de nouvelles technologies telles que les smartphones, les visiocasques ou divers types de périphériques. En parallèle, les salles de formation se voient progressivement équipées de nombreux périphériques tels que les tableaux blancs interactifs et les tablettes. Cependant ces équipements restent encore sous-utilisés. Cela peut s'expliquer par les applications mises à disposition des enseignants. Le plus souvent ces dernières ont pour objectif de proposer de nouveaux outils à une pédagogie adaptée au matériel traditionnel. Ainsi ces nouveaux supports gardent un intérêt limité et peuvent même être perçus comme contraignant pour les enseignants. On peut ainsi espérer des résultats meilleurs en co-concevant des applications avec les enseignants pour mettre en place une approche pédagogique alternative et adaptée à ces technologies, telle que celle des JENP.

2.3.3 Apports potentiels

A l'aide des JEN, les apprenants de nouvelles compétences pour affronter des situations plus complexes, pluridisciplinaires et non-déterministes rarement abordées dans les autres jeux sérieux. A l'aide de la plateforme en ligne dédiée aux JEN, les enseignants peuvent établir des relations différentes avec les apprenants aux travers des communautés. Cette approche permet d'accompagner avec plus de proximité l'apprenant dans son raisonnement. Le JENP, quant à lui, représente une évolution du JEN grâce à ses principaux apports en authenticité, en persistance et par l'intégration de nombreuses

interactions immersives. Ces évolutions permettent de confronter l'apprenant aux situations complexes par de véritables expériences. Dans les JENP, cette expérience peut être vécue dans un environnement virtuel évoluant à toute heure et partagée avec n'importe quelle entité. Le JENP peut ainsi recueillir de nouvelles traces et indicateurs permettant aux enseignants de suivre et analyser les différents comportements des apprenants face aux problèmes. Le suivi de l'enseignant avec le JEN se limitait quant à lui, à l'analyse sémantique des dialogues.

Cependant il ne semble pas pertinent de pousser des caractéristiques telles que la complexité du problème ou la pluridisciplinarité à leur maximum pour l'apport pédagogique en soit meilleur. Ainsi l'importance de chaque caractéristique du JEN doit rester adaptée au contexte du jeu. Concernant les JENP, même s'il peut être recommandé d'étendre le JEN à la fois sur la dimension sociale, spatiale et temporelle, les concepteurs peuvent prioriser la dimension la plus adaptée au scénario pédagogique. De nouveaux travaux pourraient permettre de déterminer l'influence de chaque dimension sur l'apprentissage à l'aide de JENP. De plus, pour que les enseignants puissent effectuer, eux-mêmes, ces nombreux choix de conception des JENP, il est nécessaire de leur proposer un guide spécifique.

2.4 Synthèse

L'objectif de ce chapitre est de proposer un cadre théorique pour définir et caractériser le concept de JENP. Aussi, après avoir recherché les caractéristiques des JEN existants et identifié leurs limites technologiques, nous proposons de leur associer des technologies utilisées dans les jeux pervasifs (en particulier, la réalité virtuelle, mixte et la persistance). Ces définitions et ces caractéristiques conceptuelles

fournissent aux enseignants et concepteurs de jeux sérieux, une base et des objectifs propres aux JENP.

Toutefois, cette base ne spécifie pas les outils et les modèles informatiques à mettre en place pour implémenter les objectifs de ces JENP visant l'intégration d'interactions plus authentiques et contextualisées. Pour cela, dans le prochain chapitre, nous évoquons le processus de réalisation d'un premier JENP ainsi que son expérimentation. Nous évaluerons si l'immersion et la persistance favorisent la motivation et l'engagement des apprenants dans le jeu. Nous nous intéresserons également, de façon plus générale, aux méthodes de conception de tels jeux.

Chapitre 3 : Prototypage et évaluation d'un JENP

Dans le cadre des deux premiers cycles du projet JEN.lab, nous avons animé 5 séances de co-conception par cycle, avec 2 enseignants du lycée Aristide Briand de Saint-Nazaire. Ces séances ont été réalisées afin de proposer des situations de jeu pertinentes pour l'immersion et la persistance. Suite à cela, nous avons développé un prototype de JENP et mené des expérimentations pour évaluer son effet sur la motivation et l'engagement des apprenants.

3.1 Prototype RearthM3

3.1.1 Synopsis du JENP

La planète Terre est devenue hostile à la race humaine. Les humains sont à la recherche de nouvelles ressources. Pour cela, un vaisseau spatial a été envoyé à l'autre extrémité de la galaxie. Il est maintenant en orbite autour d'une planète classée comme potentiellement habitable. À bord de ce vaisseau, des milliers de personnes espèrent de nouvelles ressources sur une nouvelle planète, et seulement une douzaine d'individus ont été désignés comme explorateurs.

Les 2 enseignants ont proposé ce synopsis du jeu après un échange avec leurs élèves. Même le scénario peut être classé en catégorie de science-fiction, les objectifs du jeu restent identiques à ceux que les experts se fixent pour la conquête sur Mars. De plus, les outils qui sont mis à la disposition des apprenants ont également des performances identiques à

des modèles actuels de la NASA⁹. Ainsi il est proposé aux apprenants une situation à la fois ludique et authentique.

3.1.2 Objectifs pédagogiques et scénario

D'un point de vue pédagogique, le jeu, nommé RearthM3, requiert des compétences en programmation et en développement durable. Dans le scénario, les apprenants agissent comme des explorateurs qui restent à bord du vaisseau spatial tout au long du jeu. Ils doivent collecter des données de la planète dans le but de choisir l'endroit le plus pertinent pour installer un camp. Trois communautés (les bâtisseurs, les technologues et les énergéticiens) imposent différentes contraintes couvrant différents aspects du développement durable. Chaque apprenant est soutenu par l'une des trois communautés et devra intégrer ses enjeux communautaires dans la décision collective finale.

Lors de la première étape du jeu, chaque apprenant est responsable de l'exploration d'une partie de la planète. Pour cette phase d'exploration, les étudiants sont aux commandes d'un astromobile (engin d'exploration pouvant être envoyé sur une planète). Leur premier objectif est de le programmer afin de répartir des capteurs à la surface d'une planète, comme illustré en figure 5. Pour cela, les joueurs ont accès à un module d'observation, (figure 6), et à un journal de bord de l'astromobile (figure 7) pour suivre sa progression mais aussi faire des observations pour anticiper les risques potentiels.

⁹ <https://www.nasa.gov/>

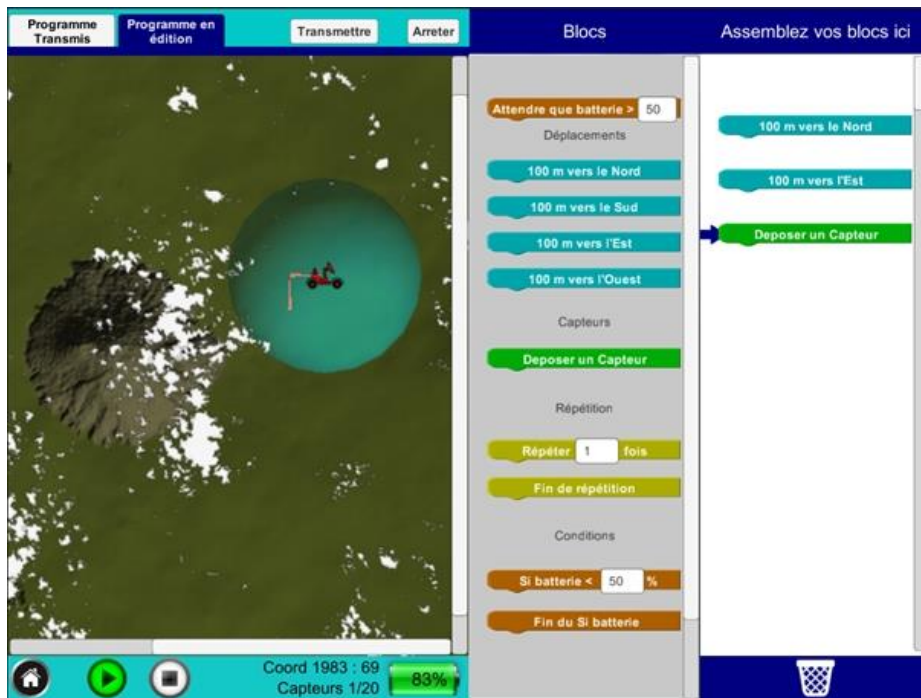


Figure 5 : Interface de programmation de l'astromobile dans RearthM3



Figure 6 : Interface d'observation pour voir l'environnement de l'astromobile

Journal de bord				
Date	Categorie	Coord	Batterie	Description
24/04/2015 13:47:0	Démarrage déplacement	1483.3.1227.9	100 %	Déplacement Nord démarré
24/04/2015 13:48:2	Déplacement réussi	1483.3.1227.9	91.97 %	Déplacement Nord réussi
24/04/2015 13:48:3	Démarrage déplacement	1483.3.1227.9	91.97 %	Déplacement Est démarré
24/04/2015 13:49:4	Déplacement réussi	1583.3.1227.9	83.95 %	Déplacement Est réussi
24/04/2015 13:49:4	Capteur déposé	1583.3.1227.9	83.95 %	Dépot d'un capteur

Figure 7 : Journal de bord des évènements de l'astromobile

La seconde étape du jeu consiste à exporter les données collectées par les capteurs, à les analyser et à les filtrer. L'apprenant mène alors une réflexion sur les conditions vitales pour l'implantation d'un camp concernant la température, l'altitude ou encore la quantité d'eau sur chaque emplacement potentiel.

Enfin, la dernière étape invite les joueurs à choisir collectivement le terrain le plus approprié pour la colonisation humaine. Pour cela, les apprenants travaillent ensemble dans leurs communautés. Chacunes d'entre elles ayant des priorités de ressources bien différentes, ils doivent trouver des arguments pour démontrer quel terrain est le plus avantageux pour eux mais aussi pour les autres. Cette dernière session se termine par une présentation orale pour chaque communauté et un vote collectif définitif pour le terrain représentant le meilleur compromis.

3.1.3 Implémentation de l'immersion et la persistance

Pour analyser les impacts de l'immersion et de la persistance, deux prototypes ont été développés. Un premier prototype avec des caractéristiques JEN se limite à des interactions WIMP¹⁰ classiques. Un deuxième prototype avec des caractéristiques JENP propose une situation plus authentique, en ajoutant plus d'immersion et de persistance (Tableau 1).

Les éléments d'immersion sont principalement ajoutés à la phase d'exploration, lorsque les apprenants doivent programmer l'astromobile.

¹⁰ WIMP : Windows Icons Menu Pointer

Les propriétés de ce robot, telles que la vitesse de déplacement, sont plus réalistes dans le JENP que dans le JEN. L'immersion est également augmentée en offrant à l'apprenant une vue en 3D de la planète explorée. En outre, les apprenants peuvent être équipés d'un visiocasque pour une vue à 360° en réalité virtuelle. L'immersion est également ajoutée en présentant une vidéo du scénario comme *teaser* lors de la première session.

La persistance a été mise en œuvre en offrant aux apprenants des JENP un monde virtuel évoluant entre deux séances de classe. Une fois qu'un programme a été soumis à l'astromobile, il fonctionne toujours même si l'application est fermée. En outre, si l'astromobile programmé s'arrête entre deux sessions, l'apprenant recevra un courrier électronique sur son adresse personnelle pour le tenir informé de la progression de l'exploration.

Tableau 1 : Principales différences entre le prototype JEN et JENP

	JEN	JENP	Caractéristiques pervasives
Teaser	Présentation orale	Vidéo du synopsis au début de la première séance	Immersion
Programmation	Reprise depuis la dernière position enregistrée	Le programme continue de s'exécuter entre 2 sessions	Persistance

Vitesse de l'astromobile	2,5 m/s	0,25 m/s	Immersion
Interface d'observation	Sonar (Interface 2D)	Laser (Interface 3D) avec vision stéréoscopique par Oculus	Immersion
Rapport de l'astromobile par email	Aucun : Nécessité d'accéder au journal dans l'application	Email donnant les informations de la dernière action réalisée	Immersion Persistance

3.2 Évaluation de l'apport du JENP

3.2.1 Hypothèses de recherche

Les impacts sur l'apprentissage des jeux sérieux en réalité mixte ont été évalués aux travers de plusieurs études (George *et al.*, 2014). Les résultats ont démontré qu'à un scénario pédagogique équivalent, l'apprentissage est meilleur lorsque le jeu est réalisé en RM plutôt que par des représentations symboliques.

Comme évoqué dans le chapitre 2, le JEN possède un ensemble de caractéristiques par rapport aux jeux sérieux, tels que la résolution de problèmes non-déterministes en s'appuyant sur des communautés. Le

JENP a quant à lui, possède des spécificités supplémentaires telles que les interactions immersives mais également des dispositifs de persistance. Ainsi ce type de prototypes étant très spécifiques, il est nécessaire de mener de nouvelles expérimentations. Notre hypothèse sera donc qu'à scénario pédagogique équivalent, l'apprenant de JEN soit meilleur lorsque le jeu est équipé à la fois de systèmes immersifs et persistants. Nous avons souhaité établir si l'augmentation de l'authenticité de la situation d'apprentissage d'un JEN, impacte la motivation et l'engagement des apprenants.

3.2.2 Protocole expérimental

Pour ce faire, nous avons comparé des groupes d'apprenants utilisant le prototype JEN avec d'autres groupes d'apprenants utilisant le prototype JENP, enrichi d'immersion et de persistance. L'expérimentation du jeu Rearth M3 a été menée dans le lycée co-concepteur du jeu, dans deux classes de formation de STI2D. 57 élèves âgés de 16 à 18 ans, avec une large majorité de garçons (seulement 4 filles) ont participé à 4 séances de 2 heures. Deux groupes de 13 et 14 élèves ont réalisé l'expérimentation en condition pervasive (c'est-à-dire en utilisant le prototype JENP). Deux groupes de 15 étudiants ont effectué l'expérimentation en condition non pervasive (c'est-à-dire en utilisant le prototype JEN). Chaque prototype permet de collecter les traces d'interactions. Un questionnaire a été remis à la fin de la dernière séance. De plus, une séance *focus group* a également été réalisée avec 10 étudiants à la fin du jeu.

3.2.3 Évaluation de la motivation

Pour évaluer la motivation des apprenants, nous avons utilisé le questionnaire de *The Situational Motivation Scale (SIMS)* (Guay *et al.*, 2000). L'objectif de ce questionnaire est de déterminer, durant une activité, le type de motivation de chaque individu. En utilisant une échelle de 1 à 7, l'individu détermine son niveau d'accord pour chacune des 16 affirmations.

50 élèves ont répondu au questionnaire: 27 pour le groupe non-pervasif, 23 pour le groupe pervasif. Les résultats du test de Levene permettent d'observer que nous avons une variance homogène pour la motivation intrinsèque ($F = 0,666$, $p = 0,418$), l'autodétermination ($F = 1,883$, $p = 0,176$), la motivation extrinsèque ($F = 0,594$, $p = 0,445$) et l'amotivation ($F = 0,568$, $p = 0,455$). Nous pouvons donc effectuer un T-Test pour chaque mesure.

Tableau 2 : Résultats sur la motivation (n=effectifs, m=moyenne, et=écart type)

	n	Motivation Intrinsèque		Autodétermination		Motivation Extrinsèque		Amotivation	
		m	et	m	et	m	et	m	et
Non-pervasifs	27	4,7	1,72	3,32	1,58	4,81	1,52	3,35	1,15
pervasifs	23	4,11	1,4	3,06	1,21	4,94	1,51	3,57	0,98

Comme le montre le tableau 2, les valeurs de la *motivation intrinsèque* sont globalement hautes pour le groupe pervasif comme le groupe non-pervasif. La motivation intrinsèque est en moyenne plus élevée pour le groupe des non-pervasifs mais de manière non-significative ($p = 0,281$). Les valeurs de l'*autodétermination* sont globalement les plus faibles du questionnaire pour les deux groupes. L'*autodétermination* du groupe

non-pervasif semble plus élevée mais encore de manière non-significative ($p = 0,531$)

Les valeurs de la *motivation extrinsèque* sont en moyenne, les plus hautes du questionnaire pour chacun des groupes. Dans ce type de motivation, même la différence reste la moins non-significative ($p = 0,882$), la moyenne du groupe pervasif, dans ce cas, supérieure à celle du groupe non-pervasif.

Les valeurs de l'*amotivation* sont globalement faibles pour les deux groupes même si la différence est également non-significative ($p = 0,429$).

Lors du *focus group*, les apprenants ont justifié que leur motivation est plus importante à apprendre au travers d'un jeu sérieux quel qu'il soit, que par une session d'enseignement traditionnelle. Même si leurs préférences sont en faveur du JENP, la différence entre les deux prototypes semble être secondaire pour eux.

3.2.4 Evaluation de l'engagement

3.2.4.1 Méthodologie

En ayant réalisé des expérimentations de jeu vidéo en condition écologique sur plusieurs séances, il est possible d'utiliser l'approche qualitative de Bouvier *et al.* (2014) pour identifier et qualifier l'engagement des apprenants. Cette approche consiste à transformer des traces brutes d'interaction contenant des informations peu significatives en traces plus sémantiques. Nous avons inclus dans le prototype du jeu, un collecteur de traces pour recueillir des informations telles que la sélection de chaque menu, les événements liés à la rédaction des programmes et les résultats des capteurs enregistrés par l'astromobile. À

partir de ces traces, nous avons composé plusieurs indicateurs de haut niveau liés à l'engagement.

Selon les travaux de Bouvier *et al.*, qui sont principalement fondés sur la théorie de l'autodétermination (Ryan et Deci, 2000), nous avons considéré trois types de comportements révélateurs d'engagement dans notre analyse: la fréquence des actions décisives, la temps de réflexion consacré à ces actions et le temps consacré aux conséquences de ces actions. Pour mesurer l'ensemble, nous avons défini 6 indicateurs: (i) le pourcentage du terrain couvert par les capteurs pour chaque élève (ii) le nombre de tests de programme effectués avant chaque transmission de de l'élève à son astromobile (iii) le nombre de programmes transmis à l'astromobile et le temps moyen entre 2 transmissions (iv) le temps passé par chaque élève dans chaque type d'interface de l'environnement d'apprentissage (v) le nombre d'actions réalisées par chaque élève dans le module de programmation (vi) le nombre de sessions effectuées par élève dans chaque module. Ces informations ont été calculées pour chaque groupe et chaque session d'apprentissage. Nous avons utilisé UTL (Iksal, 2011) (Choquet et Iksal, 2007) pour traiter les traces. Cet outil a été utile pour décrire un processus d'analyse au moyen d'associations de données intermédiaires afin de fournir des indicateurs. Les représentations graphiques des indicateurs ont été réalisées avec la bibliothèque JavaScript HighCharts¹¹.

3.2.4.2 Résultats

Le premier objectif fixé par l'enseignant est que chaque apprenant explore le plus efficacement possible leurs terrains afin de partager les informations récoltées avec les autres joueurs. Ainsi, l'indicateur

¹¹<http://www.highcharts.com/>

pouvant évaluer la réussite de chacun, illustré en figures 8 et 9, correspond à la taille du terrain exploré (taux de couverture des capteurs posés). Cet indicateur montre de meilleurs résultats chez les apprenants dans les groupes pervasifs (60,51% de couverture contre 46,77% pour le groupe non-pervasifs).

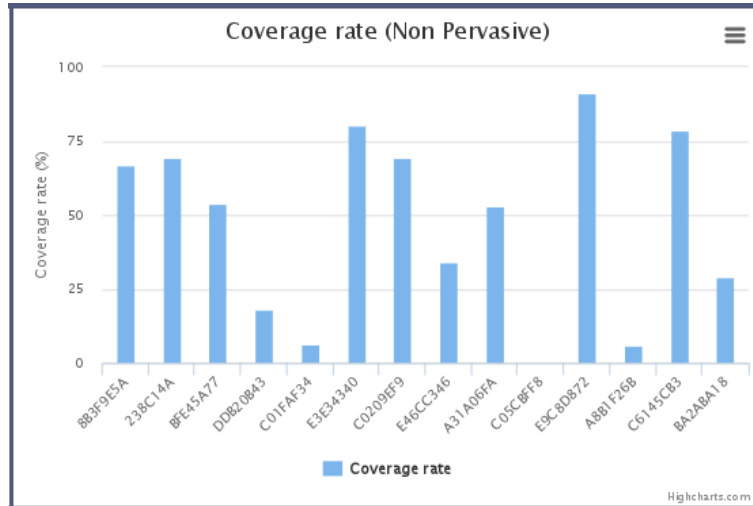


Figure 8 : Taux de terrain exploré par élève selon le groupe non-pervasif

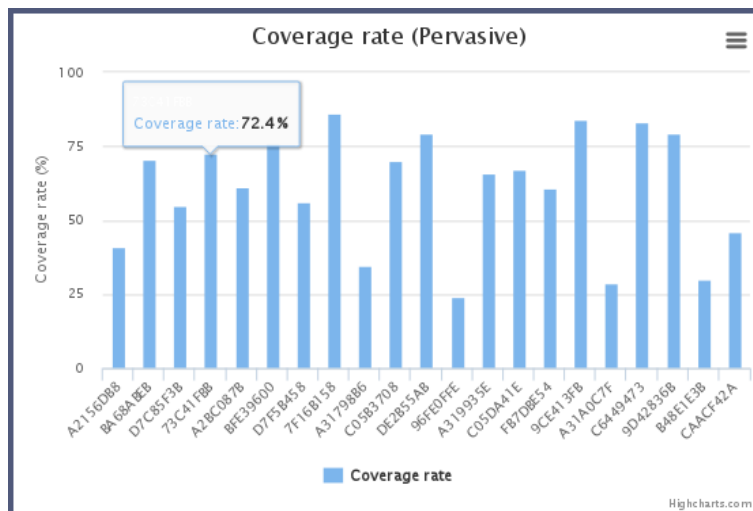


Figure 9 : Taux de terrain exploré par élève selon le groupe pervasif

Pour atteindre le principal objectif d'exploration, les apprenants possèdent de nombreuses solutions de programmation de leur astromobile. Ils rédigent un programme, peuvent effectuer autant de tests qu'il le souhaite puis le soumettent définitivement au véhicule. Si le programme ne correspond aux possibilités du véhicule tel que l'usage excessif de la batterie ou la direction vers un obstacle, le programme sera stoppé. Un nouveau programme devra alors être rédigé à partir de ce point d'arrêt. Les enseignants recommandent donc de réfléchir et tester préalablement leur solution avant de la soumettre. Chaque ligne des figures 10 et 11, correspond aux actions d'un apprenant et quantifie le nombre de tests ou simulations effectués. Chaque changement de couleur indique qu'une série de simulations se termine et qu'une transmission définitive du programme a été effectuée à l'astromobile. Les résultats démontrent que les groupes pervasifs souhaitent s'assurer de la validité de leur programme avant de le soumettre (plus de simulations mais moins de soumissions) tandis que les groupes non-pervasifs semblent avoir une approche "essai-erreur" (beaucoup de soumissions, moins de simulations).

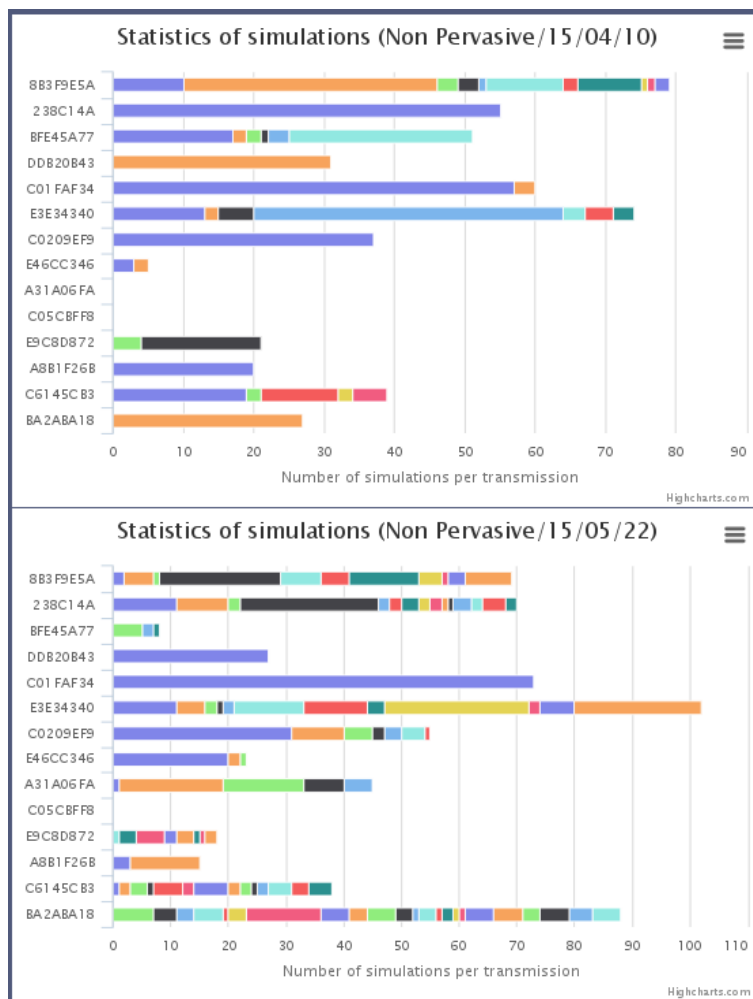


Figure 10 : Statistiques sur les simulations de chaque élève du groupe non-pervasif

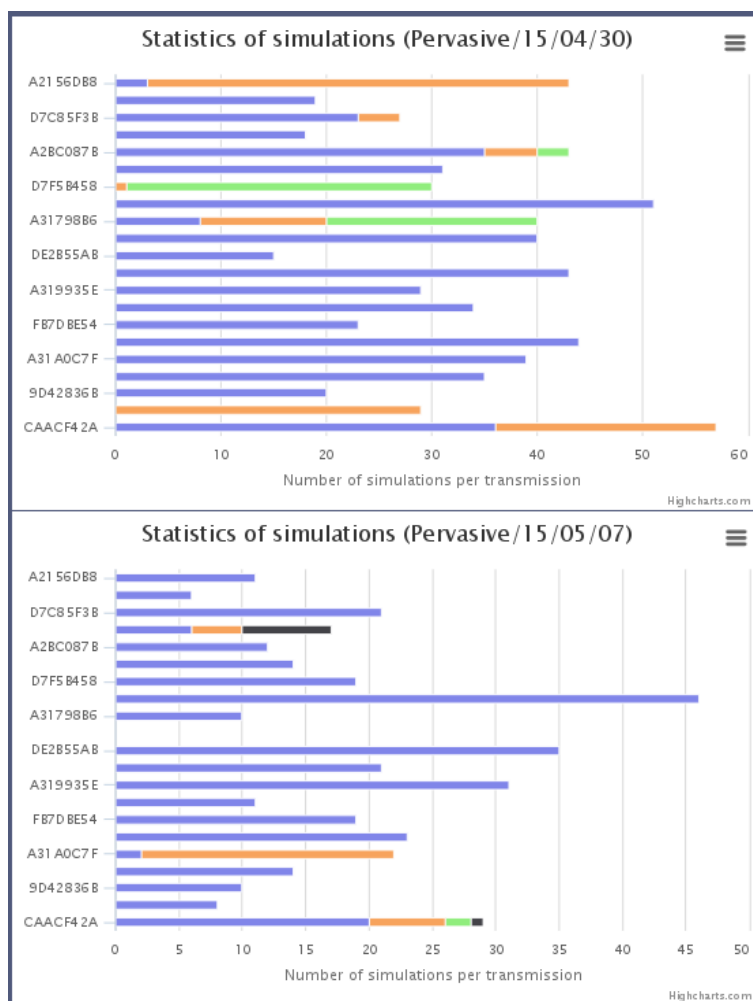


Figure 11 : Statistiques sur les simulations de chaque élève du groupe pervasif

La soumission définitive du programme à l'astromobile étant l'action la plus décisive, l'analyse de sa fréquence semble indissociable de celle du comportement de l'apprenant. Dans les figures 12 et 13, les colonnes bleues représentent la durée moyenne en secondes que chaque apprenant consacre à la préparation d'un nouveau programme. La courbe noire symbolise la somme des soumissions définitives effectuées par chaque apprenant. Ces deux types d'informations proposent des résultats opposés pour le groupe non-pervasif dans la figure 12. On constate en

effet, que moins l'apprenant a soumis de temps à ses programmes et plus fréquemment il a été nécessaire de soumettre des corrections. Comme illustré dans la figure 13, les apprenants du groupe pervasif ont une fréquence de soumission de programme plus faible, en limitant leur nombre de soumissions et consacrant plus de temps à la réalisation de chaque programme.

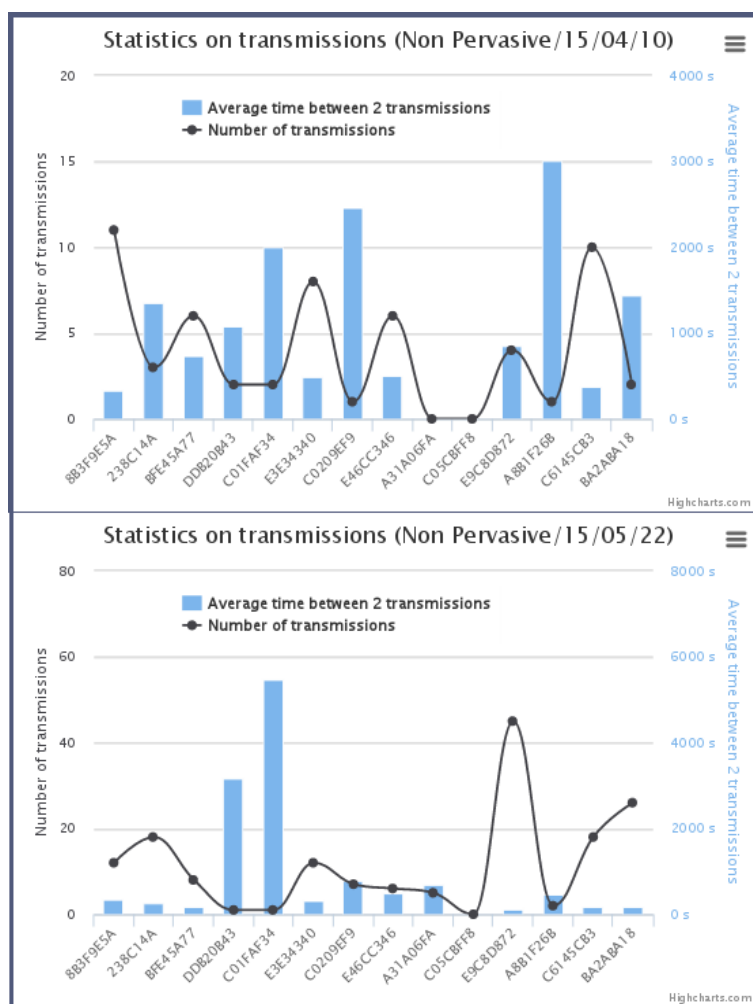


Figure 12 : Statistiques sur les transmissions de chaque élève du groupe non-pervasif

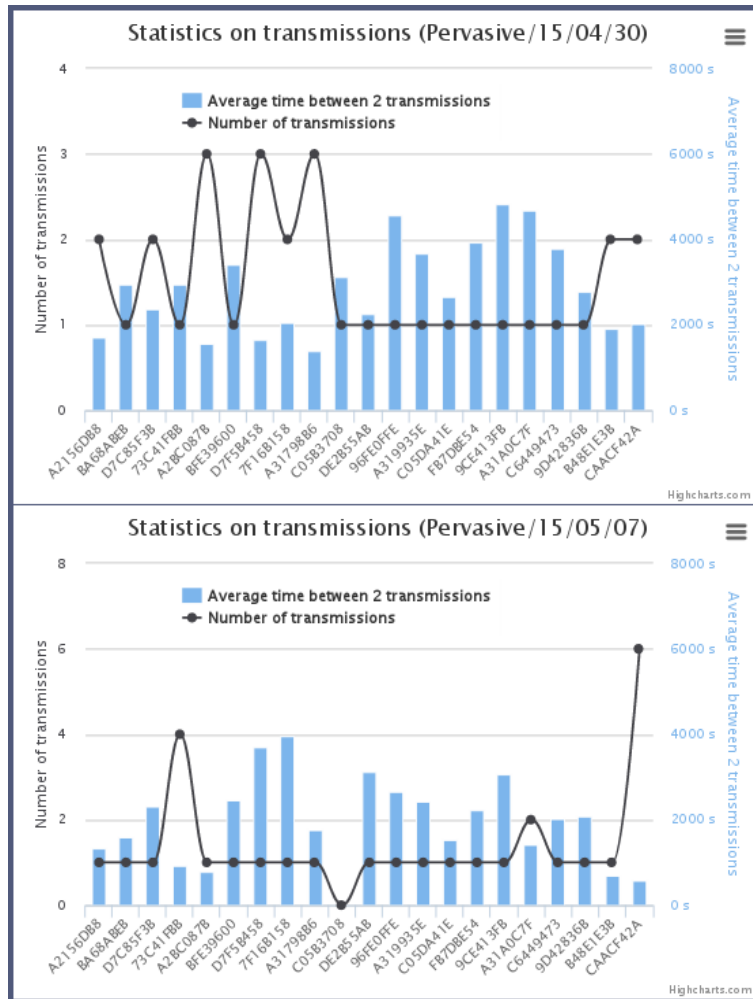


Figure 13 : Statistiques sur les transmissions de chaque élève du groupe pervasif

L'indicateur, illustré en figures 14 et 15, représente la durée en secondes que l'apprenant consacre aux interfaces de programmation, d'observation et de journal de bord. On peut constater que les interfaces d'observation et de journal de bord sont plus visitées par le groupe pervasif. Ces interfaces sont dédiées au suivi de l'astromobile. La

démarche des élèves du groupe pervasif suppose un intérêt plus important pour les résultats de leurs actions.

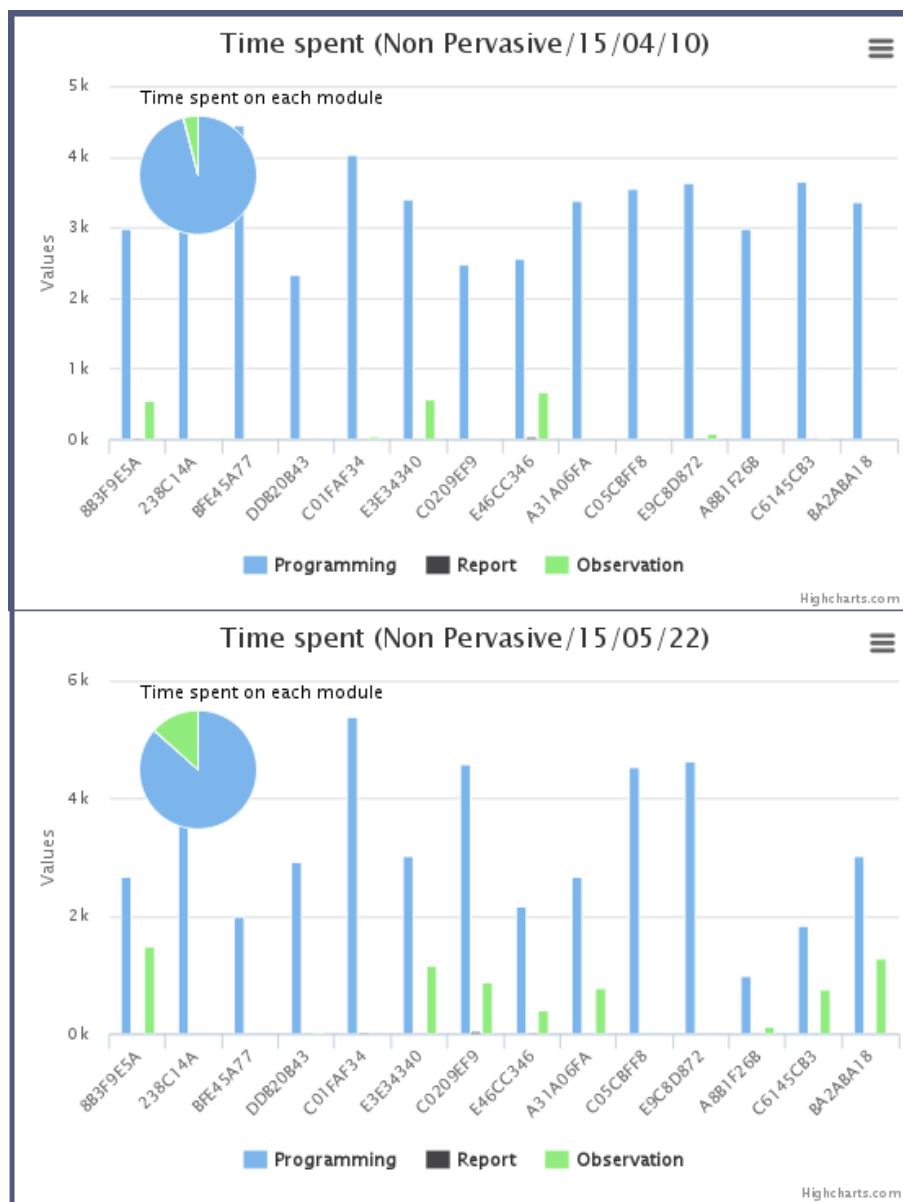


Figure 14 : Durée passée dans chaque module par étudiant du groupe non-pervasif

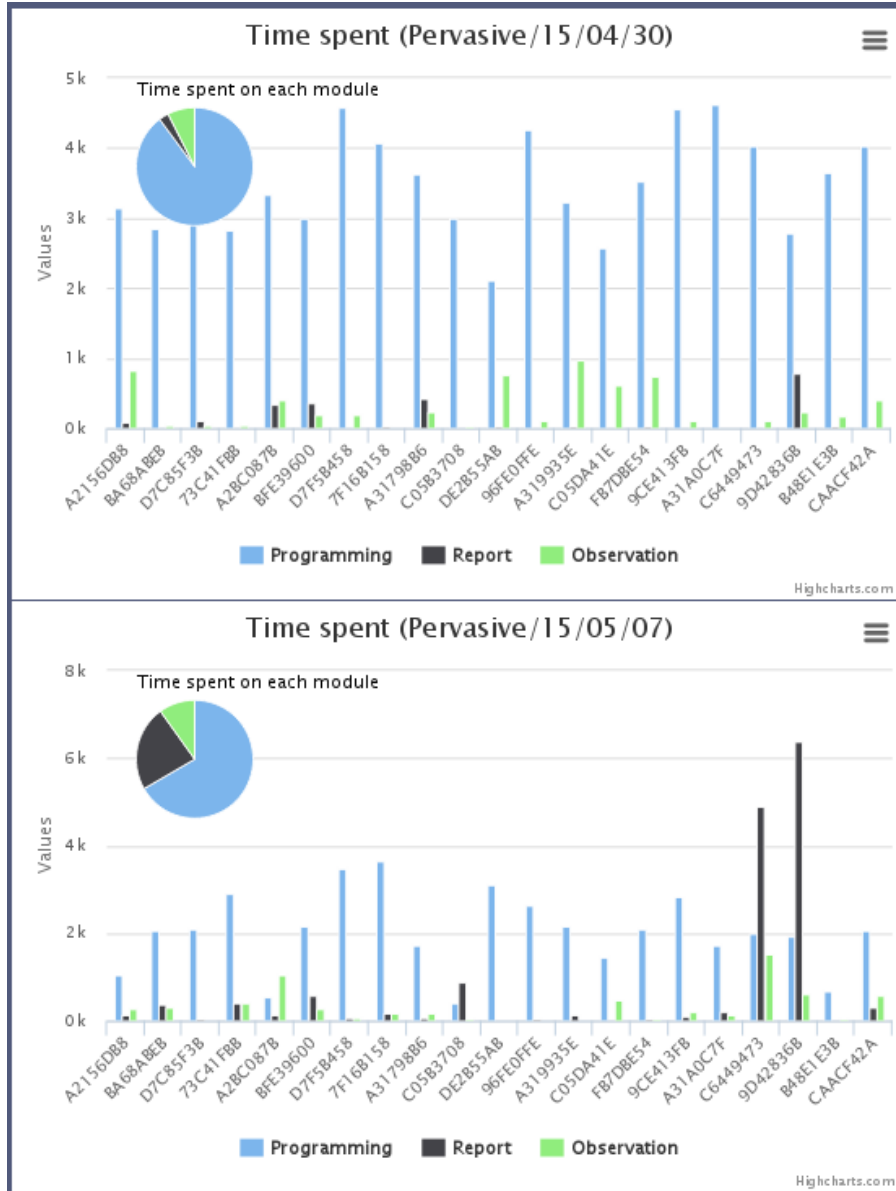


Figure 15 : Durée passée dans chaque module par étudiant du groupe pervasif

3.2.5 Conclusion

Ces premiers résultats montrent que le JENP semble favoriser l'engagement des apprenants par rapport à un JEN plus classique. Les élèves en condition pervasive passent plus de temps à réfléchir sur leurs programmes et sont plus impliqués dans le contrôle de leur déploiement. La motivation, quant à elle, ne semble pas impactée par la différence entre JEN et JENP. L'apprentissage par le jeu est de toute façon bien reçu par les élèves qu'ils aient joué en condition pervasive ou non-persive. De toute évidence, les échantillons de données étant faibles, l'interprétation de ces indicateurs ne peut pas être directement généralisée.

Lors de cette expérimentation, nous n'avons pas analysé l'engagement social dans les communautés, car aucun outil particulier n'avait été fourni aux apprenants pour échanger ensemble. Cela est devenu possible avec une nouvelle version du prototype, grâce au déploiement d'une plateforme incluant entre autres des forums et des documents partagés (Sanchez *et al.*, 2016). De plus, de nouvelles traces concernant le débat réalisé au cours de la dernière session seront collectées et permettront une analyse plus complète, notamment pour analyser le processus décisionnel collectif et les impacts des communautés.

Le protocole de l'expérimentation, composé de deux conditions dans lesquelles différents participants ont utilisé différents prototypes, peut également limiter les conclusions. Cependant, les prototypes ont été conçus pour avoir les mêmes fonctionnalités.

En conclusion, le JENP offre la possibilité d'ajouter de l'immersion et de la persistance à l'environnement virtuel d'apprentissage, en ayant un impact positif sur l'engagement des apprenants.

3.3 Réflexion méthodologique sur la co-conception de JENP

Le projet JEN.lab a permis de réaliser de nombreuses séances de co-conception avec les différents acteurs. Cette expérience sur 3 cycles nous permet de proposer des pistes de réflexion sur la co-conception de ces JENP.

3.3.1 Méthodologie des jeux sérieux immersifs

De nombreuses méthodologies de conception de jeux sérieux sont proposées dans la littérature. Beaucoup d'entre elles sont fondées sur l'utilisation de patrons de conception (Marne *et al.*, 2012) ou de modèles (Carvalho *et al.*, 2015). Une spécificité des JENP est de permettre à l'apprenant de se confronter à une situation complexe et authentique au travers de plusieurs communautés. Ces caractéristiques laissent place à de nombreux scénarios de jeux mais elles requièrent également une réflexion avancée pour les concepteurs. En effet, l'expertise d'ingénieurs pédagogiques, de *game designers* voire même des développeurs experts en RMV peuvent être nécessaire pour garantir un agencement pertinent des éléments ludiques et pédagogiques dans le scénario du JENP. Dans ce cas, la conception de ce jeu sérieux requiert une méthodologie facilitant la collaboration des différents concepteurs.

Pour ce type de collaboration, Marfisi *et al.* (2010) propose une méthodologie de conception de jeux sérieux en 7 étapes, illustrée en figure 16, permettant d'ordonner les tâches que chaque acteur (formateur et *game designer*) aura à accomplir.

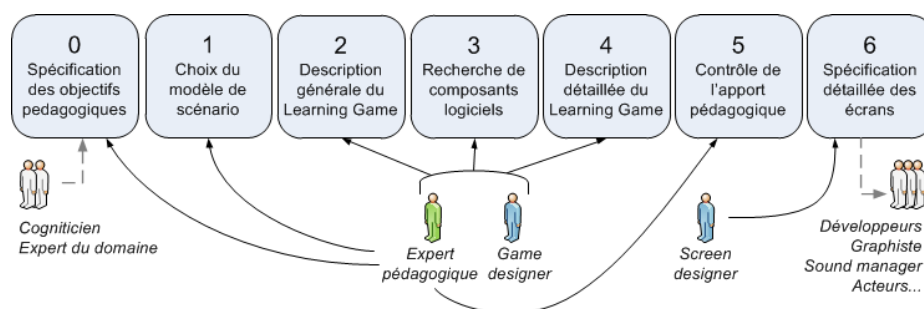


Figure 16 : Les 7 étapes pour concevoir les jeux sérieux selon Marfisi (2010)

Lors de l'étape 0, l'expert pédagogique se doit de recueillir et formaliser les objectifs pédagogiques.

L'étape 1, il propose de choisir la structure scénaristique du jeu en tenant compte du profil des apprenants et du contexte d'utilisation. Marfisi préconise l'utilisation des « jeux à mission ». Ces derniers incitent le joueur à enchaîner des étapes dans un ordre chronologique.

Lors de l'étape 2, l'expert pédagogique et le *game designer* se doivent de décrire le scénario général. Un espace séparé est proposé à chaque concepteur afin que l'un décrive la structuration pédagogique et l'autre, la mise en scène ludique du scénario. Cette description est réalisée au travers de trois niveaux de granularité. Le niveau 1 regroupe les grands modules pédagogiques et leurs mises en scène dans les missions ludiques. Le niveau 2 plus précis, permet de décomposer chaque module en actes et chaque mission en séquences. Le niveau 3, encore plus fin, décrit chaque activité pédagogique et chaque élément de l'IHM.

L'étape 3 invite expert pédagogique et designer à rechercher des composants logiciels réutilisables afin de limiter le coût de développement. Lors de l'étape 4, les mêmes acteurs détaillent la description commencée à l'étape 2. L'étape 5 est dédiée à la vérification

par l'expert pédagogique, des nouveaux éléments du scénario avec les objectifs initialement fixés dans l'étape 1. Lors de l'étape 6, l'équipe du *game designer* proposent au niveau de granularité le plus fin, technique détaillent la description de l'étape 2 à un niveau de granularité toujours plus fin tout en vérifiant la cohérence entre l'aspect pédagogique et ludique. Grâce à l'outil LEGADEE associé à cette méthodologie, les concepteurs réussissent à produire des spécifications claires pour décrire des activités plus variées.

Ce modèle a été réadapté par Orliac *et al.* (2012) à destination des jeux d'apprentissage en RMV. Ainsi le modèle f-MRLG permet entre autres, de prendre en compte des périphériques et des objets réels ainsi que les interactions qui en dépendent. Ce modèle préconise que préalablement à la formalisation, une phase de créativité soit effectuée par l'ensemble des concepteurs. Durant cette phase, les concepteurs sont invités à explorer les possibilités existantes sur les mécaniques ludiques et les dispositifs immersifs en utilisant des cartes à compléter et à positionner sur un poster. Une catégorie de cartes est dédiée aux éléments ludiques, une autre à l'apprentissage et une dernière catégorie à l'environnement du jeu. Des cartes vierges sont également mises à disposition.

Dans le projet JEN.lab, nous avons animés plusieurs séances de créativité lors de chaque cycle, dans l'objectif de pouvoir formaliser la scénarisation de nouveaux JENP. Lors de ces séances de créativité entre co-concepteurs aux profils variés, nous avons utilisé de nouvelles cartes et établi progressivement une procédure de *storyboarding* pour les JENP. Dans le cas des JENP, les apprenants peuvent participer à plusieurs activités de nature très différentes dans le même jeu. Chaque utilisateur peut appartenir à une ou plusieurs communautés. Ainsi une catégorie de cartes de type social, permet de déterminer si chaque activité est individuelle, collaborative, coopérative, etc. Pour assurer l'aspect

pervasif du jeu, il est également suggéré aux concepteurs d'effectuer des choix de cartes concernant l'immersion et de la persistance du jeu. De plus, l'ensemble des caractéristiques de JENP peuvent être gérées de façon plus performante que d'évaluer à la fin de la conception si le jeu comporte bien les fonctionnalités requises.

Pour cela, la partie 2 de notre manuscrit proposera une nouvelle méthode de réutilisation des interactions innovantes.

3.3.2 Proposition pour l'aide à la conception d'un JENP

Avant de formaliser le scénario d'un JENP dans un outil auteur, nous avons mené des séances de co-conception entre enseignants, chercheurs, ingénieur-pédagogique, *game designers* et développeurs. L'expérience acquise au cours de ces séances, nous permet de proposer de nouvelles cartes ainsi qu'une procédure de storyboarding. L'objectif de ces propositions est de faciliter les premières séances de créativité entre co-concepteurs de JENP.

3.3.2.1 Jen Cards

La conception collaborative nécessaire pour les JENP requiert de concevoir les différents aspects du jeu en utilisant un langage commun. Comme Deng préconise des objets tangibles pour concevoir les jeux sérieux, un jeu de cartes spécifique a été élaboré (Deng *et al.*, 2014). Ces cartes, nommées *jen cards* (Figure 17), sont réparties en 4 catégories : sociale, ludique, compétence et technologique (Serna *et al.*, 2015). Les cartes sociales permettent de choisir les modalités d'échanges entre les apprenants. Les cartes ludiques proposent des ressorts ludiques (être

« soumis au hasard »), et des mécaniques ludiques (ex : « élire »). Les cartes compétences définissent généralement des compétences transversales mais peuvent également être associées à une discipline. Dans le cadre de la thèse, nous avons établi la quatrième catégorie de cartes que sont les cartes technologiques. Ces cartes devront permettre à tous les co-concepteurs d'effectuer des choix en termes de technologies et d'interactions, sans connaissances préalables.

Le jeu de cartes n'est pas figé. Par exemple, de nouvelles cartes technologiques pourront être ajoutées à l'apparition de nouveaux services ou dispositifs matériels. De même, les cartes compétences requièrent de changer l'intégralité de leur contenu dès qu'un enseignant vise de nouveaux objectifs pédagogiques.

Comme l'illustre la figure 18, les cartes technologies sont regroupées en 6 catégories, telles que la localisation, l'augmentation, le partage, la communication, le réseau et la capture. Chaque carte présente un service particulier illustré par une icône et expliqué par une brève description. Ces cartes ne font pas directement référence à un seul matériel ou une technique bien précise mais une catégorie de façon à rester accessible à tous les co-concepteurs.

Afin d'assurer l'aspect pervasif du jeu conçu, certaines cartes technologiques sont associées à l'immersion, d'autres à la persistance. Plusieurs cartes, telles que celles liées aux technologies Web, sont considérées à la fois immersives et persistantes. En effet, les pages et les forums garantissent la persistance et, même si l'immersion est plus limitée sur une page Web, certains sites offrent maintenant la possibilité de visualiser des vidéos stéréoscopiques ou à 360 °.

La simplification des concepts peut également paraître comme une représentation insuffisante pour certains utilisateurs. Il peut donc être

pertinent d'ajouter des informations au dos des cartes ou bien d'y ajouter des QR codes permettant de visualiser des vidéos d'exemples.

Lors des expérimentations du JEN.lab, nous avons constaté que les principaux avantages de l'usage des cartes est d'être produites à moindre coût et de permettre aux concepteurs de s'accorder sur les concepts à intégrer dans le jeu. Le principal inconvénient reste qu'un travail de pré-sélection de l'animateur de séance s'impose sans quoi les concepteurs risquent d'être confrontés à l'analyse de centaines de cartes.

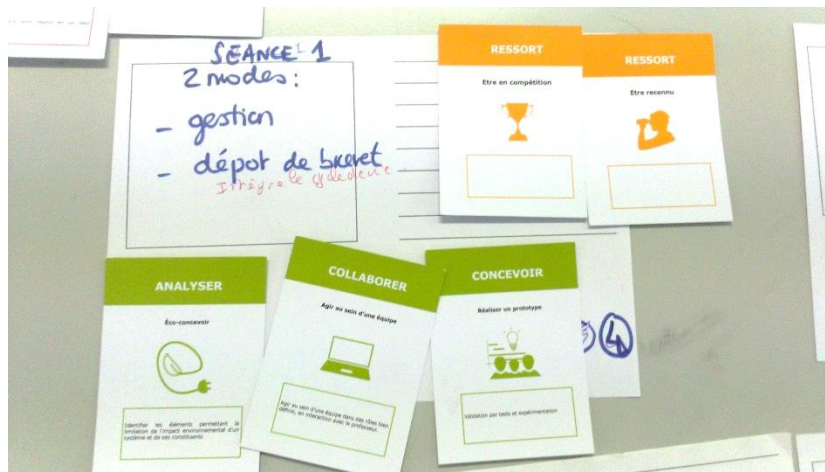


Figure 17 : Utilisation de jen cards en séance de co-conception pour le projet JEN.lab

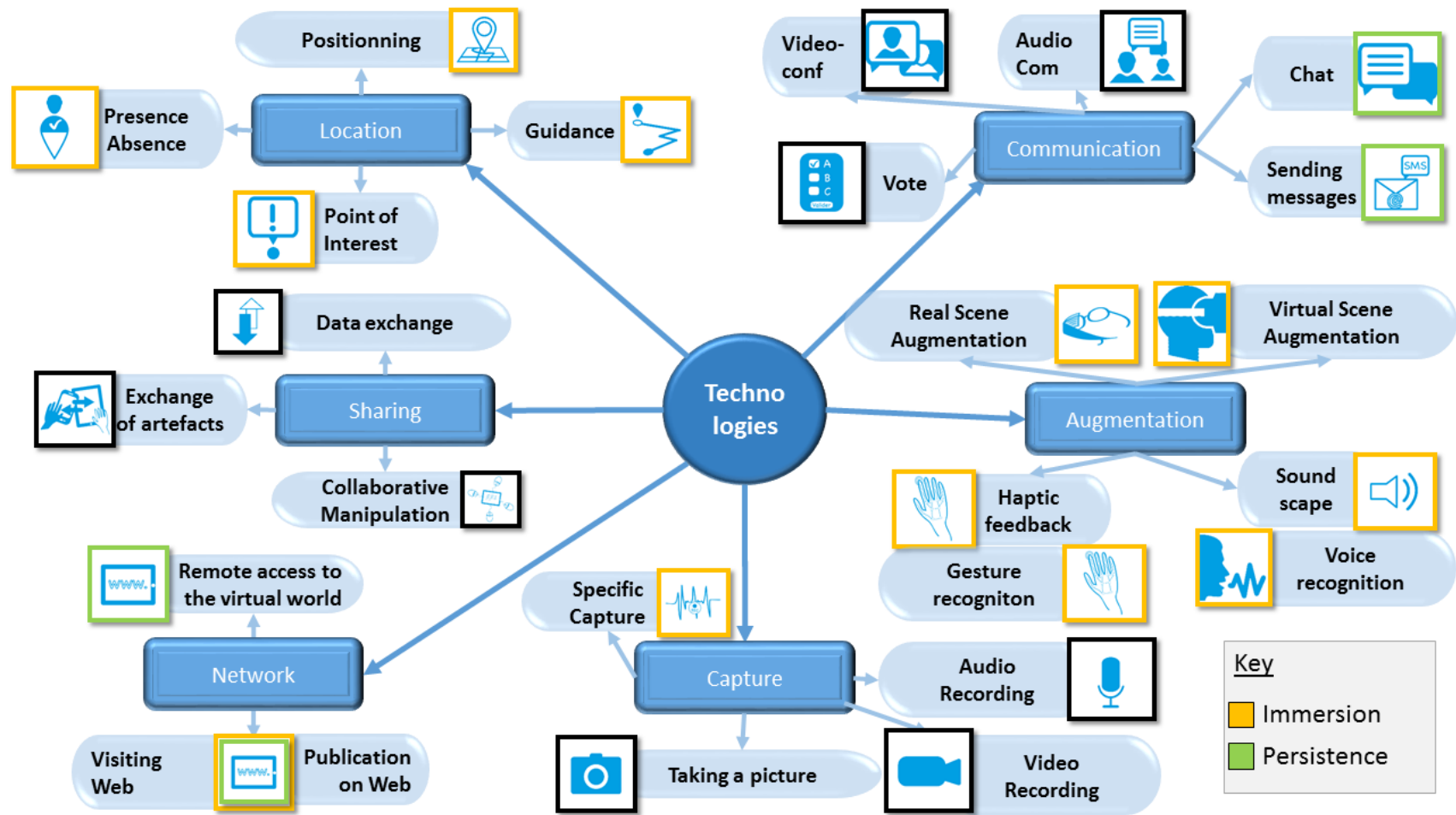


Figure 18 : Représentation de toutes les cartes technologiques

3.3.2.2 Phase de créativité des JENP

Dans le projet JEN.lab, nous avons encadré des séances de créativité avec des co-concepteurs, à l'aide des *jen cards*. Lors du premier cycle du projet, les cartes ont été mises à disposition pour aider les différents participants à communiquer librement sur des concepts communs. Sans leur fixer d'objectifs précis, les co-concepteurs ont multiplié des propositions, des alternatives scénaristiques plutôt de s'accorder et en développer une.

Ainsi lors des cycles suivants, nous avons donc suggéré des objectifs intermédiaires lors des séances de créativité de JENP. L'objectif final a été d'accompagner les co-concepteurs à la production des planches de *storyboarding* offrant une base à la formalisation du scénario.

La procédure que nous mis en place a été effectuée en 4 phases. Comme évoqué en figure 19, ces phases se sont déroulées de façon séquentielle. Cependant, lorsqu'une modification a eu besoin d'être effectuée dans une phase antérieure, nous avons vérifié l'ensemble des conséquences sur les autres phases.

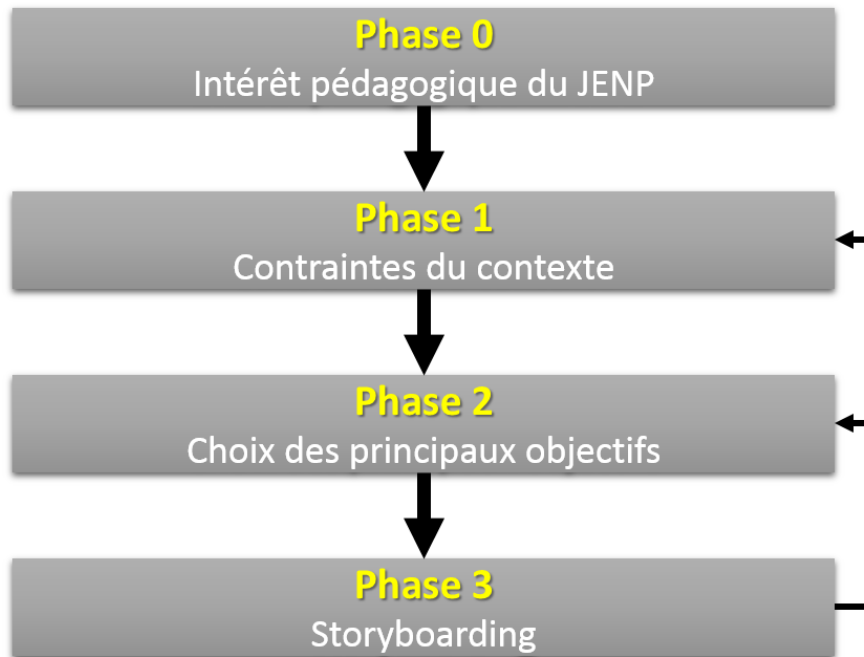


Figure 19 : Procédure des séances de créativité d'un JENP

Dans un premier temps, nous avons fait découvrir aux co-concepteurs les apports pédagogiques et les caractéristiques des JENP.

Dans un second temps, nous avons suggéré à l'ensemble des co-concepteurs d'évoquer leurs contraintes organisationnelles et budgétaires pouvant impacter les limites du jeu. Ainsi si l'enseignant n'avait pas la possibilité d'effectuer de sortie scolaire, les *jen cards* de service de localisation des technologies pouvaient être retirées pour la suite de la procédure. De même, la définition du nombre d'heures de cours et d'apprenants, nous a permis d'établir un repère pour le choix des activités durant les phases suivantes.

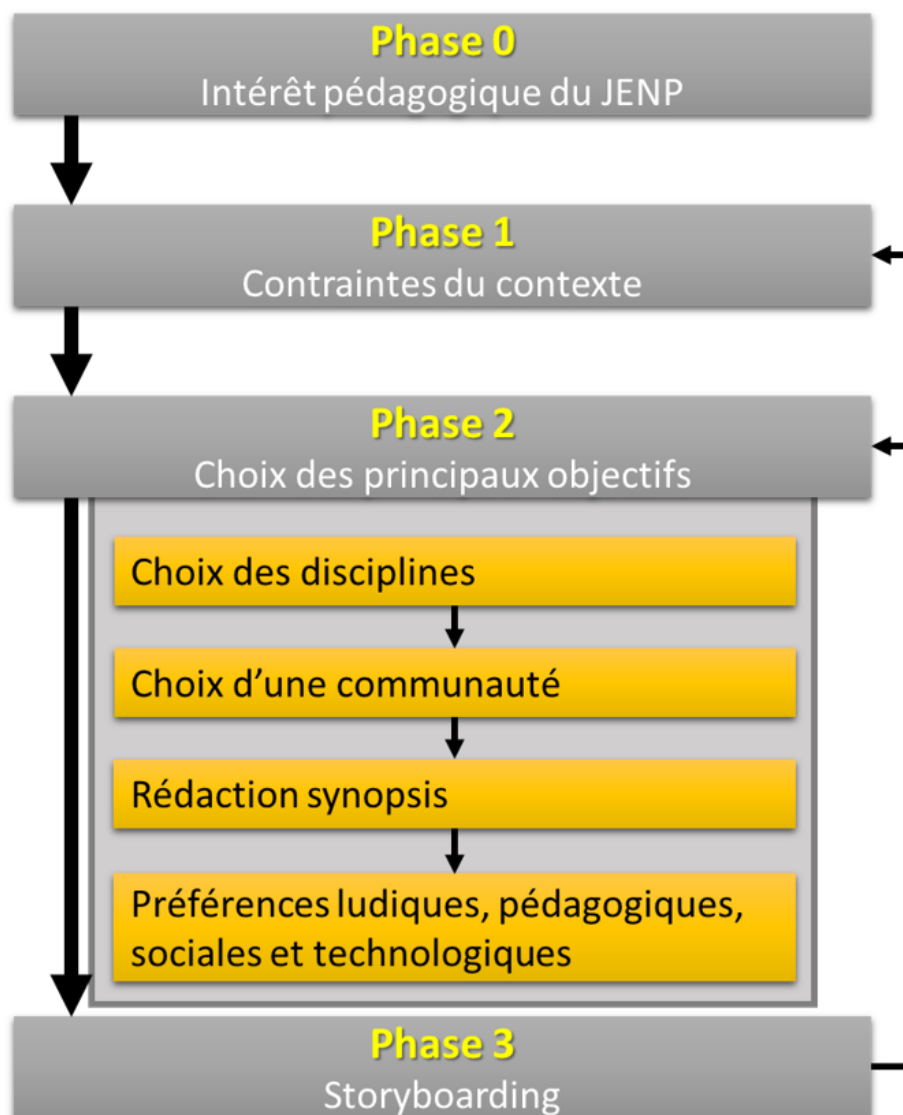


Figure 20 : Phase 2 des séances de créativité d'un JENP

Lors de la phase 2, illustrée en figure 20, nous avons eu comme objectif de définir les principaux objectifs. Pour cela, nous avons invité les co-concepteurs à choisir une liste des disciplines telles que les mathématiques, la biologie, l'économie ou tout autre. Cette démarche peut contribuer à la caractéristique de pluridisciplinarité du JENP. A la suite de ce choix, nous avons pu préparer les cartes de compétences *jen cards* correspondant aux attentes des enseignants.

Nous avons ensuite suggéré aux co-concepteurs de proposer ensuite des communautés liées aux disciplines préalablement choisies. Ce choix a pu permettre de définir le profil que l'apprenant. Ce choix invitait ainsi à la réflexion du cadre épistémique des JENP.

Dans cette même phase 2, nous avons invité les co-concepteurs à définir un contexte scénaristique potentiellement compatible avec les communautés choisies. Pour assurer la caractéristique de l'authenticité, nous avons invité les co-concepteurs à définir une base scénaristique réaliste. Cette étape a consisté à rédiger un court synopsis pouvant être retouché lors des étapes suivantes de rédaction du *storyboard*.



Figure 21 : Etape 3 de la Phase 2 pour le prototype RearthM3

Comme illustré sur la figure 21, la dernière étape de la phase 2, nous avons suggéré aux co-concepteurs, de préciser leurs préférences de concepts à l'aide des *jen cards*. Pour cela, nous avons demandé de choisir au moins deux cartes pour chaque catégorie, afin d'exprimer les compétences, les mécanismes ludiques et les concepts technologiques

préférés. La dernière phase est fondée sur ces préférences tout en laissant la liberté d'en ajouter ou en supprimer.

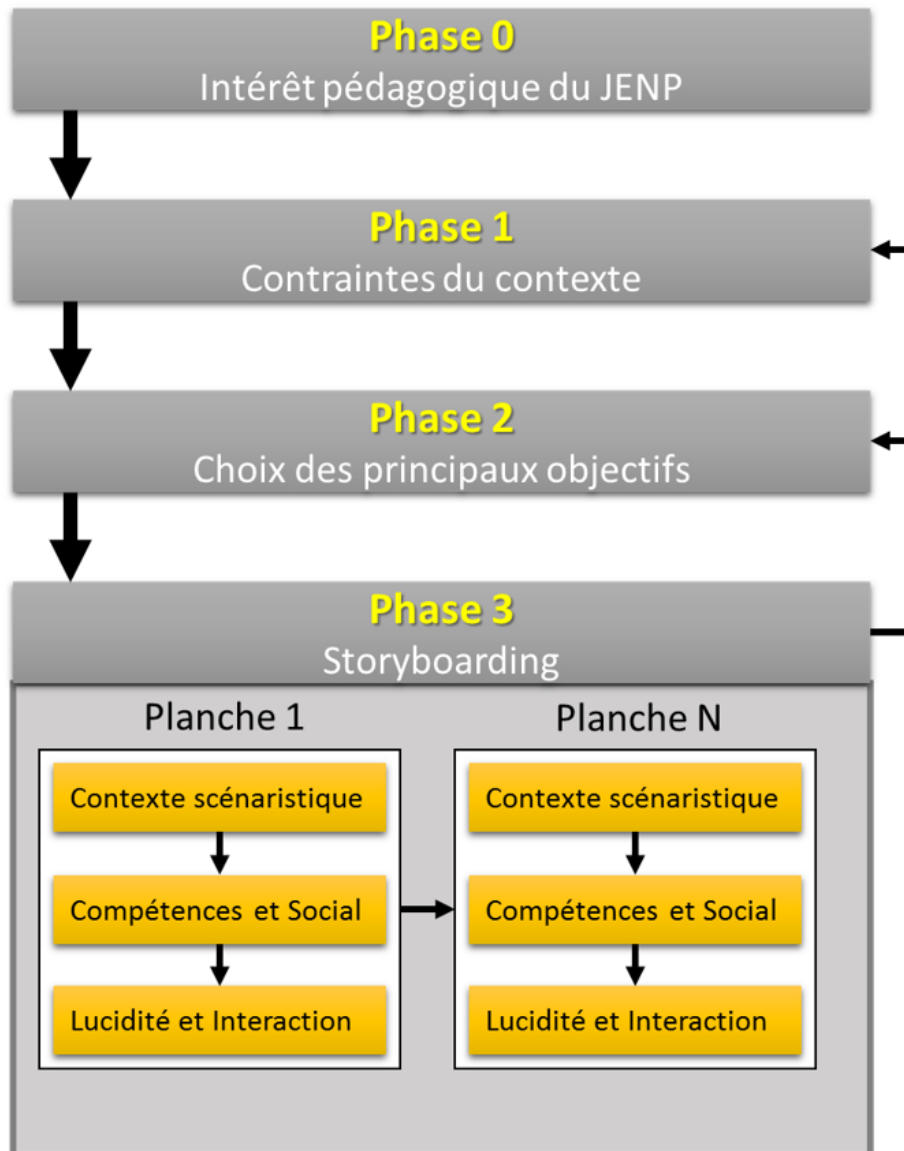


Figure 22 : Phase 3 des séances de créativité d'un JENP

Comme illustré en figure 22, la phase 3 a eu pour objectif de proposer plusieurs planches de *storyboard* correspondant chacune à une étape scénaristique.

Pour chaque planche, nous avons suggéré aux co-concepteurs d'établir un contexte scénaristique en cohérence avec le synopsis ainsi que les éventuelles communautés. De plus, nous avons conseillé que les tâches évoquées respectent l'authenticité des interactions.

Concernant l'intérêt pédagogique, nous avons proposé aux co-concepteurs d'associer à la planche, des cartes compétences et des cartes sociales. La planche de *storyboard* a également été accompagnée de schémas et de textes complémentaires. Lors de cette démarche, nous avons suggéré une réflexion sur l'aspect non-déterminisme et complexe du problème auquel l'apprenant serait confronté.

Nous avons ensuite invité les co-concepteurs à associer à la planche, les cartes ludiques et les cartes technologiques. Nous avons déconseillé d'alterner des étapes exclusivement pédagogiques et des étapes exclusivement ludiques (Widmer et Szilas, 2011). Enfin, les co-concepteurs ont sélectionné des cartes technologiques définissant les outils utiles aux tâches décrites sur la planche. De plus, nous avons suggéré d'augmenter l'authenticité de la situation de jeu par l'utilisation des cartes relatives aux systèmes immersifs et persistants.

3.4 Synthèse

Ainsi un guide de conception et des *jen cards* ont été conçus pour qu'un groupe composé de corps de métiers toujours plus large, puisse proposer un nouveau jeu. Ce processus permet également la prise en compte du matériel et des contraintes organisationnelles, facilitant la mise en place d'une pédagogie innovante adaptée à leur centre de formation. En quelques séances, les co-concepteurs ont la possibilité de produire un cahier de charges complet pour que les game-designer, développeurs et infographistes puissent maquetter et chiffrer le coût de l'implémentation.

La persistance étant implémentable par des technologies courantes, l'importance du coût du JENP reste dépendante de la capacité des développeurs de jeux sérieux à intégrer rapidement des interactions immersives.

Les expérimentations ont permis, à l'aide d'un prototype JENP dans un lycée, de démontrer l'apport des technologies innovantes au concept du JEN sur l'engagement des apprenants. A l'aide d'indicateurs d'engagement spécialement conçus, il a été démontré que le comportement des élèves a été plus favorable dans un prototype JENP que dans un prototype JEN.

PARTIE 2 : Faciliter le développement d'une interaction immersive

Dans le cadre du projet JEN.lab, la co-conception du JENP a amené les *game designers* et les enseignants à collaborer dans des séances de créativité ainsi que par une rédaction formelle du jeu à l'aide d'outil *LEGADEE*. A partir de ces informations, le développeur du JENP est parvenu à réaliser un jeu sérieux contenant des interactions immersives dans un environnement persistant.

Comme illustré sur la figure 23, à l'aide d'informations de haut niveau, les développeurs et infographistes doivent définir avec précision les éléments du graphisme, des animations et de l'interaction. L'implémentation d'interactions immersives peut nécessiter des compétences peu courantes dans la communauté des développeurs de jeux sérieux. Il devient ainsi probable qu'une sous-traitance à des experts RMV soit nécessaire à la réalisation d'un JENP. Pouvant engendrer des contraintes financières et organisationnelles, le seul développement des interactions immersives risque de rendre les JENP moins accessibles aux formateurs.

Ainsi, la seconde partie de la thèse est dédiée à la recherche d'une solution permettant aux développeurs de jeux sérieux d'implémenter des interactions immersives adaptées à leur contexte.

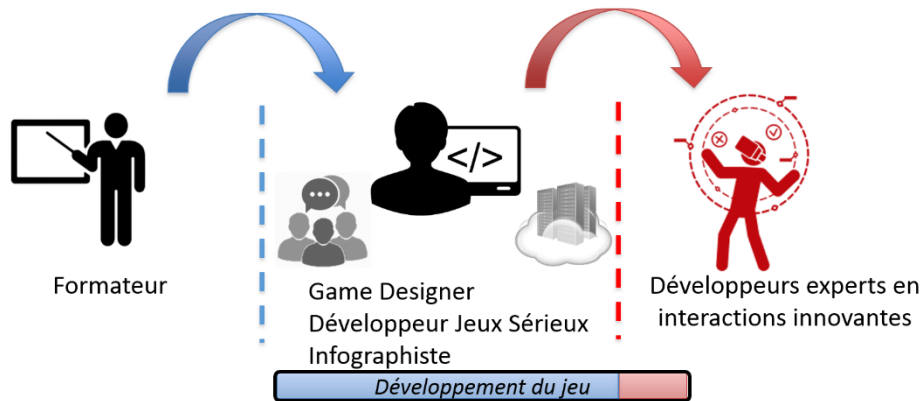


Figure 23 : Processus de développement d'un JENP sans outil dédié

Au cours de ces derniers mois, les plus grands géants du numérique se sont lancés dans une véritable course pour offrir de nouveaux périphériques de la réalité virtuelle et augmentée au grand public. Des périphériques tels que les visiocasques ont été expérimentés pour différents usages depuis plus de 30 ans, jusqu'au jour où un investissement bien plus important a été porté sur un modèle plus léger, à l'affichage plus fluide et la résolution plus importante. Ainsi grâce à ces nouveaux casques immersifs, une large majorité d'utilisateurs peuvent avoir leur cerveau trompé sur le sens de la vue même si des nausées persistent encore (Davis *et al.*, 2015). Dans ce contexte, les réflexions sur des nouveaux usages se multiplient en particulier dès lorsqu'on souhaite rendre l'utilisateur plus acteur que simple spectateur. Même si le visiocasque, ne prenant en compte que les mouvements de tête, commence à être admis comme un standard, les accessoires permettant les interactions avec le reste du corps restent encore très diversifiées. De plus même si une partie de ce type de matériel semble suffisamment mature pour être exploité tout comme les mondes virtuels (Fenn et LeHong, 2011) comme l'illustre la dernière courbe en figure 24, qu'en est-il de leurs interactions ?

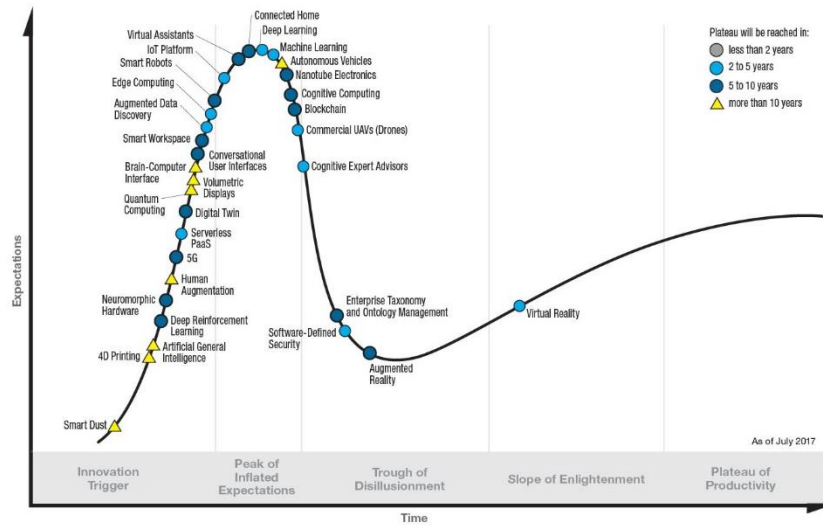


Figure 24 : Cycle du Hype selon Gartner en 2017

Le développement d'interactions peut être destiné autant aux visiocasques qu'à du matériel tactile, tracking, haptique, voire même olfactif, gustatif ou cerveau-ordinateur.

Ainsi pour répondre à ce besoin des développeurs de jeux sérieux, nous décrivons dans le chapitre 4, les avantages et limites d'outils facilitant le développement d'application immersive. Dans le chapitre 5, une proposition d'un métamodèle d'interaction immersive sera détaillée. Enfin, le dernier chapitre de cette partie décrit d'abord MIREEDGE, un outil pour les développeurs d'interactions immersives puis l'expérimentation permettant de valider les concepts mis en avant dans MIREEDGE.

Chapitre 4 : État de l'art sur les outils de développement d'interaction

Pour déterminer l'existence d'un outil adapté au profil des développeurs de jeux sérieux, il est nécessaire de définir avec précision leurs besoins. Dans un second temps, une liste de catégories d'environnements de développement permettant l'implémentation d'interactions immersives, doit être confrontée avec le besoin défini.

4.1 Besoins des développeurs

4.1.1 Évolution du développement d'interaction

On peut constater que même si la diversité des périphériques immersifs est croissante, le nombre d'applications dédiées aux visiocasques est encore très limité. Les premières applications ne permettaient que de simples mouvements de tête, laissant à l'utilisateur la seule possibilité que d'observer (Davis *et al.*, 2015). Puis d'autres ont souhaité associer au visiocasque des périphériques courants tels que les manettes de jeu. Le développement d'application est resté simple à programmer mais l'usage n'était pas toujours instinctif, il pouvait même rendre l'utilisateur malade rapidement (Tan *et al.*, 2015). Différents systèmes de détection de position et d'orientation de l'utilisateur (*tracking*), plus légers et moins coûteux ont alors ouverts de nouvelles perspectives (Hilfert et

König, 2016). Ils requièrent un effort physique limité tout en assurant une meilleure représentation cognitive. Le *tracking* tout comme la représentation des membres peut varier de la main, jusqu'à l'intégralité du corps. De plus, un même périphérique utilisant une même représentation 3D et souhaitant réaliser une même action peut être implémentée sous différentes formes d'interactions. Ainsi une manette HTC Vive, représentée par une main virtuelle 3D, peut sélectionner un objet d'une scène 3D soit en plaçant la main au contact de l'objet, soit en le pointant de l'index.

Ainsi on peut supposer que même s'il existe très peu de différences pour un infographiste entre la modélisation d'objets 3D dans une application classique et celle d'une application immersive, les concepteurs et développeurs sont confrontés à de véritables défis pour définir des interactions inédites pleinement adaptées.

Plus concrètement, les concepteurs peuvent décrire le contexte d'une scène, les différentes actions de haut-niveau et leurs conséquences au travers d'éventuels *storyboards*. Le développeur, quant à lui devra encore effectuer un grand nombre de choix pour définir le comportement exact, soit chaque action de bas niveau, que l'on souhaite affecter à chaque objet de la scène. Ce comportement peut se définir quand l'objet 3d est seul, lorsqu'il est confronté à un autre mais aussi en fonction du comportement d'un ou plusieurs utilisateurs. La différence pour le développeur entre la réalisation d'une application immersive et celle d'une application non-immersive se situe principalement sur les algorithmes d'interaction entre l'objet virtuel et l'utilisateur.

Les développeurs de jeux sérieux sans expertise en RMV (développeurs non-RMV) doivent décrire l'interaction de manière détaillée mais sans terme technique.

De plus, s'ils ont une approche itérative et participative, ils se doivent d'être capables d'effectuer un prototypage rapide. Ces développeurs non-RMV doivent également garder la possibilité d'adapter les interactions aux recommandations de l'utilisateur.

4.1.2 Vers un besoin pour tous types de développeurs

De nos jours, la majorité des développeurs aptes à gérer ce type d'interactions innovantes sont actuellement des spécialistes de concepts et matériels de réalité virtuelle, soit des développeurs réguliers en réalité mixte ou virtuelle (développeurs RMV). Leur nombre étant relativement limité, le besoin croissant de leurs prestations engendre un coût important pour de nouvelles applications. Il est ainsi nécessaire de savoir s'il existe un moyen pour les développeurs non-RMV, de réaliser des applications telles que celles évoquées dans la partie 1. Ces développeurs non-RMV peuvent quelques fois être assimilés à des débutants en réalité virtuelle or leur objectif principal n'est pas forcément d'acquérir plus de connaissances, de maîtriser plus d'outils tels que les développeurs RMV mais seulement de pouvoir ajouter des interactions RMV à leurs applications. Sans connaissance avancée sur le périphérique, une de leurs principales interrogations reste la faisabilité de l'interaction qu'ils souhaitent réaliser. Bien que la liste des interactions possibles pour chaque périphérique soit dépendante de ses caractéristiques techniques, elle l'est aussi de la créativité et l'ingénierie du développeur qui saura implémenter son usage. Ainsi il serait un avantage conséquent pour chaque développeur d'accéder à une bibliothèque proposant l'ensemble des interactions possibles à chaque périphérique.

La majorité des nouveaux développeurs 3D temps réel se risquent rarement à enrichir leur monde virtuel avec des interactions innovantes, malgré le nombre croissant de périphériques RMV sur le marché. Cependant, de nombreux protocoles ont été proposés pour faciliter l'implémentation des interactions RMV. Chacun de ces protocoles est destiné soit à un profil de développeur RMV, soit à un profil de développeur non-RMV. En ciblant un seul profil, l'outil offre des spécificités avantageuses mais au risque de limiter réellement son usage. En prenant en compte que les développeurs non-RMV ont des niveaux de compétences très variés et bien différentes des développeurs RMV, nous nous intéressons à une solution permettant de dépasser les limites liées au profil du développeur. De cette façon, il pourra être envisagé de s'appuyer sur les avantages des outils communautaires tels que la capitalisation des codes, l'évolutivité et la collaboration. Afin que chaque interaction puisse être partagée sous différentes formes, nous proposons un métamodèle spécifique.

Pour répondre aux principaux problèmes des développeurs non-RMV et des développeurs RMV, nous proposons un outil hybride prenant en charge la **facilité d'utilisation**, la **réutilisation** et la **personnalisation** des algorithmes d'interaction. Pour proposer une solution pertinente, il est d'abord nécessaire d'établir quelles catégories d'outils existants peuvent répondre à chacun de ces critères.

4.2 Environnements de développement pour applications immersives

Aujourd'hui il existe un grand panel d'outils permettant d'écrire des algorithmes d'interaction RMV. Certains outils sont des environnements de développement indépendants tandis que d'autres sont des extensions d'environnements (Kreylos, 2008). Certains sont destinés exclusivement aux profils des développeurs RMV, d'autres ciblent les développeurs non-RMV. Certains se limitent à du prototypage rapide, d'autres sont voués au développement de produits finis. Ils peuvent également être compatibles seulement avec certaines catégories de périphériques telles que celles à bas coût ou à l'inverse, les plus onéreuses. Nous avons décidé de classer les multiples outils de développement en 4 catégories : les plateformes de développement dédiées à la réalité virtuelle, les moteurs de jeu munis de bibliothèques fournies par les fabricants de périphériques, les *middleware* RMV et les outils d'assistance pour les moteurs de jeu. Il convient de noter qu'il est possible de cumuler dans les moteurs de jeux, les bibliothèques, les *middleware* et les outils d'assistance. Notre analyse se base sur les critères que devrait posséder l'outil : la facilité d'utilisation de l'environnement de développement, la réutilisation et la personnalisation d'algorithmes d'interactions.

4.2.1 Plateforme de développement dédiée à la réalité virtuelle

La catégorie des plateformes dédiées à la RMV est la plus ancienne. Il y a déjà 30 ans, des applications de réalité virtuelle ont été conçues pour

répondre à des besoins militaires essentiellement (plus qu'à ceux du divertissement ou de l'éducation). Ces applications étant vouées aux spécialistes de l'époque, les outils de développement proposaient à la fois de gérer le comportement de tous les objets, des animations simples à des interactions plus complexes. L'idée principale est de pouvoir programmer l'ensemble d'une scène au moyen d'une seule et même interface illustrée en figure 25. Ces outils essaient de proposer un maximum de fonctionnalités à un nombre très limité de programmeurs spécialisés. De ce fait, par souci de rentabilité, leurs coûts d'acquisition des licences ; de formation et de maintenance sont importants. Cependant, l'évolution de certaines plateformes de développement a été consacrée à la simplification des interfaces de programmation. Pour cela, les plus populaires tels que Virtools et Eon studio ont proposé de la programmation visuelle (Yingyan *et al.*, 2009). La programmation visuelle pouvant limiter le champ d'action, et les développeurs RMV souhaitant garder un maximum de liberté, Virtools illustré en Figure 26, permet également de coder soit même son bloc de programmation.

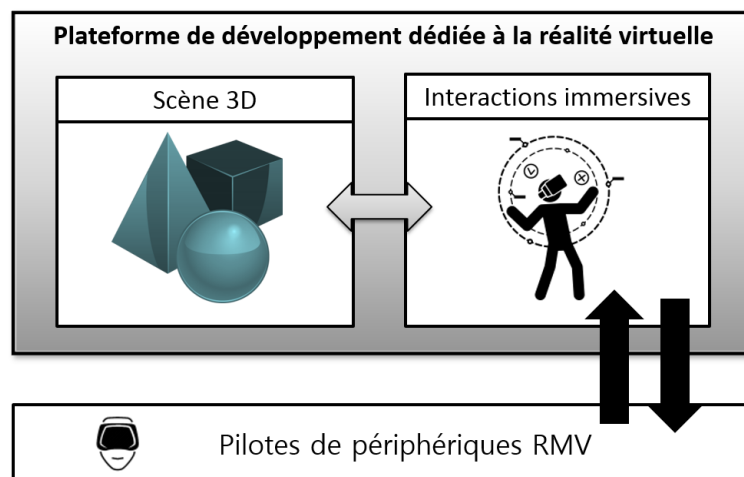


Figure 25 : Programmation d'interactions en environnement de développement dédié à la réalité virtuelle

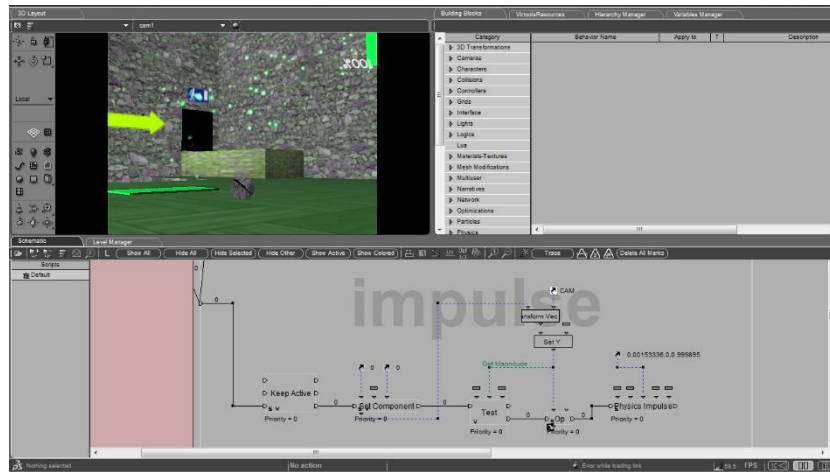


Figure 26 : Programmation d'interactions sous Virtools

En prenant en considération les nouveaux périphériques RMV, les développeurs doivent attendre voire même payer pour que les auteurs des plateformes les mettent à disposition.

4.2.2 Moteurs de jeu équipés de bibliothèques de fabricants

A l'heure où la course au jeu vidéo semblait imposer des mondes virtuels toujours plus grand et complexe pour les plus passionnés de jeux vidéo, un nouveau marché a fait son apparition, celui de la catégorie des jeux pour utilisateurs occasionnels (Juil, 2010). Ce nouveau marché a permis de mettre en avant des jeux se devant de rester financièrement plus accessibles pour les joueurs occasionnels. Pour limiter le prix d'achat, outre des méthodes de publicité, la durée de développement se doit de rester relativement faible, tout comme le coût d'achat des environnements de développement. Pour répondre à ces besoins, de petits studios et des développeurs indépendants se sont équipés de moteurs de jeu moins puissants mais financièrement plus accessibles (Eberly, 2006). Ces outils offrent des possibilités d'exporter les

applications sur Internet, sur mobile et même sur les consoles de jeu. La performance de leurs moteurs 3D temps réel a commencé à satisfaire le grand public et ces moteurs sont ainsi perçus comme une nouvelle opportunité pour chaque développeur de jeu. Pour faciliter le travail des développeurs, de célèbres industriels tels que Oculus ou HTC Vive, souhaitant un panel d'applications toujours plus large, ont proposé des bibliothèques officielles pour des moteurs tels que Unity3D et Unreal Engine 4. La figure 27 montre que ce type de bibliothèque permet d'accéder directement aux propriétés du matériel. De plus, il est à noter que les fabricants des matériels, souhaitant une large diversité d'interactions ont fourni uniquement des accès à des informations de bas niveau afin que chaque développeur crée l'algorithme qui lui convient. Les développeurs qui utilisent ces bibliothèques doivent donc posséder une expertise technique pointue permettant de prendre en compte les contraintes bas niveau des dispositifs dans leurs algorithmes. Par exemple, fournir des informations brutes sur l'état d'un bouton d'une manette ou celui d'un gyroscope d'un casque à un instant T demande encore beaucoup de réflexion pour implémenter un algorithme d'interaction permettant de déplacer un objet virtuel. À l'aide de cette méthode de développement, les développeurs RMV ont une grande liberté. *A contrario*, même si l'import de bibliothèques de fabricants de dispositifs reste simple pour les développeurs non-RMV, ils peuvent se retrouver bloqués par manque de connaissances techniques et devront rechercher des informations par leurs propres moyens ou en demandant sur les forums.

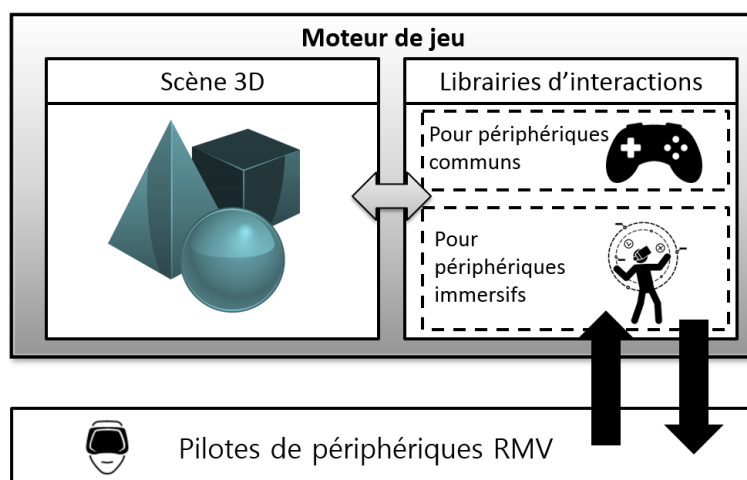


Figure 27 : Programmation d'interactions dans un moteur de jeu avec des bibliothèques de fabricants de dispositifs RMV

Ainsi malgré la complexité de cette solution, l'avantage est d'être la méthode la plus utilisée par une communauté qui souhaite s'entraider.

4.2.3 Moteurs de jeu équipés de *Middleware*

Un *middleware* en réalité virtuelle a pour rôle principal d'assurer le transfert de données entre les périphériques et l'application finale, comme l'illustre la figure 28. Un des meilleurs exemples de ce type de *middleware* utilisé depuis plusieurs décennies est VRPN (Virtual Reality Peripheral Network) (Stelzer *et al.*, 2014). Ce *middleware* offre des services simples et efficaces (Taylor *et al.*, 2010). Il aide principalement à unifier les différents protocoles d'échange requis par chaque périphérique. Il peut gérer plusieurs périphériques en même temps et peut échanger les données en mode client-serveur. VRPN est un *middleware* qui traite des données de bas niveau. Il requiert donc des compétences avancées en réalité virtuelle. Ainsi même si unifier ces données simplifie une partie de la tâche au développeur RMV, toute

l'implémentation de l'algorithme d'interaction reste encore à coder dans la plateforme de développement.

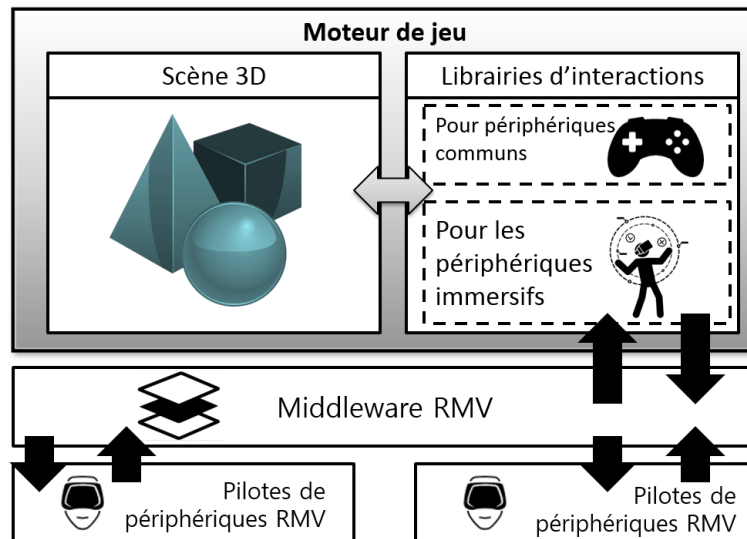


Figure 28 : Programmation des interactions dans un moteur de jeu avec un *middleware* RMV

D'autres *middleware* offrent des interfaces et des bibliothèques de plus haut niveau tel que MiddleVR (Koenig *et al.*, 2014, p.). Cet outil associé au moteur Unity3D permet de combiner les fonctionnalités de VRPN avec une interface de configuration beaucoup plus accessible. Une première étape propose, au travers d'une application extérieure illustrée en figure 29, d'étalonner le ou les périphériques que l'on souhaite utiliser. Dans un second temps, le développeur programme son interaction directement dans l'environnement de développement tel Unity3D. Tous les périphériques gérés par MiddleVR peuvent être programmés sans avoir besoin de paramétrer directement leurs pilotes. L'implémentation de l'ensemble des interactions peut ainsi être réalisée par une méthode uniforme. Ce *middleware* payant est déjà utilisé par de nombreux développeurs RMV. Si ces derniers souhaitent réutiliser leurs algorithmes d'interaction pour une nouvelle application, ils doivent les

dupliquer et les transférer manuellement d'un projet à l'autre. En ce qui concerne les développeurs non-RMV, la manipulation de la bibliothèque commune requiert des compétences avancées sur les périphériques et leurs limites. Les mises à jour du *middleware* permettent de gérer de nouveaux périphériques. Leur fréquence dépend de la capacité et de la stratégie de l'entreprise car MiddleVR est une application commerciale.

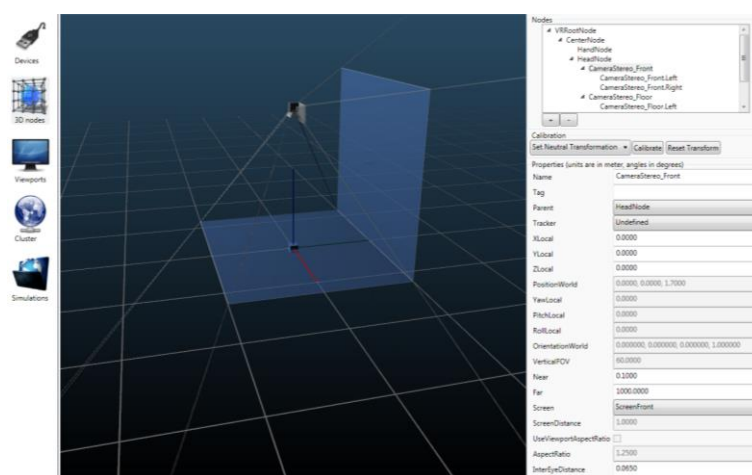


Figure 29 : Paramétrage des périphériques sous MiddleVR

Il convient de noter que, en tant qu'applications indépendantes, ces *middleware* assurent une compatibilité plus importante avec les périphériques que dans les autres catégories de solutions.

4.2.4 Moteurs de jeu équipés d'outils d'assistance

Le terme d'outil d'assistance RMV se réfère à de nouvelles bibliothèques, voire même de véritables outils qui sont intégrés à l'environnement de développement, pour faciliter l'implémentation d'interaction RMV. Contrairement aux bibliothèques des fabricants de

périphériques, ces outils proposent de manipuler des informations plus proches de l'interaction que du matériel, comme l'illustre la figure 30.

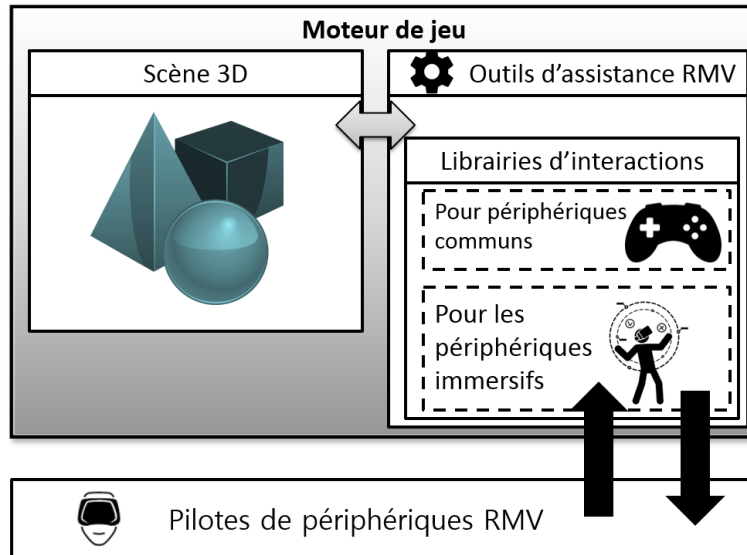


Figure 30 : Programmation d'interaction dans un moteur de jeu muni d'outils d'assistance RMV

Un outil de référence existant depuis plusieurs années et faisant l'objet de plusieurs publications est *Reality-based User Interface System* (RUIS) (Takala, 2014). RUIS est gratuit et destiné aux développeurs non-RMV. Le principal objectif de cet outil est de permettre le développement d'interaction sans connaissance avancée en réalité virtuelle. Comme illustré en figure 27, RUIS est un ensemble de composants, qui sont importés directement dans les moteurs de jeu Unity3D ou Processing. Chaque composant possède un usage aussi divers que la configuration son affichage, un autre pour la gestion de son déplacement. Dans l'exemple de la figure 31, un composant est dédié à la configuration d'une Kinect, un second à la configuration d'un Oculus et un troisième à l'interaction de déplacement dans la scène 3D. Ainsi ces composants peuvent configurer et intégrer rapidement des interactions prédéfinies que l'on souhaite ajouter à la scène 3D. Même si

le paramétrage d'un composant est une tâche relativement simple pour un développeur, des échanges et des connexions entre le composant et le reste de l'application deviennent beaucoup plus compliqués. Cela peut même susciter des modifications profondes dans d'autres scripts afin de les rendre compatibles avec le nouveau composant intégré.

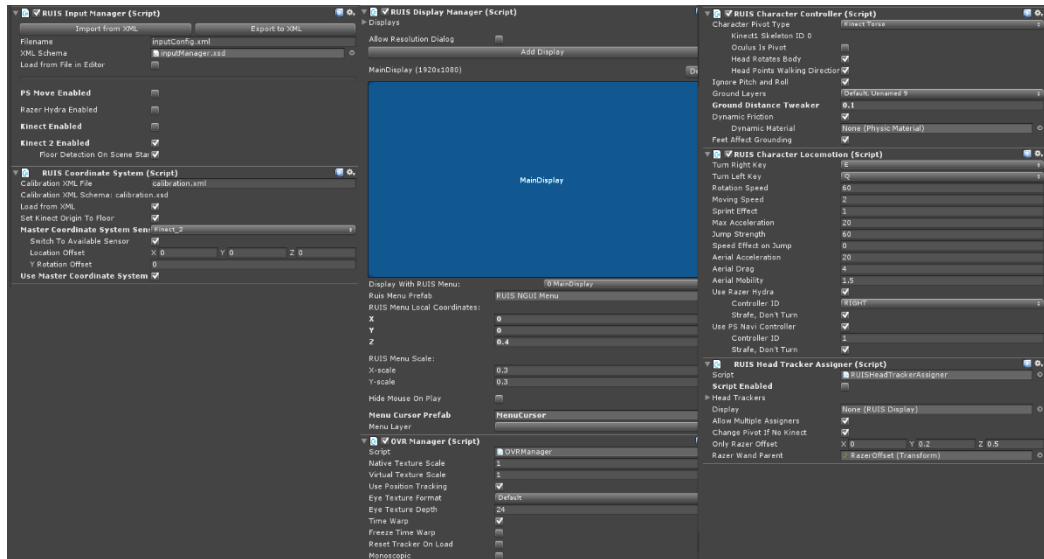


Figure 31: Interface de paramétrage RUIS sous Unity3D

L'outil RUIS gère une liste variée de périphériques : tels que la Kinect, le Razer Hydra, la Playstation Move ou encore l'Oculus. Les développeurs de ces composants sont uniquement les auteurs de RUIS. De plus, en limitant l'utilisateur à des tâches de configuration, la personnalisation de l'algorithme d'interaction en est tout autant restreinte.

4.2.5 Solutions hybrides

Certains environnements de développement peuvent également être en corrélation avec plusieurs des 4 catégories précédentes. Ainsi la plateforme Mascaret (Abidi *et al.*, 2016) est une approche fondée sur la

modélisation pour la conception d'environnement virtuels sémantiques (Chevaillier *et al.*, 2012). Elle permet de décrire le comportement, les interactions et les tâches de chaque entité de la scène 3D sans requérir de connaissance technique en réalité virtuelle. Cette approche emploie essentiellement le langage UML¹². Cependant cette solution impose de considérer chaque objet ou utilisateur comme des agents. Ainsi l'ajout d'un objet interactif implique de lui associer un algorithme compatible avec tous les autres objets de la scène.

#FIVE est une plateforme de développement d'environnement virtuel interactif et collaboratif (Bouville *et al.*, 2015). Elle est centrée sur la réutilisation par un système d'*entité-composant*. #FIVE fournit une boîte à outils qui facilite l'implémentation d'actions et de comportements d'objets ainsi que le paramétrage et la gestion d'interactions collaboratives. #FIVE propose un guide pour créer efficacement un environnement virtuel collaboratif et interactif. #SEVEN est une évolution de #FIVE par la gestion supplémentaire du scénario global en langage C# sous Unity3D. Avec une approche similaire, l'outil *WF toolkit* (Elvezio *et al.*, 2016), tout comme #SEVEN, ne permet pas aux utilisateurs, de créer eux-mêmes, de nouveaux composants.

La solution *Interaction Techniques Markup Language* (InTml) de Figueroa *et al.* (2008) est un ensemble d'outils permettant de représenter un algorithme d'interaction sous deux formes : Une représentation simplifiée de l'interaction sous forme de diagramme destinée aux concepteurs RV et un compilateur permettant de transformer le diagramme en script C++ ou Java. Ces derniers sont voués à être intégrés par les développeurs RV dans leur projet. En termes de compatibilité, InTml est une application sans dépendance avec une liste de

¹² Unified Modeling Language

périphériques préétablie. Il est à noter que son approche est fondée sur les concepts de modèles et de la programmation orientée composant. De nouveaux composants peuvent être créés par les développeurs RV. Ce type de solution peut ainsi faciliter le prototypage rapide tout en ayant un code efficace (Knott *et al.*, 2014).

Cependant la limite de InTml, est que l'utilisateur non-RV est le concepteur. Ainsi la représentation est très simplifiée car le concepteur a rarement la capacité de programmer lui-même l'interaction immersive. À l'inverse, le développeur de jeux sérieux non-RV a besoin de composer un algorithme d'interaction pleinement compatible à sa scène virtuelle. Pour rendre certains concepts plus accessibles aux programmeurs, les moteurs de jeu proposent des interfaces de programmation visuelle, telle que *Blueprints* (Sewell, 2015). Cependant, cette interface garde une compatibilité limitée avec les moteurs de jeu et les périphériques RMV.

4.3 Synthèse

Nous confrontons les 4 catégories d'outils de développement d'applications immersives dans le tableau 3, notamment en ce qui concerne les critères liés aux besoins identifiés précédemment.

Tableau 3 : Comparaison des outils de développement RMV selon les besoins des développeurs non-RMV

<i>Implémentation de l'algorithme d'interaction</i>	Plateforme de développement dédiée à la RMV	Moteur de jeu		
		avec les bibliothèques des fabricants RMV	avec les <i>Middleware</i> RMV	avec des outils d'assistance RMV
Facilité d'utilisation pour les	Requiert des connaissances en	Requiert des connaissances en dispositif immersif	Requiert des connaissances en	Facile mais uniquement

développeurs non-RMV	dispositif immersif		dispositif immersif	pour les usages prévus
Capitalisation du code pour les développeurs RMV	En exportant soi-même son code et en le plaçant dans un magasin en ligne	En exportant soi-même son code et en le plaçant dans un magasin en ligne	Seuls les auteurs de l'outil peuvent partager	Seuls les auteurs de l'outil peuvent partager
Échanges entre les développeurs –non -RMV et les développeurs RMV	Collaboration possible par les interfaces de programmation visuelle	Le développeur-non-RMV doit être assisté par le développeur RMV	Le développeur-non-RMV doit être assisté par le développeur RMV	Collaboration possible mais dans un périmètre restreint
Possibilité de réutilisation	Liste de blocs disponible	Recherche, duplication et réécriture des scripts	Recherche, duplication et réécriture des scripts	Liste de composants disponible
Évolution (compatibilité avec de nouveaux périphériques et leurs pilotes)	Requiert des mises à jour ou des paiements dans le magasin en ligne	Dépend de la réactivité du fabricant	Requiert des mises à jour de l'outil	Requiert des mises à jour de l'outil
Gratuité	Rarement	De plus en plus	Quelques fois mais très coûteux dans les autres cas	Couramment
Exemples	3dvia studio Eon Reality studio Virtools	Unity3D; Unreal Engine avec des SDK tels que Oculus ou HTC Vive	VRPN, MiddleVR	RUIS, ARToolkit, FreeVR

Il est à noter que les plateformes dédiées offrent souvent la possibilité de réutilisation à l'aide de blocs de programmation. En ce qui concerne les moteurs de jeu, les bibliothèques des fabricants RMV permettent de personnaliser pleinement les interactions de périphériques récents. Les *middleware* facilitent la standardisation mais seulement pour les développeurs RMV. Les moteurs de jeu associés à des outils d'assistance

RMV ont l'avantage d'offrir une facilité d'utilisation, mais seulement pour des contextes préalablement définies.

A l'inverse, une solution hybride telle que InTml semble pouvoir cumuler les critères évoqués. Cependant elle reste destinée exclusivement au profil du développeur non-RMV. Nous pouvons supposer qu'à partir d'une architecture similaire à InTml et de modélisations réadaptées, une nouvelle solution pourrait répondre à la fois aux besoins des développeurs RMV et des développeurs non-RMV.

Chapitre 5 : Modélisation d'une interaction 3D immersive

Pour envisager de modéliser autrement les algorithmes d'interactions 3D immersives, il est nécessaire de mieux cerner, la définition, les intérêts et les limites des interactions 3D.

5.1 Interaction 3D

Le paradigme d'interaction est reconnu comme un ensemble de techniques permettant à un utilisateur de réaliser une tâche dans un environnement virtuel.

La première classification de Hand (1997) complétée par Bowman (1999) regroupe les techniques d'interaction 3D selon les tâches de navigation, manipulation, de sélection et de contrôle de l'application. D'autres catégories ont également pu être suggérées telles que celles des changements d'échelle, menus virtuels et autres widgets.

Il reste cependant important de distinguer deux types d'interaction 3D, les immersives et les non-immersives. Dans notre cas, le concept d'immersion est associé à l'usage d'un ou plusieurs périphériques RMV et non à celui des sensations pouvant elles, être associées au concept de présence (Mestre *et al.*, 2006).

Concernant les interfaces non-immersives, les travaux de Jankowski et Hachet (2015) proposent une classification détaillée au travers de nombreuses catégories. Ainsi pour la seule navigation de l'utilisateur

dans l'environnement 3D : les sous-catégories liées au mouvement général d'exploration, au mouvement de recherche de cible, à la saisie de coordonnées et à la construction de trajectoire possèdent chacune plusieurs techniques. Ces dernières proposent des solutions bien distinctes pour faciliter entre autres, le point de vue de l'utilisateur, son mouvement et ses limites dans l'espace. Il est aujourd'hui possible d'évaluer la facilité d'utilisation de ces techniques par des méthodes qui leur sont spécifiques. Enfin, des interactions allant au-delà du clavier, souris et tactile ont prouvé leur efficacité tels que l'interface cerveau ordinateur et la navigation de graphiques 3D sur le Web.

Concernant les interactions immersives, selon le traité de la réalité virtuelle (Fuchs et al., 2006a), les activités des utilisateurs en RMV sont décomposables en quelques comportements de base : les Primitives Comportementales Virtuelles (PCV) dont les catégories sont proches de celles de Hand et Bowman. D'un point de vue matériel, l'interaction 3D est liée à un ou plusieurs périphériques d'entrée ainsi qu'un ou plusieurs périphériques de sortie. D'un point de vue logiciel, l'interaction 3D est liée à une technique d'interaction logicielle offrant à l'utilisateur un scénario d'utilisation pour accomplir une tâche. Cette partie logicielle peut inclure une Aide Logicielle Comportementale (ALC) permettant de combler les défauts des dispositifs technologiques. Par exemple, la saisie d'un objet virtuel à partir des coordonnées de chaque doigt, ne peut pas le limiter à attendre un positionnement similairement à la réalité. Ainsi pour faciliter cette saisie, il peut être simulé une forme de « magnétisme » entre l'objet virtuel et la position de la main.

Les technologies restant encore peu matures pour retranscrire avec exactitude le geste réel dans l'environnement virtuel, Burkhardt (2006b) envisage deux approches. La première invite les concepteurs, malgré les limites matérielles, à établir un système exploitant le geste « naturel ».

Les facteurs de réalisme tels que la qualité et la précision de l'interprétation des gestes restent indispensables. La seconde approche possible est d'avoir recours à des métaphores, même si cela peut nuire parfois à l'utilisabilité. Ces métaphores peuvent être associées à la manipulation d'objets virtuels, à un déplacement dans l'espace ou simplement à contribuer à l'animation d'un avatar. Ces métaphores ne sont autres que des modèles d'interaction que l'utilisateur devra reproduire. Ainsi face à un système de tracking de corps, laisser des deux pieds côte à côte peut être associé à la position d'arrêt et placer un pied devant l'autre peut être associé à l'action de marcher. A l'inverse de l'approche demandant un geste naturel similaire à celui effectué dans le monde réel, les métaphores requièrent que les concepteurs communiquent des règles à l'utilisateur. Elles ont cependant l'avantage majeur de réduire le temps de réalisation de la tâche ainsi que les efforts physiques à fournir par l'utilisateur.

Concernant les tâches les plus complexes, d'après une analyse comparative (Ouramdane *et al.*, 2009), la précision, la sécurité et la charge cognitive des interactions 3D doivent encore être optimisées. Les industriels ne priorisent pas encore tous ces critères, or dans des domaines tels que la santé, l'aérospatial et les milieux hostiles à l'homme, ce type d'interaction devient cependant une véritable alternative.

5.2 Modélisation de l'interaction

Selon Fuchs (2006b), il existe trois niveaux d'immersion et d'interactions : le niveau sensori-moteur, le niveau cognitif et le niveau fonctionnel. Au premier niveau, seules les actions physiques de l'utilisateur avec les entités du monde virtuel sont considérées. Un

second niveau prend en compte l'ensemble du processus cognitif de l'utilisateur. Dans le dernier niveau, l'objectif n'est pas de proposer seulement une immersion cognitive à l'utilisateur mais de considérer l'ensemble du processus de l'activité dans le monde virtuel.

Pour concevoir, implémenter ou étudier une interaction, il est nécessaire de commencer par la décrire. Cette description peut être réalisée en utilisant une multitude d'informations et présentée sous une infinité de formes. Pour que ces descriptions puissent être exploitables, il est nécessaire de simplifier leur représentation en ne faisant apparaître que les éléments utiles leur destinataire, ce que l'on appelle des modèles. Deux catégories de modèles existent, les modèles d'interaction utilisés pour la conception et les modèles architecturaux destinés au développement de l'application.

Dans le domaine de l'informatique et plus particulièrement celui de la réalité mixte, Chalon a proposé IRVO, un formalisme facilitant la conception d'IHM de systèmes de réalité mixte (Chalon, 2004). Dans ce dernier, la frontière entre le monde virtuel et le monde réel est représentée de façon explicite. Les entités réelles et virtuelles y sont réparties de chaque côté, les objets mixtes sont représentés comme un ensemble d'objets réels et virtuels. Une extension de ce formalisme, appelé IRVO+, ajoute la notion de zone aux utilisateurs, objets et outils déjà présents, comme illustré dans la Figure 32. Son niveau de représentation le rend principalement utile pour les phases de conception. IRVO se situe à un niveau d'abstraction tel qui permet uniquement de visualiser l'ensemble de la tâche sans détailler les échanges de données, le paramétrage des entités ni le séquençement des actions indispensable à la création d'un algorithme d'interaction.

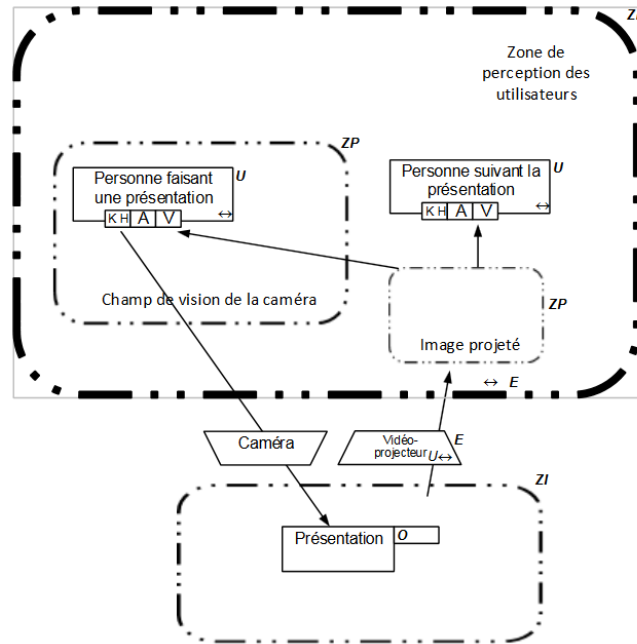


Figure 32 : Exemple de modèle d'interactions gestuelles avec le formalisme IRVO+ (Delomier, 2013)

Pour décrire de façon plus détaillée une interaction dans un environnement virtuel, les formats X-VRML et le X3D (Walczak et Cellary, 2003) proposent de décrire en XML, les objets, les animations et interactions d'un environnement virtuel. L'avantage de cette description est d'être à la fois précise et universelle même elle présente l'inconvénient d'avoir été utilisée uniquement dans des scènes peu interactives. Néanmoins, cette structuration X3D permet par l'approche VR-Wise (Pellens *et al.*, 2007) de produire des représentations graphiques simplifiés de comportement d'agents virtuels, comme illustré en figure 33 ainsi que des d'interactions, comme illustré en figure 34.

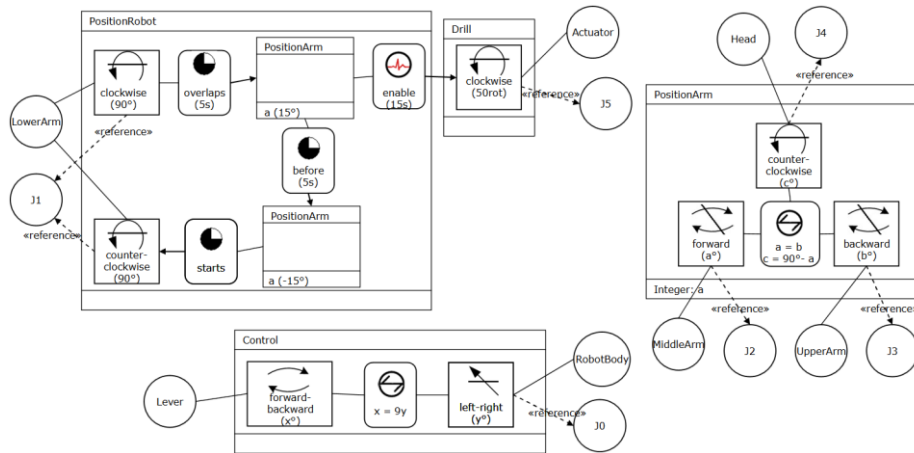


Figure 33 : Exemple de modèle de comportement avec le formalisme VR-Wize (Pellens *et al.*, 2007)

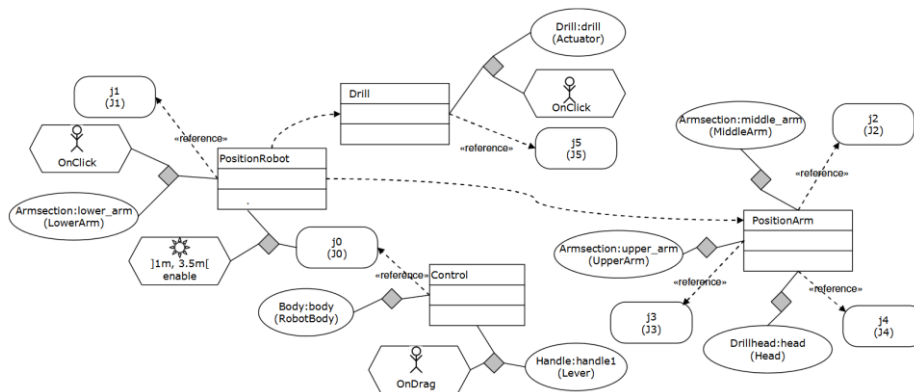


Figure 34 : Exemple de modèle d'interaction avec le formalisme VR-Wize (Pellens *et al.*, 2007)

Cet outil facilite la collaboration entre l'utilisateur final et l'expert en réalité virtuelle lors de séances de conception. Concernant le développeur de l'interaction RMV, cette production se limite à fournir des informations afin de coder par lui-même l'interaction dans son environnement de développement.

Cependant, les travaux de Radkowski démontrent que la conception de diagramme d'activités à l'aide du langage UML¹³ permet de concevoir des modèles de comportements facilitant directement la tâche du développement de l'application (Radkowski et Hilus, 2011). L'avantage de cette approche est de transformer un modèle en script à intégrer dans une application RV existante. Sa principale limite est de ne produire que la programmation événementielle dans un nombre limité d'environnements de développement.

Toujours structuré par l'UML, le métamodèle *Virtual Environment supporting Human Activities* (VEHA) conçu pour des environnements virtuels de grande taille, permet d'exécuter en temps réel des modèles d'environnements métier indépendants des plateformes (Chevaillier *et al.*, 2009). Un de ses principaux objectifs est la trace d'activités dans les environnements virtuels. Son concept repose sur celui de l'agent artificiel ou humain. L'agent doit connaître en détail l'environnement avec lequel il doit interagir. Opérationnel sur plusieurs plateformes, ce métamodèle peut être utilisé pour faciliter la conception et le suivi d'applications immersives, plus particulièrement ceux d'environnements multi-agents. Cependant il impose au développeur de structurer l'intégralité des animations et interactions de sa scène avec le concept d'agent, qui bien que théoriquement possible dans de nombreux cas, ne correspond à la logique de tous les développeurs.

¹³ Unified Modeling Language

5.3 Proposition d'un métamodèle pour les interactions immersives

Comme défini dans les chapitres précédents, le besoin auquel nous souhaitons apporter une nouvelle solution est prioritairement celui du développeur non-RMV, en laissant également la possibilité au développeur RMV de s'approprier l'outil. Cet acteur peut en effet rencontrer des difficultés pour développer une interaction 3D adaptée au périphérique immersif qu'il n'utilise qu'occasionnellement. Comme soulevé dans la littérature, les concepteurs et développeurs peuvent rarement se contenter de mettre en place des interactions de base mis à disposition par un industriel ayant produit le périphérique. Il est le plus souvent nécessaire d'en connaître les limites et de mettre en place de nouvelles métaphores d'interaction adaptées à la situation visée.

De plus, avant l'implémentation des interactions, le développeur non-RMV peut avoir mis en place une scène contenant un ensemble d'objets. Le comportement de chacun de ces objets peut avoir été développé avec une logique de programmation très différente. Ainsi le développeur se doit de coder une interaction n'imposant pas de reprogrammer les comportements des objets. Concernant le développeur RMV, nous souhaitons lui laisser un accès direct au code source généré à partir de l'interaction modélisée afin qu'il puisse l'adapter finement à son besoin. Essentiellement destiné à des développeurs, cette modélisation peut être effectuée à un niveau de granularité faible pour définir un véritable algorithme. Sa limite de précision est fixée par la possibilité de rester réutilisable pour un maximum d'environnement de développement.

Pour répondre à ces critères, nous avons établi un métamodèle d'interaction immersive, formalisé en UML et représenté schématiquement dans la figure 35.

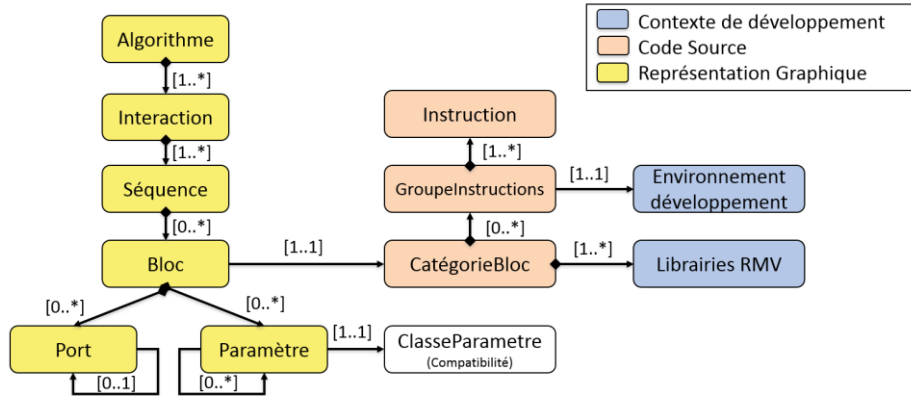


Figure 35 : Représentation simplifiée du métamodèle d'interaction

5.3.1 Représentation visuelle de l'interaction

Souhaitant placer ce métamodèle au plus proche de la logique du développeur, notre première entité est bien l'**algorithme**. Bien qu'habituellement chaque interaction possède son propre algorithme, nous laissons la possibilité d'en regrouper plusieurs pour optimiser le temps de modélisation. Ainsi l'ouverture et la fermeture d'un tiroir virtuel pourront être regroupées dans un même modèle. Chaque **interaction** est composée d'une ou plusieurs **séquences** d'actions. Il est ainsi possible de distinguer les étapes d'initialisation, de traitement continu et de procédures événementielles. Le contenu de ces séquences est principalement un séquençage d'actions et d'opérations logiques, regroupé sous le terme de bloc.

Ce **bloc** est l'élément graphique de base de nos modèles. Il est associé à une image et à un texte court pour être représenté simplement. L'intérêt du bloc prend tout son sens lorsqu'il représente en une seule entité des

calculs tels que la distance entre la main et un objet donné ou encore l'état d'ouverture de la main détecté par le périphérique. Comme une entité de la programmation orientée composant, le bloc doit contenir un service bien précis et être réutilisable. Il se doit d'être utile à un grand nombre d'algorithmes et être potentiellement compatible avec de nombreux environnements de développement. L'ordre de séquençement de ces blocs est établi par l'association de leur **port**. Le premier bloc est quant à lui, identifié par le lien qui relie son port à celui de sa séquence. Les blocs possèdent également des **paramètres**, qui eux ont pour seule vocation le partage de valeurs. Chacune de ces valeurs possède un type, nommé **classe de paramètre**, assurant la pertinence d'un échange entre deux paramètres. Les valeurs des paramètres sont actualisées à chaque utilisation du bloc dans l'algorithme. Dans la littérature, d'autres modèles d'interactions unissent les concepts de port et paramètre dans une même représentation. Or cela peut imposer de réitérer des calculs dans le seul but de réutiliser le résultat. Notre objectif final étant de fournir un script à un développeur, il est souhaitable d'orienter notre choix sur celui offrant le meilleur niveau de performance à son algorithme. En contrepartie, nos modèles risquent de devoir représenter plus de liens entre les blocs.

5.3.2 Représentation scriptée

Les utilisateurs ciblés étant également des développeurs experts, nous souhaitons aller au-delà de la représentation graphique. Ainsi la spécificité de notre métamodèle est de transformer l'association de blocs, par une génération de lignes de code produisant un seul et même script. Ce dernier peut ainsi être repris voire même modifié pour réaliser une interaction particulière. Pour cela, comme l'illustre la figure 35, chaque bloc est associé à des groupes d'instructions (cf. annexe 1 pour une

version détaillée du métamodèle). Ces groupes d'instructions sont les codes sources pouvant faire référence à des bibliothèques de périphériques RMV et à des environnements de développement. Ainsi chaque bloc pourra décliner ses codes sources afin de fournir celui qui correspond à l'environnement de développement. Cela permet également de prendre en compte les différentes versions des bibliothèques pouvant être utiles à un même bloc. Cela permet ainsi d'assurer toujours plus de compatibilité.

Il est à noter qu'un bloc peut être associé à une ou plusieurs bibliothèques laissant l'opportunité de créer des blocs spécifiques à des dispositifs d'interaction associés à plusieurs périphériques. Concernant les environnements de développement, outre leurs bibliothèques internes, il a été pris en compte que certains offrent la possibilité d'utiliser plusieurs langages de programmation pour écrire un même algorithme. Un développeur pourra ainsi exploiter le langage qu'il aura lui-même utilisé dans le reste de son application.

Par exemple, un bloc peut être dédié au calcul du nombre de doigts tendus détectés par le périphérique LeapMotion. Un paramètre d'entrée est alors consacré au degré d'inclinaison minimal définir la tension du doigt. La valeur du paramètre de sortie correspond au nombre de doigts détectés comme tendus. L'algorithme peut être souhaité sur des environnements tels Unity3D et Unreal Engine. Dans le cas de Unity3D, le code peut être en langage C# tout comme en Javascript. Au début du script généré, un groupe d'instructions dédié à l'import de bibliothèques se doit d'être en corrélation avec le pilote du périphérique LeapMotion utilisé. Un second groupe d'instructions initialise les variables du calcul en prenant en compte le paramètre d'entrée. Enfin un dernier groupe d'instructions est consacré au calcul à effectuer en continue et mettant à jour le paramètre de sortie.

5.4 Synthèse

Nous sommes aujourd'hui en capacité de classer les interactions 3D en catégories telles que la navigation, la manipulation, la sélection et le contrôle de l'application. Il est devenu possible de les décrire sous forme de modèles d'interaction par des formalismes tels que IRVO ou VEHA. Ces différentes représentations d'interactions ciblent chacune un intérêt spécifique tel que la représentation simplifiée de l'IHM en co-conception ou encore le fonctionnement de systèmes multi-agents. Notre principal objectif reste de faciliter le développement d'interactions 3D immersives aux utilisateurs non-RMV. Plus concrètement nous souhaitons qu'un développeur puisse décrire rapidement, précisément et sans connaissance technique approfondie en RMV, son interaction 3D immersive en apportant un minimum de modifications à son projet existant. Nous souhaitons également qu'un développeur RMV puisse générer le script correspondant afin de le modifier à sa guise. Pour cela, nous proposons un métamodèle d'interaction offrant à la fois une représentation simplifiée de l'interaction mais également une transformation sous forme de codes sources. Enfin, concernant les EVAH, cette solution permet également le suivi de l'activité de l'apprenant par la création de blocs qui leur seront dédiés.

Chapitre 6 : Conception et évaluation de MIREEDGE

Comme mentionné dans le chapitre 4, l'approche hybride InTml est celle qui répond au plus grand nombre de nos critères mais elle n'est pas adaptée au profil des développeurs non-RMV. Le métamodèle du chapitre 5 a permis de proposer une approche théorique plus adaptée. Dans ce chapitre, nous détaillons donc les caractéristiques de la solution nommée MIREEDGE (*Mixed and virtual Reality DEvelopment tool for Game Engine*) ainsi que l'évaluation que nous avons conduite.

6.1 Besoins identifiés et choix de conception

Notre proposition s'inspire de la programmation orientée composant (Brugali et Scandurra, 2009). Dans cette approche, un composant regroupe un ensemble de fonctionnalités répondant à un objectif précis. Le composant se doit également d'être réutilisable. Cela se traduit par le fait que certaines interfaces proposent de gérer des composants sous forme de blocs pouvant être associés et réutilisés. Cette méthode présente l'avantage d'augmenter la productivité et la qualité (Mohagheghi et Conradi, 2007).

Cependant, deux limites sont reconnues à cette solution : le manque de flexibilité des blocs et le besoin de produire de nombreux blocs compatibles avec les diverses plateformes et périphériques.

Nous entendons par manque de flexibilité, la nécessité de recréer un bloc tout entier même si nous souhaitons ajouter une seule fonction basique

non prévue initialement. Pour pallier cette limite, nous proposons que l'assemblage de blocs puisse être transformé en code source (Biernacki *et al.*, 2008).

Le défi de la production de blocs compatibles avec les plateformes et les périphériques, peut être relevé par la communauté de développeurs, à condition qu'elle y trouve suffisamment de bénéfices en retour. Augmenter le potentiel de réutilisabilité semble donc indispensable.

6.1.1 Facilité d'utilisation

Comme nous l'avons décrit précédemment (cf. Chapitre 4), dans un outil tel que RUIS, les activités des développeurs se limitent au paramétrage. Afin d'élargir leur champ d'action, nous pouvons envisager que les développeurs non-RMV puissent programmer. Ainsi grâce à leurs compétences en développement et sans être confrontés à des termes techniques en RMV, ils pourraient personnaliser davantage leurs interactions. Pour cela, les développeurs non-RMV doivent être assistés sur la logique et la syntaxe pour créer des algorithmes d'interactions via une interface simplifiée (Resnick *et al.*, 2009). Nous supposons, qu'après une prise en main suffisante, certains développeurs utilisant cette interface simplifiée pourraient réussir à produire des algorithmes RMV avancés. Un outil tel que Virtools a déjà prouvé qu'il était possible de produire de tels algorithmes avec efficacité (Yingyan *et al.*, 2009). Une grande partie de cette efficacité pour développeur non-RMV est due à l'utilisation d'un langage visuel (Bentrad *et al.*, 2011). Toutefois, la programmation visuelle peut également être perçue comme un ensemble de boîtes noires (Sewell, 2015) avec ses avantages et ses limites. Bien qu'une boîte noire présente l'avantage de masquer la complexité en simplifiant l'appel à de nombreuses fonctions et variables, le champ d'action du développeur s'en trouve limité. L'impossibilité d'accéder

aux paramètres internes suggère que le développeur ne puisse pas produire l'algorithme initialement envisagé. Pour surmonter cette limite, une alternative peut être de conceptualiser le processus de développement en deux étapes : la première permettrait aux développeurs d'écrire leurs algorithmes avec un outil de programmation visuelle, tandis que la seconde leur permettrait d'accéder à la version scriptée du précédent algorithme (du code) afin d'effectuer les ultimes ajustements.

6.1.2 Capitalisation du code

Les développeurs RMV produisent eux-mêmes des algorithmes d'interaction pour satisfaire des demandes spécifiques. Ils peuvent écrire ces algorithmes intégralement ou essayer d'en trouver des similaires ou proches sur des forums, ou dans des anciens projets. Plus récemment, un nouveau type de plateforme en ligne tel GitHub (Dabbish *et al.*, 2012) permet de diffuser le code source. Si cette plateforme permet effectivement de partager l'intégralité d'un projet, les fonctionnalités de recherches proposées ne facilitent pas l'identification d'algorithmes d'interaction précis parmi le code source du projet.

Ainsi, pour la réutilisation de code, habituellement les développeurs limitent leurs recherches à leurs propres projets antérieurs. Analyser le code de projets inconnus implique de devoir comprendre les objectifs du projet ainsi que la logique de chaque auteur et de ce fait, devient trop coûteux en temps.

Afin d'optimiser cette identification d'algorithmes, la solution que nous souhaitons concevoir devra permettre de stocker tout ou partie d'algorithme dans une base de données dont la structure sera suffisamment détaillée et générique pour assurer la normalisation de la description des interactions. Les développeurs RMV pourront ainsi

alimenter et exploiter une banque d'algorithmes afin de ne pas programmer à nouveau ce qui a déjà été pensé par d'autres ou par soi-même.

6.1.3 Partage de connaissances

Internet est aujourd'hui le principal vecteur du partage de connaissances. Dans les échanges les plus communs, un développeur non-RMV pose des questions sur un forum et doit attendre la réponse d'un ou plusieurs développeurs RMV. La réponse du développeur RMV est souvent chronophage, il met à disposition ses connaissances de façon pédagogique et sans attendre quoi que soit en retour. Le développeur non-RMV pourra les consulter sans contrepartie et autant de fois qu'il le souhaite (Plonka *et al.*, 2015).

D'ailleurs, les plateformes web de partage de code de type GitHub permettent des échanges relativement structurés entre développeurs grâce, notamment, à un suivi de bugs et des fonctionnalités de l'application partagée. Toutefois ces plateformes restent destinées à des développeurs aux profils relativement comparables.

Or, dans notre cas, nous souhaitons que les développeurs RMV et les développeurs non-RMV apportent chacun leurs contributions dans une véritable collaboration (Sowe *et al.*, 2008). Cela devra se traduire dans notre solution logicielle par trois caractéristiques majeures. Premièrement, le développeur RMV devra partager un modèle d'interaction qu'il pense pertinent, dans un formalisme de description. Deuxièmement, notre logiciel permettra d'associer le code fourni par un développeur RMV à un bloc visuel, paramétrable et accessible par un développeur non-RMV. Troisièmement, la communauté composée des développeurs non-RMV et des développeurs RMV pourra multiplier les tests.

6.1.4 Réutilisation des algorithmes

Actuellement, en se limitant aux algorithmes d'interaction, un script - quel que soit son objectif - peut être remplacé par un autre dès lors qu'ils utilisent le même langage et le même moteur. Ainsi, deux algorithmes établis selon la même logique et ayant le même objectif ne peuvent pas être associés s'ils possèdent un langage ou s'ils sont issus de moteurs différents. C'est cette limite que nous souhaitons lever grâce à notre solution logicielle en permettant d'associer des algorithmes aux logiques et objectifs communs, sous une représentation schématique commune. Concrètement, cela apparaîtra sous forme de blocs de programmation suffisamment générique pour être réutilisable d'un moteur à l'autre. Cela sera possible par la nature du lien entre la représentation et le code source dans le métamodèle détaillé dans le chapitre précédent.

6.1.5 Capacités d'évolution

Les outils RMV sont également limités par le nombre de périphériques avec lesquels ils sont compatibles. Certains outils imposent cette limite pour éviter une surcharge de mémoire de l'environnement de développement : ces outils intègrent des bibliothèques, des ressources et/ou des pilotes pour chaque périphérique avec lequel ils sont compatibles, même si un seul sera *in fine* choisi par l'utilisateur pour l'application qu'il développera.

Une quantité de mémoire importante est requise pour les pilotes de certains périphériques. Les utilisateurs ciblés par ces outils étant des développeurs, nous supposons qu'ils ont les capacités nécessaires et suffisantes pour installer le pilote du périphérique et importer la bibliothèque fournie par le fabricant. Ainsi notre solution ne nécessitera pas de mises à jour importantes liées à ces pilotes.

Avec notre solution, une simple connexion à un serveur spécifique sera suffisante pour charger des données sur les derniers périphériques, les derniers moteurs et les derniers algorithmes d'interactions immersives.

6.1.6 Synthèse

Nous venons de décrire les choix de conception faits pour notre solution logicielle à partir des avantages et limites des outils existants. Elle repose sur l'idée que la réutilisation d'interactions RMV par la communauté peut être facilitée par un outil qui offre plusieurs usages à un même algorithme, graphiquement pour les développeurs non-RMV et textuellement pour les développeurs RMV. Concrètement, notre solution devra permettre de :

- Télécharger et concevoir facilement son algorithme par programmation visuelle à l'aide d'une bibliothèque adaptée aux besoins de l'utilisateur ;
- Proposer une base d'interactions immersives contenant un maximum d'interactions pour un nombre important de périphériques et compatibles avec de nombreux moteurs ;
- Transformer la représentation graphique réalisée en un script totalement rééritable directement dans le moteur de jeu de l'utilisateur. Pour ce faire, la solution est fondée sur le concept de programmation d'interactions génériques partagées;
- Promouvoir la capitalisation du code et le partage des connaissances, impliquant la communauté des développeurs RMV afin de faire face à la quantité importante d'algorithmes à produire. Ce principe permet de traduire le même algorithme d'interaction pour une multitude de moteurs de jeu. Notons toutefois qu'il n'est pas nécessaire qu'un algorithme soit traduit

pour tous les moteurs pour commencer à la partager avec l'ensemble de la communauté.

- Être compatible avec un maximum de moteurs (sans dépendance à un moteur particulier). En effet, le développement d'une nouvelle interface graphique complète, l'échange fréquent de requêtes avec une base de données externe et la possibilité de générer des nouveaux scripts, sont des fonctionnalités implémentables dans une application indépendante. Cependant même si des extensions peuvent être développées pour certains moteurs de jeu, leurs fonctionnalités restent bien plus limitées.

6.2 Caractéristiques de MIREEDGE

En nous fondant sur le concept de modélisation d'interactions en environnement virtuel (Pellens *et al.*, 2007) et sur les solutions d'environnement de développement RMV mentionnées précédemment, (cf. chapitre 4) nous proposons l'outil de développement RMV : *Mixed and virtual Reality DEvelopment tool for Game Engine* (MIREEDGE). Le principe de fonctionnement est illustré sur la Figure 36. L'objectif de cette solution logicielle est de permettre aux développeurs de récupérer un algorithme d'interaction dans une bibliothèque et de le reprogrammer visuellement, pour finalement générer un script correspondant dans le moteur de jeu utilisé par le développeur.

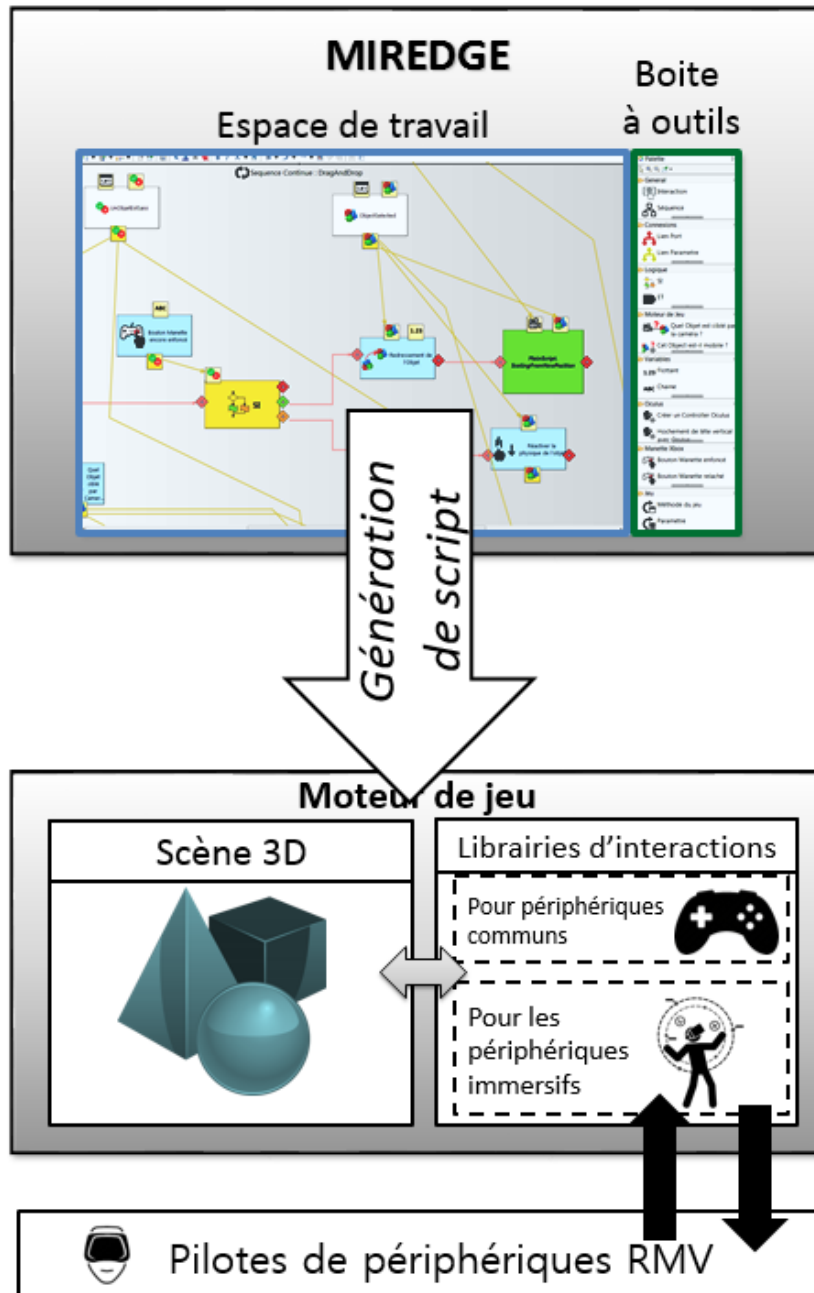


Figure 36 : Programmation visuelle et génération de script, une solution hybride avec MIREEDGE

6.2.1 Programmation Visuelle

MIREEDGE propose une représentation sous la forme d'un organigramme de programme comme illustré dans la Figure 37. Chaque fonction est représentée par une icône. La séquence du programme et la propagation des variables sont effectuées en connectant les blocs par des liens. Un premier type de lien permet de définir l'ordre d'exécution des blocs. Un second type de lien permet le transfert de valeurs depuis le résultat d'un bloc vers l'entrée d'un autre bloc. Des contrôles automatiques sont effectués pour assurer la cohérence entre le type de lien et la nature du point de connexion.

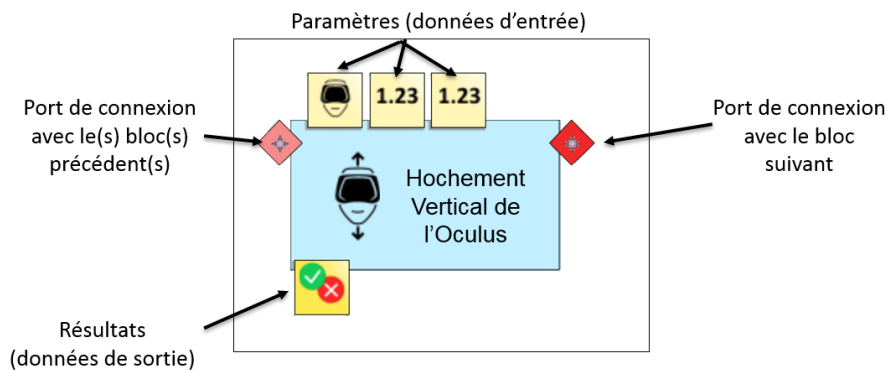


Figure 37 : Représentation d'un bloc MIREEDGE

Il existe 3 catégories de blocs :

- La première catégorie, représentée en bleu, comprend des méthodes simplifiées pour communiquer avec des périphériques RMV. Ces méthodes sont utilisées pour transmettre des ordres d'exécution aux périphériques ou pour collecter des informations sur leurs états ou leurs propriétés. Une majorité de ces blocs est dédiée à des composants d'interactions complexes, tel que le hochement de tête pour l'Oculus illustré en Figure 37.

- La seconde catégorie, représentée en jaune, contient des éléments logiques de programmation. Ceux-ci permettent d'ajouter des conditions ainsi que des répétitions. Cette catégorie est essentielle pour répondre à la complexité d'un algorithme tout en restant simple pour les développeurs non-RMV.
- La troisième catégorie, représentée en vert, permet aux développeurs de se référer à des fonctions de scripts déjà existantes dans leurs projets de moteur de jeu. Le développeur doit donc indiquer à quelle fonction de quel script du projet, ce bloc fait référence. Les paramètres d'entrée et de sortie sont également à indiquer. Il est à noter que l'utilisation de ce type de bloc peut limiter la réutilisation de l'algorithme.

La figure 38 présente la représentation graphique d'un algorithme sous MIREEDGE.

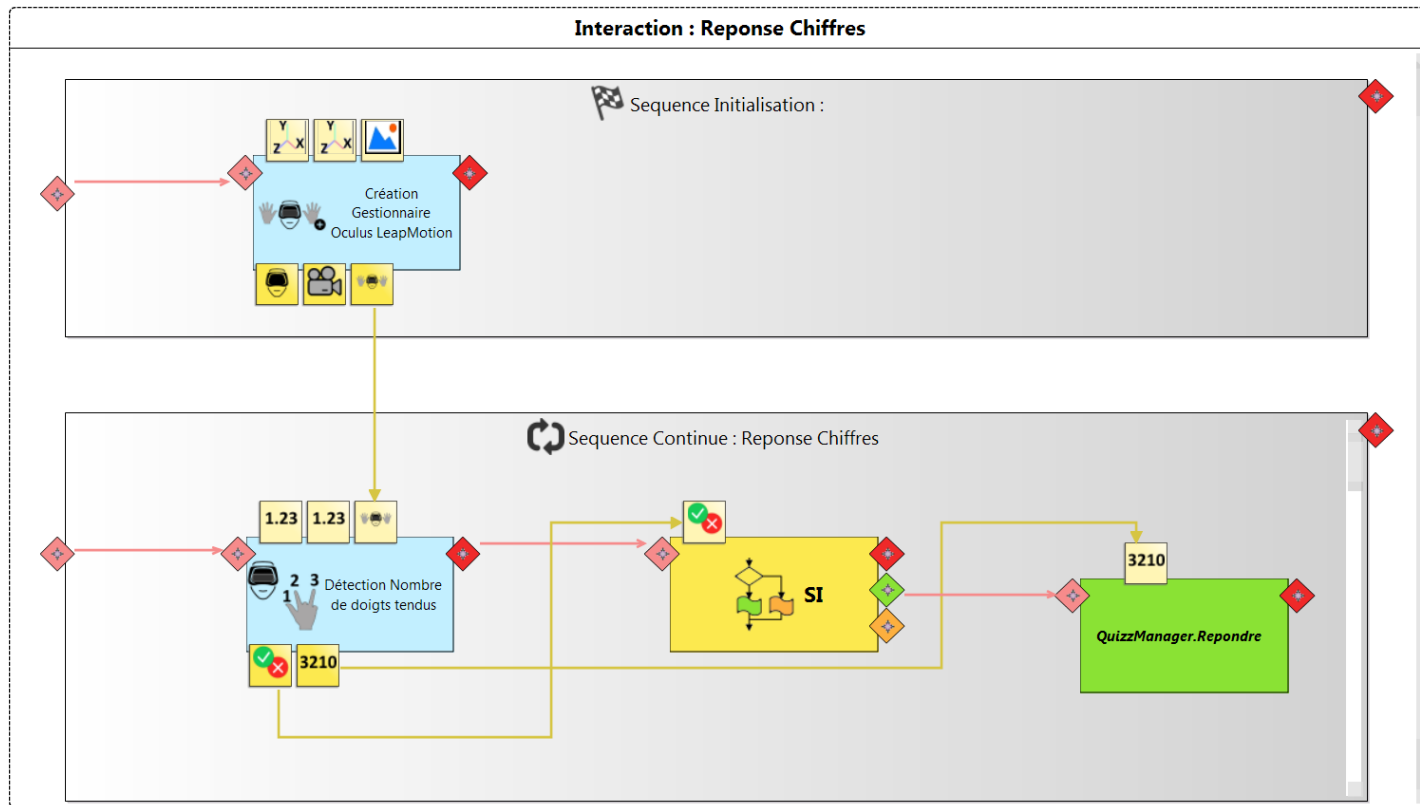


Figure 38 : Exemple d’algorithme sous MIREEDGE

Dans l'exemple ci-dessus, l'algorithme permet d'appeler la fonction *Repondre* de la classe *QuizManager* déjà existante dans le projet sous une condition. Pour appeler cette fonction, le dispositif Oculus Rift et LeapMotion doit détecter des doigts tendus. Cette détection est effectuée dans le bloc bleu et retourne deux informations : un booléen indiquant si le processus de détection est terminée ainsi qu'un entier correspondant au nombre de doigts détectés comme tendus. Le bloc jaune *SI* permet de ne déclencher la fonction *Repondre* que sous la condition que le processus de détection soit terminé. Le bloc vert permet de déclencher la fonction *Repondre* du projet qui possède en paramètre d'entrée, le nombre de doigts détectés comme tendus.

Comme il peut y avoir de nombreux blocs dans la bibliothèque, il peut être nécessaire de les filtrer à partir du contexte d'usage préalablement identifié. Par conséquent, certains blocs sont dédiés à certains types d'actions, tels que le déplacement d'un objet 3D, la sélection d'une zone, la modification de la vue, etc. Nous pouvons également prendre en compte le matériel et le logiciel que l'utilisateur aura à sa disposition et nécessaire pour l'utilisation de certains blocs, à savoir :

- Le périphérique d'entrée : une caméra Kinect, un détecteur LeapMotion, un contrôleur Oculus, etc.
- Le périphérique de sortie : un casque Hololens, des lunettes de RA Epson, un visiocasque Oculus, etc.
- Le moteur de jeu pour lequel le script sera généré. Même si dans l'idéal, un bloc d'interaction se doit d'être compatible avec les principaux moteurs de jeu.

6.2.2 Outil communautaire pour la création de composants d'interactions

Pour qu'une communauté de développeurs adopte une solution logicielle, la transparence doit être absolue (Dabbish *et al.*, 2012). Ainsi, si MIREEDGE est disponible en open source, un plus grand nombre de développeurs souhaiteront contribuer à la base d'algorithmes d'interactions immersives.

Une limite majeure des environnements de développement RMV existants (cf. chapitre 4) est de n'être compatible qu'avec une liste relativement limitée et figée de périphériques. Pour pallier cela, les caractéristiques de MIREEDGE présentées dans la Figure 39 permettent aux développeurs de la communauté RMV de créer eux-mêmes de nouveaux composants d'interactions dès la sortie de nouveaux périphériques. Dans un premier temps, les développeurs choisissent une représentation graphique et indiquent le type et la valeur par défaut de chaque paramètre d'entrée et de sortie. Dans un second temps, les développeurs disposent d'un script contenant des variables de classe portant les noms des paramètres précédemment saisis. Ils se doivent alors de compléter ce script en intégrant le code qu'ils souhaitent partager. Ces deux étapes s'appuient sur le métamodèle décrit dans le chapitre 5. De plus, dans la logique communautaire de MIREEDGE, ces nouveaux composants seront évalués par des pairs afin d'en assurer la pertinence.

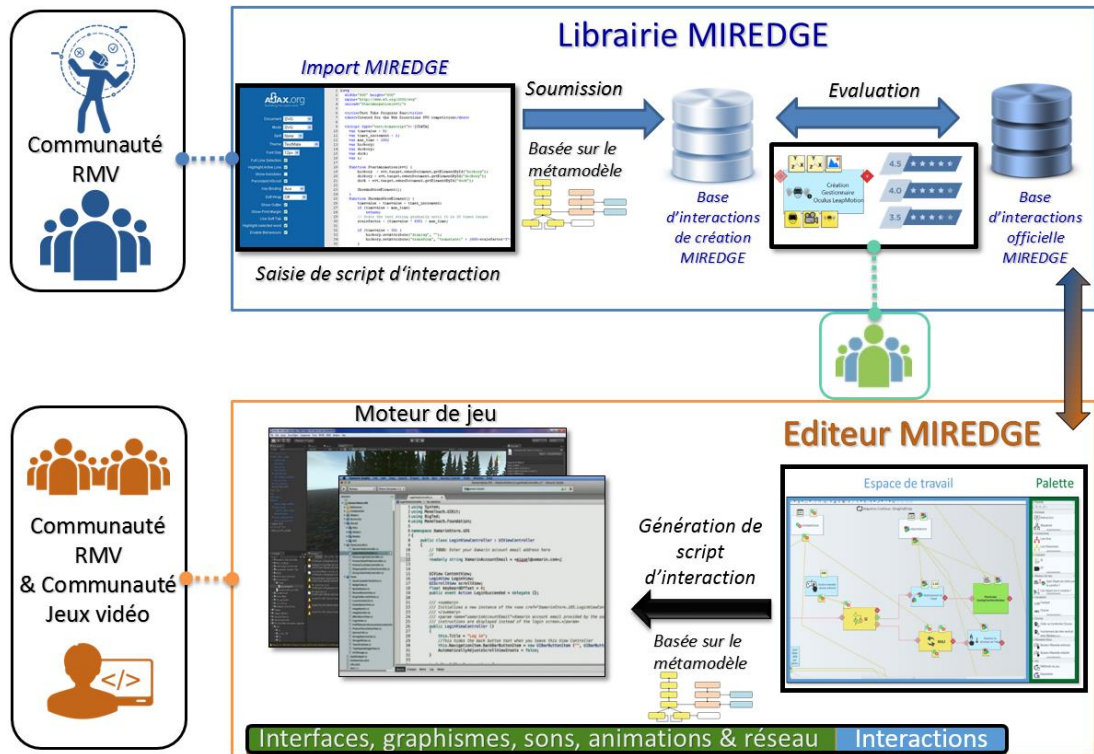


Figure 39 : Processus de développement communautaire d'une interaction immersive sous MIREEDGE

6.2.3 Génération de script

La dernière étape de MIREEDGE repose sur la génération du script correspond à l'algorithme réalisé en programmation visuelle. Ce script est exporté dans la scène courante du projet du moteur de jeu. Ce dernier intègre le nouveau script comme une nouvelle ressource réutilisable. Il est possible de modifier ce script dans le moteur de jeu, comme tout autre code. Cependant, la version actuelle de MIREEDGE ne permet pas de reprendre un script modifié dans l'éditeur visuel.

Les scripts générés contiennent un code documenté et commenté. Cela constitue une valeur pédagogique puisqu'elle donne aux développeurs

non-RMV la possibilité de consulter voire d'interpréter les instructions du code généré.

Pour transformer la structure des entités graphiques en lignes de code, nous avons considéré les algorithmes produits sous MIREEDGE comme des modèles, dérivés du métamodèle présenté au chapitre 5. Dans ces modèles, chaque bloc graphique correspond à différents ensembles de lignes de code. Ces lignes permettent aux utilisateurs de déclarer ou d'utiliser des bibliothèques, des méthodes ou des variables. Ils peuvent être attachés à un ou plusieurs périphériques. Il peut exister des versions du même bloc dans différents langages et pour différents moteurs.

6.2.4 Implémentation de MIREEDGE

Pour implémenter MIREEDGE, deux approches étaient possibles : soit développer l'intégralité des fonctionnalités, soit adapter un outil existant. Or la création d'un environnement de développement spécifique et performant ne peut s'effectuer que durant de longs mois par une équipe d'experts aux compétences variées. De plus, la maintenance de ce type d'outil demande tout autant de temps et de compétences.

Concernant l'adaptation d'outils existants, nous avons constaté que les environnements de programmation visuelle sont peu paramétrables pour intégrer les fonctionnalités de MIREEDGE. À l'inverse, les éditeurs génériques de modèles proposent de nombreuses fonctionnalités mais doivent être adaptés à un usage précis. Le *framework Sirius* s'appuie sur l'environnement de développement *Eclipse* pour proposer un atelier de modélisation graphique (Jäger *et al.*, 2016). Grâce à cela, nous avons défini les règles de modélisation auxquelles le développeur sera contraint pour réaliser son algorithme d'interactions immersives. Ces règles ont été définies par création d'un métamodèle spécifique à MIREEDGE (cf. annexe 1). Une personnalisation de la représentation graphique de

chaque type d'entité (les séquences, les blocs, les paramètres, les ports) a également été définie sous *Sirius*.

Concernant la génération du code, *Sirius* rassemble l'ensemble des informations du modèle réalisé par l'utilisateur dans un seul fichier XML. En se basant sur le métamodèle, une extension d'*Eclipse* nommée *Epsilon* permet de transposer des informations issues d'un modèle dans un fichier texte. Grâce à cela, nous avons pu transposer les blocs utilisés dans le modèle d'interactions en lignes de code de notre base d'interactions. Pour générer le script attendu par le développeur, il a été nécessaire de tenir compte des liens de séquencement et de paramétrage des blocs. De plus, plusieurs blocs identiques pouvant être utilisés dans un même algorithme, un travail a été nécessaire pour éviter les doublons de noms de variables et de fonctions (cf. annexe 2).

6.3 Évaluation de MIREEDGE

Nous présentons dans cette section, une évaluation de MIREEDGE, en comparaison avec Unity3D, environnement de développement connu des deux communautés : développeurs RMV et développeurs non-RMV (Jokela *et al.*, 2003).

6.3.1 Objectif

L'objectif de notre étude est d'évaluer l'efficacité, l'efficacité et la satisfaction relatives à MIREEDGE chez des développeurs RMV et des développeurs non-RMV lors de la réalisation de tâches de développement d'interactions à intégrer à un projet existant. Pour ce faire, nous avons fait le choix de comparer MIREEDGE et Unity3D pour chaque communauté de développeurs. Même si MIREEDGE aurait pu être comparé à d'autres outils tels que *RUIS* ou *ARToolkit*, développer

directement sous Unity3D reste la méthode la plus utilisée par les développeurs de chaque communauté pour effectuer ce type de tâche.

6.3.2 Méthode

6.3.2.1 Participants

Un premier groupe de quinze participants (âgés de 21 à 27 ans, moyenne = 22, écart type = 1,5), nommé le groupe non-RMV, formé pour développer des jeux sérieux mais sans expérience de développement d'interaction RMV, a participé à l'expérimentation. Le deuxième groupe était composé de seize autres participants (âgés de 21 à 27 ans, moyenne=24, écart type=1,8) formés pour développer des applications RMV, sera appelé le groupe RMV.

Ces participants ont été répartis entre deux conditions expérimentales : 1/ Les développeurs utilisant MIREEDGE pour implémenter les interactions demandées (condition avec MIREEDGE) ; 2/ Les développeurs utilisant uniquement Unity3D et les bibliothèques existantes (condition sans MIREEDGE).

Ainsi, quatre groupes expérimentaux ont été impliqués dans l'évaluation : 8 développeurs non-RMV en condition avec MIREEDGE, et 8 développeurs RMV en condition avec MIREEDGE, 7 développeurs non-RMV en condition sans MIREEDGE et 8 développeurs RMV en condition sans MIREEDGE.

6.3.2.2 Matériels et mesures

Chaque participant possédait un PC sur lequel Unity3D v5 était installé, accompagné d'un libre accès à internet. À côté de cela, deux périphériques étaient disponibles : le visiocasque Oculus DK2 et le contrôleur LeapMotion. Cette combinaison offre des interactions très

spécifiques (Petry et Huber, 2015). Les pilotes de ces périphériques sont déjà installés et le SDK inclus dans le projet Unity3D mis à disposition. Les participants de la condition expérimentale sans MIREEDGE utilisaient exclusivement Unity3D, tandis que les participants de la condition expérimentale avec MIREEDGE utilisaient notre solution logicielle. La version de MIREEDGE fournie aux développeurs avait fait l'objet d'une inspection ergonomique et d'améliorations en conséquence. L'inspection ergonomique consiste à revoir l'interface de MIREEDGE afin de vérifier qu'elle réponde à un ensemble de critères ergonomiques (Nogier, 2008), de manière à détecter les aspects positifs et négatifs de l'utilisabilité (Baccino *et al.*, 2005).

Nous avons mesuré l'efficacité et l'efficacé comme indicateurs à partir de traces numériques et vidéos, ainsi que la satisfaction à partir d'un questionnaire (cf. annexe 3). Ce questionnaire était composé de :

- 22 questions inspirées du CSUQ (*Computer Usability Satisfaction Questionnaire*) (Lewis, 1995),
- 2 questions ouvertes sur les aspects positifs et ceux à améliorer dans la version actuelle de MIREEDGE,
- 1 question sur la priorisation de 15 fonctionnalités que nous avons envisagées dans le cadre de la finalisation de MIREEDGE (par ex., générateur de diagramme à partir de scripts).

6.3.2.3 Tâches

Les participants devaient implémenter 3 interactions à partir d'un projet existant pendant une durée fixée (1 heure), en développant les interactions les unes après les autres. Le projet existant proposait une scène où l'utilisateur faisait face à un avatar. L'avatar émettait uniquement des informations textuelles dans une bulle de dialogue. L'avatar prenait l'initiative de poser des questions, offrant 3 possibilités

de réponse à l'utilisateur. Les deux premières interactions permettaient de répondre aux questions de l'avatar. La troisième interaction permettait de changer d'avatar et ainsi d'être confronté à d'autres questions, comme illustré en Figure 40.



Figure 40 : Interface utilisée pour réaliser la tâche 1

La première interaction - correspondant à une tâche ayant un degré de complexité faible - à implémenter permettait de répondre oui ou non à la première question de l'avatar. Cette réponse devait être faite à l'aide d'un hochement de tête vertical ou horizontal détecté par l'utilisation du gyroscope du visiocasque. Cette détection pouvait être paramétrée en fonction de la durée du mouvement ainsi que de l'ampleur minimale du mouvement de la tête de l'utilisateur.

La seconde interaction - correspondant à une tâche ayant un degré de complexité modérée - permettait à l'utilisateur d'indiquer un nombre entre 1 et 4 lui permettant de choisir entre 4 propositions. Pour cela, l'utilisateur devait placer le nombre de doigts tendus correspondant devant la caméra infrarouge LeapMotion. Cette détection devait être paramétrable en définissant la durée de détection des doigts ainsi que l'inclinaison minimale définissant un doigt comme tendu.

La troisième interaction - correspondant à une tâche ayant un degré de complexité élevée - consistait à changer l'avatar intervieweur. Dans la scène prévue, 4 avatars avaient été placés tout autour de l'utilisateur.

Chaque avatar était un intervieweur qui posait des questions sur des thèmes différents. Si l'utilisateur souhaitait changer de questionnaire, il était nécessaire de sélectionner et relancer un nouveau dialogue avec un autre avatar. Pour ce faire, les utilisateurs devaient faire pivoter leur visiocasque pour regarder leur nouvel interlocuteur et le pointer du doigt. Pour les participants en condition « sans MIREEDGE », les développeurs disposaient d'une bibliothèque contenant des fonctions permettant de détecter des mouvements de tête et des doigts. Leur production finale était un script directement écrit en langage C# dans l'éditeur MonoDevelop.

Pour les participants en condition « avec MIREEDGE », les développeurs avaient une palette de blocs correspondant à chacune des fonctions de la bibliothèque à disposition des « sans MIREEDGE ». Dans l'espace de travail MIREEDGE, ils devaient connecter les blocs disponibles pour définir une séquence d'interaction. Enfin, ils pouvaient lancer la procédure de génération de script qui envoyait directement le script correspondant dans le projet Unity.

Tous les participants pouvaient tester leur script à l'aide de l'Oculus DK2 et de la LeapMotion dès qu'ils le souhaitaient, afin de mieux comprendre et apporter des modifications, si nécessaire.

6.3.3 Résultats

6.3.3.1 Efficacité

Notre principal indicateur pour mesurer l'efficacité est le pourcentage de tâches accomplies avec succès par l'utilisateur, synthétisé dans le tableau 4. Les résultats suggèrent qu'aucune des tâches n'était ni trop facile ni trop difficile car elles ont toutes pu être exécutées par au moins une personne et jamais par tous les utilisateurs.

De plus, nous constatons logiquement que le nombre de participants ayant réussi diminue à mesure que les tâches augmentent. Ainsi, certaines conditions ont peu de participants pour la troisième interaction. Pour objectiver ce fait, nous avons réalisé un tableau de contingences croisant les groupes expérimentaux et le degré de complexité des tâches.

Tableau 4 : Nombre de participants par groupes expérimentaux en fonction du degré de complexité de la tâche

Groupes	Aucune Tâche	Tâche simple	Tâche modéré	Tâche complexe	Tout
Groupe RMV /sans MIREEDGE	1	1	3	3	8
Groupe RMV /avec MIREEDGE	4	2	1	1	8
Groupe non-RMV /sans MIREEDGE	2	4	1	0	7
Groupe non-RMV /avec MIREEDGE	2	2	4	0	8
Tout	9	9	9	4	31

Il y avait une association intermédiaire entre le groupe expérimental (niveau d'expertise et avec/sans MIREEDGE) et le nombre de tâches effectuées par les participants (Cramér $v_2 = 0,14$ ¹⁴). Le Taux de Liaison (TDL)¹⁵, mesurant l'association entre les modalités de deux variables

¹⁴ Le V_2 de Cramér estime la force globale du lien entre deux variables (Cramér, 2016). Le V_2 de Cramér se calcule en divisant le ϕ^2 par le ϕ^2 max. Le ϕ^2 max est la plus petite dimension du tableau moins 1. Compris entre 0 et 1, la liaison est considérée forte pour V_2 Cramér $> 0,16$, faible pour V_2 Cramér $< 0,04$, et intermédiaire entre les deux (Wolf et Corroyer, 2004).

¹⁵ Les Taux de Liaison sont des écarts relatifs à l'indépendance. Ils s'obtiennent par la comparaison entre les données observées et celles qui auraient été obtenues si les deux facteurs étudiés étaient indépendants. Il y a attraction lorsque le taux de liaison est

(e.g. le groupe RMV sans MIREEDGE pour la tâche simple accomplie) (Cramér, 2016), révèle une attraction positive notable :

- Les développeurs RMV n'utilisant pas MIREEDGE et réalisant la tâche simple et la tâche modérée (TDL=+ 0,29),
- Les développeurs RMV utilisant MIREEDGE et réalisant aucune tâche (TDL=+0,72),
- Les développeurs non-RMV utilisant MIREEDGE et la tâche simple et la tâche modérée (TDL=+0,72),
- Les développeurs non-RMV n'utilisant pas MIREEDGE et réalisant la tâche simple (TDL=+0,97),
- Les développeurs RMV n'utilisant pas MIREEDGE et réalisant la tâche simple, la tâche modérée et la tâche complexe (TDL=+1,91).

A l'inverse, il y a une répulsion entre :

- Les développeurs non-RMV n'utilisant pas MIREEDGE et réalisant la tâche simple, la tâche modérée et la tâche complexe (TDL=-0,51),
- Les développeurs RMV sans MIREEDGE et réalisant aucune tâche (TDL=-0,57),
- Les développeurs RMV sans MIREEDGE et réalisant la tâche simple (TDL=-0,57),
- Les développeurs RMV utilisant MIREEDGE et réalisant la tâche simple et la tâche modéré (TDL=-0,57),

positif, et répulsion lorsqu'il est négatif. L'attraction est dite remarquable pour $TDL \geq 0,25$.

- Les développeurs non-RMV n'utilisant pas MIREEDGE et réalisant la tâche simple, la tâche modérée et la tâche complexe (TDL=-1,00),
- Les développeurs non-RMV utilisant MIREEDGE et réalisant la tâche simple, la tâche modérée et la tâche complexe (TDL=-1,00).

Les principales attractions sont résumées dans la Figure 41.

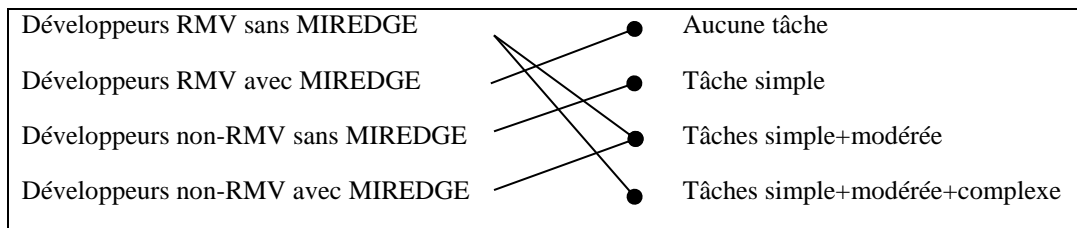


Figure 41 : Les principales attractions entre les groupes et les tâches réalisées par les participants

Ces données permettent de valider l'hypothèse selon laquelle, sans MIREEDGE, la majorité des développeurs RMV possède des compétences pour accomplir les tâches modérée (i.e., 3/8 participants) et complexe (i.e., 3/8 participants) contrairement aux développeurs non-RMV. Un autre résultat intéressant est que la moitié des développeurs non-RMV a réussi à accomplir la tâche ayant un degré de complexité modérée grâce à l'outil MIREEDGE (i.e., 4/8 participants), alors que la majorité des développeurs non-RMV n'utilisant pas MIREEDGE dans l'étude ne parvient à accomplir que la tâche simple (i.e., 4/7 participants). On peut également noter que la moitié des développeurs RMV utilisant MIREEDGE bloque lors de la première tâche et ne parvient pas à réaliser la tâche simple. Une explication possible est la différence entre la nouvelle pratique qu'induit MIREEDGE et les automatismes de ces développeurs RMV. En outre, ils ont acquis, lors de leur formation, des

principes et des méthodes incitant à rédiger des scripts en utilisant directement les bibliothèques des périphériques.

En résumé, les développeurs RMV sont plus efficaces en rédigeant leurs algorithmes d'interactions directement à l'aide de Unity3D et des bibliothèques standards de périphériques, tandis que les développeurs non-RMV semblent plus efficaces à l'aide de MIREEDGE.

Précisons toutefois que cette expérimentation n'a été réalisée que sur une courte durée : une heure ne permettant pas, de déterminer dans quelle mesure les habitudes des développeurs RMV peuvent évoluer.

6.3.3.2 Efficience

Si la réussite d'une tâche est un critère intéressant, le temps de réalisation de la tâche l'est tout autant (i.e., indicateur classiquement utilisé pour mesurer l'efficience). Nous avons donc mesuré le temps nécessaire aux développeurs pour accomplir chaque tâche, dans chaque condition expérimentale. Pour cela, nous avons suffisamment de données à comparer pour les tâches à degré de complexité simple et modéré dans les Tableaux 5 et 6. Les données correspondent au temps de réalisation exprimé en secondes. L'heure de début correspond à l'entrée de la première instruction ou au placement du premier bloc. Le temps de fin correspond à la dernière action nécessaire pour que l'algorithme soit fonctionnel pour cette tâche.

Tableau 5: Temps de réalisation de la tâche simple (en secondes)

Groupe	Moyenne	Minimum	Maximum	Écart type
Développeurs RMV/sans MIREEDGE	744	1987	1327	522
Développeurs RMV/avec MIREEDGE	1059	2576	1818	1073
Développeurs non- RMV/sans MIREEDGE	1712	2629	2195	460
Développeurs non- RMV/avec MIREEDGE	1203	1993	1510	351

Nous constatons que le groupe des développeurs RMV réalise la tâche simple plus rapidement sans MIREEDGE alors que celui des développeurs non-RMV la réalise plus rapidement grâce à MIREEDGE. En d'autres termes, les développeurs non-RMV sont plus efficaces grâce à notre outil alors que les développeurs RMV le sont sans MIREEDGE.

Tableau 6 : Temps de réalisation de la tâche modérée (en secondes)

Groupe	Moyenne	Minimum	Maximum	Ecart type
Développeurs RMV/sans MIREEDGE	285	1431	635	419
Développeurs RMV/avec MIREEDGE	146	467	284	165
Développeurs non- RMV/sans MIREEDGE	376	376	376	
Développeurs non- RMV/avec MIREEDGE	115	266	186	73

Pour la tâche dite modérée, le groupe des développeurs non-RMV est plus efficace à l'aide de MIREEDGE. En ce qui concerne le groupe des développeurs RMV ayant acquis une première maîtrise de l'outil via la première tâche, il devient plus efficace avec MIREEDGE qu'avec

l'utilisation des bibliothèques sous Unity3D. Ainsi, on peut supposer qu'après une phase d'apprentissage, les développeurs RMV trouvent également un intérêt à modéliser sous MIREEDGE.

6.3.4 Satisfaction et autres retours subjectifs

Les résultats du questionnaire sont cohérents avec les résultats précédents sur l'efficacité et l'efficience. En effet, ils montrent que MIREEDGE a été perçu comme étant plus satisfaisant par la communauté des développeurs non-RMV que par celle des développeurs RMV (Tableau 7), probablement pour la raison des habitudes et des schèmes d'utilisation acquis sous Unity3D par les développeurs RMV.

Tableau 7 : Moyenne, écart-type, min et max des scores obtenus aux échelles de Likert à 10 points (22 questions du *CSUQ*)

	Développeurs non-RMV	Développeurs RMV
Moyenne	6,18	3,77
Ecart-type	1,99	2,08
Minimum	0	0
Maximum	10	9

Ce questionnaire nous a également permis d'avoir des retours subjectifs au-delà de la satisfaction. En ce qui concerne les points positifs, la représentation « *flow-based programming* » a été appréciée par les développeurs non-RMV tout comme les développeurs RMV. Les développeurs non-RMV ont également souligné la simplicité de l'interface.

Parmi les retours négatifs, un problème soulevé par les participants (et par l'inspection ergonomique préalable) a été l'absence d'assistance pour identifier les erreurs d'algorithmique. Par exemple, un bloc sans aucune connexion ne sera pas pris en compte lors de la génération de script. Ainsi, aucun blocage ou erreur ne sera provoqué mais cela pourrait engendrer un questionnement voire une confusion dans la représentation de l'algorithme pour l'utilisateur. Ainsi la création d'un module de vérification quant à la cohérence des liens entre blocs sera une priorité dans les évolutions de MIREEDGE.

6.4 Limites et évolutions

Une première limite est relative au fait que nous avons dû fournir un objectif identique et des ressources équivalentes à tous les participants afin que les résultats soient pertinents dans le cadre temporel de l'étude limité à 1 heure. En outre, dans la condition sans MIREEDGE, les bibliothèques nécessaires à la mise en œuvre de l'algorithme ont été fournies directement aux développeurs. Or habituellement, les développeurs recherchent préalablement des exemples dans des projets antérieurs ou des forums. En ce qui concerne la condition avec MIREEDGE, nous avons demandé aux développeurs de programmer l'intégralité d'un algorithme. Or MIREEDGE permet également de partager et donc de réutiliser des algorithmes existants. Aussi, nous pensons que la tâche aurait pu être accomplie beaucoup plus rapidement avec l'accès à une base d'algorithmes d'interactions.

Une deuxième limite est liée à la formation de MIREEDGE qui a été très courte. Il est probable que des expérimentations à plus long terme auraient pu démontrer un gain d'efficacité et d'efficience plus important que dans la présente étude.

Une troisième limite provient du fait que certaines fonctionnalités restent difficiles à implémenter dans notre solution logicielle, malgré le fait que notre ambition était de cumuler un maximum de qualités des outils de développement RMV existants. Rappelons que le concept de réingénierie est un processus utilisé par les développeurs consistant à améliorer le code et les commentaires déjà présents (Perez-Castillo *et al.*, 2011).

Chaque bloc possède deux représentations, l'une visuelle, l'autre scriptée. Le passage de la représentation visuelle à la représentation scriptée requiert une opération de transformation. L'opération inverse, de la représentation scriptée vers la représentation visuelle, n'est pas implémentée. Ainsi si l'utilisateur a modifié son script généré dans le moteur de jeu, la représentation graphique sous MIREEDGE ne sera pas mise à jour. De plus, si l'utilisateur régénère le code depuis MIREEDGE, ses modifications seront écrasées. Toutefois, si l'utilisateur détecte une véritable anomalie dans la génération, il pourra proposer une nouvelle version à associer au bloc correspondant. Cette nouvelle version devra quant à elle être validée par la communauté pour être intégrée dans la base des interactions officielles de MIREEDGE.

Une quatrième limite est l'absence de débogage. Idéalement, il aurait fallu envisager une coloration des blocs utilisés pendant l'exécution du script dans le moteur de jeu. Cette fois encore, le choix de ne pas prioriser l'implémentation de cette fonctionnalité est la conséquence d'une réflexion de conception. Pour cela, nous nous sommes appuyés sur l'idée selon laquelle le débogage est nécessaire dès lors que des erreurs peuvent être commises. Or en utilisant la programmation visuelle, grâce à la logique du métamodèle et aux diverses règles de restriction de l'interface, les erreurs sont grandement limitées, tout comme dans Scratch (Resnick *et al.*, 2009).

Enfin, même si l'expérience a été effectuée sur un nombre très limité d'interactions, le métamodèle doit théoriquement permettre de représenter un large éventail d'interactions. Ainsi il serait intéressant d'étudier une plus grande variété d'algorithmes d'interaction tels que la sélection, la navigation et la manipulation (Jankowski et Hachet, 2013). Nous pouvons également nous demander si la programmation visuelle sera toujours aussi efficace et efficiente pour des interactions plus avancées (Wonner *et al.*, 2013) ou des dispositifs spécifiques tels que l'interface cerveau-machine (Mercier-Ganady *et al.*, 2013).

6.5 Synthèse

MIREEDGE peut répondre aux besoins d'un large spectre d'utilisateurs : des développeurs RMV jusqu'aux développeurs non-RMV. Ces utilisateurs ont un objectif commun, qui est la réalisation rapide d'une application RMV performante. Notre solution vise à répondre à cet objectif commun en permettant à ces deux communautés de collaborer sur le même concept: la réutilisation d'algorithmes d'interactions immersives.

Pour ce faire, le développeur peut programmer graphiquement un algorithme d'interaction et le convertir en un script entièrement éditable dans son projet actuel, grâce à un métamodèle et une base d'interactions immersives.

Pour évaluer MIREEDGE, des expérimentations ont été menées et ont confirmé un gain en efficacité et en efficacité, en particulier pour les développeurs non-RMV.

Toutefois, nous avons conscience que de nouvelles études devront être effectuées à plus grande échelle pour confirmer la compatibilité avec des interactions immersives plus complexes. Une étude connexe mériterait

d'être conduite afin d'identifier les conditions requises pour que les développeurs RMV soient prêts à partager leurs connaissances et celles nécessaires pour que les développeurs non-RMV soient prêts à affronter les nouveaux périphériques à l'aide d'un assistant tel que MIREEDGE.

Bilan et perspectives

Les objectifs de cette thèse ont été de vérifier le potentiel de l'immersion et la persistance sur l'engagement et la motivation des JENP ainsi que faciliter le développement d'interactions immersives. Offrir à l'apprenant un contexte authentique peut l'aider à se préparer à de futures situations. De même, que la collaboration entre 2 communautés de développeurs peut apporter une complémentarité utile pour affronter de plus grands défis. Pour résumer notre démarche, nous rappellerons dans ce chapitre nos contributions tant pratiques que théoriques ainsi que les limites et perspectives identifiées pour poursuivre nos travaux.

7.1 Contributions

7.1.1 Apports pour la recherche en environnements informatiques pour l'apprentissage humain

Dans cette thèse, suite à notre collaboration avec David W. Shaffer dans le projet JEN.lab, nous avons dans un premier temps souhaité étudier le concept peu connu des JEN (jeux épistémiques numériques). Nous avons affiné la définition des JEN en établissant **une liste de caractéristiques sous-jacentes**. Parmi ces jeux, nous avons identifié des principes simples tels que la nécessité qu'un apprenant s'entraîne à affronter des problèmes pluridisciplinaires, complexes, non-déterministes au travers d'une communauté, le tout dans un contexte authentique. Or dans les expérimentations de Shaffer, les apprenants utilisent des interfaces laissant peu de place à ce dernier aspect.

Pour améliorer l'authenticité du scénario, nous avons proposé un **nouveau concept de jeu**, le JENP (JEN pervasif). Ce dernier a vocation à étendre le réalisme du JEN d'un point de vue spatial, social et temporel, en grande partie en s'appuyant sur des technologies adaptées. En ayant un intérêt pédagogique au travers d'un univers persistant, interactif et immersif, ce type de jeu s'inscrit **dans les EVAH (Environnement Virtuel pour l'Apprentissage Humain)**.

Afin de démontrer son apport, nous avons mené des **expérimentations** en milieu écologique, portant sur **l'engagement** de l'apprenant. Pour cela, nous avons comparé les comportements des apprenants utilisant deux versions d'un même jeu, un JEN et JENP. Nous avons ainsi pu montrer que la version JENP conduit à un meilleur engagement de l'apprenant. La motivation a également été évaluée mais, bien que présente dans les deux cas, nous n'avons pas mesuré de différence significative entre les deux groupes.

Pour que le JENP soit accepté par ses futurs utilisateurs, il a été nécessaire que l'enseignant s'implique pleinement dans le processus de conception. Les enseignants ont pu exprimer leurs préférences technologiques sans devoir en maîtriser les spécificités techniques. Des systèmes immersifs ou persistants ont ainsi été recommandés pour que le JENP soit pleinement adapté à l'environnement dans lequel il sera déployé.

7.1.2 Apports pour la recherche en réalité mixte et virtuelle

Il est important de noter que cette thèse, nous a offert l'opportunité de **développer un véritable prototype JENP**, permettant d'être directement confronté aux problèmes techniques que cela implique. Face

à cela, nous avons constaté que la prise en charge de la persistance par la mise en place de serveurs ne faisait appel qu'à des compétences relativement courantes. Cependant, l'implémentation de la dimension immersive requiert des compétences plus rares sur les technologies innovantes. Ainsi nous avons supposé qu'un développeur de jeux sérieux est en capacité de développer l'ensemble du JENP à l'exception de ses **interactions immersives**. Les développeurs de jeux sérieux formant une communauté importante, leur permettre de développer des JENP, de façon autonome, deviendrait un avantage fort pour que la diffusion de ce type de jeu devienne envisageable dans de nombreux contextes d'enseignement. Pour atteindre cet objectif, nous avons émis l'hypothèse que des échanges efficaces au sein d'une plus large communauté pourraient être une solution pour ces développeurs non-RMV. De plus, notre **analyse des outils existants de développement RMV** permettant d'assister les développeurs RMV ou les développeurs non-RMV, nous a menée à la conclusion qu'aucun ne permet de cumuler **partage et réutilisabilité**.

Pour répondre à ce besoin, nous avons proposé **un métamodèle** pour créer et réutiliser rapidement des modèles d'interactions immersives.

A partir de ce dernier, nous avons conçu **un nouvel outil de développement d'interaction commun à différents moteurs de jeu**, nommé **MIREEDGE**. L'objectif de cet outil est de permettre aux développeurs RMV de capitaliser leurs réalisations et aux développeurs non-RMV de bénéficier d'une bibliothèque d'interactions immersives toujours plus grande. Afin d'être accepté rapidement par les développeurs, MIREEDGE propose une comptabilité avec les moteurs de jeu existants, le tout dans une approche *open-source*. La seconde caractéristique importante de MIREEDGE est sa capacité à **transformer un modèle d'interaction** (réalisable rapidement par de la

programmation visuelle) **en un script** directement exploitable et modifiable dans un développement déjà existant dans un moteur de jeu. Nos **expérimentations de MIREEDGE** ont permis de comparer l'efficacité, l'efficacité et la satisfaction du développement de plusieurs tâches avec et sans l'outil pour des groupes de développeurs RMV et des groupes de développeurs non-RMV. Nos conclusions permettent de confirmer que, pour les développeurs non-RMV, MIREEDGE leur permet **d'implémenter des interactions immersives** qu'ils parviennent rarement à réaliser avec des outils courants. Quant aux développeurs RMV, dès lors qu'ils acceptent un raisonnement de modélisation bien différent de leur habitude, MIREEDGE peut leur permettre de **réaliser une tâche plus rapidement**.

Pour MIREEDGE, le principe se résume donc à ne plus réinventer la roue et à rendre accessible à tous les interactions immersives. A l'avenir, les développeurs RMV pourraient ainsi se consacrer à des interactions immersives plus avancées et les développeurs non-RMV pourraient accéder aux interactions utilisées couramment par les développeurs RMV. De plus, en étant utilisé par un plus grand nombre, les algorithmes d'interactions les plus courants pourraient être largement testés et ainsi fiabilisés.

7.2 Limites et Améliorations

7.2.1 JENP

7.2.1.1 Scénarisation

Une piste de recherche mène également à offrir la possibilité à l'enseignant de modifier en temps réel, la difficulté des activités du

JENP. Le JENP offre pour cela de nombreuses opportunités mais un travail préalable se doit d'être effectué sur les conséquences de chaque modification.

7.2.1.2 Co-conception

Concernant la co-conception, les enseignants ont parfois trouvé des difficultés à choisir leurs *jen cards*. Une potentielle solution serait de développer une application tablette identifiant les cartes par QR code. Cela pourrait permettre aux enseignants de visualiser des vidéos d'exemple afin de mieux les assister dans leurs choix.

Le guide de co-conception, quant à lui, n'a pas été validé dans l'usage et s'est limité à une version textuelle. Ce travail a déjà été repris pour mettre en place un outil en ligne assistant le concepteur pas à pas dans sa démarche de conception du JENP. Même si l'aspect tangible des cartes est favorable à la créativité, l'outil numérique facilite la diffusion et l'exploitation de la production.

7.2.1.1 Extensions

Comme l'aspect pervasif a bien été défini comme des extensions spatiale, temporelle et sociale, une plateforme de gestion de jeu en ligne vient d'être finalisée par notre équipe de recherche. Cette dernière peut également jouer un rôle tout aussi important que les périphériques d'immersion et les serveurs en ligne. Il est donc possible de comparer plus directement les différences entre JEN et JENP dans leurs globalités. Lors de nos dernières expérimentations, des contraintes techniques ont nécessité que les élèves n'accèdent à des interfaces 3D qu'en salle de classe. Avec les avancées d'HTML5, il devient possible que la plateforme puisse proposer un accès en tout lieu et à tout heure

(Pakkanen *et al.*, 2017). Cela repose encore de nouvelles questions d'organisation et de pédagogie aux enseignants.

7.2.2 MIREEDGE

7.2.2.1 Miredge Creator

L'architecture du processus complet de MIREEDGE démarre sur l'intégration d'algorithmes dans une base de données. Sur la logique du métamodèle, cette base est en cours de développement et un premier prototype a permis de démontrer qu'il est possible pour un développeur RMV de mettre à disposition un algorithme et de l'associer à une représentation graphique. Pour cela, une application en ligne permet d'analyser automatiquement un langage connu et de découper logiquement l'algorithme afin de le rendre réutilisable. Le développeur doit pour cela associer des variables de son algorithme aux paramètres qu'ils souhaitent associer à son bloc. Cela a été mis en place pour les langages Javascript et C# à destination de Unity3D.

Bien qu'un premier prototype de MIREEDGE Creator ait été développé, il reste encore à le soumettre à la communauté et à en expérimenter son utilisation.

7.2.2.2 Indépendance MIREEDGE

Les développeurs souhaitant déboguer en mode *pas à pas* dans leur moteur de jeu comprendraient mieux le code si ce même *pas à pas* était représenté en temps réel sur les blocs de l'interface MIREEDGE. Cela ne semble pas impossible mais représente un défi de communication entre MIREEDGE et les éditeurs de script des moteurs de jeu.

7.2.2.3 Interface de MIREEDGE

La prise en main de l'outil MIREEDGE a été également identifiée comme un point clé de son acceptation par les développeurs. Une amélioration de sa présentation et l'ajout d'un tutoriel pourraient être bien utiles, selon les personnes ayant participé aux expérimentations.

Concernant la programmation graphique, de nombreuses améliorations de l'interface de MIREEDGE pourraient être envisagées. En effet, par ce type de représentation, les algorithmes au contenu conséquent peuvent manquer de lisibilité. Les principales pistes envisagées sont le regroupement des liens entre blocs ainsi que le concept d'encapsulation. Par ce dernier il deviendrait possible de regrouper plusieurs blocs sous l'apparence d'un seul qui cumulerait les paramètres d'entrée et de sortie de ceux qu'il contient. Ce regroupement en unité cohérente d'un point de vue de l'interaction pourrait aussi être une force pour la réutilisabilité au sein d'une communauté de développeurs.

Pour devenir toujours plus attractif, il a également été envisagé que notre outil n'utilise que la technologie web. MIREEDGE fonctionne actuellement sous Eclipse pour bénéficier de bibliothèques adaptées à la modélisation et à la génération de scripts. Or Eclipse requiert une quantité de mémoire importante pour représenter une interface simple. Il restera donc à réfléchir sur une alternative moins coûteuse en termes de ressources.

7.2.2.4 Partage d'informations

Pour une utilisation large de MIREEDGE, l'activité des communautés de développeurs est centrale. Pour garantir cela, il est nécessaire d'établir une véritable réflexion sur leurs règles d'échanges. Ainsi lorsqu'un script et sa représentation sont proposés par un développeur, il ne doit pas être

intégré directement dans la bibliothèque de tous les autres. Il semble nécessaire de mettre en place des modérateurs qui jugeront si cette proposition peut être considérée comme valide. Pour cela, des critères doivent être établis pour savoir comment définir si un bloc est acceptable pour la communauté. D'autres règles peuvent être nécessaires pour savoir comment réévaluer un bloc qui ne fait qu'évoluer.

MIREEDGE a été expérimenté avec une bibliothèque de blocs très limitée et sans repartir d'un modèle d'algorithme d'interaction existant. Cependant, il a été conçu pour avoir une bibliothèque de blocs et de modèles d'interaction toujours plus grande. Ainsi, même si l'on estime que l'utilisateur peut limiter sa liste de blocs en indiquant les périphériques matériels qu'il utilisera, il peut être nécessaire d'envisager des mots-clés lorsqu'il recherchera à réutiliser un modèle.

7.2.2.5 Nouvelles évaluations de MIREEDGE

L'évaluation de MIREEDGE s'étant limitée à une expérimentation sur une tâche d'une heure seulement, il pourrait être pertinent d'évaluer son utilité sur une durée plus importante. Cela permettrait de vérifier si l'outil peut être accepté définitivement ou non par les développeurs non-RMV et les développeurs RMV. Cette démarche permettrait d'en connaître les forces et les faiblesses au quotidien ainsi que les usages émergents.

Enfin, les interactions immersives de la base de données MIREEDGE pourraient être accompagnées de listes d'usages, de contextes, de limites techniques, ainsi que de retours d'expérience utilisateur pour chacune d'elles. De nouvelles classifications pourraient ainsi être proposées en reliant plus directement chaque interaction immersive à ses apports pédagogiques.

7.3 Perspectives de recherche

7.3.1 Expérimentations des JENP

Concernant les prototypes ayant permis nos expérimentations, ils continuent à être utilisés au lycée Aristide Briand de Saint Nazaire, avec de plus en plus d'enseignants et d'élèves chaque année (150 élèves en 2017).

À la découverte des caractéristiques que nous avons identifiées pour les JEN(P), Shaffer a évoqué la nécessité de mener de nouvelles expérimentations permettant d'identifier quelle forme d'influence possède chacune d'elle sur l'apprentissage. Ainsi si le JEN possède une pluridisciplinarité forte ou traite de problèmes peu complexes, il peut être intéressant d'en connaître les conséquences sur le formateur et l'apprenant.

D'autres expérimentations pourraient également être menées pour déterminer l'impact des différents types d'interactions immersives utilisées par l'apprenant et l'enseignant dans le cadre des JENP.

7.3.2 Métamodèle de MIREEDGE

Le métamodèle de MIREEDGE a été conçu pour répondre à un besoin bien précis, celui de modéliser uniquement des interactions immersives. Cependant, il pourrait être utile pour modéliser davantage de comportements d'objets en environnement virtuel. Il peut être envisageable de chercher à élargir le métamodèle dès lors que cela ne nuit pas à son objectif premier.

7.3.3 Réingénierie par MIREEDGE

MIREEDGE a actuellement la capacité de transformer un modèle graphique d'algorithme d'interaction immersive en script. Cependant, transformer un script existant en représentation visuelle n'est pas encore possible sur MIREEDGE. Pour y parvenir, il doit d'abord être prouvé que cette transformation est théoriquement possible en se basant sur ce même métamodèle ou le faire évoluer en conséquence si nécessaire. D'un point de vue pratique, même si *Epsilon* a permis la transformation inverse, il n'est pas en capacité d'effectuer ce type de transformation.

Cependant si une solution est développée et que le contenu de la base d'interactions est suffisant, on pourrait alors envisager que les développeurs reprennent de nombreux scripts d'interactions existants pour en obtenir des représentations visuelles. Ces interactions pourraient alors être réutilisées plus facilement pour mettre à jour d'anciens projets et en créer de nouveaux.

Références

- Abidi, M.-A., Lyonnet, B., Chevaillier, P., Toscano, R., 2016. Contribution of Virtual Reality for Lines Production's Simulation in a Lean Manufacturing Environment. *Int. J. Comput. Theory Eng.* 8, 182–189.
- Alvarez, J., 2007. Du jeu vidéo au serious game : approches culturelle, pragmatique et formelle (Thèse de doctorat). Toulouse 2, Toulouse 2.
- Ardito, C., Lanzilotti, R., Raptis, D., Sintoris, C., Yiannoutsou, N., Avouris, N., Costabile, M.F., 2011. Designing Pervasive Games for Learning, in: Marcus, A. (Ed.), *Design, User Experience, and Usability. Theory, Methods, Tools and Practice, Lecture Notes in Computer Science*. Springer Berlin Heidelberg, pp. 99–108.
- Baccino, T., Bellino, C., Colombi, T., 2005. Mesure de l'utilisabilité des Interfaces. *Hermès Sci. - Lavoisier* 1–250.
- Bell, M.W., 2008. Toward a definition of "virtual worlds." *J. Virtual Worlds Res.* 1.
- Bentrad, S., Meslati, D., others, 2011. Visual Programming and Program Visualization—Towards an Ideal Visual Software Engineering System. *IJIT-ACEEE Int. J. Inf. Technol.* 1, 56–62.
- Biernacki, D., Colaço, J.-L., Hamon, G., Pouzet, M., 2008. Clock-directed modular code generation for synchronous data-flow languages, in: *ACM Sigplan Notices*. ACM, pp. 121–130.
- Blach, R., Landauer, J., Rösch, A., Simon, A., 1998. A highly flexible virtual reality system. *Future Gener. Comput. Syst.* 14, 167–178.
- Bonsignore, E., Hansen, D., Kraus, K., Ruppel, M., 2012. Alternate Reality Games as Platforms for Practicing 21st-Century Literacies. *Int. J. Learn. Media* 4, 25–54.
- Boots, N.K., Strobel, J., 2014. Equipping the Designers of the Future: Best Practices of Epistemic Video Game Design. *Games Cult.*
- Bouvier, P., Sehaba, K., Lavoué, E., 2014. A trace-based approach to identifying users' engagement and qualifying their engaged-behaviours in interactive systems: application to a social game. *User Model. User-Adapt. Interact.* 24, 413–451.
- Bouville, R., Gouranton, V., Boggini, T., Nouviale, F., Arnaldi, B., 2015. # FIVE: High-level components for developing collaborative and interactive virtual environments, in: *Software Engineering and Architectures for Realtime Interactive Systems (SEARIS), 2015 IEEE 8th Workshop On*. IEEE, pp. 33–40.

- Bowman, D.A., 1999. Interaction techniques for common tasks in immersive virtual environments. Georgia Institute of Technology.
- Brugali, D., Scandurra, P., 2009. Component-based robotic engineering (Part I) [Tutorial]. *IEEE Robot. Autom. Mag.* 16, 84–96.
- Carvalho, M.B., Bellotti, F., Berta, R., De Gloria, A., Sedano, C.I., Hauge, J.B., Hu, J., Rauterberg, M., 2015. An activity theory-based model for serious games analysis and conceptual design. *Comput. Educ.* 87, 166–181.
- Chalon, R., 2004. Réalité mixte et travail collaboratif: IRVO, un modèle de l'interaction homme-machine. Ecole Centrale de Lyon.
- Chevallier, P., Querrec, R., Septseault, C., 2009. VEHA, un métamodèle d'environnement virtuel informé et structuré. *Tech. Sci. Inform.* 28, 715–740.
- Chevallier, P., Trinh, T.-H., Barange, M., De Loor, P., Devillers, F., Soler, J., Querrec, R., 2012. Semantic modeling of Virtual Environments using MASCARET. *IEEE*, pp. 1–8.
- Choquet, C., Iksal, S., 2007. Modeling Tracks for the Model Driven Re-engineering of a TEL System. *J. Interact. Learn. Res.* 18, 161.
- Collins, A., Ferguson, W., 1993. Epistemic forms and Epistemic Games: Structures and Strategies to Guide Inquiry. *Educ. Psychol.* 28, 25–42.
- Cook, S.W., Mitchell, Z., Goldin-Meadow, S., 2008. Gesturing makes learning last. *Cognition* 106, 1047–1058.
- Cramér, H., 2016. *Mathematical Methods of Statistics (PMS-9)*. Princeton University Press.
- Dabbish, L., Stuart, C., Tsay, J., Herbsleb, J., 2012. Social coding in GitHub: transparency and collaboration in an open software repository, in: *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*. ACM, pp. 1277–1286.
- David, B., Yin, C., Chalon, R., 2010. Contextual Mobile Learning Strongly Related to Industrial Activities: Principles and Case Study. *ArXiv10010642 Cs*.
- Davis, S., Nesbitt, K., Nalivaiko, E., 2015. Comparing the onset of cybersickness using the Oculus Rift and two virtual roller coasters, in: *Proceedings of the 11th Australasian Conference on Interactive Entertainment (IE 2015)*. p. 30.
- Delomier, F., 2013. *Jeux Pédagogiques Collaboratifs Situés: Conception et mise en œuvre dirigées par les modèles*. Ecole Centrale de Lyon.
- Deng, Y., Antle, A.N., Neustaedter, C., 2014. Tango cards: a card-based design tool for informing the design of tangible learning games. *ACM Press*, pp. 695–704.

- Donovan, M.S., Bransford, J.D., Pellegrino, J.W., 1999. How people learn. Retrieved March 8, 2006.
- Eberly, D.H., 2006. 3D game engine design: a practical approach to real-time computer graphics. CRC Press.
- Egenfeldt-Nielsen, S., 2006. Overview of research on the educational use of video games. *Digit. Kompet.* 1, 184–213.
- Elvezio, C., Sukan, M., Feiner, S., 2016. A framework to facilitate reusable, modular widget design for real-time interactive systems, in: *Software Engineering and Architectures for Realtime Interactive Systems (SEARIS), 2016 IEEE 9th Workshop On. IEEE*, pp. 1–7.
- Fenn, J., LeHong, H., 2011. *Hype Cycle for Emerging Technologies, 2011*.
- Figueroa, P., Bischof, W.F., Boulanger, P., Hoover, H.J., Taylor, R., 2008. Intml: A dataflow oriented development system for virtual reality applications. *Presence Teleoperators Virtual Environ.* 17, 492–511.
- Fischbach, M., Wiebusch, D., Latoschik, M.E., 2017. Semantic Entity-Component State Management Techniques to Enhance Software Quality for Multimodal VR-Systems. *IEEE Trans. Vis. Comput. Graph.* 23, 1342–1351.
- Fuchs, P., Moreau, G., 2003. *Le traité de la réalité virtuelle, Le traité de la réalité virtuelle*. Ecole des Mines de Paris, Paris.
- Fuchs, P., Moreau, G., Berthoz, A., 2006a. *Le traité de la réalité virtuelle volume 1 : L'Homme et l'environnement virtuel, Mathématique et informatique*. Presse des Mines.
- Fuchs, P., Moreau, G., Burkhardt, J.-M., 2006b. *Le traité de la réalité virtuelle Volume 2 - L'interfaçage : l'immersion et l'interaction en environnement virtuel, Mathématique et informatique*. Presse des Mines.
- Funke, J., 2010. Complex problem solving: a case for complex cognition? *Cogn. Process.* 11, 133–142.
- George, S., Michel, C., Serna, A., Bisognin, L., 2014. Évaluation de l'impact d'un jeu sérieux en réalité mixte 21.
- González, M.A., Santos, B.S.N., Vargas, A.R., Martín-Gutiérrez, J., Orihuela, A.R., 2013. Virtual Worlds. Opportunities and Challenges in the 21st Century. *Procedia Comput. Sci.* 25, 330–337.
- Guay, F., Vallerand, R.J., Blanchard, C., 2000. On the Assessment of Situational Intrinsic and Extrinsic Motivation: The Situational Motivation Scale (SIMS). *Motiv. Emot.* 24, 175–213.
- Hand, C., 1997. A survey of 3D interaction techniques, in: *Computer Graphics Forum*. Wiley Online Library, pp. 269–281.

- Hatfield, D., Shaffer, D.W., 2006. Press play: Designing an epistemic game engine for journalism, in: Proceedings of the 7th International Conference on Learning Sciences. International Society of the Learning Sciences, pp. 236–242.
- Heyselaar, E., Hagoort, P., Segaert, K., 2017. In dialogue with an avatar, language behavior is identical to dialogue with a human partner. *Behav. Res. Methods* 49, 46–60.
- Hilfert, T., König, M., 2016. Low-cost virtual reality environment for engineering and construction. *Vis. Eng.* 4, 2.
- Hinske, S., Lampe, M., Magerkurth, C., Röcker, C., 2007. Classifying pervasive games: on pervasive computing and mixed reality. *Concepts Technol. Pervasive Games- Read. Pervasive Gaming Res.* 1, 20.
- Iksal, S., 2011. A declarative and operationalized language for learning systems Analysis: UTL. *Inf. Soc. -Soc. 2011 Int. Conf. On* 62–66.
- Jäger, S., Maschotta, R., Jungebloud, T., Wichmann, A., Zimmermann, A., 2016. Creation of domain-specific languages for executable system models with the Eclipse Modeling Project, in: Systems Conference (SysCon), 2016 Annual IEEE. IEEE, pp. 1–8.
- Jankowski, J., Hachet, M., 2015. Advances in interaction with 3D environments, in: Computer Graphics Forum. Wiley Online Library, pp. 152–190.
- Jankowski, J., Hachet, M., 2013. A Survey of Interaction Techniques for Interactive 3D Environments. Presented at the Eurographics 2013 - STAR.
- Jenkins, H., Purushotma, R., Robison, A.J., Weigel, M., Clinton, K., 2009. Confronting the challenges of participatory culture: Media education for the 21st century. *MacArthur Found. Publ.* 1, 1–59.
- Jokela, T., Iivari, N., Matero, J., Karukka, M., 2003. The standard of user-centered design and the standard definition of usability: analyzing ISO 13407 against ISO 9241-11, in: Proceedings of the Latin American Conference on Human-Computer Interaction. ACM, pp. 53–60.
- Juul, J., 2010. *A Casual Revolution: Reinventing Video Games and Their Players*. MIT Press.
- Kebritchi, M., Hirumi, A., 2008. Examining the pedagogical foundations of modern educational computer games. *Comput. Educ.* 51, 1729–1743.
- Kim, J., Lee, E., Thomas, T., Dombrowski, C., 2009. Storytelling in new media: The case of alternate reality games, 2001–2009. *First Monday* 14.

- Knott, T., Weyers, B., Hentschel, B., Kuhlen, T., 2014. Data-flow oriented software framework for the development of haptic-enabled physics simulations, in: *Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)*, 2014 IEEE 7th Workshop On. IEEE, pp. 65–72.
- Koenig, S., Ardanza, A., Cortes, C., De Mauro, A., Lange, B., 2014. Introduction to Low-Cost Motion-Tracking for Virtual Rehabilitation, in: Pons, J.L., Torricelli, D. (Eds.), *Emerging Therapies in Neurorehabilitation*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 287–303.
- Kreylos, O., 2008. Environment-Independent VR Development, in: *Advances in Visual Computing*. Presented at the International Symposium on Visual Computing, Springer, Berlin, Heidelberg, pp. 901–912.
- Lewis, J.R., 1995. IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use. *Int J Hum-Comput Interact* 7, 57–78.
- Loiseau, M., Lavoué, E., Marty, J., George, S., 2013. Raising awareness on Archaeology: A Multiplayer Game-Based Approach with Mixed Reality. Presented at the Proceedings of the 7th European Conference on Games Based Learning, pp. 336–343.
- Loup, G., George, S., Serna, A., 2015. Fondements et caractérisation des jeux épistémiques numériques pervasifs, in: 7ème Conférence Sur Les Environnements Informatiques Pour l'Apprentissage Humain. Agadir, Morocco.
- Loup, G., Serna, A., Iksal, S., George, S., 2016. Immersion and Persistence: Improving Learners' Engagement in Authentic Learning Situations, in: *ECTEL 2016*. Presented at the ECTEL 2016, Springer, Cham, pp. 410–415.
- Loup-Escande, É., Burkhardt, J.-M., Richir, S., 2013. Anticiper et évaluer l'utilité dans la conception ergonomique des technologies émergentes : une revue, *Anticipating and evaluating the usefulness of emerging technologies in ergonomic design: a review of usefulness in design*. *Trav. Hum.* 76, 27–55.
- Lourdeaux, D., 2001. *Réalité virtuelle et formation : conception d'environnements virtuels pédagogiques (phdthesis)*. École Nationale Supérieure des Mines de Paris.
- Marfisi-Schottman, I., George, S., Tarpin-Bernard, F., 2010. Tools and methods for efficiently designing serious games, in: *Proceedings of the 4th European Conference on Games Based Learning ECGBL*. pp. 226–234.

- Marne, B., Wisdom, J., Huynh-Kim-Bang, B., Labat, J.-M., 2012. The Six Facets of Serious Game Design: A Methodology Enhanced by Our Design Pattern Library, in: 21st Century Learning for 21st Century Skills, Lecture Notes in Computer Science. Presented at the European Conference on Technology Enhanced Learning, Springer, Berlin, Heidelberg, pp. 208–221.
- Mercier-Ganady, J., Loup-Escande, É., George, L., Busson, C., Marchal, M., Lécuyer, A., 2013. Can We Use a Brain-computer Interface and Manipulate a Mouse at the Same Time?, in: Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology. ACM, New York, NY, USA, pp. 69–72.
- Mestre, D., Fuchs, P., Berthoz, A., Vercher, J.L., 2006. Immersion et présence. *Traité Réal. Virtuelle* Paris Ecole Mines Paris 309–38.
- Milgram, P., Kishino, F., 1994. A taxonomy of mixed reality visual displays. *IEICE Trans. Inf. Syst.* 77, 1321–1329.
- Mohagheghi, P., Conradi, R., 2007. Quality, productivity and economic benefits of software reuse: a review of industrial studies. *Empir. Softw. Eng.* 12, 471–516.
- Montola, M., 2005. Exploring the edge of the magic circle: Defining pervasive games, in: Proceedings of DAC. p. 103.
- Montola, M., Stenros, J., Waern, A., 2009. *Pervasive Games: Theory and Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Nogier, J.-F., 2008. *Ergonomie du logiciel et design web - 4e éd.: Le manuel des interfaces utilisateur*. Dunod.
- Orliac, C., Michel, C., George, S., 2012. An Authoring Tool to Assist the Design of Mixed Reality Learning Games, in: Ravenscroft, A., Lindstaedt, S., Kloos, C.D., Hernández-Leo, D. (Eds.), 21st Century Learning for 21st Century Skills, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 441–446.
- Ouramdane, N., Otmane, S., Mallem, M., 2009. Interaction 3D en Réalité Virtuelle-Etat de l'art. *Tech. Sci. Inform.* 28, 1017–1049.
- Pakkanen, T., Hakulinen, J., Jokela, T., Rakkolainen, I., Kangas, J., Piippo, P., Raisamo, R., Salmimaa, M., 2017. Interaction with WebVR 360 video player: Comparing three interaction paradigms, in: 2017 IEEE Virtual Reality (VR). Presented at the 2017 IEEE Virtual Reality (VR), pp. 279–280.
- Papargyris, A., Poulymenakou, A., 2005. Learning to fly in persistent digital worlds: the case of Massively Multiplayer Online Role Playing Games. *ACM SIGGROUP Bull.* 25, 41–49.

- Pellens, B., De Troyer, O., Kleiner mann, F., Bille, W., 2007. Conceptual modelling of behaviour in a virtual environment. *Int. J. Prod. Dev.* 4, 626–645.
- Perez-Castillo, R., Guzman, I.G.R. d, Piattini, M., Ebert, C., 2011. Reengineering Technologies. *IEEE Softw.* 28, 13–17.
- Perkins, D.N., 1997. Epistemic games. *Int. J. Educ. Res.* 27, 49–61.
- Petry, B., Huber, J., 2015. Towards effective interaction with omnidirectional videos using immersive virtual reality headsets. *ACM Press*, pp. 217–218.
- Plonka, L., Sharp, H., van der Linden, J., Dittrich, Y., 2015. Knowledge transfer in pair programming: An in-depth analysis. *Int. J. Hum.-Comput. Stud.* 73, 66–78.
- Posada, J., Toro, C., Barandiaran, I., Oyarzun, D., Stricker, D., de Amicis, R., Pinto, E.B., Eisert, P., Döllner, J., Vallarino, I., 2015. Visual computing as a key enabling technology for industrie 4.0 and industrial internet. *IEEE Comput. Graph. Appl.* 35, 26–40.
- Radkowski, R., Hilus, M., 2011. Unified Modeling Language to Enhance the Specification of Discrete Event Systems for Virtual Reality Applications 31–40.
- Raybourn, E.M., 2014. A new paradigm for serious games: Transmedia learning for more effective training and education. *J. Comput. Sci.* 5, 471–481.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., Kafai, Y., 2009. Scratch: Programming for All. *Commun ACM* 52, 60–67.
- Richards, D., Taylor, M., 2015. A Comparison of learning gains when using a 2D simulation tool versus a 3D virtual world: An experiment to find the right representation involving the Marginal Value Theorem. *Comput. Educ.* 86, 157–171.
- Richert, A., Shehadeh, M., Plumanns, L., Groß, K., Schuster, K., Jeschke, S., 2016. Educating engineers for industry 4.0: Virtual worlds and human-robot-teams: Empirical studies towards a new educational age, in: 2016 IEEE Global Engineering Education Conference (EDUCON). Presented at the 2016 IEEE Global Engineering Education Conference (EDUCON), pp. 142–149.
- Ryan, R.M., Deci, E.L., 2000. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *Am. Psychol.* 55, 68–78.
- Salmani Nodoushan, M.A., 2009. The Shaffer–Gee perspective: Can epistemic games serve education? *Teach. Teach. Educ.* 25, 897–901.

- Sanchez, E., Jouneau-Sion, C., Delorme, L., Young, S., Lison, C., Kramar, N., 2012. Fostering Epistemic Interactions with a Digital Game A Case Study about Sustainable Development for Secondary Education.
- Sanchez, É., Monod-Ansaldi, R., 2015. Recherche collaborative orientée par la conception. *Educ. Didact.* 9, 73–94.
- Sanchez, E., Piau-Toffolon, C., Oubahssi, L., Serna, A., Marfisi-Schottman, I., Loup, G., George, S., 2016. Toward a Play Management System for Play-Based Learning, in: *European Conference on Technology Enhanced Learning*. Springer, pp. 484–489.
- Schultze, U., Michael Leahy, M., 2009. *The Avatar-Self Relationship: Enacting Presence in Second Life*.
- Serna, A., Tabard, A., Emin-Martinez, V., 2015. JEN. cards: un outil pour faciliter la conception collaborative de learning games, in: *Proceedings of the JEN Workshop, EIAH Conference*.
- Sewell, B., 2015. *Blueprints Visual Scripting for Unreal Engine*. Packt Publishing Ltd.
- Shaffer, D.W., 2006. Epistemic frames for epistemic games. *Comput. Educ., Virtual Learning?* 46, 223–234.
- Shaffer, D.W., 2006. Epistemology: The Debating Game, in: *How Computer Games Help Children Learn*. Palgrave Macmillan, New York, pp. 17–40.
- Shaffer, D.W., Gee, J.P., 2005. *Before Every Child Is Left Behind: How Epistemic Games Can Solve the Coming Crisis in Education*. WCER Working Paper No. 2005-7. Wisconsin Center for Education Research.
- Shaffer, D.W., Hatfield, D., Svarovsky, G.N., Nash, P., Nulty, A., Bagley, E., Frank, K., Rupp, A.A., Mislevy, R., 2009. Epistemic Network Analysis: A Prototype for 21st-Century Assessment of Learning. *Int. J. Learn. Media* 1, 33–53.
- Sowe, S.K., Stamelos, I., Angelis, L., 2008. Understanding knowledge sharing activities in free/open source software projects: An empirical study. *J. Syst. Softw.* 81, 431–446.
- Stedmon, A.W., Stone, R.J., 2001. Re-viewing reality: human factors of synthetic training environments. *Int. J. Hum.-Comput. Stud.* 55, 675–698.
- Steed, A., 2008. Some useful abstractions for re-usable virtual environment platforms. *Softw. Eng. Archit. Realt. Interact. Syst.-SEARIS*.
- Stelzer, R., Steindecker, E., Arndt, S., Steger, W., 2014. Expanding VRPN to Tasks in Virtual Engineering, in: *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*.

- American Society of Mechanical Engineers, p. V01BT02A026–V01BT02A026.
- Takala, T.M., 2014. RUIS: A Toolkit for Developing Virtual Reality Applications with Spatial Interaction, in: Proceedings of the 2Nd ACM Symposium on Spatial User Interaction, SUI '14. ACM, New York, NY, USA, pp. 94–103.
- Tan, C.T., Leong, T.W., Shen, S., Dubravs, C., Si, C., 2015. Exploring Gameplay Experiences on the Oculus Rift. ACM Press, pp. 253–263.
- Taylor, R.M., Jerald, J., VanderKnyff, C., Wendt, J., Borland, D., Marshburn, D., Sherman, W.R., Whitton, M.C., 2010. Lessons about Virtual-Environment Software Systems from 20 years of VE building. *Presence Camb. Mass* 19, 162–178.
- Tchounikine, P., 2009. Précis de recherche en Ingénierie des EIAH.
- Tramberend, H., 1999. Avocado: a distributed virtual reality framework. Presented at the Proceedings IEEE Virtual Reality, pp. 14–21.
- Walczak, K., Cellary, W., 2003. X-VRML for advanced virtual reality applications. *Computer* 36, 89–92.
- Warburton, S., 2009. Second Life in higher education: Assessing the potential for and the barriers to deploying virtual worlds in learning and teaching. *Br. J. Educ. Technol.* 40, 414–426.
- Widmer, D.S., Szilas, N., 2011. Un temps pour jouer, un temps pour écouter? L'effet de la position des explications théoriques dans un jeu pédagogique sur l'immersion et les performances en algèbre élémentaire, in: *Environnements Informatiques Pour l'Apprentissage Humain, Conférence EIAH'2011*. Editions de l'UMONS, Mons 2011, pp. 423–433.
- Wolf, M., Corroyer, D., 2004. *L'Analyse Statistique des Données en Psychologie (Statistical data analysis in psychology - translated by authors)*. Armand Colin, Paris.
- Wonner, J., Grosjean, J., Capobianco, A., Bechmann, D., 2013. Bubble Bee, an Alternative to Arrow for Pointing out Directions, in: *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology*. ACM, New York, NY, USA, pp. 97–100.
- Yingyan, T., Jun-sheng, Z., Zhijun, G., 2009. Design and Implementation of Interactive 3D Scenes Based on Virtools, in: *2009 International Forum on Computer Science-Technology and Applications*. Presented at the 2009 International Forum on Computer Science-Technology and Applications, pp. 87–89.
- Zyda, M., 2005. From visual simulation to virtual reality to games. *Computer* 38, 25–32.

Production personnelle

Publications et communications scientifiques

Loup, G., Serna, A., George, S. (soumis, EuroVR). A Community-Based Mixed and Virtual Reality Development Tool.

Sauret, F., Emin-Martinez, V., **Loup, G.**, Oubahssi, L., Piau-Toffolon, C., Sanchez, E., Serna, A. (2017). REARTH un exemple de Jeu Épistémique Numérique, de la conception à l'expérimentation. In: CETSIS 2017, Le Mans (France).

Loup, G., Serna, A., Iksal, S., George, S. (2016). Immersion and Persistence: Improving Learners' Engagement in Authentic Learning Situations. In: Proceedings of ECTEL 2016, Lyon (France). pp. 410-415.

Sanchez, E., Piau-Toffolon, C., Oubahssi, L., Serna, A., Marfisi, I., **Loup, G., George, S.** (2016). Toward a Play Management System for Game-Based Learning. In: Proceedings of ECTEL 2016, Lyon (France). pp. 484-489.

Loup, G. (2016). Environnement de développement pour l'intégration de composants de réalité mixte dans les jeux sérieux. In: Proceedings of RJC EIAH 2016, Montpellier (France). pp. 137-142.

Loup-Escande, E., Lotte, F., **Loup, G.**, Lécuyer, A. (2015). User-Centred BCI Game Design, in Handbook of Digital Games and

Entertainment Technologies, edited by Ryohei Nakatsu, Matthias Rauterberg and Paolo Ciancarini. pp. 225-250.

Loup, G., George, S., Serna, A. (2015). Fondements et caractérisation des jeux épistémiques numériques pervasifs. In: Proceedings of the 7ème conférence sur les EIAH. Agadir (Maroc). pp. 41-52.

Ateliers

Loup, G., George, S., Serna, A. (2017). Conception, développement et évaluation de jeux sérieux immersifs. IHM 2017, Poitiers (France)

Loup, G., George, S., Serna, A. (2017). Conception, développement et évaluation de jeux épistémiques numériques pervasifs : quelques pistes de réflexion. ORPHEE Rendez-vous 2017, Atelier : Réalités mixtes, virtuelles et augmentées pour l'apprentissage : perspectives et challenges pour la conception, l'évaluation et le suivi, Font-Romeu (France)

Démonstrations

Loup, G., George, S., Serna, A. (2017). MIREEDGE : un outil communautaire pour l'implémentation d'interactions immersives. IHM 2017, Poitiers (France)

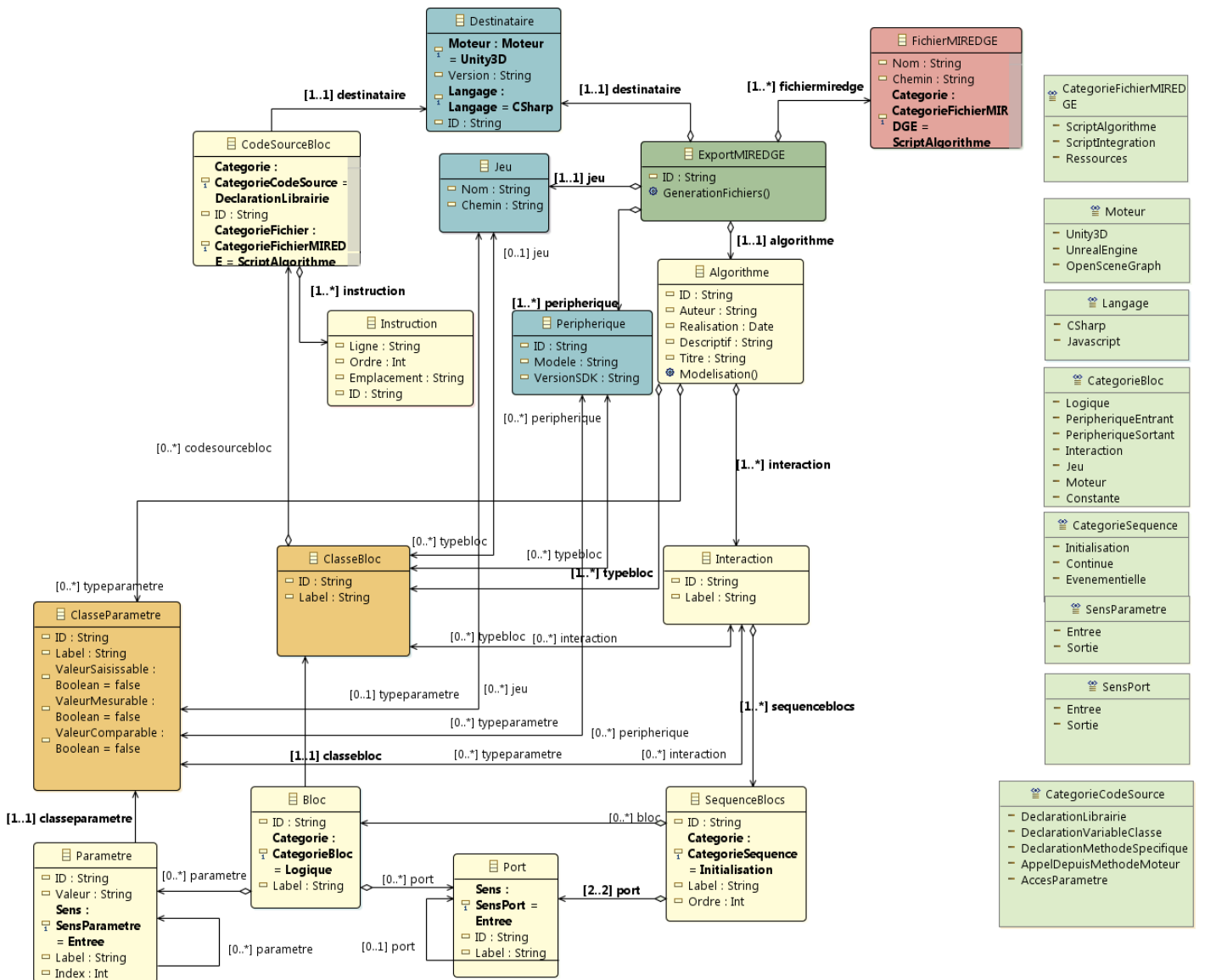
Salons

RearthM3 exposé lors de la journée d'inauguration de l'institut Claude Chappe au Mans 2015, à la fête de la science à Laval en 2015 et 2016, ainsi qu'à Laval Virtual 2017

MIREEDGE exposé à Laval Virtual 2017

Annexes

Annexe 1 : Métamodèle détaillé de MIREEDGE



Annexe 2 : Extrait de l'algorithme de génération de code pour Unity3D v5 en C#

```
[%
(FichierMIREEDGE.all.selectOne(fm|fm.Categorie=CategorieFichierMIREEDGE#ScriptAlgorithme).Chemin+"/MIREDD
GE/Scripts/"+FichierMIREEDGE.all.selectOne(fm|fm.Categorie=CategorieFichierMIREEDGE#ScriptAlgorithme).Nom).
println();
var BlocSpaceLevel=0;
%]
[%//Librairies
var libRequired = LibrairiesRequired();// Suppression des doublons
    for (lib in libRequired) {%]
[%=lib%]
    [%}
%]

[%//Déclaration de la classe
%]
public class [% var
className=FichierMIREEDGE.all.selectOne(fm|fm.Categorie=CategorieFichierMIREEDGE#ScriptAlgorithme).Nom;
className=className.replace(".cs","");%][%=className%] : MonoBehaviour {

[%//Déclaration des Variables de classe
BlocSpaceLevel=1;
var varClassRequired = VariablesClasseRequired();
for (varClass in varClassRequired) {%]
    [%=varClass%]
    [%}
%]

[%//Déclaration de la méthode Start de Unity => Ensemble des séquences d'initialisation
BlocSpaceLevel=2;
%]
    void Start() {
    [%//Rechercher les instructions sans emplacement des blocs de la séquence d'initialisation
    %]
    [% for (interact in Algorithme.all.first().interaction) {
        for (seq in interact.sequenceblocs.select(sq|sq.Categorie=CategorieSequence#Initialisation)) {
            var firstBlocID_OfSeq : String;
            firstBlocID_OfSeq = seq.getFistBlocID();
            if (firstBlocID_OfSeq.length()==0){
                ("Séquence "+seq.Label+" : Aucun premier bloc connecté !").println();
            } else {
                var blocToGeneration : Bloc = seq.bloc.selectOne(b|b.ID=firstBlocID_OfSeq);
                blocToGeneration.generateNextBloc(seq,BlocSpaceLevel);
            }
        }
    }%]
    [%}%]
    [%}%]
    [%//Rechercher les instructions de blocs utilisés dans d'autres séquences et qui spécifie un emplacement Start
    var instructionsRequired = InstructionsRequired("Start");
    for (instructionRequired in instructionsRequired) {
        //Space Manager
        for (i in Sequence{1..2}){
            instructionRequired=" "+instructionRequired;
        }
    }
}
```

```

    %][%=instructionRequired%]
    [% }
%]
}

void Update() {
    [%//Rechercher les instructions sans emplacement des blocs de la séquence continue
    BlocSpaceLevel=2;
    %]
    [%for (interact in Algorithme.all.first().interaction) {
        for (seq in interact.sequenceblocs.select(sq|sq.Categorie=CategorieSequence#Continue)) {
            var firstBlocID_OfSeq : String;
            firstBlocID_OfSeq = seq.getFistBlocID();
            if (firstBlocID_OfSeq.length()==0){
                ("Séquence "+seq.Label+" : Aucun premier bloc connecté !").println();
            } else {
                //("firstBlocID_OfSeq : "+firstBlocID_OfSeq).println();
                var blocToGeneration : Bloc =
seq.bloc.selectOne(b|b.ID=firstBlocID_OfSeq);
                //("blocToGeneration : "+blocToGeneration.Label).println();
                blocToGeneration.generateNextBloc(seq,BlocSpaceLevel);
            }
        }
    }
    [% }%]
    [% }%]

    [%//Rechercher les instructions de blocs utilisés dans d'autres séquences et qui spécifie un
emplacement Update
    var instructionsRequired = InstructionsRequired("Update");
    for (instructionRequired in instructionsRequired) {
        //Space Manager
        for (i in Sequence{1..2}){
            instructionRequired=" "+instructionRequired;
        }
    }
    %][%=instructionRequired%]
    [% }
%]
}

    [%//Rechercher les instructions sans emplacement des blocs des Sequences Evenementielles
    BlocSpaceLevel=2;
    %]
    [%for (interact in Algorithme.all.first().interaction) {
        for (seq in interact.sequenceblocs.select(sq|sq.Categorie=CategorieSequence#Evenementielle)) {[%]
    public void [%=seq.Label%]() {
[%
        var firstBlocID_OfSeq : String;
        firstBlocID_OfSeq = seq.getFistBlocID();
        if (firstBlocID_OfSeq.length()==0){
            ("Séquence "+seq.Label+" : Aucun premier bloc connecté !").println();
        } else {
            //("firstBlocID_OfSeq : "+firstBlocID_OfSeq).println();
            var blocToGeneration : Bloc = seq.bloc.selectOne(b|b.ID=firstBlocID_OfSeq);
            //("blocToGeneration : "+blocToGeneration.Label).println();
            blocToGeneration.generateNextBloc(seq,BlocSpaceLevel);
        }
    }
    [% }%]
}

[% }%]

```

```
[% }%]
```

```
[%//Méthodes spécifiques utilisées dans des blocs Gérer les doublons  
var instructionsFunctionsSpe = FunctionSpeRequired();  
for (instructionFunctionSpe in instructionsFunctionsSpe) {%][%=instructionFunctionSpe%]  
[% }  
%]  
}
```

Annexe 3 : Questionnaire de satisfaction

Questionnaire

Sujet ID :

Condition ID :

Ce questionnaire a pour but de recueillir vos impressions suite à la réalisation de la tâche de développement.

Pour les questions de 1 à 22, cochez la nuance entre les deux extrêmes de l'échelle. Pour les questions 23, 24 et 26, essayez de lister vos propositions, le plus clairement possible. Pour la question 25, cochez le degré d'importance qui correspond le mieux à la fonctionnalité associée. Gardez à l'esprit qu'il n'y a pas de bonne ou mauvaise réponse. Seule votre opinion compte !

1. En général, je suis satisfait(e) de la facilité d'utilisation de ce système

	0	1	2	3	4	5	6	7	8	9	10	
Pas du tout d'accord	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tout à fait d'accord

2. Ce système est simple à utiliser

	0	1	2	3	4	5	6	7	8	9	10	
Pas du tout d'accord	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tout à fait d'accord

3. J'ai complété mon travail correctement en utilisant ce système

	0	1	2	3	4	5	6	7	8	9	10	
Pas du tout d'accord	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tout à fait d'accord

4. J'ai été en mesure de réaliser rapidement la consigne avec ce système

	0	1	2	3	4	5	6	7	8	9	10	
Pas du tout d'accord	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tout à fait d'accord

5. J'ai réalisé la consigne efficacement en utilisant ce système

	0	1	2	3	4	5	6	7	8	9	10	
Pas du tout d'accord	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tout à fait d'accord

6. Je me sens à l'aise avec ce système

	0	1	2	3	4	5	6	7	8	9	10	
Pas du tout d'accord	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tout à fait d'accord

7. J'ai eu de la facilité à apprendre comment utiliser ce système

	0	1	2	3	4	5	6	7	8	9	10	
Pas du tout d'accord	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tout à fait d'accord

8. Je crois être devenu(e) rapidement efficace en utilisant ce système

	0	1	2	3	4	5	6	7	8	9	10	
Pas du tout d'accord	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tout à fait d'accord

9. Les messages d'erreur présentés par ce système m'indiquent clairement comment résoudre les problèmes

	0	1	2	3	4	5	6	7	8	9	10	
Pas du tout d'accord	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tout à fait d'accord

10. Lorsque j'ai fait une erreur d'utilisation sur ce système, il m'a été facile et rapide de la corriger

	0	1	2	3	4	5	6	7	8	9	10	
Pas du tout d'accord	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tout à fait d'accord

11. Les outils d'aide disponibles sur ce système (tels que l'aide en ligne, les messages à l'écran et autres informations) sont utiles

	0	1	2	3	4	5	6	7	8	9	10	
Pas du tout d'accord	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tout à fait d'accord

12. J'ai facilement trouvé l'information que je cherchais

	0	1	2	3	4	5	6	7	8	9	10	
Pas du tout d'accord	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tout à fait d'accord

13. L'information fournie avec ce système est facile à comprendre

	0	1	2	3	4	5	6	7	8	9	10	
Pas du tout d'accord	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tout à fait d'accord

14. L'information disponible sur ce système contribue à me soutenir dans la réalisation de la consigne

	0	1	2	3	4	5	6	7	8	9	10	
Pas du tout d'accord	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tout à fait d'accord

15. L'organisation de l'information dans le système est claire

	0	1	2	3	4	5	6	7	8	9	10	
Pas du tout d'accord	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tout à fait d'accord

16. L'interface de ce système est agréable

	0	1	2	3	4	5	6	7	8	9	10	
Pas du tout d'accord	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tout à fait d'accord

17. J'aime utiliser l'interface de ce système

	0	1	2	3	4	5	6	7	8	9	10	
Pas du tout d'accord	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tout à fait d'accord

18. Ce système possède toutes les fonctions et le potentiel correspondant à mes attentes

	0	1	2	3	4	5	6	7	8	9	10
--	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	-----------

Pas du tout d'accord **Tout à fait d'accord**

19. En général, je suis satisfait(e) de ce système.

0 1 2 3 4 5 6 7 8 9 10

Pas du tout d'accord **Tout à fait d'accord**

20. Je pense que le système est utile dans mon travail

0 1 2 3 4 5 6 7 8 9 10

Pas du tout d'accord **Tout à fait d'accord**

21. Utiliser le système me permettrait de développer plus rapidement des « serious game »

0 1 2 3 4 5 6 7 8 9 10

Pas du tout d'accord **Tout à fait d'accord**

22. Utiliser le système augmenterait ma productivité

0 1 2 3 4 5 6 7 8 9 10

Pas du tout d'accord **Tout à fait d'accord**

23. Inscrivez les principaux aspects négatifs

--

24. Inscrivez les principaux aspects positifs

--

25. Vous devez associer à chaque fonctionnalité proposée dans le tableau ci-après un degré d'importance.

Fonctionnalités	Inutile	Peu important	Moy. important	Important	Très important
Fonctionnalités déjà implémentées					
Blocs de variables	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Blocs de logique	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Blocs des périphériques	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Blocs du jeu existant	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Structuration Interaction/Séquence	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Liens de séquençement (ports)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Liens de paramétrage	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Génération du script	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Export direct dans le moteur de recherche	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Paramétrage automatique dans le moteur de jeu	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fonctionnalités non implémentées dans l'outil actuel mais envisagées					
Filtrer la palette en fonction des périphériques ciblés	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prévoir une banque d'algorithmes d'interactions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Editeur de blocs	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Générateur de diagramme à partir de scripts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Choix de la langue des commentaires	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

26. Selon vous, quelles fonctionnalités non proposées dans l'application mériteraient d'être ajoutées ?

Thèse de Doctorat

Guillaume LOUP

Conception et développement d'interactions immersives pour jeux sérieux

Résumé

Cette thèse s'inscrit dans le domaine de l'ingénierie des Environnements Virtuels pour l'Apprentissage Humain (EVAH) et a été financée par le projet ANR JEN.lab. Les jeux épistémiques numériques (JEN) sont une catégorie des jeux sérieux dédiés à la résolution de problèmes complexes, pluridisciplinaires et non-déterministes. L'objectif de ces JEN est de proposer des situations d'apprentissage authentiques du point de vue des interactions qu'elles permettent, les apprenants pouvant alors construire et ancrer des connaissances dans leur contexte d'usage.

La première partie de la thèse est consacrée à une méthodologie permettant de concevoir des JEN aux interactions enrichies sur le plan spatial, temporel et social, nommés jeux épistémiques numériques pervasifs (JENP). Des expérimentations ont été menées en milieu écologique et ont montré un apport pédagogique par une augmentation de l'engagement des apprenants. La conception des prototypes expérimentés a mis en évidence que le développement informatique des interactions immersives d'un EVAH demeure actuellement réservé à des spécialistes en Réalité Mixte ou Virtuelle (RMV).

Ainsi la seconde partie de la thèse a été dédiée à la proposition d'un environnement de développement nommé MIREEDGE pouvant être utilisé par des développeurs non spécialistes en interactions immersives. Une interface de programmation visuelle permet aux développeurs de modéliser des interactions sans connaissance spécifique en RMV par réutilisation de blocs. Les principaux blocs sont destinés à représenter visuellement une fonctionnalité ou un algorithme d'un périphérique de RMV. Des blocs logiques permettent aux développeurs d'établir un algorithme d'une interaction particulière par composition. Enfin d'autres blocs permettent de relier le nouvel algorithme à d'autres méthodes existantes dans le projet où l'interaction sera intégrée. Des expérimentations ont permis d'évaluer l'efficacité et l'efficacité de MIREEDGE pour le profil des développeurs RMV ainsi que celui des développeurs non-initiés à la RMV.

Les deux profils de développeurs sont complémentaires dans MIREEDGE, les premiers proposant de nouveaux blocs d'interactions à l'ensemble de la communauté, les seconds, en plus grand nombre, pouvant mener à un assemblage optimal. Les développeurs RMV ayant une nécessité à poursuivre leur algorithme au-delà de la programmation visuelle, MIREEDGE permet une génération automatique de code fournissant le script correspondant et totalement rééditable dans leur moteur de jeu. L'approche MIREEDGE, dans son ensemble, s'inscrit dans une architecture logicielle permettant le partage rapide d'interactions selon un méta-modèle. Son indépendance logicielle permet de garantir une large compatibilité avec les environnements de développement et les périphériques, ouvrant ainsi des perspectives sur l'étude de logiques d'interactions universelles.

Mots clés

Jeux sérieux, Jeux épistémiques numériques, Interactions immersives, Développeur de jeux sérieux, Programmation visuelle, Moteur de jeu, Développeur en réalité virtuelle

Abstract

This thesis is in the field of Virtual Environments for Human Learning (VEHL). It was funded by the JEN.lab national ANR project, which aims to create Digital Epistemic Games (DEG), a category of Serious Games dedicated to solving complex, multidisciplinary and non-deterministic problems. The objective of DEGs is to propose authentic learning situations in terms of interactions so that learners can construct and anchor knowledge in their context of use.

In the first part of this thesis, we propose a methodology for designing pervasive DEGs (i.e., DEGs with spatially, temporally or socially enriched interactions). Several experiments we conducted in ecological environments showed the educational benefits of DEGs through an increase of the learner's engagement. However, the design of these DEGs put forward the fact that the development of immersive interactions for VEHLs currently requires the skills of Mixed or Virtual Reality (MVR) experts.

In the second part of this thesis, we therefore propose a development environment named MIREEDGE, intended for developers who are not MVR specialists. A visual programming interface allows developers to model interactions without specific knowledge, by using predefined blocks. The main blocks represent a functionality or an algorithm of a MVR device. The logical blocks allow developers to create an algorithm for a specific interaction. Finally, other blocks can be used to connect these algorithms to the existing methods of the project that will integrate these interactions. We led experimentations in order to evaluate the effectiveness and efficiency of MIREEDGE for MVR specialist and non-specialist developers.

The two developer profiles are complementary in MIREEDGE. Indeed, the MVR specialists create new interaction blocks for the whole community, and the MVR non-specialists, much more numerous, assemble these blocks in optimal ways. As MVR specialists need to define the algorithms in the visual blocks in detail, MIREEDGE automatically generates the corresponding code scripts that can be edited in their game engine. The MIREEDGE approach, in general, offers a software architecture that allows sharing interactions according to a meta-model. Its software independence ensures wide compatibility with development environments and MVR devices. This opens perspectives for the study of universal interactions.

Key Words

Serious Games, Pervasive Games, Digital Epistemic Game, Immersive interactions, Serious Game developer, Visual programming, Game engine, Virtual reality developer