



HAL
open science

Un jeu de rôles dans le laboratoire Arcade : une autre façon d'enseigner la programmation

Viviane Guéraud

► **To cite this version:**

Viviane Guéraud. Un jeu de rôles dans le laboratoire Arcade : une autre façon d'enseigner la programmation. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1988. Français. NNT : . tel-00332109

HAL Id: tel-00332109

<https://theses.hal.science/tel-00332109>

Submitted on 20 Oct 2008

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.

THESE

présentée par

Viviane GUERAUD

pour obtenir le titre de **DOCTEUR**
de **L'UNIVERSITE JOSEPH FOURIER - GRENOBLE I**

(arrêté ministériel du 5 Juillet 1984)

Spécialité : **INFORMATIQUE**

**UN JEU DE RÔLES DANS LE LABORATOIRE ARCADE :
UNE AUTRE FAÇON D'ENSEIGNER LA PROGRAMMATION**

Thèse soutenue le 3 Février 1989

Composition du Jury:

Président :	SCHOLL Pierre-Claude
Rapporteurs :	BOUSSARD Jean-Claude VIVET Martial PERROT Jean
Examineurs :	CRAMPES Michel PEYRIN Jean-Pierre

Thèse préparée au sein du Laboratoire de Génie Informatique de Grenoble



Je tiens à remercier,

Monsieur Pierre-Claude SCHOLL, Professeur, Directeur de l'U.F.R. d'Informatique à l'Université Joseph Fourier - Grenoble I, de me faire l'honneur de présider le jury de cette thèse. Il a eu la gentillesse de m'accueillir il y a quelques années dans l'équipe de recherche en Enseignement Assisté par Ordinateur qu'il dirigeait.

Monsieur Jean-Claude BOUSSARD, Professeur en Informatique à l'Université de Nice, de l'intérêt qu'il porte aux travaux de notre groupe. Je lui suis très reconnaissante d'avoir accepté d'être rapporteur de cette thèse.

Monsieur Martial VIVET, Professeur en Informatique à l'Université du Mans d'avoir bien voulu exercer le rôle de rapporteur. Je lui sais gré de l'intérêt qu'il a manifesté pour mon travail.

Monsieur Jean PERROT, Professeur à la Faculté des Lettres, des Sciences de l'Homme et des Sociétés, Directeur du Laboratoire de Recherche sur le Jeu et le Jouet à l'Université Paris-Nord, d'avoir accepté la charge de rapporteur. Je lui suis reconnaissante d'avoir bien voulu s'intéresser à mon travail d'informaticienne.

Monsieur Michel CRAMPES, Ingénieur, Responsable Etude et Développement en Ingénierie Pédagogique au sein de la société SYSECA TEMPS REEL à Saint-Cloud, d'avoir accepté de prendre connaissance de ce travail et de participer à ce jury. Ses travaux ont été pour moi un encouragement à poursuivre ma propre recherche.

Monsieur Jean-Pierre PEYRIN, Maître de Conférences à l'Université Joseph Fourier - Grenoble I, Directeur de la MST ESI, et responsable du groupe ARCADE, de l'enthousiasme avec lequel il défend toujours ses idées et les nôtres. Malgré les nombreuses responsabilités dont il a par ailleurs la charge, il sait se montrer toujours disponible. Il a été un directeur de thèse très précieux, prodiguant les conseils et les encouragements nécessaires. Qu'il trouve ici le témoignage du plaisir avec lequel j'ai travaillé sous sa direction. Qu'il soit remercié de la confiance qu'il m'a faite en acceptant de diriger cette thèse, et de l'aide chaleureuse qu'il m'a apportée tout au long de ce travail.

Les autres membres du groupe ARCADE qui, chacun à leur manière ont contribué à donner vie au laboratoire Arcade. Tous m'ont aidée à mener à bien cette recherche, qu'ils en soient remerciés:

Jean-Michel CAGNAT, Maître de Conférences à l'Université Joseph Fourier - Grenoble I, directeur de la MIAGE pour les conseils avisés et les critiques toujours pertinentes qu'il m'a donnés, ainsi que pour le temps qu'il a toujours accepté de me consacrer.

Sylvie PAINVIN, psycho-pédagogue, pour l'énergie avec laquelle elle m'a toujours épaulée. Je n'oublie aucune des nombreuses pierres qu'elle a apportées à mon travail, pas plus que la disponibilité et la bonne humeur dont elle a toujours fait preuve à mon égard.

Inggriani LIEM, pour l'aide et le soutien moral qu'elle m'a accordés, ainsi que pour les multiples attentions dont elle sait chaque jour combler son entourage, avec le sourire et la discrétion qui la caractérisent.

Je tiens aussi à remercier

L'ensemble des membres du Laboratoire de Génie Informatique dirigé successivement par Monsieur Jacques Mossière et Monsieur Yves Chiaramella. Et tout particulièrement Madame Liliane Di Giacomo, Madame Nicole Hervé et le service comptable pour les services rendus.

L'ensemble du personnel de la médiathèque pour son indispensable travail.

Monsieur Joseph Sifakis et le secrétariat du 3ème cycle informatique, et plus généralement l'ensemble du personnel gérant le 3ème cycle à l'Université Joseph Fourier qui ont eu la charge administrative de cette thèse.

Les membres du service de reprographie pour le soin et l'efficacité qu'ils ont apportés au tirage de cette thèse.

Je tiens enfin à remercier

Tous mes proches, parents ou amis, qui à un moment ou à un autre, m'ont facilité la tâche en me libérant de quelque souci ou de quelque contrainte.

et surtout,

Claude pour son constant soutien de tous ordres,
Maïca pour sa joie de vivre,
et tous deux pour toutes choses qui ne sauraient être exprimées ici.

Plan

Introduction

Chapitre I. Une aide pour l'étude de la programmation: le laboratoire Arcade

- I.1. Le laboratoire Arcade tel que nous le rêvions
- I.2. Apprendre et enseigner dans le laboratoire Arcade
- I.3. Approche de la programmation dans le laboratoire Arcade
- I.4. Description du laboratoire
- I.5. Richesse du laboratoire
- I.6. Extensibilité du laboratoire

Chapitre II. Une activité particulière dans le laboratoire Arcade: le jeu

- II.1. L'activité de jeu
 - II.1.1. Définitions et caractéristiques
 - II.1.2. Typologie des jeux
 - II.1.3. Les jeux de rôles
 - II.1.4. Les jeux sur ordinateur
- II.2. Le rôle pédagogique du jeu
 - II.2.1. Une vue bibliographique de la question
 - II.2.2. Notre point de vue

Chapitre III. Jeu et programmation

- III.1. La programmation et son univers
- III.2. La programmation est-elle un jeu ?
- III.3. Jeu et étude de la programmation
 - III.3.1. Le jeu sujet d'études
 - III.3.2. Le jeu outil d'apprentissage
 - III.3.2.1. Des réalités: le jeu des plaquettes, ChipWits, JAC
 - III.3.2.2. Des attentes

Chapitre IV. "Toi, Moi et Lui", un jeu de rôles pour l'étude de la programmation

IV.1. Motivations pédagogiques

IV.2. Du changement de domaine au jeu de rôles

IV.3. Un jeu de rôles dans l'enseignement de la programmation

IV.4. "Toi, Moi et Lui" ou l'analogie avec la programmation

IV.5. Toi, Moi, Lui ou un autre: quel rôle pour l'ordinateur ?

IV.6. Les différentes utilisations pédagogiques de "Toi, Moi et Lui"

IV.6.1. Quel meneur de jeu ?

IV.6.2. Meneurs de jeu, créez votre propre scénario !

IV.6.3. Meneurs de jeu et enseignants: pour un transfert vers la programmation

IV.7. Prolongements

Conclusion

Références bibliographiques

Annexes

I. Jouons avec "Toi, Moi et Lui", manuel d'utilisateur

II. Jouons avec "Toi, Moi et Lui", livret du meneur de jeu

III. Description technique du jeu "Toi, Moi et Lui"

Introduction

Face à l'importance que prend actuellement l'informatique dans notre société, les défis à relever sont de trois ordres: démythifier l'ordinateur aux yeux de tous, permettre au plus grand nombre d'utiliser l'ordinateur facilement et à bon escient, et enfin donner une formation de qualité aux informaticiens.

Ce travail se situe dans l'axe de formation des informaticiens, tout particulièrement dans le contexte des formations universitaires. Son objectif n'est pas de modifier le contenu de l'enseignement proposé, ni de modifier la forme de cet enseignement. Il s'agit plutôt d'apporter une aide aux étudiants, et donc aux enseignants, en proposant aux premiers de nouvelles activités qu'ils puissent pratiquer en complément de la formation classique qu'ils reçoivent. Ces activités doivent leur permettre une approche complémentaire des concepts et méthodes abordés en cours et mis en pratique au niveau des travaux dirigés et des travaux pratiques.

Notre groupe a choisi d'utiliser l'ordinateur comme support aux activités proposées. Si l'ordinateur et son emploi constituent le thème de l'enseignement, les étudiants n'utilisent concrètement la machine que pour écrire un nombre limité de programmes dans le cadre des classiques travaux pratiques. Dans le même temps, l'enseignement traditionnel se heurte à des difficultés, voire à des impossibilités de présentation de certains aspects dynamiques, tels que l'exécution d'un programme, l'évolution des structures de données, les relations entre l'analyste et le programmeur etc. Nous avons donc voulu étudier et proposer diverses façons d'utiliser l'ordinateur comme aide à l'enseignement de la programmation.

D'autres travaux participent à l'enseignement de l'informatique par des moyens informatiques, selon diverses spécialisations: la conduite de projet (JAC, [Crampes 87]) et le génie logiciel (IPHIGENIE [Coulette 87]), un langage de programmation (LISP Tutor [Anderson 85]), un environnement de programmation ([Comar 87]), une aide à l'implantation de données abstraites (SAIDA [Grandbastien 88]), une aide à la correction de programmes (PROUST [Jonhson 85]), etc. Notre propre objectif situe notre travail au carrefour de ces différents axes.

Nous souhaitons en effet proposer aux étudiants un large éventail d'activités sur l'ensemble des concepts et aussi des méthodes nécessaires à la programmation depuis l'énoncé du problème décrit dans un cahier des charges jusqu'à l'écriture d'un programme dans un langage. Nous avons voulu proposer aux étudiants un lieu (ou plutôt l'image informatisée d'un lieu) où ils puissent pratiquer selon leurs besoins, leurs doutes ou leurs envies, ces diverses activités. Nous avons souhaité que ce lieu soit agréable, qu'il offre de bonnes conditions de travail, et nous l'avons baptisé "laboratoire informatisé pour l'enseignement de la programmation" ou laboratoire Arcade en reprenant le nom de notre projet.

L'élaboration et la conception générale du laboratoire ont constitué un travail collectif de quatre personnes (J.M. Cagnat, S. Painvin, J.P. Peyrin et moi-même) auquel j'ai activement participé dès le départ. Une telle étape de définition de projet tire sa substance de la confrontation d'idées, d'orientations et de désirs. Cette élaboration étant faite [Cagnat 86b], les quatre participants précédents et I. Liem ont pu alors œuvrer à la construction concrète du laboratoire, dans des voies parallèles.

Parmi les activités que nous souhaitions offrir aux étudiants dans le laboratoire, les modes de travail tels que manipulation, jeu et visualisation étaient privilégiés; ils constituent autant d'axes de recherche.

Ma propre recherche s'est centrée sur l'activité de jeu. Elle a commencé par l'appropriation du domaine "jeu" à partir d'un travail bibliographique. Cette réflexion a été menée avec la collaboration de S. Painvin, psychologue [Guéraud 86b].

Ces bases m'ont ensuite permis d'aborder de façon constructive le thème central de la thèse qu'est l'activité de jeu au service de l'enseignement de la programmation. Prenant en compte l'existant dans ce domaine et les besoins ressentis dans l'enseignement classique de la programmation, ce travail a débouché sur la conception, la spécification et la réalisation d'un jeu de rôles qui met en œuvre, en les modélisant, les différentes attitudes d'un programmeur lors de la résolution d'un problème.

Le premier chapitre de la thèse présente les idées fortes qui nous ont guidé dans l'élaboration du laboratoire d'enseignement de la programmation: notre rêve, nos choix pédagogiques, notre approche de la programmation. Je décris ensuite le laboratoire actuel, et je tente d'analyser toute la richesse qu'il peut présenter pour chacun des intervenants: étudiants, enseignants et même logiciels qui y sont placés. Enfin, j'essaie d'en montrer la généralisation possible.

Le second chapitre repose sur l'investigation que j'ai menée dans le domaine du jeu. Je présente tout d'abord une caractérisation de l'activité de jeu, une typologie des jeux, différentes définitions ainsi que des résultats opératoires pour la suite de mon travail, notamment en ce qui concerne les jeux sur ordinateur. J'aborde aussi la délicate question du rôle pédagogique du jeu. Après une revue bibliographique du sujet, je précise la position que je défends dans notre contexte.

Le troisième chapitre étudie les rapports entre jeu et programmation. Je montre en quoi l'activité de programmation peut offrir des occasions de jeu. J'étudie aussi la place occupée par le jeu dans l'enseignement de la programmation, en particulier comme outil d'apprentissage. Je présente et tente d'analyser les quelques jeux à vocation pédagogique rencontrés dans ce domaine. Enfin, vu l'activité de programmation, je définis quels types de jeu pourraient intervenir dans son enseignement.

Le quatrième chapitre présente le jeu de rôles que j'ai conçu et réalisé dans le cadre du laboratoire Arcade. Ce jeu repose sur une analogie entre la création d'un programme et la création d'une bande dessinée. Je présente les motivations pédagogiques qui ont été à l'origine de sa conception, ainsi que les raisons qui m'ont conduit à placer les étudiants dans un domaine autre que celui de la programmation. Le jeu réalisé est décrit; les analogies avec le travail de programmation sont précisées. Enfin, les diverses utilisations pédagogiques du jeu sont discutées.

Nous suggérons au lecteur qui voudrait appréhender le contenu de cette thèse sans vouloir en connaître avant tout les fondements d'ordre psycho-pédagogiques de ne pas aborder en profondeur le paragraphe 1.2 ni le chapitre II lors d'une première lecture.

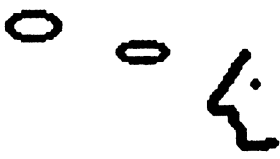
Chapitre I

**Une aide pour l'étude de la
programmation :**

le laboratoire ARCADE



Si Arcade m'était conté...



Grenoble. Septembre 85.
Deux enseignants en programmation
sont à la recherche de leurs étudiants.

POP!

OUAH!

Ah! Les voilà! Ils sont plus motivés
et plus actifs qu'en cours...

Bong!

Et si on créait un lieu
de travail sympa, où ils
pourraient manipuler, vi-
sualiser, jouer... à leur
rythme, selon leur façon
d'apprendre?

Un laboratoire
informatisé
de
programmation

Bientôt le travail commence.

On pourrait faire ça... et ça...

Oh non! Bof!

Mais si!

Zut! ça a déjà
été fait!

Mais le travail
continue. Chacun
apporte une pièce au
puzzle.

Des tris, des trains
et des machines...

Des cubes, des arbres
et des fractals...

Toi, moi et lui...

Sans oublier Knuth,
Morris et Pratt...

Eh! j'attends
ta maquette.

J'ai fini.

Enfin...
Presque!

Mars 88. Le labo Arcade est enfin réalisé.

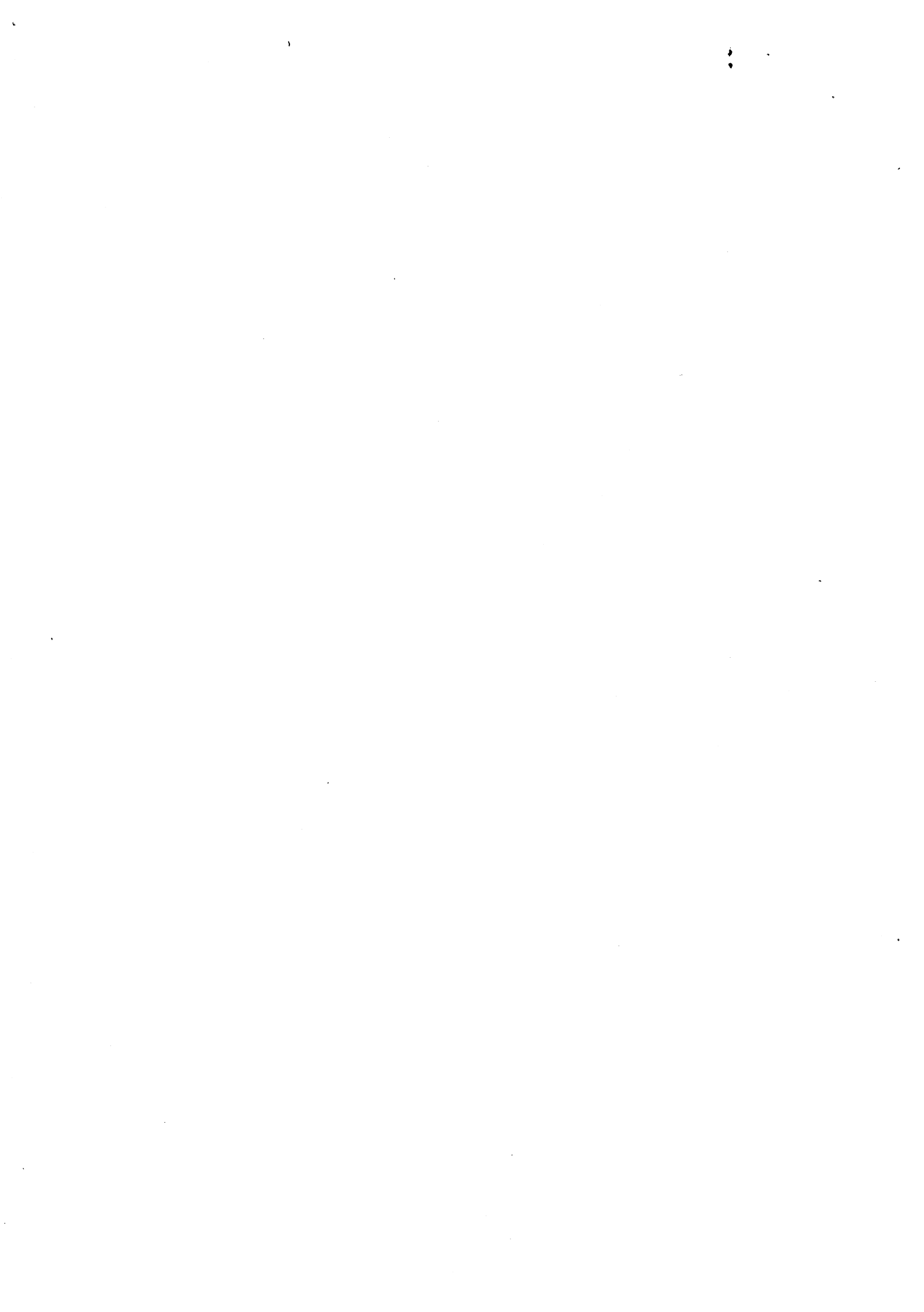
Bip!

Clicloc!

A toi!

Henri,
apporte-moi
un café...

Ça marche!



Chapitre I. Une aide pour l'étude de la programmation: le laboratoire Arcade

Ce chapitre présente les idées fortes qui nous ont guidé dans l'élaboration du laboratoire d'enseignement de la programmation: notre rêve, nos choix pédagogiques, notre approche de la programmation. Nous décrivons ensuite le laboratoire actuel et nous tentons d'analyser toute la richesse qu'il peut présenter pour chacun des intervenants: étudiants, enseignants et même logiciels qui y sont placés. Enfin, nous essayons d'en montrer la généralisation possible.

Nous avons donc voulu étudier et proposer diverses façons d'utiliser l'ordinateur comme aide à notre enseignement de la programmation. Notre objectif n'a pas été de développer un cours complet et autonome et nous n'envisageons pas que l'étudiant puisse partir avec une disquette et revenir quelques mois après passer l'examen ! Nous voulions plutôt lui proposer des logiciels venant **enrichir** l'enseignement traditionnel tel qu'il existe dans notre environnement: cours, travaux dirigés, travaux pratiques.

I.1. Le laboratoire Arcade tel que nous le rêvions

Dans certaines disciplines (physique, chimie), il existe des "**laboratoires**" spécialement équipés dans lesquels les étudiants se livrent à un certain nombre d'activités. Ces laboratoires sont des lieux d'expériences et de recherches où les étudiants se rendent pour étudier et approfondir certains concepts présentés en cours. Ils y trouvent du matériel lourd, installé en permanence dans certaines salles (exemple: salle d'électrostatique, salle d'hydraulique). Ils y trouvent aussi du matériel à utiliser pendant l'expérience (flacons de produits divers en Chimie,...), ainsi que des appareils de mesure transportables.

Pourquoi ne pas avoir notre "**laboratoire de programmation**" [Cagnat 86a] ?

Imaginons ce laboratoire comme un ensemble de bâtiments. Chaque bâtiment serait dédié à un thème particulier de programmation: le bâtiment "récusivité", le bâtiment "tri", le bâtiment "arbres binaires",... Ce laboratoire serait totalement informatisé. Les logiciels réalisés constitueraient les étages, les salles, les équipements lourds, le matériel du laboratoire.

D'autres modèles connus dans des domaines divers (Cité des Sciences et de l'Industrie, arcades de jeux, jeux d'aventures) nous ont inspiré dans la conception de notre laboratoire.

La Cité des Sciences et de l'Industrie de La Villette est un lieu culturel de divertissement, d'information et de formation. De nombreux thèmes y sont illustrés par un ensemble d'activités réparties dans d'immenses salles et offertes au public. Le visiteur est libre de choisir les thèmes et activités qui l'intéressent. C'est lui qui construit son cheminement personnel vers une meilleure compréhension du thème choisi. L'ensemble privilégie la participation active du visiteur à des manipulations et expériences diverses [Guéraud 86a].

Dans un genre assez différent, nous avons les **arcades**, où billards électriques, baby-foot, et jeux vidéo se côtoient dans une salle généralement enfumée et très fréquentée. Ils sont à la disposition des joueurs qui viennent y passer un moment de détente, passant d'une machine à une autre au gré de leurs envies. La liberté de circulation, la facilité d'accès aux machines et la diversité des jeux proposés sont des éléments très attractifs pour le joueur, qui ainsi peut exercer ses capacités de raisonnement et d'observation, de même que ses réflexes.

Considérons enfin un **jeu d'aventures**, tel que Transylvania (commercialisé par Penguin Software), dans lequel le joueur est transporté dans un espace imaginaire qui le confronte à de multiples dangers (vampires et autres loups-garous) au cours de sa recherche d'une princesse en danger.

Selon les commandes qu'il donne au système, le joueur se retrouve dans divers lieux. Libre à lui d'aller et venir comme bon lui semble. Il faut imaginer les décisions à prendre et comment tirer le meilleur parti des objets qu'il voit au hasard de ses déplacements. Un exemple: je vois un crapaud, puis plus tard (ou avant) du papier attrape-mouches, puis plus tard (ou avant) des mouches, et tout ceci parmi d'autres objets. L'idée peut alors me venir d'attraper les mouches avec le papier pour les donner au crapaud (lequel, pour me remercier, me fournit alors une indication précieuse dans ma quête de la princesse).

En écartant les aspects anecdotiques des jeux de ce type, on constate qu'ils font appel à une démarche intéressante: il y a un problème à résoudre; des éléments de solution sont dispersés dans un ordre quelconque; il faut remarquer ces éléments, en choisir certains et les combiner pour progresser (peut-être) en direction de la solution. Il n'y a pas de guidage formel, mais une démarche par essais-erreurs mettant en œuvre des fonctions de raisonnement déductif, analogique,...

Ces divers exemples sont pour nous des sources d'inspiration, des modèles. Certes ces modèles ont leurs limites; leurs buts ne sont pas identiques et ne visent pas le même type d'apprentissage; des adaptations seront nécessaires. Mais nous aimerions que notre laboratoire soit une **synthèse** de ces modèles: qu'il soit d'un abord plaisant, qu'il laisse une grande liberté à son utilisateur, qu'il l'implique dans des activités nombreuses et variées. Nous voudrions que la manipulation, le jeu et la visualisation y soient des modes de travail privilégiés pour l'étudiant. Nous rêvons que l'étudiant puisse se déplacer à sa guise, qu'il choisisse, en fonction des problèmes rencontrés ou des thèmes qui l'intéressent, d'aller explorer les salles susceptibles de l'aider à progresser vers une solution ou vers une meilleure compréhension.

I.2. Apprendre et enseigner dans le laboratoire Arcade

Au delà de notre rêve, nous avons voulu préciser les choses au niveau pédagogique. Un apprentissage peut-il se faire dans le cadre que nous rêvons pour notre laboratoire ? Quel apprentissage visons-nous et quel enseignement

voulons-nous donner à travers les activités proposées ? Cette étude de nature pédagogique a été réalisée dans notre équipe par Sylvie Painvin, psychologue, et a fait l'objet du rapport "Voulez-vous apprendre avec moi ?" [Painvin 86]. Nous présentons dans ce paragraphe les résultats de cette étude qui nous semblent pertinents.

Différents facteurs interviennent simultanément quand un être humain est en train d'apprendre: ils sont d'ordre physiologique, social, affectif, intellectuel, moral,... La psychologie apporte des réponses sur ces différents plans séparément, mais sa principale difficulté reste d'élaborer un modèle global qui puisse rendre compte de l'interaction des facteurs et des variations individuelles. Il ne s'agit pas de faire ici l'inventaire et l'analyse de toutes ces réponses (et des questions restant sans réponses), aussi l'accent a-t-il été mis sur quelques caractéristiques de l'activité "apprendre" qui nous paraissent déterminantes et globalisantes.

Notre analyse de l'activité "apprendre"

- apprendre, c'est être en action

"Apprendre n'est pas le passif d'enseigner, c'est un acte qu'on exerce sur soi-même", nous dit Reboul [Reboul 80]. Ajoutons que dans le plus petit processus humain (cognitif, affectif, social) il y a activité: par exemple, percevoir, mémoriser, comprendre, ressentir comportent une part de choix et de construction, donc d'action. Selon le type d'activités mis en jeu, l'action prend des formes différentes, elle n'est pas seulement extérieure, visible, elle est aussi intérieure.

La motivation joue un rôle important dans toute activité. Elle recouvre les facteurs qui "déclenchent l'activité du sujet, la dirigent vers certains buts, lui permettent de se prolonger si ces buts ne sont pas immédiatement atteints et l'arrêtent le moment venu" [Reuchlin 78].

L'étude des motivations humaines est difficile. La place de l'imaginaire, l'accès à l'information et son traitement, les influences sociales jouent un rôle essentiel dans l'attitude de l'homme face à une tâche, une idée, une valeur [Reuchlin 78].

- apprendre, c'est être en relation

Qu'ils soient comportementalistes, cognitivistes ou psychanalystes, tous les psychologues soulignent l'importance de la relation de l'individu au monde. L'enfant apprend parce qu'il est à l'écoute de ce qui l'entoure, parce qu'il veut le comprendre, le maîtriser ou le modifier. Il s'y cherche et il veut s'en différencier. L'existence d'"enfants sauvages" montre bien la nécessité de la présence d'autrui dans le développement de l'enfant.

Rogers a beaucoup insisté sur la qualité de l'écoute dans la relation interpersonnelle: qu'il soit thérapeute ou enseignant, celui qui aide doit être à l'écoute positive de la personnalité et des expériences d'autrui. Son rôle est celui d'une "personne-ressource", reflétant et reformulant les expressions (émotions, questions, réponses) de celui qui cherche à comprendre [Rogers 70].

- apprendre, c'est être en congruence

Etre congruent, cela veut dire être en relation avec ses propres sentiments, c'est-à-dire les repérer, les accepter, le cas échéant les communiquer [Rogers 70]. Apprendre requiert d'intégrer, donc d'accueillir et de faire sien ce qui arrive. La non-connaissance de soi, la non-confiance en soi, le renoncement à sa propre personnalité ne permettent que des apprentissages superficiels, éphémères ou conventionnels. Pouvoir répéter la pensée du maître, ce n'est pas avoir appris.

Chacun apprend en fonction de sa propre personnalité. Certains ont besoin de méthode, de guide (inhibition à l'action, manque de confiance en soi, recherche de

la perfection ou de la vérité). D'autres préfèrent agir, essayer, pratiquer. Diverses expériences ont montré que le travail isolé favorise l'apprentissage des introvertis, alors que le travail en groupe favorise celui des extravertis. Certains apprennent vite, d'autres sont plus lents. Certains acceptent, d'autres contestent. Toutes ces attitudes sont trop souvent ignorées par les éducateurs, qui préfèrent projeter sur l'apprenant réel leur propre modèle d'apprenant idéal, ou voir leurs élèves à travers les yeux des autres [Rosenthal 71].

- apprendre, c'est tenir compte du temps

Si on apprend à tout âge, on apprend surtout quand on est jeune. Le fait d'apprendre est une activité nécessaire, vitale pour le jeune (enfant ou animal). Le jeune apprend davantage, plus vite (mémoire non structurée, hyperactivité,...). On n'apprend pas les mêmes choses et de la même manière, à des âges différents.

Faire des expériences et les intégrer prend du temps. Apprendre demande du temps, car il faut relier les choses entre elles, laisser se faire un travail subconscient. Les intuitions ne se programment pas. Les travaux de psychologie expérimentale ont également montré que l'apprentissage massé (en continuité) était moins efficace que l'apprentissage distribué (interruption de la tâche par un changement d'activité ou un repos), pour des tâches de type résolution de problèmes [Reuchlin 78].

- apprendre, c'est accepter le changement

Apprendre provoque un changement. Cette notion de changement est centrale, car on a appris quand quelque chose a changé en soi: une connaissance, une compréhension, un schème d'action, un mode de pensée, une aptitude, une défense émotionnelle,...

Le changement passe toujours par une phase de destruction et de reconstruction, c'est pourquoi Reoul peut dire qu'apprendre, c'est aussi désapprendre. C'est accepter le risque de changer quelque chose, perdre une structure antérieure, gagner une nouvelle structure, et tout rejouer l'instant suivant.

Apprendre, c'est aussi accepter le plaisir et le pouvoir attachés au savoir, accepter de s'affirmer, de se battre pour ses idées, ses convictions, sa propre pensée. Lewin a montré que tout individu oppose une résistance au changement, le sien propre ou celui d'autrui. Le changement est signe de vie, mais il dérange celui qui ne fait pas confiance à la vie.

Notre approche de l'enseignement

S. Painvin nous suggère que l'enseignant, le thérapeute et l'artiste sont de la même famille et poursuivent des buts voisins (sinon identiques). L'enseignant ne doit-il pas alors dialoguer avec ses cousins, jusqu'à leur emprunter les idées, techniques et valeurs adéquates à sa propre fonction ? Des éléments provenant des trois courants: pédagogie, psychologie, spectacle, sont rassemblés ici.

L'enseignant dont nous rêvons ne prétend pas détenir la vérité, mais il a des convictions, une foi fondées sur sa propre expérience ou celle d'autrui. Il n'applique pas de recettes, mais il suit des lignes directrices, opératoires. Il n'hésite pas à s'engager, à chercher, à innover, à s'adapter.

- l'enseignant a un rôle à jouer

Il y a deux types d'éducation: l'éducation spontanée (dans la vie) et l'éducation intentionnelle (dans des institutions). On peut vivre sans école, par l'entremise de la famille, des amis qui "enseignent" (montrent) ce qu'il faut savoir: informations et

savoir-faire peuvent être appris sur le tas. Mais pour révéler toutes les potentialités humaines, notamment accéder à la synthèse, à l'abstraction, à la création, l'enseignement (qu'il soit scolaire, professionnel ou initiatique, collectif ou individuel) est indispensable.

L'enseignant a des devoirs, des pouvoirs et des limites. Le rôle de l'enseignement est de préparer à la vie en créant des situations qui simulent celles de la vie sans se confondre avec elles: faire progresser l'élève, le conduire à la maîtrise, tout en le protégeant. C'est ce que Reboul appelle la simulation enseignante. Dans ce cadre, l'enseignant doit assurer une discipline, évaluer les résultats de son enseignement, programmer la matière dont il a la charge, motiver ses élèves, choisir en permanence ce qu'il peut et doit dire, ce qu'il peut et doit faire. Le contexte institutionnel (horaires, programmes, locaux, règlements) et les propres aptitudes de l'enseignant sont autant de limites ou de ressources pour mener à bien ce travail.

- l'enseignant est à l'écoute de la personne globale et différente qu'est l'élève

Prenons l'exemple du cinéma, de la thérapie. Tout le monde est d'accord pour qu'il existe des films de types différents, de formes ou de sujets variés. L'oeuvre est le produit de la personnalité du créateur, qui se projette (en partie) dedans et qui vise un certain but. Elle est perçue par le spectateur qui en filtre, interprète et assimile le message en fonction de sa propre histoire. La rencontre du cinéaste avec le spectateur se fait ou ne se fait pas. Dans le cas du cancer, immunothérapie, radiologie, chirurgie, chimiothérapie, psychothérapie peuvent être utilisées selon le degré de gravité de la maladie, la disponibilité des techniques, le choix du médecin, celui du patient. Dans le domaine de la psychothérapie aussi, il existe de nombreuses approches: le premier entretien a précisément pour but de "voir si ça peut marcher ensemble". En est-il autrement en matière de relation pédagogique ? Il faut des méthodes pédagogiques variées (pour l'enseignant, pour l'élève, pour la recherche même).

Or l'enseignement français privilégie les méthodes traditionnelles, analytiques, passives, centrées sur les mécanismes individuels de la pensée. Il privilégie actuellement les technologies au détriment de la relation enseignant-enseigné, les fonctions intellectuelles au détriment des fonctions sociales, affectives ou physiologiques, ce qui nie à la fois l'interaction des fonctions psychologiques et la tonalité dominante des élèves. Selon le niveau d'"apprentissage" où l'on se place, on peut observer une prépondérance de certaines fonctions, cependant, "sans l'activité globale et motivée, l'apprentissage est impossible" [Reboul 80].

- l'enseignant accompagne l'élève

- en lui proposant des repères, des modèles et des méthodes

Le rôle des repères est de permettre la structuration des savoirs, leur mémorisation et leur assimilation, de façon économique (retenir l'essentiel) et intelligente (saisir les liens). La maîtrise de repères sur les choses et les idées provenant du monde extérieur favorise la construction de l'identité psychologique de l'être humain. D'où l'importance des rites de passage d'un niveau à un autre (évaluation de l'élève; changement de sujets ou d'approches).

Les modèles (au sens de réalisation exemplaire) sont utiles, car ils donnent un but à la tâche et permettent de se situer par comparaison. Ils doivent être manipulés et critiqués, et susciter qu'à terme l'élève crée lui-même, en se démarquant des modèles d'autrui. Dans l'apprentissage des langues, on observe que les élèves se fabriquent des grammaires transitoires, imparfaites, mais qui reflètent l'état de leurs connaissances. Ces grammaires évoluent avec l'apprentissage, jusqu'à tendre vers une grammaire correcte.

Le rôle des méthodes (de travail global, de résolution de tâches précises) est d'éviter une perte de temps excessive, en faisant profiter l'élève des expériences accumulées par d'autres, et d'aider l'élève dans ses difficultés. Tout le problème consiste à sélectionner la méthode qui correspond à telle tâche, à telle situation d'enseignement, à tel élève, à tel enseignant... Sans parler du fait que ne pas intervenir auprès de l'élève peut être considéré comme une "méthode" pédagogique ou éducative, il semble que tout enseignement implique l'idée de méthode, à savoir une intention consciente, réfléchie et organisée de transmettre des savoirs dans un contexte spécifique.

- en le faisant agir

"C'est en forgeant qu'on devient forgeron". "La théorie naît de la pratique". "Comprendre, c'est toucher du doigt". C'est le principe qui est mis en œuvre à la Cité des Sciences et de l'Industrie de La Villette à Paris [Guéraud 86a]. Manipulation, jeu, simulation sont autant de formes d'action nécessaires pour s'approprier un savoir. Elles permettent de découvrir par soi-même, d'expérimenter, de poser des hypothèses, de faire des déductions et des analogies. Dans l'action, l'être est impliqué de manière globale. Il crée librement sa propre pensée.

L'enfant et le petit animal apprennent à vivre en jouant, en manipulant des objets. Ils imitent, répètent, font comme si leur activité avait un but réel: du gazouillis à la parole, de la poupée au bébé, de la bagarre à la chasse, le jeu fait acquérir des aptitudes. Les rites, les contes de fées, les mythes aussi simulent la réalité en la transposant dans un espace protégé, réglé, irréel; ce sont des élaborations poétiques (compréhensibles par tous) des fantasmes humains. La fascination actuelle des jeunes (et des adultes !) pour les jeux vidéo ou jeux de rôle à fantaisie médiévale ou de science-fiction exprime le même besoin: évoluer dans un lieu autre où l'on peut

faire comme si, sans danger. Jouer et apprendre, c'est aller du désordre (déséquilibre) vers l'ordre (équilibre).

Winnicott, parlant du processus de thérapie, dit: la créativité du patient, le thérapeute qui en sait trop peut la lui dérober. Ce qui importe, ce n'est pas tant le savoir du thérapeute, que le fait qu'il puisse cacher son savoir ou se retenir de proclamer ce qu'il sait: "Le moment-clé est celui où l'enfant se surprend lui-même, et non celui où je fais une brillante interprétation" [Winnicott 71].

- en le laissant se tromper

Les enseignants n'aiment en général pas l'échec. Or l'échec existe et l'élève le perçoit. Abolir l'échec, c'est refuser la vie. Le nier ou le relativiser, c'est tromper l'élève. Au lieu d'éliminer l'échec de la pédagogie, Reboul propose de susciter une pédagogie de l'échec [Reboul 80]:

- en évitant l'échec irrémédiable (on peut recommencer)
- en tenant compte des facteurs extrascolaires de l'échec
- en ne renonçant pas à évaluer, mais en n'enfermant pas l'élève dans son échec
- en montrant à l'élève son propre rôle dans son échec
- en lui montrant que son échec concerne ce qu'il a fait et non ce qu'il est (même si sa personne entière est engagée).

L'échec ne se manifeste pas de la même façon selon ce que l'on apprend: il peut être erreur, incapacité ou incompréhension (sa réussite, c'est la compréhension).

Pour Freud, les erreurs ont un sens [Freud 68]: lapsus et actes manqués (oublis) sont liés à des représentations et affects refoulés. Ces erreurs, si elles perturbent l'activité scolaire, ne sont pas négatives en soi puisqu'elles permettent de se rendre compte que de l'inconscient demande à être entendu. L'affectif participe aux activités intellectuelles: vouloir enfouir des savoirs qui ne peuvent être intégrés est aberrant.

- en le surprenant et en le provoquant

Apprendre nécessite de modifier l'équilibre antérieur des structures cognitives concernées. Cela ne peut se faire que dans l'inconfort et la rupture:

Inconfort, parce que changer est anxiogène, demande de l'énergie, des efforts, parce que changer demande de choisir, de se situer, de s'engager.

Rupture, parce que réfléchir, résoudre des paradoxes, dépasser les contrastes exigent de prendre de la distance, de se décentrer (comme l'a bien montré la théorie de la forme).

Apprendre se fait souvent par la souffrance, mais s'il existe une "école de la souffrance" (la souffrance subie permet de comprendre les autres et soi-même), ce n'est pas la souffrance par elle-même qui éduque, mais cette forme particulière de souffrance qu'est l'épreuve. L'épreuve fait prendre conscience des limites, et en même temps sélectionne les conduites qui permettent de la surmonter [Reboul 80].

- en donnant sa place au plaisir

Le plaisir d'apprendre tient à la curiosité naturelle de l'individu, qui peut cependant diminuer ou disparaître si l'environnement est hostile (interdits, contraintes, uniformité). L'enseignant doit accepter de jouer, de mettre en scène, de faire du

spectacle. Outre son talent personnel de comédien inhérent au choix de ce métier, les technologies actuelles peuvent l'aider et le permettront de plus en plus. "Si la formation veut trouver un nouveau souffle, elle doit chercher du côté du lude, du spectacle" [Béville 86].

En résumé

Chacun apprend en fonction de sa propre **personnalité** d'où l'intérêt d'utiliser des méthodes pédagogiques variées, de présenter des activités de natures diverses, et de laisser à l'étudiant le choix de son cheminement propre (mais sans le perdre!). Pour apprendre, il faut être motivé. Si les facteurs de **motivation** ou de **démotivation** sont nombreux et difficiles à cerner, il n'en reste pas moins qu'il faut tenter, au niveau où l'on intervient, de faire naître et de maintenir cette motivation: attirer, **séduire**, stimuler la curiosité, **défier**, mais aussi **sécuriser** par l'univers protégé que doit être le lieu d'apprentissage. Il est important de noter la place d'éléments subjectifs dans l'apprentissage, intervenant au niveau de la motivation, mais aussi au niveau du travail subconscient nécessaire au changement de structuration... Enfin, nous n'oublierons pas qu'"Apprendre, c'est aussi désapprendre" et qu'il faut oser **surprendre** l'étudiant, risquer de le mettre momentanément dans l'inconfort par un changement de niveau, de point de vue, voire de domaine. L'enseignant et l'enseignement proposé doivent être des **ressources** pour l'étudiant, au service de celui-ci.

Nous voulons nous appuyer sur cette approche pédagogique pour mettre en œuvre tant la structure globale du laboratoire, que l'approche de chacun des logiciels que nous y proposerons, tout en laissant la place à la créativité et aux intuitions de chaque concepteur.

I.3. Approche de la programmation dans le laboratoire

Pendant très longtemps, l'enseignement de la programmation a été réduit à l'enseignement d'un langage. Les élèves apprenaient "comment dire", à eux d'imaginer "quoi dire". Ce stade est maintenant dépassé et nous pouvons constater un consensus des enseignants ([Meyer 78], [Arsac 80], [Lucas 83], [Ducrin 84], [Laurent 85],...) qui s'intéressent dorénavant plutôt aux formes de raisonnement et aux méthodes de travail en vue d'une bonne conception des programmes. Là est bien le cœur de la programmation, comme en témoigne Anna Gram dont le souci est de rendre la démarche informatique systématique par l'étude des principaux raisonnements qui sont à la base de la construction de logiciels. [Gram 86].

Quelle programmation enseigner ? Programmation impérative, programmation fonctionnelle, logique ou orientée-objets ? Ces différents types de programmation ne sont pas autant d'ennemis, prêts à s'entretuer sous les emportements de leurs supporters, mais représentent bien plusieurs modes de pensée. Ont témoigné dans ce sens au Colloque francophone sur la Didactique de l'Informatique [Didactique 88], aussi bien la communication de J.F. Cloutier [Cloutier 88] que la synthèse du colloque faite par son président J. Arsac.

L'idéal serait qu'un même individu, capable de pratiquer ces différents modes de pensée, sache choisir en fonction de la nature du problème qu'il a à résoudre, le mode qui conviendra le mieux. Il est évident que selon la formation de l'individu, l'un ou l'autre de ces modes de pensée lui sera plus familier, plus aisé. Il est donc d'autant plus important de ne pas proposer un seul mode de pensée, et les enseignants s'attachent de plus en plus à présenter ces différents types de programmation au cours des formations universitaires.

Ne pouvant pas tout faire tout de suite dans notre laboratoire, nous avons dû faire un choix, et nous nous sommes plutôt intéressés à l'enseignement de la

programmation impérative. D'abord, comme nous l'avons fait remarquer plus haut, nous pensons que la programmation impérative reste toujours d'actualité. De plus, l'ambiance impérative est encore bien présente dans le domaine de la programmation. Enfin, la programmation impérative conduit aux langages les plus répandus, et donc répond à l'attente des étudiants ainsi qu'à celle des collègues d'autres disciplines. Il est naturellement souhaitable d'élargir par la suite notre travail, dans le but d'aborder au sein du laboratoire Arcade les différents types de programmation.

Dans ce contexte de programmation impérative, nous nous intéressons à tous les concepts, méthodes, ou comportements nécessaires à son apprentissage.

Pour un thème donné, nous souhaitons proposer aux étudiants des activités leur permettant plusieurs approches, plusieurs points de vue. Nous voulons que l'étudiant puisse visualiser un algorithme donné de diverses manières (une vue peut-être très proche de l'exécution ou plus éloignée) selon diverses options (choix des données,..), qu'il puisse comparer l'efficacité d'algorithmes traitant un même sujet, qu'il ait la possibilité de tester sa compréhension de l'algorithme, qu'il soit amené à choisir les "outils" nécessaires pour faire fonctionner un algorithme donné, qu'il puisse découvrir l'idée d'un algorithme, soit par des expérimentations libres, soit selon une démarche guidée...

I.4. Description du laboratoire

A partir de nos motivations premières, et des réflexions diverses que nous venons de présenter, le laboratoire a pris forme. Nous souhaitons maintenant en donner une description très concrète [Cagnat 88c].

Le laboratoire: un lieu d'accueil

Le rôle premier et fondamental du laboratoire est d'être un lieu d'accueil:

- Accueil des logiciels qui y sont placés.
- Accueil de l'étudiant qui vient y pratiquer un ensemble d'activités.
- Accueil de l'enseignant qui y cherche des compléments à ses cours.

Pour remplir au mieux ce rôle, le laboratoire fournit:

- Un local agréable et bien situé qui met en valeur les logiciels.
- Une bonne organisation d'ensemble permettant au visiteur de se diriger en fonction de ses intérêts, de découvrir l'existence des logiciels appropriés et de les utiliser facilement.
- De bonnes conditions de travail qui facilitent l'apprentissage et augmentent le plaisir d'apprendre.

Même si notre laboratoire privilégie la liberté de l'étudiant, les besoins de **communication** entre enseignant et étudiant ne disparaissent pas totalement. Quelques mécanismes simples sont utilisés à cet effet:

- Des listes d'exercices suggérés stimulent l'imagination de certains étudiants, ou attirent leur attention sur des difficultés précises.
- Un mécanisme de type "courrier des lecteurs" ou "forum" qui permet des échanges de vues tout en restant informel, redonne une certaine chaleur humaine à l'environnement. Il fournit de plus à notre groupe la possibilité de recueillir facilement les remarques et les suggestions des utilisateurs.

La liberté laissée à l'étudiant d'organiser ses activités à son gré s'oppose à l'idée de "contrôler" le cheminement de chacun. Rien n'empêche toutefois l'enseignant d'aiguiller ses étudiants vers telle ou telle activité en fonction de

l'avancement de son cours, l'étudiant restant libre d'aller, de sa propre initiative, explorer des "régions" inconnues. L'idée de fournir dans le laboratoire-même des plans de parcours (des guides "touristiques" ou "thématiques") est à l'étude.

Les logiciels développés par notre groupe sont évidemment placés dans le laboratoire, mais le laboratoire reste un lieu **ouvert**. Nous rencontrons parfois des logiciels qui, par certains aspects, illustrent des concepts en rapport avec le thème de notre laboratoire, même si ce n'est pas leur principal objectif. Nous proposons donc des activités offertes par des logiciels commercialisés ou du domaine public. Nous les désignons par le terme de "logiciels invités", par analogie avec les "professeurs invités" ou les "conférenciers invités". On retrouve bien ici l'aspect fondamental de lieu d'accueil.

La structure du laboratoire: bâtiments, salles, activités

Conformément à la métaphore spatiale utilisée, le laboratoire se présente comme un ensemble de bâtiments, dans lesquels on doit se rendre pour accéder à diverses activités.

- **Le paysage:** le premier écran propose un paysage semé de bâtiments, dont le nom renseigne sur le contenu. Il suffit de cliquer sur un bâtiment pour pénétrer à l'intérieur.

- **Les bâtiments:**

+ Le bâtiment "Accueil": il permet à un utilisateur qui vient pour la première fois, de consulter quelques écrans d'informations sur les principales conventions employées.

+ Un bâtiment standard contient une salle (et une seule, pour l'instant). Toutes les salles ont le même aspect général. Dans chaque salle, apparaissent les noms des différentes activités proposées (accompagnés d'icônes dont le symbolisme deviendra plus significatif avec la pratique des activités).

+ Le bâtiment "Langages", pour une vraie programmation. Illustration parfaite de notre politique d'invitations, la salle "Langages" propose un interpréteur Pascal (de Think Technologies) et un interpréteur Basic (de Microsoft). Pour faciliter son travail, l'étudiant a accès à des fichiers qui contiennent le texte de plusieurs des programmes étudiés dans les autres salles. Il peut ainsi examiner leur code, suivre leur exécution normalement ou en pas à pas, les modifier, écrire des variantes, etc.

+ Des bâtiments spéciaux: bibliothèque, cinéma,...

- Une activité type:

Le choix d'une activité conduit à un écran de présentation, similaire pour toutes les activités. On y trouve toujours:

+ une (très) brève description de l'activité, qui suffit au novice pour se faire une idée de ce que comporte cette activité. Il peut ainsi explorer rapidement la salle pour voir les ressources qu'elle offre. L'utilisateur régulier trouve là une confirmation qu'il s'agit bien de l'activité qu'il veut pratiquer.

+ une icône permettant de lancer effectivement l'activité. Le mode d'emploi propre au logiciel choisi entre alors en vigueur. Lorsque l'utilisateur "quitte" le logiciel, il revient sur ce même écran de présentation.

+ selon les activités, une ou plusieurs autres icônes donnent accès à des conseils d'utilisation, à des idées d'exercices, au courrier des utilisateurs, etc.

Le contenu actuel (et provisoire) du laboratoire

Le laboratoire est entièrement réalisé sur du matériel Apple Macintosh. Il contient les salles et les logiciels suivants:

Salle Caractères

- Compter les A, Compter les LE, Machines Caractères
(illustration directe de certains chapitres de [Scholl 88])

Salle Exploration de Tables

- Exploration de Tables, Pattern Matching, Recherches Min et Max
(illustration d'algorithmes de recherche dans un tableau)

Salle Graphes et Arbres

- Parcours de graphes, Tri topologique
(visualisation d'algorithmes manipulant des pointeurs)
- Arbres binaires
(pratique libre d'algorithmes de parcours et d'insertion)
- Logiciel invité: Binary Trees

Salle Langages (Logiciels invités)

- Interpréteur Macintosh Pascal (de Think Technologies, Inc)
- Interpréteur Microsoft Basic (de Microsoft)
- Tangram [Lucci 89]
(aide à l'écriture à partir d'une bibliothèque de schémas et d'algorithmes en notation algorithmique interprétable)

Salle Récursivité

- Dessins Récursifs [Liem88b]
(découverte de la définition et du programme récursif)
- Tours de Hanoï [Guéraud 85]
(découverte guidée, du jeu à l'algorithme récursif)
- Logiciels invités: Towers of Hanoï, MandelZoom, Tree, Munching Squares

Salle Spécifications

- Toi, Moi et Lui (jeu de rôles, à 3 joueurs) [Guéraud 88]
- Anagramme (bande dessinée illustrant un travail de spécifications et de recherche d'algorithme)

Salle de Tris [Painvin 88b]

- La Gare de Triage [Painvin 88a]
(dessin animé où l'on applique à des wagons le tri par minimum)
- Tri Guidé (apprentissage guidé du tri par minimums) [Cagnat 88a]
- Meccano de Tri [Cagnat 88b]
(pratique libre de divers algorithmes de tri, sur des cubes)
- Tris Internes [Liem 88a]
(visualisation et confrontation de 7 algorithmes de tri)
- Tri en BD
(compléter la bande dessinée en respectant l'algorithme du tri par minimum)

Différentes salles du laboratoire Arcade

Tri

Les principaux tris

Le tri par minimums

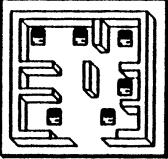
Meccano de Tri

Tris Internes

Tri Guidé

La Gare de Triège

Tri en 3D

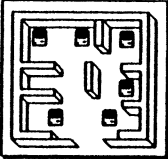


Exploration de Tables

Pattern Matching

Recherches Min et Max

Exploration de Tables



Récurtivité

Dessins Récuratifs

Tours de Hanoi

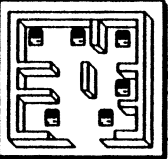
Logiciels invités

Towers of Hanoi

Tree

Punching Squares

Mando Zoom

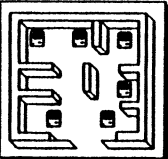


Caractères

Machines Caractères

Compter les A

Compter les LE



Langages

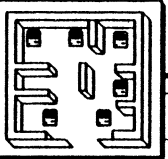
Logiciels invités

Algo-Assistance

Tangram

Pascal

Basic



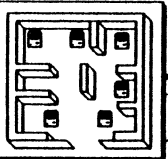
Graphes et Arbres

Parcours de Graphes

Tri Topologique

Arbres Binaires

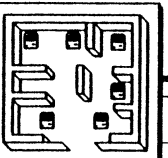
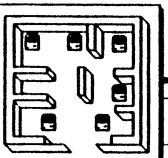
Logiciel invité Binary Trees



Spécifications

Toi, moi et lui

Anagramme

Logiciel par logiciel

Afin de caractériser le type de travail proposé à l'élève par le laboratoire, voici une description de certains des logiciels que nous avons réalisés et qui ont déjà fait l'objet de quelques expérimentations avec des étudiants.

🍏 Gare de triage: dessin animé

Ce dessin animé est centré sur l'algorithme de tri par minimums successifs et est construit sur un double objectif pédagogique: enseigner et amuser. La fantaisie utilisée est le triage de wagons en gare, par 2 cheminots, et la formulation de cette activité par un dialogue musical entre un étudiant-analyste et un professeur-programmeur.

Plusieurs activités sont présentées à l'étudiant:

- l'observation du processus tel qu'il pourrait se passer dans la réalité
- l'analyse des différents composants du processus
- leur codage dans un langage de programmation
- l'observation de l'exécution de la procédure de tris par minimums

L'étudiant est spectateur individuel du film qui se déroule devant lui. Il peut, grâce à un tableau de commande, arrêter le film, revenir en arrière, continuer, changer de vitesse de projection, avancer en pas à pas.

🍏 Tri Guidé: apprentissage par manipulations guidées

Il s'agit ici de guider un processus d'apprentissage relativement autonome en proposant différents degrés de liberté et différents niveaux de détails.

On fournit des objets, et des opérations sur ces objets qui permettent la mise en œuvre d'un certain algorithme. L'étudiant peut, par menu, se placer dans différents

modes de travail:

- *démonstration automatique* : il voit fonctionner l'algorithme, sans intervention de sa part.

- *mode suggéré* : seule l'opération nécessaire est activable. L'idée est de faire découvrir l'enchaînement des opérations tel qu'il est dans l'algorithme.

- *mode vérifié* : toutes les opérations sont proposées, mais seule la bonne opération sera acceptée. Il s'agit ici de s'assurer que l'ordre des opérations a été bien compris.

- *mode libre* : tout est permis, que cela corresponde ou non à l'algorithme. (L'étudiant teste ainsi s'il est capable de répéter correctement tout l'algorithme, sans aucune aide.)

L'algorithme choisi est un tri par minimums successifs avec un tableau de Données et un tableau de Résultats. Des icônes placées entre les deux tableaux permettent d'effectuer les opérations suivantes:

- rechercher le minimum dans le tableau Données;
- transférer un élément du tableau Données dans le tableau Résultats (avec remplissage de haut en bas du tableau Résultat);
- éliminer la valeur sélectionnée dans le tableau Données (la case passe visuellement en noir).

Un deuxième niveau permet un apprentissage plus approfondi en obligeant à faire en détail la recherche du minimum.

🍷 Meccano de Tri: manipulations algorithmiques

L'idée est de mettre à la disposition de l'étudiant un ensemble d'outils permettant de réaliser des tris d'objets représentés à l'écran par des cubes (dont la valeur est invisible). Ces outils sont de divers niveaux: déplacer un cube vers une zone de

travail; échanger deux cubes; rechercher le cube contenant la plus petite valeur; etc. L'étudiant peut choisir librement des outils parmi ceux qui lui sont proposés. Il doit ensuite travailler uniquement avec les outils qu'il a initialement choisis. Selon les choix faits, il devient possible (ou non) de faire, plus ou moins rapidement, des tris suivant tel ou tel algorithme.

Dans l'état actuel, le logiciel permet de pratiquer la plupart des algorithmes classiques non récurifs: tri par minimums (ou par maximums) successifs; tri par insertion; tri par bulles; tri de Shell; tri par fusion; etc.

Différents usages pédagogiques sont possibles: expérimentation libre; essai de reproduction d'un algorithme imposé (défini par son nom ou par une description de son principe). Notons que dans ce logiciel l'étudiant reste libre dans ses manipulations. On ne cherche ni à le guider vers une solution, ni à vérifier si ses manipulations sont conformes à un algorithme particulier.

🍏 Tris Internes: observation d'exécution

Ce logiciel permet l'observation de l'exécution des principaux algorithmes de tris internes.

Les tris choisis sont ceux présentés dans [Wirth 86]. Nous avons choisi ce livre particulier parce qu'il est très connu et que le chapitre sur les tris nous est apparu exemplaire d'un point de vue pédagogique, même si les algorithmes ne sont pas toujours présentés dans leur version la plus efficace.

Le principe de la visualisation d'un tableau a été tiré d'un travail antérieur [Brown 85] réalisé à l'Université Brown aux Etats-Unis: un élément du tableau est symbolisé par un trait proportionnel à la valeur de cet élément. Ainsi le tri est exécuté sur une représentation graphique du tableau.

Le logiciel réunit deux applications:

- la première permet de commander l'exécution d'un algorithme de tri: l'étudiant choisit un algorithme; il choisit également les caractéristiques des données (aléatoires, ordonnées, inversées, égales ou bien introduites par lui-même) ainsi que le nombre d'éléments. Il décide de la vitesse de l'exécution. Il peut à tout moment recommencer, ou changer l'algorithme et le jeu de données.

- la seconde permet de comparer les algorithmes de tri à l'aide d'une évaluation du temps d'exécution et des nombres de comparaisons et d'affectations. L'évaluation est présentée par des histogrammes.

🍏 Dessins récursifs: travail sur la récursivité

Le but de ce logiciel est d'offrir à un étudiant un certain nombre d'outils pour qu'il découvre, à sa manière, la définition récurrente d'un dessin géométrique (et donc le programme récursif qui le construit).

Les dessins proposés sont de difficultés variées: carrés imbriqués, étoile, flocon, triangles imbriqués, courbe de Peano, courbe de Sierpinski, courbe de Hilbert.

Le logiciel propose à l'étudiant:

- une lecture structurée de l'algorithme.
- une observation de l'exécution de l'algorithme (l'étudiant définit la vitesse d'exécution, la profondeur de la récursivité, et le mode de l'observation: normal, ou par niveau).
- un guidage dans la compréhension de la description récursive (comparaison ou superposition de niveaux consécutifs, repérage d'un niveau à l'intérieur d'un niveau différent, règles de construction).
- un exercice de renforcement (l'étudiant désigne, l'un après l'autre, tous les éléments du tracé qui doivent être effectués: un système de points décomptés en cas d'erreur donne un aspect de jeu à l'exercice).

I.5. Richesse du laboratoire

Suite à cette description concrète (mais partielle) du laboratoire, essayons d'analyser toute la richesse qu'il peut représenter pour chacun des intervenants: étudiants, enseignants et même logiciels qui y sont placés.

Pour l'étudiant

- l'attrait de la présentation du laboratoire et la facilité d'accès à l'intérieur de celui-ci permise par l'interface proposée l'invitent à l'explorer.
- la structuration du laboratoire et la facilité d'accès lui permettent un cheminement aisé vers les thèmes qui l'intéresse.
- la totale liberté de circulation lui permet de se créer un cheminement personnel, répondant à ses propres motivations. Nous avons vu en effet que l'apprentissage dépend de la personne, de son intérêt à un instant donné, de sa curiosité. Certains étudiants préfèrent par exemple attendre qu'un thème ait été abordé en cours avant de s'aventurer sur les logiciels qui le traitent, d'autres au contraire seront capables d'une grande énergie pour aborder certains thèmes non encore présentés en cours. Il est intéressant de permettre ces différentes approches.
- le nombre important de thèmes abordés et leur diversité lui permet un choix effectif en fonction de ses motivations et le place ainsi dans de bonnes conditions d'apprentissage.
- le nombre et la variété des activités proposées sur un même thème lui permettent d'avoir plusieurs approches, plusieurs points de vue du sujet. Ceci présente toujours une richesse pour un apprentissage et facilite une maîtrise du sujet: un enfant par exemple aborde la lecture à la fois en lisant, en parlant et en écoutant. Ces approches

sont complémentaires, elles apportent toutes quelque chose en elles-mêmes aussi bien que comme pièces d'un puzzle. Les différents logiciels proposés dans le laboratoire sur le thème tri illustrent bien cette idée [Painvin 88b].

- la variété des niveaux auxquels peut être abordé un thème, lui permet de trouver chaussure à son pied. Certaines activités sont plus proches de la familiarisation, d'autres nécessitent une compréhension fine des concepts.

Il n'est pas simple de définir le niveau d'un logiciel. Un même logiciel peut en effet présenter des activités de niveaux différents: c'est le cas de "Tri guidé" avec ses différents modes de travail [Cagnat 88a]. De plus, une même activité peut être pratiquée à des niveaux différents selon la connaissance qu'on a du sujet. C'est le cas par exemple de "Tris internes" [Liem 88a] qui propose la visualisation de différents algorithmes de tris: soit on ne connaît pas les algorithmes de tris, on observe alors leur fonctionnement global, leur performance et du coup on aura envie (ou non) de s'intéresser de plus près à ces algorithmes, soit on connaît les algorithmes et on essaie de retrouver leurs caractéristiques dans le fonctionnement observé [Liem 89]. Les exercices suggérés dans la pratique des activités peuvent aussi être de niveaux très différents comme dans "Gare de triage" [Painvin 88a].

- la liberté d'action à l'intérieur d'un logiciel favorise une démarche active.

- les notes, les suggestions d'exercices, le manuel d'utilisation et le courrier des lecteurs qui accompagnent les logiciels sont là pour l'aider en cas de difficulté ou d'incertitude, voire de désaccord !

Pour l'enseignant

Les caractéristiques précédentes lui permettent d'utiliser les ressources du laboratoire comme le ferait un étudiant. Les expériences faites avec nos collègues montrent qu'ils y trouvent eux-même un profit.

De plus, l'enseignant peut vouloir utiliser les activités proposées dans le laboratoire au profit de son cours. Il peut le faire de diverses manières, selon qu'il choisit de:

- utiliser directement un logiciel en cours afin de montrer quelque chose. Des expériences ont été faites dans ce sens avec le logiciel "Dessins récursifs". Les résultats sont très encourageants.

- guider les étudiants (selon leur bon vouloir) à des moments précis vers tel ou tel logiciel, dans le but de

- faciliter ensuite la présentation d'une notion,

- illustrer une notion qu'il vient de présenter,

- faire travailler les étudiants sur un thème donné,

- leur donner l'occasion d'explorer un domaine qu'il n'aborde que partiellement en cours,

- leur donner des points de vue différents, complémentaires, peut-être apparemment opposés, sur un même sujet,

- leur donner l'occasion de s'exercer sur un sujet...

- le considérer comme un outil de plus mis à la disposition des étudiants, en libre service, et auquel il peut faire référence comme on fait référence à un ouvrage ou à une situation vécue dans l'univers de la formation...

L'enseignant peut aussi intervenir à l'intérieur du laboratoire en complétant la liste de suggestions d'exercices.

Pour les logiciels

- le laboratoire joue le rôle d'écrin en fournissant un environnement d'utilisation confortable, agréable. Le logiciel est ainsi mis en valeur.

- la présence d'autres logiciels, en particulier sur le même thème qu'un logiciel donné, fournit un message implicite à l'utilisateur: le logiciel considéré n'est plus censé fournir une réponse complète, absolue à tous les problèmes du thème traité. Il représente une approche qui, ainsi remise à sa place, peut être mieux appréciée.
- l'ensemble des logiciels qui participent à la compréhension d'un même thème tire des bénéfices de cette association. Une articulation plus étroite peut être explicitement établie entre divers logiciels d'un même thème. C'est le cas du logiciel "Dessins récurifs" et du logiciel "Animations". Les deux logiciels en tirent une plus-value.

I.6. Extensibilité du laboratoire

Les richesses que présente le laboratoire Arcade ne sont en fait liées à la programmation que par l'objet des activités proposées.

Vers d'autres branches de l'informatique

Imaginons maintenant que le laboratoire se préoccupe de l'enseignement des systèmes, des bases de données, d'intelligence artificielle... La structure d'accueil du laboratoire peut s'étendre sans devoir se modifier en profondeur. Il s'agit d'ajouter de nouveaux bâtiments, éventuellement de les regrouper en quartiers. Le plus gros travail reste naturellement d'imaginer les activités et d'élaborer les logiciels correspondants pour les nouvelles branches de l'informatique abordées.

Une telle extension est actuellement envisagée. Les enseignants concernés seraient naturellement mis à contribution: ils repèrent bien ce qui leur rendrait service pour leurs cours, et le laboratoire actuel les aide à imaginer des activités qui pourraient être proposées aux étudiants.

Vers d'autres disciplines

En allant beaucoup plus loin, la structure du laboratoire pourrait être transposée vers d'autres disciplines. Mais pour cela, il faudrait nous allier d'autres compétences...

Chapitre II

Une activité particulière dans le laboratoire ARCADE:

le jeu



Chapitre II. Une activité particulière dans le laboratoire Arcade: le jeu

Nous voulons que les activités proposées dans le laboratoire soient plaisantes. Nous souhaitons même aller plus loin et proposer certaines activités sous la forme de jeux. Evidemment, au premier abord, ça ne semble pas très sérieux. Le jeu peut-il apporter quelque chose dans un apprentissage, et surtout peut-il avoir un apport, et lequel dans l'apprentissage de la programmation ? Pour répondre à ces questions il nous a fallu remonter un peu plus en amont et nous plonger un peu dans l'univers "jeu".

Ce chapitre repose sur l'investigation que nous avons faite dans ce domaine. Nous présentons tout d'abord une caractérisation de l'activité de jeu, une typologie des jeux, différentes définitions ainsi que des résultats opératoires pour la suite de ce travail, notamment en ce qui concerne les jeux sur ordinateur. Nous abordons aussi la délicate question du rôle pédagogique du jeu. Après une revue bibliographique du sujet, nous précisons la position que nous défendons dans notre contexte.

La notion de jeu est universelle. De tout temps, en tout lieu, on joue. Le jeu a pourtant longtemps échappé aux préoccupations des historiens qui maintenant encore s'intéressent davantage aux jouets qu'aux jeux. L'origine des jeux, le plus souvent rituelle ou religieuse, se perd dans la nuit des temps (jeu de la poupée, jeu de marelle du 4ème millénaire avant J.C...). Les jeux se sont transmis à travers les siècles, ou à travers le monde. Certains sont restés intacts, d'autres ont évolué dans le principe, dans la forme ou avec les progrès techniques, d'autres encore ont complètement disparu, sans qu'on puisse vraiment connaître les critères de succès ou d'échec. Pour plus de détails, le lecteur pourra consulter [Caire-Jabinet 81] ou [Caillois 67b].

Le jeu est rarement affaire de solitaire. De nombreux jeux nécessitent partenaires, adversaires, spectateurs, supporters, voire structures et sont tout à fait insérés dans la vie quotidienne. Sociologues et ethnologues se sont donc intéressés au jeu et ont tenté d'étudier la liaison entre les jeux et la culture d'une société. Caillois, dans "Les jeux et les hommes" [Caillois 67a] constate que les ressorts qui nous poussent à jouer (besoin de s'affirmer, goût du défi, poursuite de la chance, joie d'improviser...) se retrouvent dans la vie réelle et font fonctionner la société. Il affirme que culture et jeux sont intimement liés et que chacun agit sur l'autre. Pour lui, "expliquer les jeux par la culture, et la culture par les jeux sont deux opérations complémentaires, également fécondes, si elles ne se veulent pas exclusives". Le jeu serait donc facteur et image des cultures. Cependant, la multitude et la diversité des jeux présents simultanément dans une culture font qu'il est difficile d'établir le diagnostic d'une civilisation à partir de ces jeux. D'autant plus que certains jeux développent les valeurs institutionnelles alors que d'autres servent de défoir à des instincts non reconnus.

Parallèlement, les éthologues (citons Tinbergen, Lorenz, Chauvin) constatent la présence du jeu chez les espèces animales en milieu naturel. Les auteurs s'accordent sur le fait que le jeu soit en relation étroite avec le degré d'organisation de l'espèce, les animaux s'adonnant d'autant plus au jeu qu'ils occupent un rang élevé dans la classification zoologique. Chaque espèce possède en règle générale ses jeux caractéristiques. Les jeux peuvent être très élaborés et importants en nombre, ils sont aussi variés (combats amicaux, jeux de cache-cache, imitations, ...). Le jeu s'observe davantage chez les jeunes animaux. Sous son apparente gratuité, il a alors une utilité fondamentale pour la survie de l'individu et de l'espèce.

Dans une perspective plus individuelle, les psychologues étudient le jeu en tant qu'activité dans une perspective épistémologique ou diagnostique; ils peuvent également l'utiliser à des fins de thérapie ou d'expression [Guéraud 86b].

Plus proches de nous, les mathématiciens s'intéressent aussi au jeu, que ce soit pour l'étude des probabilités, ou par le biais des récréations mathématiques. En ce qui concerne la théorie mathématique des jeux, il s'agit d'une analyse mathématique qui s'intéresse aux problèmes de décision dans des situations où il y a conflit [Vajda 68], [Dami 75]. La théorie repose sur deux postulats: la possibilité d'une information totale (qui épuise les données utiles) et la concurrence d'adversaires dont les initiatives sont toujours prises en connaissance de cause, dans l'attente d'un résultat précis. Ceci ne se rencontre pas dans la vie réelle, pas plus que dans l'univers séparé des jeux. Pour ces raisons, si nous reconnaissons à cette théorie un grand intérêt mathématique et des applications notamment dans le domaine de l'économie, nous ne pensons pas qu'elle puisse nous être utile dans notre approche du jeu.

Le domaine du jeu est donc vaste, et intéresse de nombreuses disciplines. Il nous a fallu nous frayer un chemin dans ce domaine. Les questions qui ont guidé notre progression sont de deux ordres. Les premières tournent autour du jeu seul: qu'est-ce qui caractérise une activité de jeu ? Comment se repérer parmi la multitude de jeux existants ? Existe-t-il une classification ? Qu'est-ce qu'un jeu de rôle ? Que penser des jeux sur ordinateur ?... Les secondes mettent en jeu deux activités, le jeu et l'étude, et cherchent à savoir ce qui peut les lier ou les opposer.

I.1. L'activité de jeu

I.1.1. Définitions et caractéristiques

Nous avons vu que la notion de jeu est universelle, et qu'elle intéresse, chacune à sa façon, de nombreuses disciplines. Mais interrogeons-nous maintenant sur ce que révèle ce petit mot "jeu", par sa définition et ses différents emplois. Car "si l'on

peut dire que l'enfant joue, que le musicien joue, que le bois joue, cette identité de termes ne doit-elle pas être posée comme révélatrice d'une parenté de sens ?" [Henriot 76].

Définitions

Le terme jeu désigne avant tout une "activité physique ou mentale purement gratuite, qui n'a, dans la conscience de celui qui s'y livre, d'autre but que le plaisir qu'elle procure" (Petit Robert). Cette activité n'a pas de conséquence pour la vie réelle comme le traduit l'expression "ce n'est qu'un jeu". Le jeu est plaisant, il peut aller jusqu'à passionner, obstiner... on peut se piquer au jeu ! (attention, si on se prend au jeu, on ne joue plus).

Jeu désigne aussi cette activité, lorsqu'elle est organisée selon un certain nombre de règles, arbitraires, impératives et sans appel. Il faut "jouer le jeu" (même le tricheur joue le jeu car il feint de respecter les règles) ou ne pas jouer du tout. Le jeu est dans ce cas limité par des règles. Il a souvent d'autres limites, qui le séparent de la réalité: limites dans le temps, ou dans l'espace (terrain de jeux, table de jeux, maison de jeu) et il ne s'agit pas d'être hors-jeu !

Le même terme désigne ce qui sert à jouer (jeu de cartes, jeu d'échecs...) ; par extension, il s'agit d'un appareillage complet auquel on ne peut enlever ou rajouter aucun élément sans fausser le jeu (jeu de voile, jeu d'orgue). Ce qui révèle une notion de totalité fermée, complète au départ et immuable.

Le jeu signifie aussi la manière dont on joue, le style d'un interprète; le joueur est lié par des règles, l'interprète par un texte ou une partition, mais ils n'en demeurent pas moins libres de manifester leur personnalité. On peut "jouer franc jeu" ou "double jeu", "jouer serré" ou "jouer au plus fin"; l'acteur peut avoir un

jeu sobre, ou pathétique; on parle aussi du jeu de jambes d'un boxeur ou du jeu de mains d'un pianiste...

Cette idée de latitude, de facilité de mouvement, de liberté utile apparaît clairement lorsqu'on parle du jeu d'un engrenage. Il y a à la fois assemblage et parfaite horlogerie, élasticité et marge de mouvement.

Une idée de chance est aussi liée au jeu, comme en témoigne par exemple l'expression "avoir beau jeu". Parallèlement apparaît la notion de risque avec les locutions "mettre en jeu", "jouer gros jeu", "jouer sa carrière"; On prend le risque si "le jeu en vaut la chandelle".

Nous nous intéressons maintenant au jeu en tant qu'activité. La question que nous nous posons ici est de savoir si la définition donnée plus haut suffit pour reconnaître à coup sûr l'activité de jeu ? Qu'est-ce qui caractérise vraiment l'activité de jeu ? Comment la reconnaître ? Est-ce à l'air hilare du supposé joueur ?

Caractéristiques

Il n'y a jeu en fait que si la personne qui agit a conscience de jouer. Il est donc difficile pour un observateur de se rendre compte véritablement. Il tentera de le faire en se mettant à la place du joueur, mais ne possède pas pour cela de critères objectifs. En sont témoins la difficulté de reconnaître parfois le jeu dans l'activité d'un étranger par exemple, ou encore dans l'activité d'un animal.

Le joueur a **conscience** du fait qu'il joue. Même chez les animaux, on peut remarquer une certaine conscience du jeu; le chat, par exemple, est capable de faire patte de velours lorsqu'il joue avec vous. Si une personne n'a pas conscience de jouer, elle ne joue pas; qu'elle ait un jeu de cartes dans les mains par exemple, ne

change rien. Si le joueur perd cette conscience à un moment donné, il croit alors à la réalité de ce qu'il est en train de vivre et ne joue plus.

Ceci ne signifie pas que le joueur doit rester à l'écart du jeu, sans s'y impliquer. Son **engagement** est au contraire nécessaire au jeu. Il doit s'engager suffisamment pour entrer dans le jeu et en même temps, sous peine d'en sortir, il doit conserver un certain recul par rapport à son jeu: "ce n'est qu'un jeu".

Mais s'il est vrai que "le jeu n'est pas dans la chose mais dans l'usage qu'on en fait" [Henriot 76], il n'en reste pas moins que certaines choses permettent qu'il y ait activité de jeu et d'autres non. La question est donc de savoir ce qui caractérise une activité-jeu potentielle et Roger Caillois, dans son livre "les jeux et les hommes", nous donne une réponse [Caillois 67a].

Un reflet des principales caractéristiques du jeu est déjà apparu en évoquant les différents emplois de ce mot. Nous avons vu qu'on ne joue que pour son plaisir, et de son propre gré. Le jeu doit être une activité **libre**; on entre dans le jeu parce qu'on le désire, on sort du jeu quand on le veut (au risque toutefois de mécontenter ses éventuels partenaires, de donner une mauvaise image de soi, c'est le "pouce" des enfants).

Le jeu offre des limites d'espace et de temps, et est ainsi maintenu à l'écart de la vie réelle, il est **séparé**. Cette frontière entre la vie réelle et le jeu doit être particulièrement nette, sous peine de ruiner la nature même du jeu: les enfants qui jouaient "à la bagarre" deviennent violents, méchants et envoient des coups bas, celui qui se déguise croit à la réalité du rôle et c'est l'aliénation... Si l'on se prend au jeu, si l'on croit à la réalité de ce que l'on devrait seulement jouer, il n'y a plus de jeu.

La définition première du jeu contient l'idée de gratuité. Le jeu est une activité **improductive**. Il est occasion de dépense pure (temps, énergie, ingéniosité, adresse). C'est aussi vrai dans les jeux de hasard et les paris où il n'y a pas création d'argent mais simplement déplacement. Chaque partie repart sur les mêmes bases. Seule l'habileté du joueur a pu évoluer. Notons que si le jeu ne produit rien d'extérieur à lui-même, il n'en modifie pas moins les qualités ou les aptitudes du joueur. Et en quelque sorte "le joueur est l'œuvre de son jeu" [Henriot 76].

Le jeu doit aussi être une activité **incertaine**: lorsque l'issue ne fait plus de doute, le jeu s'arrête (on abat les cartes). Il faut un renouvellement constant et imprévisible de la situation sans quoi il ne peut plus y avoir de jeu. Celui-ci consiste justement à inventer une réponse à la situation qui se présente. L'incertitude doit rester même si l'on s'entraîne pour avoir les meilleures chances de gagner. La victoire doit être accessible mais incertaine. Le risque doit demeurer.

L'activité de jeu est **réglée**, par des règles précises, arbitraires mais irrécusables (échecs, bridge, ballon prisonnier...) ou **fictive**, dans ce deuxième cas, on se conduit "comme si", c'est ce qui tient lieu de règle (jeux de rôle, imitation, "tu serais l'élève et moi la maîtresse"...).

Il faudra donc nous assurer que les activités que nous proposons sous forme de jeux respectent ces caractéristiques (activité libre, séparée, improductive, incertaine, réglée ou fictive) sans lesquelles il ne peut y avoir de jeu. Il s'agit d'organiser l'activité pour que le jeu puisse naître.

II.1.2. Typologie des jeux

Considérons maintenant qu'il n'y a pas un jeu mais des jeux; la multitude et la variété des jeux existants donnent un peu le vertige et font ressortir le besoin d'une **classification**. Si nous pouvons nous appuyer sur une telle classification, nous pourrions essayer de voir quels types de jeux proposer dans notre laboratoire.

Il n'est pas simple de découvrir un principe de classement. De multiples points de vue sont possibles, et le vocabulaire courant emploie en fait plusieurs classifications concurrentes. On parle en effet de jeux de cartes, de jeux d'adresse, de jeux de plein air, de jeux de société, de jeux d'enfants... mélangeant du même coup les critères de répartition: instrument du jeu, qualité requise, lieu du jeu, nombre de joueurs, âge des joueurs...

Parmi les différentes typologies rencontrées ([Aveline 61], [Caillois 67a], [Caire-Jabinet 81]), celle proposée par Caillois a retenu notre attention car elle regroupe les jeux selon le type de comportement induit chez le joueur. Ce critère peut effectivement nous permettre de situer les jeux par rapport aux comportements visés chez l'étudiant en programmation.

Caillois [Caillois 67a] définit ainsi 4 catégories fondamentales: compétition (âgon), hasard (alea), simulacre (mimicry) et vertige (ilinx).

Les noms d'origine étrangère indiqués ici entre parenthèses sont les noms employés par l'auteur. Celui-ci précise qu'il a emprunté à telle ou telle autre langue le vocable le plus significatif, afin d'éviter que chaque ensemble se trouve marqué par la qualité particulière d'un des éléments qu'il réunit, et que les termes prêtent ainsi à confusion.

- **compétition** (âgon): il s'agit d'un combat où l'égalité des chances est artificiellement créée. La rivalité s'exerce sur une qualité précise, dans des limites définies; le joueur ne doit compter que sur ses propres capacités. L'âgon implique discipline et persévérance. Les jeux de compétition incluent les compétitions sportives mais aussi le jeu d'échecs, l'alpinisme, les mots croisés...

- **hasard** (alea): La victoire est ici fondée sur une décision indépendante du joueur. Le joueur compte sur tout, excepté sur lui. Il risque un enjeu puis il ne fait qu'attendre, dans l'espoir et le tremblement, l'arrêt du sort. Citons les jeux de dés, la roulette, la loterie, le loto...

- **simulacre** (mimicry): le jeu consiste à devenir soi-même un personnage illusoire et à se conduire en conséquence. Dans le cadre du jeu, le plaisir est d'être autre ou de se faire passer pour autre, avec toute la maîtrise et l'art que cela nécessite. Le carnaval, les jeux de rôle, tous les jeux où l'enfant imite l'adulte, etc, font partie des jeux de simulacre.

- **vertige** (ilinx): ces jeux reposent sur la poursuite du vertige. Il s'agit de s'abandonner à une sorte d'étourdissement qui anéantit la réalité. Le trouble que provoque le vertige est recherché pour lui-même: c'est notamment la cas du toboggan, des manèges, de la balançoire, et des nombreuses machines à vertige que l'on trouve dans les parcs d'attraction.

Naturellement de nombreux jeux combinent entre eux, à des degrés divers, ces différents ressorts. Toutes les alliances ne sont cependant pas possibles, toutes n'ont pas la même force. La maîtrise de soi réclamée par l'âgon par exemple ne peut se combiner avec la recherche du vertige. Par contre beaucoup de jeux combinent âgon et alea, faisant appel à la fois au hasard et aux qualités du joueur. C'est la cas par exemple de la plupart des jeux de cartes.

La typologie présentée par Caillois va en fait plus loin et s'intéresse, selon un deuxième axe, à la façon dont on joue à l'intérieur d'un même type de jeu. Prenons un exemple pour illustrer cela. Il y a mille façons de courir par jeu: du "allez on court", sans règle, sans but précis, pour le bonheur de l'agitation (paidia); en passant par "le dernier arrivé a perdu" où le jeu est encore improvisé mais offre un but précis; jusqu'au "je cours le 100m dans la compétition, mardi prochain" qui suggère des règles précises, un entraînement (ludus).

C'est sur cette typologie que nous avons choisi d'appuyer la suite de notre réflexion. Elle sera utilisée soit pour classifier des jeux existants, soit pour déterminer les types de jeu que nous souhaitons proposer dans le laboratoire Arcade.

II.1.3. Les jeux de rôles

Les jeux de rôles font partie des jeux de simulacre définis plus haut. Pourquoi s'intéresser particulièrement à eux ? Si les jeux de rôles font depuis longtemps partie de l'univers des psychologues [Mucchielli 83], un plus large public est actuellement en train de découvrir ces jeux, comme en témoignent les revues spécialisées de jeux. Mais en dehors des personnes qui pratiquent le jeu de rôles, peu dans le grand public savent ce que c'est.

"De nombreux joueurs même parmi les habitués des jeux les plus complexes, demeurent résolument méfiants à l'égard des jeux de rôles. Sans doute parce qu'ils craignent de s'égarer dans une activité aux règles obscures et ésotériques, réservée aux seuls initiés. Pourtant les jeux de rôles... ce n'est pas sorcier" nous expliquent Thierry Paunin, Pierre Rosenthal et Gildas Sagot dans Jeux et Stratégies [Paunin 86].

Il s'agit de rassembler **un certain nombre de participants** (au minimum 2, maximum 10-12). N'importe qui peut participer à un jeu de rôles, sans distinction d'âge, aucune connaissance particulière n'est nécessaire. Le jeu de rôles est essentiellement un **jeu de communication**.

L'un des joueurs anime la partie: c'est le **meneur de jeu**. Il va faire jouer un scénario qu'il s'est procuré tout fait ou qu'il a lui-même conçu. A lui de décrire la situation initiale aux autres joueurs, et de leur exposer la mission à remplir. Ses talents de conteur et de comédien contribuent à ce que les joueurs s'imaginent réellement dans un monde autre. Tout au long de la partie, il veille à l'application des règles, il répond aux joueurs en leur indiquant les conséquences de leurs actions; il doit être capable d'arbitrer, d'animer et d'improviser tout à la fois.

Chaque participant a un **rôle à tenir**. Il va d'abord créer le **personnage** qu'il va incarner, en lui attribuant un certain nombre de caractéristiques (caractéristiques physiques, capacités, motivations, profession, talents...). Toutefois, certains jeux de rôles sont inspirés de personnages connus du cinéma, de la littérature ou de la BD: il ne s'agit pas alors de créer le personnage mais de faire jouer James Bond ou Superman ! Dans tous les cas, le joueur doit entrer dans la peau de son personnage. Donnez-lui vie, ne craignez surtout pas d'entrer dans des détails superflus, inventez-lui des tics, des lubies, un accent... nous conseillons [Paunin 86]. Agissez en fonction des caractéristiques et de la psychologie de votre personnage. Sans quoi le jeu perd un de ses principaux attraits: le meneur de jeu doit y veiller.

Si la pratique du jeu de rôles fait un large appel à l'imagination, on ne fait pas plus "n'importe quoi" dans un jeu de rôles qu'au go ou au bridge. Le jeu de rôles possède **des règles**. Certaines concernent la création des personnages; d'autres précisent le monde dans lequel se passe le jeu; enfin les règles de simulation

permettent de gérer la partie et les événements divers. Ces règles doivent être parfaitement connues du meneur de jeu. Par contre, il n'est la plupart du temps pas indispensable que les autres joueurs les connaissent.

Le scénario est la base de l'aventure que vont vivre les joueurs. La plupart des jeux de rôles incluent un scénario d'initiation; le meneur de jeu peut avec l'expérience l'agréments ou en créer de nouveaux. Classiquement, un scénario comprend un historique exposant le contexte, l'atmosphère, ainsi que des éléments précis de l'aventure; une introduction plaçant les joueurs dans ce contexte, avec un certain nombre d'informations; la description des personnages secondaires; une chronologie ainsi qu'une description complète des lieux incluant les conséquences des actions que pourront y entreprendre les joueurs.

Quel est le but du jeu ? Il s'agit avant tout de jouer le rôle d'un personnage et de vivre à travers lui une aventure. Mais au delà, chaque scénario propose aux joueurs un but précis, une mission à accomplir. L'objectif est collectif. Ce sera au meneur de jeu d'imposer un juste milieu entre l'attitude la plus collective qui risque d'engendrer des parties un peu trop sages et le chacun pour soi qui tournera au règlement de compte permanent.

Les jeux de rôles sont de plus en plus sophistiqués. La traditionnelle table de jeu ne suffit plus aux joueurs ! Certains passent de la table à la rue, à la ville et déjà au pays et même à plusieurs pays. Un vieux château, deux ou trois meneurs de jeu, une demi-douzaine de personnages non joueurs ayant un rôle fixé, un scénario et deux ou trois groupes de cinq ou six aventuriers, déguisements et matériel... sont les ingrédients d'une aventure qui peut s'étendre sur un ou deux jours (Jeux & Stratégie n°43 et n° 45)... N'y a-t'il pas cependant un risque d'aller trop loin, en se coulant trop dans la réalité, détruisant du même coup l'univers protégé qui permet le jeu ? N'est-ce pas le cas de ce "jeu" où le but est de se rendre le plus rapidement

possible dans le village espagnol, italien ou portugais tiré au sort, le tout sans argent ni papiers bien sûr !

Certains jeux de simulation sont en fait des jeux de rôles. C'est le cas notamment des jeux d'entreprise [Mucchielli 83]. Les jeux reposant sur une simulation comportent selon D. Vitale [Vitale 84], trois grandes familles:

- les "wargames" qui mettent deux camps face à face, dans un duel sans merci; ce genre de duel s'inspire le plus souvent d'un conflit historique, mais il peut aussi apparaître dans le domaine du fantastique ou de la science-fiction.
- les jeux de pouvoir, où les affrontements et les alliances successives entre plusieurs protagonistes ne laisseront finalement qu'un seul vainqueur.
- les jeux de rôles, où le succès de l'aventure commune dépend avant tout d'une coopération exemplaire entre les personnages qu'animent les joueurs.

En vingt ans, souligne D. Vitale, ces jeux ont déjà exploré un nombre impressionnant de mécanismes ludiques inédits. Cette nouvelle génération de jeux a remis à jour l'essence même de l'activité ludique humaine: l'alliance de l'intelligence et de l'imagination.

II.1.4. Les jeux sur ordinateur

Nous utiliserons ici l'appellation "jeu sur ordinateur" pour tout jeu dont le support est l'ordinateur alors que l'appellation "jeu vidéo" sera plutôt réservée aux jeux dont le support est une console de jeu, ou un écran de télévision.

Selon Gros et Pernelet, les jeux vidéo représentent économiquement, technologiquement et sociologiquement une grande aventure, tout ce qu'il y a de plus sérieuse [Gros 83]. Les jeux vidéo ont en fait été les précurseurs des jeux sur ordinateur. Au départ, on a surtout assisté à une transposition des jeux vidéo vers le

support ordinateur. Ceci permettait d'exploiter immédiatement "des histoires toutes prêtes, déjà testées auprès du grand public, aux graphismes définis et aux options et déclinaisons cernées" [Chazoule 84]. Parallèlement de nouveaux jeux ont été trouvés, utilisant les possibilités nouvelles offertes par les ordinateurs; notamment des jeux d'aventure et de stratégie ont été développés.

Le premier intérêt des jeux sur ordinateur est de démythifier l'ordinateur aux yeux du grand public. L'avènement du micro-ordinateur y est pour beaucoup: l'ordinateur est devenu accessible à la famille, son usage banalisé. L'ordinateur familial est le plus souvent utilisé pour jouer [Chazoule 84]. Nous souhaitons maintenant essayer de comprendre ce qui rend les jeux sur ordinateur si attrayants.

C'est ce que tente de faire T.W. Malone dans sa thèse [Malone 80]. Bien qu'il note de grandes différences dans les préférences des gens, il parvient à la conclusion que pour intéresser les joueurs, un jeu sur ordinateur doit:

- présenter un défi:

Pour cela, il faut que le jeu propose des buts que le joueur n'est pas sûr d'atteindre. Dans cette optique, un jeu peut présenter des buts de difficulté ou bien de niveau variables(par exemple un niveau peut porter sur l'adresse considérée, un autre niveau peut porter sur la rapidité avec laquelle le but est atteint); il doit alors être facile pour le joueur de se générer des buts de difficulté appropriée. Le jeu peut aussi utiliser le hasard ou bien cacher délibérément de l'information qui sera révélée à petites doses.

Dans tous les cas, au cours de la partie, le joueur doit constamment être tenu au courant de sa performance et de l'écart qui le sépare encore du but à atteindre.

Malone note encore que les buts à court terme sont supérieurs aux buts à long terme pour maintenir l'intérêt et susciter la performance et que les buts discontinus motivent davantage les joueurs.

- utiliser une **fantaisie**:

Les jeux doivent montrer ou évoquer des objets physiques ou des situations sociales qui peuvent varier du complètement possible au complètement impossible.

Malone distingue les fantaisies intrinsèques des fantaisies extrinsèques. Une fantaisie extrinsèque dépend de l'adresse mais l'adresse ne dépend pas de la fantaisie. Citons le jeu du pendu utilisé par exemple en arithmétique: une réponse fausse (ou juste) influe sur le monde fantaisiste en ajoutant (ou pas) un élément dans la pendaison; par contre savoir ou non calculer est totalement indépendant de la représentation du pendu.

Prenons tout de suite un exemple pour comprendre ce que peut être une fantaisie intrinsèque. Le jeu "Darts" consiste à donner la fraction correspondante à une position donnée sur un segment unité, position marquée par un ballon à crever; la réponse est symbolisée par une flèche qui atteint et crève, ou non, le ballon; cette visualisation indique donc dans quelle mesure la réponse est inexacte (beaucoup trop haut, un peu trop bas...). Ici, la fantaisie permet de donner visuellement le but, les actions du joueur ont donc un effet direct et logique dans la fantaisie, la fantaisie dépend de l'adresse, et l'adresse dépend de la fantaisie. Il s'agit d'une fantaisie intrinsèque. Les problèmes sont présentés en termes des éléments du monde fantaisiste choisi. Les réponses sont reflétées dans le monde fantaisiste, qu'elles soient justes ou fausses ainsi que la manière dont elles sont éventuellement fausses.

Il est préférable d'utiliser une fantaisie intrinsèque. Comme nous venons de le voir, le grand intérêt d'une telle fantaisie est de présenter visuellement un but, de répondre à l'action du joueur par un effet visuel à la fois logique et constructif lui donnant des éléments pour améliorer ses performances. Une telle fantaisie peut être inventée soit en simulant une situation où l'adresse est utilisable, soit en inventant une situation qui implique une analogie avec l'adresse considérée.

Malone insiste sur l'importance des aspects émotionnels de la fantaisie: la fantaisie utilisée doit plaire au joueur. Il serait donc souhaitable de pouvoir adapter la fantaisie au goût et à l'humeur du joueur considéré. Ce qui en tous cas présente un défi certain pour les concepteurs; reste à savoir si le niveau de difficulté est approprié !...

- éveiller puis satisfaire la **curiosité** du joueur:

La curiosité est activée par la nouveauté, la complexité, l'effet de surprise, l'incongruité, le manque apparent de cohérence... On peut distinguer en fait deux sortes de curiosité: la curiosité sensorielle et la curiosité cognitive.

La curiosité sensorielle s'attache aux évènements techniques, changement dans la lumière, le son, changement de plan... Ces évènements attirent notre attention, quelque soit le contenu proposé. Les effets audio-visuels, peuvent intervenir comme décoration, ce qui augmente en général l'intérêt initial du jeu mais peut devenir ennuyant par la suite; il serait donc intéressant que de tels effets soient optionnels. Les effets audio-visuels peuvent aussi intervenir comme renforcement de la fantaisie, comme récompense, ou comme système de représentation et c'est sans doute leur meilleure utilisation puisqu'ils permettent alors de donner une information beaucoup plus efficacement qu'avec des nombres ou du texte.

La curiosité cognitive: il s'agit du désir d'améliorer ses structures de connaissance. Pour éveiller celle du joueur il est souhaitable de lui présenter juste assez d'informations pour que sa vision des choses lui semble incomplète, non cohérente ou qu'il ne découvre pas de loi simple rendant compte de leur diversité. Pour maintenir cette curiosité éveillée, l'information renvoyée au joueur doit le surprendre (au moins à prime abord). Mais s'il est important de montrer des choses nouvelles et surprenantes, il ne faut pas bien sûr qu'elles soient absolument incompréhensibles: l'information donnée doit toujours être constructive.

Ces informations pratiques sur ce qui fait un "bon" jeu sur ordinateur nous seront d'une grande utilité, notamment pour comprendre le ressort de quelques jeux que nous étudierons, et pour nous guider dans la création de jeux pour notre laboratoire.

II.2. Le rôle pédagogique du jeu

II.2.1. Une vue bibliographique de la question

Il semble globalement admis que le jeu a une importance fondamentale dans le développement du jeune enfant. Le jeu revêt une valeur adaptative chez l'enfant comme chez l'animal d'ailleurs: il est une préparation à la vie.

Lorsqu'il s'agit d'enfants en âge scolaire, le rôle pédagogique du jeu est plus controversé, les opinions et les expériences divergent. Citons-en quelques-unes.

Selon **Freinet**, il faut exclure le jeu de toute stratégie éducative sérieuse. En effet pour lui, l'enfant mérite mieux que le jeu [Chateau 54]. Il faut proposer à celui-ci du "travail-jeu", c'est à dire un travail donnant autant de satisfaction que le jeu, plutôt que du "jeu-travail" où le jeu prend la place du travail [Leif 76].

Claparède propose que le jeu soit une clef de voute de l'école active parce qu'il voit le jeu comme pré-exercice [Chateau 54]. **Montessori**, suivie de **Audemars** et **Lafendel**, est une des premières à utiliser le jeu dans ses classes maternelles et d'arriérés mentaux. L'utilisation du jeu à l'école est issue de ces expériences faites dans le cadre de l'enseignement spécialisé [Medici 69].

Chateau a une position plus nuancée [Chateau 54]. Pour lui, le travail scolaire doit dépasser le jeu, et rester à mi-chemin entre jeu et travail social. Le jeu a cependant un grand intérêt pédagogique et éducatif, car il est épreuve et habitue l'enfant à remplir une tâche, il lui apprend à respecter certaines consignes, à fournir un effort et à avoir une activité de groupe.

"Qui dit jeu dit à la fois effort et liberté". Jouer, c'est chercher une jouissance morale. C'est cette jouissance que **Chateau** veut transporter dans son éducation. Pour cela il faut présenter à l'enfant des obstacles à franchir, et des obstacles qu'il veuille franchir. Il s'agit donc de présenter la tâche scolaire comme désirable, comme un obstacle à franchir librement; et ce n'est pas un mince problème.

L'inconvénient du jeu est qu'il reste hors du temps et de l'espace réel; le groupe de jeu se suffit à lui-même. **Chateau** conclut donc qu'une éducation basée uniquement sur le jeu reste insuffisante et que le travail scolaire doit être un pont jeté du jeu au travail (mais dans les grandes classes il sera proche du travail).

Leif et **Brunelle** défendent le jeu comme devant être une discipline en soi [Leif 76]. Le jeu est pour eux un réservoir de créativité et de spontanéité. "Les jeux préparant à la vie de demain devront développer essentiellement l'initiative individuelle. L'enseignement est dans une grande mesure très insuffisant pour préparer à l'avenir... Le jeu en faisant appel à l'activité personnelle, permet une

prospection de l'avenir en invitant l'enfant à explorer toutes ses possibilités et pas seulement celles choisies par la société adulte actuelle".

Les réflexions ou expériences précédentes concernent plutôt les enfants en âge scolaire. On trouve peu de littérature générale sur la valeur pédagogique du jeu dans les formations d'adultes (études universitaires, formations continues...), comme si le jeu dans ce contexte restait suspect, voire tabou.

Brisant ce silence pudique, **Béville** présente une véritable plaidoirie en faveur du jeu dans la formation [Béville 86]: "Si les hommes accomplissent les plus grands exploits pour leur plaisir, pourquoi ne pas en faire officiellement un objectif?".

Il en présente les différents **usages** possibles. Le jeu pédagogique peut permettre au formé d'acquérir un **savoir**, en lui permettant "de retrouver des connaissances qu'il personnalise et de faire, éventuellement, des découvertes"; il peut aussi permettre de vérifier son savoir, ou plutôt en allant un peu plus loin, la mise en pratique de ce savoir. Le jeu pédagogique a aussi un intérêt particulier dans l'apprentissage ou la pratique d'une **méthode**; une méthode ne prend un sens réel que lorsque l'on sait en quelque sorte la dépasser, en étant aussi capable de prendre en compte la particularité d'une situation et peut-être de passer outre. Ceci introduit l'incertitude, incertitude que le jeu sait gérer. Enfin, le jeu pédagogique peut intervenir dans la formation de la **personnalité**, avec un éveil des facultés intellectuelles, une formation à la prise de décision, un développement de l'imagination, un développement de la volonté et d'autres traits de caractère en plaçant les personnes dans des situations appropriées (situations de rapidité, situations anormales "l'improvisation ne s'improvise pas"...).

Selon Béville, il n'y a probablement pas de jeux distractifs qui ne soient instructifs, mais pour qu'ils remplissent ce rôle il faut:

- premièrement reconnaître l'intérêt du jeu et l'expliciter,
- deuxièmement faire reconnaître cet intérêt par les formés, formateurs, responsables.

Il dégage les avantages et les inconvénients du jeu pédagogique, les facilités et les difficultés de sa mise en oeuvre, son efficacité et les déviations possibles (tableau reproduit ci-dessous). Soulignant le caractère positif de la motivation, du réalisme et du concret des situations, des expériences faites en commun, de l'attitude active, il met cependant en garde contre la difficulté de trouver le jeu adapté et d'en maîtriser les apports, et contre la prédominance du "jouer" sur le "apprendre".

D'autre part, nous avons vu que T.W. Malone dans sa thèse [Malone 80], nous apporte des éléments sur les caractéristiques qui font qu'un jeu sur ordinateur attire et satisfait les joueurs: présentation d'un défi clair et de difficulté appropriée, utilisation d'une fantaisie dite intrinsèque cad liée étroitement à l'habileté concernée, éveil et satisfaction de la curiosité du joueur.

T.W. Malone montre ensuite qu'il y a en général correspondance dans la poursuite simultanée des deux objectifs suivants: apprendre dans un environnement motivant et jouer dans ce même environnement.

Par exemple, il est souhaitable, pour augmenter le défi, de proposer des niveaux de difficulté variable; ceci augmente du même coup le potentiel d'apprentissage.

De même, les fantaisies intrinsèques utilisées dans un jeu stimulent l'intérêt; elles présentent aussi l'avantage de susciter chez le joueur des analogies fructueuses pour son apprentissage. Ainsi les fantaisies utilisées permettent souvent d'indiquer comment utiliser l'habileté concernée dans la vie réelle; elles peuvent aussi, par les

Evaluation du jeu pédagogique selon G.Béville

Avantages

Motivation (plaisir)
 Expérience personnelle vécue
 Situation dynamique
 Méthode active par essais-erreurs
 Auto-formation
 Auto-évaluation
 Aspect concret, visuel
 Réalisme (simulation)
 Adaptation aux évolutions technologiques
 Préparation à l'action en milieu incertain

Facilités

Possibilité de moduler en fonction des participants, du temps, du lieu
 Polyvalence
 Possibilités d'encouragement (compétition, coopération, souvenirs, récompenses)
 Travail en groupe
 Moyen de communication
 Expérience en commun

Efficacité

Rapport qualité / prix : excellent
 Rapport résultat / temps : douteux
 Rapport jeu / autres moyens : variable

Inconvénients

Evaluation subjective
 Possibilité de refuser de jouer
 Manque de répétition, de recul
 Nécessité pour le formateur, d'avoir joué au préalable
 Peur de l'incertitude

Difficultés

Manque de temps pour les commentaires
 Objectifs, résultats secondaires souvent plus importants que les buts premiers
 Complexité sous-jacente
 Difficulté à trouver, à créer le jeu adapté
 Contradiction entre les souhaits exprimés "pas d'école" et le désir réel "pas de casse-tête"

Déviations

Discours au lieu du jeu
 Jeu aux dépens de la formation

analogies proposées aider le joueur-apprenant à utiliser d'anciennes connaissances pour comprendre de nouvelles choses. Enfin le monde fantaisiste peut proposer à l'étudiant des images concrètes, qui seront plus facilement mémorisées.

L'éveil de la curiosité, notamment cognitive, nécessaire dans le jeu sur ordinateur, a aussi un rôle déterminant dans l'apprentissage; en effet on ne peut apprendre que si l'on a envie d'en savoir plus.

Toutefois, certaines caractéristiques peuvent rendre un environnement plus drôle mais moins efficace du point de vue de l'apprentissage. Imaginons par exemple qu'un jeu utilise une fantaisie déclenchant une catastrophe spectaculaire lorsque le joueur échoue; le risque est alors que le joueur-apprenant ait pour but de voir cette catastrophe, s'efforçant de perdre plutôt que d'utiliser ses compétences de façon constructive.

Il faut aussi prendre garde aux fantaisies choisies dans les jeux utilisés à des fins pédagogiques; il convient de proposer des fantaisies attirantes pour la population ciblée sans quoi les élèves se désintéresseront du jeu.

Nous avons vu que le défi présenté doit être de difficulté appropriée et que l'information présentée en réponse aux actions du joueur doit le renseigner clairement sur sa performance. Cette information doit donc être claire et toujours constructive: il faut particulièrement prendre garde à ne pas décourager le joueur-apprenant.

II.2.2. Notre point de vue

A travers les réflexions et les expériences de ces auteurs, il nous est apparu acquis que le jeu pouvait avoir un apport pédagogique. Les positions respectives de

chacun de ces auteurs tiennent davantage à la place, importante ou non, à accorder au jeu dans un apprentissage. Les méthodes pédagogiques dites "nouvelles" utilisent le jeu comme moyen fondamental de développement de l'enfant et d'apprentissage alors que Freinet, dans la hantise que le jeu prenne la place du travail, veut proposer aux élèves du "travail-jeu", c'est-à-dire un travail donnant autant de satisfactions que le jeu.

Notre position n'a pas à être si tranchée. Il n'est pas question pour nous de remplacer les formations actuelles proposées aux étudiants par des formations où tout serait jeu ! Notre objectif n'est d'ailleurs pas de transformer ces formations. N'oublions pas que le laboratoire que nous voulons offrir aux étudiants représente quelque chose en plus, au côté des formations traditionnelles; et que, parmi les nombreuses activités proposées dans ce laboratoire, nous voulons en présenter certaines sous forme de jeu. Nous espérons ainsi que les étudiants profiteront de leur temps de loisir pour apprendre. En deça des grands débats, la question que nous nous posons est donc toute simple: nos étudiants peuvent-ils étudier en jouant ? Autrement dit, le jeu et l'étude sont-ils des activités compatibles ?

Lorsque le jeu est utilisé dans un apprentissage, le premier but du jeu n'est pas d'apprendre, néanmoins, en jouant, nous développons des compétences et nous apprenons quelque chose. L'apprentissage y est incident, sinon accidentel. Par contre, dans le cas de l'étude, l'apprentissage est intentionnel. En effet, c'est le but de l'étude d'apprendre, d'acquérir une connaissance, un savoir, une maîtrise sur un sujet précis, par l'observation de phénomènes, l'expérimentation, la confrontation d'idées, le raisonnement. Au delà de cette différence première d'objectifs, que penser de ces deux activités ?

Prononcez le mot jeu et demandez autour de vous les mots qui lui sont associés. Vous entendrez sans doute les qualificatifs : amusant, futile, libre, gratuit, puéril,

facile... alors que les mots étude ou travail auront un tout autre résultat: sérieux, important, obligatoire, difficile... De là à penser qu'il y a une opposition farouche entre ces deux activités, il n'y a qu'un pas. Selon l'opinion publique, le jeu est opposé à l'étude.

Mais revenons sur ces qualificatifs. Oui, le jeu est libre, gratuit, mais nous avons vu qu'il n'était pas facile (sinon il y a certitude et le jeu disparaît): il faut à la fois disposer d'un jeu à sa taille, mais qui demande un effort. L'expression "jeu d'enfant" indique une facilité pour un adulte qui ne l'utilise donc plus comme jeu. Il y a du plaisir dans le jeu, mais ce n'est pas un amusement. Le jeu serait-il puéril ? Non, les adultes jouent aussi. Quant à la futilité, nous avons vu que le jeu est un des principaux moyens d'adaptation de l'être humain à la vie.

Pour nous, jeu et étude présentent de nombreux points communs. Notons que nous utilisons ici les caractéristiques de l'activité d'étude présentées par S. Painvin dans son rapport "Voulez-vous apprendre avec moi" [Painvin 86], dont nous avons redonné les grandes lignes dans le chapitre I.

- Jeu comme étude sont séparés

Le jeu est séparé, c'est-à-dire qu'il ne doit pas être la vie réelle. Nous avons vu également [Painvin 86] que "le rôle de l'enseignement est de préparer à la vie, en créant des situations qui simulent celles de la vie sans se confondre avec elles" [Reboul 80].

- Jeu comme étude sont libres

Dans le jeu, il y a effort, difficulté mais elle est librement surmontée. Le jeu "s'arrête" si les joueurs ne veulent plus jouer. De même, on peut contraindre une personne à suivre un enseignement, mais on ne peut la contraindre à apprendre. Il faut un engagement libre de la personne pour qu'apprendre soit possible.

- Jeu comme étude sont improductifs

Le jeu ne crée rien. Il ne crée pas d'œuvre extérieure, sinon le joueur lui-même (le joueur est l'oeuvre de son jeu). Dans l'étude, ce qui est produit, ce ne sont pas non plus les solutions aux problèmes posés, mais bien le changement qui s'est fait chez l'étudiant.

- Jeu comme étude sont incertains

Le résultat du jeu doit toujours être incertain, comporter une part de risque. Dans l'apprentissage et l'étude, l'erreur est toujours possible, et même souhaitable (on apprend dans la vie comme à l'école, à partir de ses propres réussites et erreurs). Apprendre, c'est accepter de modifier une certitude ou une incertitude antérieures. C'est prendre le risque de changer.

- Jeu comme étude sont réglés ou fictifs

Dans les jeux réglés, il y a règlement et nombre fixe de partenaires. Dans les jeux fictifs, il y a fantaisie et improvisation. L'étude comporte également ces 2 aspects: règle sous la forme de programmes scolaires, d'exposés magistraux ou d'exercices, fiction sous la forme de création collective, de débats ou de jeux de rôle.

- Jeu comme étude supposent une relation avec un autre

Dans le jeu, l'importance de l'émulation est indéniable, il y a toujours présence ou évocation d'un autre, qu'il faut éblouir, battre, convaincre. L'activité "apprendre" implique toujours une relation interindividuelle, qu'elle soit à dominante intellectuelle, affective ou corporelle. Autrui peut prendre diverses formes: personne, livre, ordinateur, objets. On apprend et on joue avec autrui.

- Jeu comme étude supposent un engagement et une distance

Le joueur doit s'engager suffisamment pour entrer dans le jeu, mais ne doit pas perdre la conscience du fait que ce n'est qu'un jeu (sous peine de sortir du jeu). L'étudiant doit s'engager pour que l'apprentissage soit possible, et en même temps conserver assez de recul pour repérer et accepter l'erreur et en tirer profit, ainsi que pour élaborer sa propre pensée (notamment en ayant un esprit actif et critique).

- Jeu comme étude sont des activités subjectives

Il est difficile de savoir si une personne joue, car il n'y a pas de manifestations uniques de l'état de jeu: chaque joueur s'exprime à sa façon et le jeu suppose souvent des tensions, des efforts, et pas toujours une joie béate. Pour des raisons analogues, il est difficile de savoir si une personne est en train d'étudier. En outre, le plaisir (pour le jeu), la connaissance (pour l'étude) varient en cours de processus: il y a des moments où l'on ne joue plus, où l'on se fâche, et où l'on désapprend, où plus rien n'est clair parce qu'il faut restructurer le tout pour prendre en compte les nouvelles données.

- Jeu comme étude requièrent la motivation de celui qui joue ou étudie.

Le jeu ne devant pas être obligatoire, il ne se maintient que par le plaisir qu'on y prend. Il reste alors à la merci de l'ennui, de la satiété ou d'un simple changement d'humeur... N'est-ce pas vrai aussi de l'étude ? De plus, "même dans une technique ludique simple, il arrive que les participants soient tous prêts à jouer aux devinettes, mais beaucoup moins disposés à passer à la réflexion ou à l'abstraction" [Béville 86]. Ceci explique que de nombreux animateurs n'utilisent le jeu qu'en phase de sensibilisation et d'initiation. L'étude comporte également des paliers de difficultés la rendant inaccessible à beaucoup.

Nous pensons donc que les activités de jeu et d'étude sont tout à fait compatibles, et nous voulons, dans notre contexte d'étudiants, donner une place au jeu aux côtés de l'étude. Comme le dit Caillois "rien autant que le jeu n'exige d'attention, d'intelligence et de résistance nerveuse" [Caillois 67a]. Le joueur (apprenant) est généralement motivé; il pratique une méthode active et s'investit beaucoup dans l'activité, le jeu permettant une relation étroite entre la théorie et la pratique [Béville 86].

Les **bénéfices** que nous attendons en offrant des activités de jeu aux étudiants, sont donc de les motiver, de renforcer leur participation active, d'associer théorie et pratique, de créer des associations d'idée, des analogies, un vécu profitables à leur apprentissage. Nous nous appuyerons sur l'analyse de **R. Caillois**, en ce qui concerne l'activité de jeu: comment la reconnaître, comment la permettre; et nous utiliserons sa typologie. Nous tirerons profit des apports de **G. Béville** quant aux usages pédagogiques du jeu dans la formation. Enfin, nous utiliserons d'autant plus volontiers les résultats de **T.W. Malone** à propos des jeux sur ordinateur (défi, fantaisie, curiosité) qu'il a lui-même, dans un souci pédagogique, montré la correspondance qui s'établit généralement entre apprendre dans un environnement motivant et jouer dans ce même environnement. Nous saurons aussi écouter leurs mises en garde respectives.



Chapitre III

Jeu et programmation



Chapitre III. Jeu et programmation

Nous explorons dans ce chapitre les différents liens pouvant exister entre le jeu et la programmation. Nous montrons en quoi l'activité de programmation peut offrir des occasions de jeu. Nous étudions aussi la place occupée par le jeu dans l'enseignement de la programmation, que ce soit comme sujet d'études ou comme outil d'apprentissage. Nous présentons et tentons d'analyser les quelques jeux à vocation pédagogique rencontrés en programmation. Enfin, vu la nature de l'activité de programmation, nous définissons quels types de jeu pourraient intervenir dans son enseignement.

Afin de mieux réfléchir aux interactions que l'univers de la programmation peut avoir avec le jeu, nous commençons par faire ici un survol rapide de l'univers de la programmation.

III.1. La Programmation et son univers

L'activité de programmation dont nous parlons ici part d'un problème et va jusqu'à sa résolution par programme sur un ordinateur. Elle inclut les travaux de spécifications, d'analyse et de programmation proprement dite. Nous avons déjà évoqué les différentes approches de cette activité: programmation impérative, fonctionnelle, logique et orientée objets.

Programmer c'est connaître les fonctionnalités de l'ordinateur, les principes de l'écriture des programmes, puis passer de nombreuses heures à définir, ordonner, composer... pour obtenir finalement un "programme" qui s'exécute, peut-être en quelques minutes, et qui paraît éventuellement très simple après coup.

Pour illustrer autrement cette difficulté de l'activité de programmation qui consiste à élaborer longuement un produit qui ne prend réellement corps qu'au

cours d'une exécution finale (par rapport au processus de production), prenons l'exemple d'un compositeur de musique symphonique. Son travail consiste à élaborer, grâce à ses nombreuses connaissances techniques et grâce à son imagination, un texte (la partition) dont la lecture, par un orchestre, produira un jour de la musique; des mois de travail pour la main pour quelques minutes de plaisir (si tout va bien) pour l'oreille.

On imagine bien les questions, les doutes, les angoisses, et les feuilles froissées qui peuvent accompagner une telle composition. Il en va de même pour le "programmeur" qui conçoit le produit, analyse le problème, élabore l'algorithme puis le programme, pour enfin le voir "tourner". Il est difficile de toujours bien vivre toute cette période d'analyse, d'élaboration de la solution, d'accepter les reculs apparents qui ne sont en fait que la maturation du travail. Il faut se contenter pendant un temps plus ou moins long de résultats abstraits, savoir ne pas se précipiter trop vite vers le concret, c'est-à-dire vers le codage d'un algorithme qui ne serait pas encore viable dans le but de voir une exécution sur l'ordinateur.

Le développement de prototype permet toutefois d'avoir des vues d'un produit informatique avant son achèvement, c'est le rôle joué par le piano pour le compositeur symphonique. Notons que le développement de prototype n'est pas immédiat non plus, et qu'un prototype ne constitue qu'une approximation du produit final. Toutes les fonctions sont implémentées mais pas dans le détail (prototype horizontal) ou bien, certaines fonctions seulement sont implémentées, dans leur forme finale (prototype vertical) [Choppy 85].

La façon de programmer et donc le temps nécessaire et le soin apporté à la période de conception et d'analyse sont fonction de la grosseur du problème, de l'enjeu concerné, du professionnalisme des programmeurs impliqués. Prenons quelques exemples. Un amateur devant un micro-ordinateur peut taper quelques lignes d'instructions, lancer l'exécution qui est immédiate, facile et peu coûteuse,

voir le résultat et rejouer ! Un étudiant devant programmer le tracé d'un dessin précis sur table traçante, aura tendance à avoir une attitude plus minutieuse, vu la longueur de l'exécution. En allant beaucoup plus loin, on imagine la rigueur qui est de mise dans le développement de projets de contrôle de centrales nucléaires par exemple.

L'activité de programmation nécessite obligatoirement des moyens matériels et des moyens logiciels (systèmes d'exploitation, environnements de programmation, ateliers de production,...) qui font partie de l'univers du programmeur.

Rappelons aussi que le programmeur évolue au sein d'une culture informatique, avec le sens large que revêt le mot culture, son côté noble, mais aussi son jargon et ses manies, et nous verrons en quoi le jeu peut s'infiltrer à divers niveaux dans cette culture.

III.2. La programmation est-elle un jeu ?

La plupart des programmeurs ont déjà eu la sensation de jouer en mettant au point un programme et Jacques Arsac n'hésite pas à affirmer "j'ai nombre de collègues pour lesquels la programmation est un jeu, comme pour moi" [Arsac 85]. Cela ne signifie pas que la programmation est pour eux de toute simplicité et réduite à un jeu d'enfant. Il ne s'agit pas non plus de dire que la programmation est en fait un jeu à part entière et que les programmeurs ont bien de la chance de pouvoir jouer à longueur de journée. Un travail ne peut pas être considéré globalement comme une activité libre, séparée de la vie réelle, incertaine et improductive.

La programmation: une activité source de jeux

En fait, l'activité de programmation peut à certains moments s'isoler de son contexte de travail et être vécue comme un jeu. Ceci est tout à fait évident lors de la mise au point d'un programme. A certains moments, il ne s'agit plus de terminer ce programme assez vite pour respecter les délais de livraison, mais bien d'arriver à bout de ce programme qui vous défie en ne voulant pas faire ce que vous voulez lui faire faire ! Et le soir, le programmeur reste encore sur sa machine se proposant de dénicher cette dernière bogue avant de partir en vainqueur ! Il agit alors comme le joueur à l'apparition de "Game Over" sur son écran vidéo, indication qui comme l'affirme Gros [Gros 83] est généralement précédée de "cette fois-ci si je n'y arrive pas j'arrête", et généralement suivie de "bon, j'essaie encore une partie mais c'est la dernière"... Cette activité là est devenue, pour un temps, jeu avec ses caractéristiques (libre, séparée, improductive, incertaine, réglée).

En allant un peu plus loin, nous nous rendons compte que l'activité de programmation n'offre non pas une mais de multiples occasions de jeu. Elle peut en fait donner lieu à des jeux de types différents selon la typologie de Caillois [Caillois 67a], présentée au chapitre 2. Prenons des exemples illustrant chacun de ces types de jeu possibles dans l'activité d'un programmeur, sans nous préoccuper de la qualité de son travail.

L'activité de programmation peut très facilement induire un comportement de type "**compétition**", si celui qui "programme" se fixe des buts en terme de réussite, de temps de programmation, de performance du programme obtenu... Là encore, il ne s'agit pas de dire que chaque fois qu'un programmeur veut obtenir un programme de bonne qualité, dans de courts délais, il a une activité de jeu. Mais dans ce contexte, il peut être incité par moments à se donner des défis partiels, et à relever ces défis en terme de jeu.

La programmation peut aussi donner lieu à des jeux de "hasard". Il peut s'agir par exemple du jeu qui consiste à se placer devant un micro-ordinateur sans bien connaître un langage de programmation, à tenter sa chance quelques fois pour trouver la syntaxe correcte d'une instruction, au lieu de la chercher sur un manuel. Nous pouvons aussi donner l'image du "programmeur" qui, négligeant les étapes précédentes, tape rapidement quelques lignes d'instruction, et déclenche la commande d'exécution, attendant ainsi le verdict de la machine. Ce "programmeur" n'a-t-il pas un comportement proche de celui d'un joueur devant une machine à sous?

Notons que ce type de comportement ne peut être pratiqué que dans le cadre de très petits programmes où il n'est pas nécessaire de réfléchir longuement pour entrevoir une solution, dans un contexte d'accès facile et interactif à l'ordinateur, et dans le cas de programmes dont l'exécution est rapide. On imagine mal, en effet, un joueur déclenchant en quelques secondes le fonctionnement de sa machine à sous, la regardant tourner pendant de longues heures, avant d'obtenir le verdict final, et rejouant immédiatement.

Examinons le cas du jeu de Loto: le joueur effectue ses choix à l'avance, le mécanisme qui détermine les résultats se déclenche à un moment précis bien que différé, et a un résultat très rapide et définitif. Le programmeur soumettant un travail en différé à l'ordinateur peut se trouver en fait dans une situation comparable. Le jeu possible ne consiste alors pas à mettre complètement un programme au point en un certain nombre de passages en machine, comme le jeu du joueur de Loto ne consiste pas à gagner une certaine somme dans l'année. Dans les deux cas, le jeu est ponctuel, attaché à un essai précis, même si on rejoue plusieurs fois.

Le programmeur est aussi souvent tenté d'avoir une vision exécutoire de son programme: il peut aller jusqu'à le faire "tourner" à la main. En cela, il se substitue à l'ordinateur, adoptant ainsi un comportement de type "simulacre".

La différence entre le programmeur, son programme et l'ordinateur qui l'exécute est parfois très floue. Notons que cette différence peut s'accroître en cas de désaccord. Le programmeur, prenant alors à témoin un collègue s'exclame: "regarde, moi, je fais ceci (il s'agit d'entendre "moi, je veux faire ceci") et lui (l'ordinateur), il fait cela"... L'ordinateur est alors personnalisé et rendu responsable des événements inattendus, alors que c'est le programme et donc le programmeur qui en sont à l'origine.

Lors de la spécification d'un produit, le programmeur peut aussi vouloir "se mettre à la place de" l'utilisateur visé. Les questions que le programmeur est amené à se poser sont alors de natures diverses. Il peut s'imaginer futur utilisateur de ce produit, et essayant en tant que tel de le définir: "quelles sont les fonctionnalités nécessaires, qu'est-ce qui me serait commode ?". Il peut aussi s'imaginer utilisateur devant un travail précis à effectuer avec le produit, ce qui peut lui permettre de se rendre compte qu'il manque peut-être certaines fonctionnalités, que l'interface peut être améliorée... Ceci correspond bien au jeu de simulacre: il s'agit de coller au mieux à la peau du personnage utilisateur, sans se prendre réellement pour lui car l'utilisateur réel connaît mieux le travail à faire avec le produit, et c'est bien à lui qu'est destiné le produit. Un tel simulacre peut aussi favoriser le dialogue entre le concepteur et l'utilisateur.

Un autre simulacre, peut aider à spécifier correctement un travail; ce simulacre consiste alors à se mettre à la place de celui qui va recevoir les spécifications, il s'agit de faire abstraction de ce que nous avons dans la tête pour ne raisonner que sur les documents que nous transmettons.

Le programmeur peut être pris de vertige devant l'ampleur d'un problème: par exemple, il ne sait pas comment aborder l'analyse, comment restructurer un algorithme qui ne le satisfait pas... Mais ce vertige n'est pas souhaité, recherché. Par contre imaginons un programmeur, ayant fait une mauvaise analyse d'un problème, ayant programmé selon cette analyse, et constatant que ce programme ne répond pas ou mal à son problème et "craque" de toutes parts. Il est alors possible que ce programmeur, dans l'angoisse de devoir tout reprendre à zéro, cherche alors volontairement un certain abrutissement, tentant de "rafistoler" morceau après morceau ce programme. Il préfère ce vertige devant la machine plutôt que de voir la réalité en face et de reprendre les choses posément, devant le listing ou l'algorithme. Il a alors un comportement qui correspond au comportement des joueurs dans les jeux de "vertige".

Du côté de la "bonne" programmation:

Devant ces différentes façons possibles de jouer au cours d'une activité de programmation, le lecteur peut avoir une drôle d'image du programmeur et de son métier. Nous avons en fait recensé ici quelques façons dont on peut jouer en programmant, toutes façons de programmer confondues: d'une approche empirique jusqu'à une approche structurée et modulaire de la programmation.

Illustrons cela grâce à une métaphore [Peyrin 88]. Imaginons que nous souhaitions construire un abri. Nous commencerions par empiler des pierres pour faire des murs. Si un mur venait à s'écrouler, nous le recommencerions en le consolidant par un bâton. Peu à peu, à force de tâtonnements, nous aurions construit nos quatre murs. Il nous resterait le toit et nous aurions du mal car c'est plus difficile. Mais nous en viendrions à bout grâce à plein de petites astuces que nous aurions découvertes au fur et à mesure de notre construction. Il nous serait sans doute difficile de garantir la résistance au moindre changement de temps, mais le résultat pourrait être très joli... Et nous pourrions avoir eu de nombreuses

occasions de jeu au cours de cette construction. De toute évidence, celui qui construit un bâtiment de quelque envergure ne s'y prend pas du tout de la même façon. Nous pouvons comprendre que si ce constructeur professionnel rencontre également au cours de ce travail des occasions de jeu, celles-ci seront sûrement différentes de celles rencontrées par l'amateur évoqué précédemment.

Un "bon" programmeur refuse de s'en remettre au hasard ou bien de s'abandonner à un quelconque vertige. Il ne croit qu'en la maîtrise de lui-même et de ses programmes. Il ne devrait donc pas, dans l'exercice de ses fonctions, s'adonner à des jeux de type "hasard" ou de type "vertige".

Par contre des jeux de "compétition" peuvent être compatibles avec son comportement de programmeur, grâce à la rigueur, à la reconnaissance des vertus de l'entraînement et à la maîtrise de soi qu'ils confèrent. Ceci ne veut pas dire pour autant que tous les jeux de compétition pouvant être pratiqués au cours de l'activité de programmation sont favorables à une "bonne" programmation.

Examinons maintenant les jeux de simulacre: ceux-ci nécessitent maîtrise de soi et imagination. Certains de ces jeux de simulacre peuvent donc apparaître au cours du travail de programmation, sans remettre en cause les qualités de celui-ci: c'est le cas du jeu "et si j'étais utilisateur...". Par contre, un "bon" programmeur sait maintenir une distance avec l'ordinateur: il ne raisonne pas en termes d'exécution de ses programmes et ne saurait donc s'adonner au jeu "et si j'étais l'ordinateur"...

La plupart des occasions de jeux dans l'activité de programmation déjà citées ont lieu lors du travail du programmeur sur l'ordinateur, en particulier lors de l'implémentation et de la mise au point des programmes. Du côté de la "bonne" programmation, il ne s'agit pas d'écrire n'importe quoi et de s'en remettre à la machine pour savoir ce qui ne va pas. Un "bon" programmeur connaît les vertus de la rigueur et de la patience, il sait se relire et se corriger. Le travail de mise au point

ne constitue qu'une petite partie de sa tâche et ne doit pas remettre en cause la sémantique du programme.

La programmation, c'est aussi un jeu sur ordinateur !

Le travail de mise au point d'un programme constitue effectivement un bon jeu sur ordinateur, et même peut-être un des meilleurs. Nous avons exposé au chapitre 2 les résultats du travail de Malone en ce qui concerne les caractéristiques que doivent présenter des jeux sur ordinateur pour être attrayants. Essayons maintenant d'analyser l'activité de programmation sur ordinateur en fonction de ces caractéristiques.

Le défi proposé est évident, il faut que le programme tourne et qu'il fournisse les résultats attendus. De plus, il est facile de générer d'autres buts de différents niveaux que ce soit en termes de produit fini (état de marche, rapidité d'exécution, espace mémoire nécessaire) ou en terme du travail nécessaire pour l'obtenir (temps de programmation...). Le "joueur" reçoit fréquemment une information sur sa performance (erreurs de compilation, erreurs d'édition de lien, erreurs à l'exécution...). Lors de la mise au point, cette information est d'ailleurs souvent trompeuse quant à la proximité du but visé, disons plus exactement qu'elle laisse au programmeur les illusions qu'il veut bien nourrir. Le voici tenu en haleine, croyant toujours que, les erreurs repérées une fois corrigées, il aura gagné, alors que l'avancement est souvent plus progressif !

La mise au point de programmes provoque bien des effets de surprise en révélant au programmeur ce que le programme fait effectivement et non ce qu'il souhaite lui voir faire. Voilà sa curiosité en éveil ! Il s'agit de comprendre maintenant ! L'estime de lui-même est en jeu !

Les choses sont moins évidentes en ce qui concerne la **fantaisie**. Toutefois, il semble que le programmeur puisse projeter sur son activité une fantaisie fonction de sa personnalité et fonction de l'application qu'il est en train de programmer. La fantaisie réside aussi dans le fait que le programmeur donne très souvent vie au couple ordinateur-programme, l'accusant verbalement de tous les maux à la découverte d'erreurs.

La fantaisie n'est pas forcément propre à l'activité même de mise au point, elle peut aussi par exemple être induite par l'interface privilégiée par le type d'ordinateur utilisé, ou par l'environnement de programmation utilisé: ne citons qu'un exemple, explosif, celui de la bombe de Macintosh, bombe apparaissant lorsqu'une grave erreur système est intervenue.

La fantaisie est aussi présente dans la culture informatique, elle apparaît dans le vocabulaire même des programmeurs, ne serait-ce que par le fameux "bug" qui mite tant de programmes. L'image de ce "bug" convient si bien aux informaticiens qu'ils ont souvent du mal à utiliser le terme français bogue correspondant.

L'activité de mise au point d'un programme constitue donc effectivement un bon jeu sur ordinateur. D'autres activités liées à l'ordinateur, de par le défi qu'elles présentent peuvent être vécues comme jeux. Tel (ancien) logiciel ne fonctionne pas sur tel (nouvel) ordinateur ? Recherchons vite, dans une activité fébrile, si nous ne pouvons pas configurer un système qui puisse concilier les deux... Notre lecteur de disquettes ne veut plus relire la disquette qu'il a pourtant configurée l'instant d'avant ? Cherchons à savoir ce qu'il s'est produit, invitons les collègues à résoudre la devinette, manipulons cette disquette... même si cette disquette ne contient rien qui ne soit sauvegardé ailleurs. La liste reste ouverte, et le lecteur, fort de ses propres expériences pourra sans doute l'alimenter.

Des jeux pervers:

Citons, en marge de ces activités honnêtes, certains jeux pervers qui ont trait à l'informatique. Il s'agit tout d'abord du "piratage" de logiciels, pratiqué si longtemps impunément, d'autant plus que circulaient officiellement sur le marché des logiciels de copie. Le jeu consistait alors à trouver le logiciel de copie capable d'éliminer la protection du logiciel X, quel que soit d'ailleurs l'intérêt réel, souvent mineur, que représentait le logiciel X pour le "pirate".

Un autre jeu pervers, souvent pratiqué par de jeunes passionnés, dans un but purement ludique, consiste à trouver grâce aux systèmes de réseaux, un accès à des informations auxquelles ils ne devraient théoriquement pas pouvoir accéder. Il s'agit de se frayer un chemin, de déterminer des mots de passe,...

L'existence de tels jeux pose bien le problème de la sécurité informatique, et prouve qu'elle n'est pas satisfaisante, comme en témoignent encore l'apparition récente des virus informatiques et les dangers qu'ils représentent.

III.3. Jeu et étude de la programmation

Les étudiants en programmation en tant qu'apprentis-programmeurs, ont accès à la part de jeu sous-jacente à l'activité de programmation déjà évoquée. Le niveau de but alors fixé n'est pas le même, mais le processus est bien du même ordre. L'activité prise globalement n'est pas plus libre, car l'étudiant met en jeu, à terme, son diplôme. Comme pour le programmeur, le jeu ne peut donc s'installer que ponctuellement dans son activité.

La question est maintenant de savoir si le jeu apparaît en d'autres occasions pour l'étudiant, du fait qu'il se situe en phase d'apprentissage de la programmation. Afin de répondre à cette question, nous pouvons envisager les choses selon deux

axes distincts. Le jeu est-il présent en tant que sujet d'étude dans une formation à la programmation ? Ou bien le jeu est-il utilisé comme outil d'apprentissage dans les formations à l'informatique ?

III.3.1. le jeu, sujet d'études: une formation spécifique au métier de concepteur de jeux sur ordinateur ?

Le jeu pourrait intervenir dans la formation en tant que sujet d'études; il ne s'agit pas de concurrencer des formations en sciences humaines, mais bien de former des étudiants au (dur?) métier de concepteur de jeux sur ordinateur. Une telle formation n'existe pas (à ma connaissance) ni à part entière, ni comme spécialité proposée à l'intérieur d'une formation informatique existante, alors que cela se justifierait tout à fait, comme le montre Verhoeff dans son article "Recreational Informatics, a discipline of its own" [Verhoeff 83].

Le concepteur doit en effet avoir une mentalité particulière, être capable d'une approche artistique, avoir une bonne dose de créativité. Pour Crawford, la conception d'un jeu est une forme d'art [Crawford 82]. Mais le concepteur doit aussi posséder des techniques de programmation particulières: animations graphiques et sonores, techniques d'intelligence artificielle, optimisation des vitesses d'exécution (un joueur n'attend pas !). De plus, le concepteur de jeux sur ordinateur doit proposer une interface particulièrement conviviale; il est impensable d'envisager une session de formation à un jeu alors que cela se fait encore couramment pour des applications professionnelles ! Le concepteur doit donc allier autant de qualités scientifiques à une approche artistique.

Il existe en fait peu de littérature concernant la conception de jeux sur ordinateur, comme le déplore Verhoeff [Verhoeff 83]. Parfois nous découvrons quelque titre alléchant comme "Une méthode de conception de logiciels de jeux scientifiques", titre d'une communication [Reinhorn 87], présentée au colloque

EAO 87. En fait l'auteur présente un guide d'entretien avec un expert de la matière concernée, qui vise à déterminer si la matière se prête plus facilement à la création d'un jeu de simulation, d'un jeu d'aventures, d'un jeu de construction... Les thèmes traités en exemple sont très vastes: l'évolution du climat, les origines de la vie, les origines de l'univers. Une fois cet axe délimité grâce à la méthode annoncée, les concepteurs se mettent au travail... Il ne s'agit donc que très faiblement d'une méthode de conception: la méthode proposée a plutôt pour objectif de trouver sur un vaste sujet, une idée de départ pouvant donner naissance à un jeu. Cette idée sera alors développée grâce au travail des concepteurs.

Mais qui sont-ils ces concepteurs ? "A la fois experts informaticiens, graphistes de l'électronique et scénaristes délirants, ils inventent des programmes de plus en plus sophistiqués. Sérieux, ils avancent dans des forêts de circuits imprimés; fantasques, ils imaginent des planètes de bandes dessinées. L'ennui, c'est qu'ils sont inaccessibles. On ne les voit jamais. On ne les connaît pas..." [Gros 83]. Gros et Pernelet expliquent le mystère qui entoure ces concepteurs par le fait que la conception de jeux ne s'improvise pas. Ces concepteurs sont très précieux, les sociétés se les "arrachent". Ils sont donc cachés et protégés par leurs employeurs.

Pamela Clark déplore le fait que les concepteurs actuels se sont intéressés à l'informatique avant de s'intéresser au jeu et souhaite que la tendance s'inverse [Clark 82]. Mais ne risque-t-on pas ainsi de tomber cette fois sur les limites techniques d'un tel concepteur ?

Une meilleure approche serait de faire collaborer un "scénariste" qui devra connaître les effets possibles, et un technicien-programmeur, spécialiste des techniques utilisées et possédant le sens des thèmes artistiques. Cette tendance est amorcée dans un domaine plus général que celui de la création de jeux sur ordinateur, si l'on considère la formation de "concepteur médiatique", assurée par Paris X [Thomas 86]. Cette licence prépare au métier de "chef de projet pour le

développement de produits et de services de communication interactifs", ce chef de projet travaillant entre les détenteurs du contenu (professeurs, banque, entreprise...) et les informaticiens réalisateurs.

Verhoeff attire aussi notre attention sur la nécessité de l'existence d'une approche scientifique des jeux vidéo, pour contrebalancer une exploitation purement commerciale du marché [Verhoeff 83]. S'il est vrai que l'on peut faire confiance aux joueurs pour laisser tomber d'eux-mêmes les jeux les plus bêtes, encore faut-il diversifier suffisamment les jeux qu'on leur propose. Or, il est plus facile pour les concepteurs d'adapter une idée qui, commercialement a fait ses preuves, et ensuite de programmer, que de plancher sur un nouveau concept, programmer et prier pour le meilleur [Williams 82] !

Il semble donc qu'une recherche scientifique en ce domaine serait nécessaire et qu'elle pourrait donner des bases utiles à une formation à la conception de jeux sur ordinateur. Une telle démarche aurait l'avantage de faire collaborer des experts de plusieurs disciplines, de rendre le jeu moins suspect aux yeux des scientifiques, et de permettre à ceux-ci une approche plus artistique, donc plus créative.

III.3.2. le jeu outil d'apprentissage

Revenons maintenant à nos préoccupations d'origine. Nous avons vu au chapitre 2 que le jeu pouvait être utilisé à des fins pédagogiques dans un enseignement. Examinons ce qu'il en est actuellement dans l'enseignement de la programmation.

III.3.2.1. Des réalités:

Dans une formation à la programmation, le jeu peut apparaître comme **sujet de réflexion**, que le jeu soit sujet d'exercice, sujet de travaux pratique, ou support de

présentation de concept. Les manuels d'enseignement sont révélateurs de cet état de fait. Choisir un jeu comme sujet d'exercice peut être un choix a priori comme dans le livre "Jeux et casse-tête à programmer" de J. Arzac [Arsac 85] ou plus occasionnel.

La plupart des manuels d'enseignement utilisent certains jeux comme les Tours de Hanoi, le jeu du Taquin, le problème des huit reines ou le problème du commis voyageur pour introduire les concepts de programmation correspondants. D'autres jeux sont cités de façon moins systématique: la caverne d'Ali Baba, le jeu de dés [Pair 88a], le jeu africain Kalaha [Pair 88]a, [Boussard 83], Tic Tac Toe [Boussard 83], [Aho 83]... Soulignons que le livre "Programmation avancée" [Boussard 83] utilise ainsi de nombreux jeux tout au long de la progression qu'il propose: le jeu du cavalier, le code secret, le mot le plus long, puzzle géométrique,...

Le jeu choisi est alors "décortiqué" pour être programmé, il est sujet de réflexion, il se prête à la mise en œuvre des concepts et méthodes présentées. Il s'agit bien d'un exercice. Le fait qu'il s'agisse de programmer un jeu peut motiver l'étudiant et lui offrir un plaisir supplémentaire. C'est ce que nous suggère Jacques Arzac lorsqu'il écrit "celui qui programme ses jeux s'amuse deux fois" [Arsac 85]. Mais l'exercice n'a rien à voir avec la pratique d'un jeu pédagogique, lors de laquelle l'étudiant peut apprendre. Le jeu n'est pas considéré dans ce cas comme un outil d'études.

D'autre part, certains langages de programmation présentent un aspect ludique par le côté très visuel de l'exécution de programmes. Citons Logo [Avram 84], et tout particulièrement le Logo-Tortue [Berdonneau 84]. L'aspect ludique que rencontrent les enfants lors de la manipulation de la tortue a permis d'aborder la programmation avec eux.

Davantage destinés à nos étudiants, apparaissent quelques jeux à vocation pédagogique, ayant trait à la programmation. Etudions-les maintenant.

a) Le Jeu des plaquettes:

Ce jeu est quelquefois utilisé en cours. Il ne nécessite aucune compétence informatique, ni même d'ordinateur, et tente de faire découvrir un peu aux joueurs ce qu'est la programmation.

Selon la description donnée dans [Scholl 88], ce jeu nécessite 5 joueurs (ou groupes de joueurs) et un meneur de jeu. Chaque joueur reçoit 10 plaquettes numérotées de 0 à 9. Une partie se joue en 5 manches. Une manche se déroule de la manière suivante:

- le meneur de jeu annonce un nombre compris entre 0 et 45.
- sans se consulter, les joueurs présentent, au signal du meneur de jeu, une de leurs plaquettes.
- le meneur de jeu a perdu si la somme des valeurs des cinq plaquettes présentées est égale au nombre annoncé.

Les joueurs constatent vite qu'ils ne gagnent à coup sûr que s'ils conviennent d'une marche à suivre. Ils sont alors amenés à inventer un algorithme et à le décrire.

Plusieurs idées d'algorithme germent alors dans la tête des joueurs. Nous assistons alors à diverses prises de pouvoir: prise de pouvoir d'un joueur dans son groupe, prise de pouvoir d'un groupe sur les autres groupes. Dans ce contexte, les idées sont explicitées, plus ou moins clairement, les algorithmes, plus ou moins justes, sont proposés.

Exprimons les idées qui sous-tendent les solutions généralement proposées. Les deux premières nécessitent de définir un ordre sur les joueurs. Il peut s'agir de:

- répartir les valeurs affichées, équitablement entre les joueurs: la base de l'algorithme est une division par 5 du nombre annoncé N . Notons $N=5*q+r$ avec $0 \leq r < 5$: si le reste r est nul, chacun présente une plaquette de valeur q , sinon les joueurs du n° 1 au n° r présentent une plaquette de valeur $q+1$ ($q+1$ est toujours inférieur ou égal à 9), les autres une plaquette de valeur q .

- répartir les valeurs affichées non nulles sur un nombre minimum de joueurs: la base de l'algorithme est une division par 9 du nombre annoncé. Notons $N=9*q+r$ avec $0 \leq r < 9$, si le quotient q est nul, le 1er joueur présente la plaquette r , les suivants présentent 0. Sinon, les joueurs du n° 1 au n° q ($0 < q \leq 5$) présentent la plaquette 9, le joueur n° $q+1$, s'il existe, présente la plaquette r , les suivants s'ils existent présentent 0.

La troisième solution ne nécessite pas d'ordre sur les joueurs mais elle en particularise un. Il s'agit de:

- répartir équitablement les valeurs entre 4 joueurs, le dernier complétant la somme: la base de l'algorithme est une division par 4 du nombre annoncé. Notons $N=4*q+r$ avec $0 \leq r < 4$. Si $q \leq 9$, les 4 joueurs présentent la plaquette q , le dernier présente la plaquette r . Sinon ($q > 9$), les 4 joueurs présentent la plaquette 9, le dernier joueur présente la plaquette de valeur $(N-9*4)$ (cette valeur est inférieure ou égale à 9 car $N \leq 45$)

La quatrième solution proposée ici attribue à l'avance un rôle précis à chacun. Il s'agit de:

- préparer un tableau indiquant ce qui doit être fait par chaque joueur pour chaque nombre que le meneur de jeu peut annoncer. Signalons que les joueurs n'envisagent en général pas spontanément cette solution, qui pour être efficace, n'en est pas

moins lourde à installer, et réduit ensuite leur fonction à celle d'un robot. Néanmoins, cette solution pourrait être retenue dans des contextes d'application particuliers.

A partir de ce qu'ont vécu les joueurs pendant la partie, le meneur de jeu, en général l'enseignant, peut expliciter ce qui s'est passé et faire le lien avec la programmation en faisant ressortir:

- l'intérêt d'un point de vue efficacité, de résister à la tentation de commencer très vite la partie afin de passer du temps à se mettre d'accord sur la stratégie à employer. Ceci constitue bien comme nous l'avons montré plus haut, une des difficultés rencontrées par le programmeur qui doit consacrer beaucoup de temps à la conception, à l'analyse, à l'élaboration de l'algorithme, à l'écriture du programme avant de voir concrétiser tout ce travail par l'exécution de ce programme.
- la nécessité de la communication entre les partenaires, les difficultés de cette communication (difficulté de décrire un algorithme de façon précise et non ambiguë, difficulté de comprendre un algorithme énoncé par un autre...), ses pièges (les sous-entendus, l'impression de "tout" expliquer et de "tout" comprendre par des exemples...).
- la différence entre un ordinateur qui applique aveuglément et fidèlement un algorithme et un joueur qui peut se tromper dans l'application de l'algorithme ou qui prendra la liberté d'improviser face à un cas limite non décrit par l'algorithme.
- la diversité des solutions trouvées et leur intérêt relatif à la façon dont on joue, à la fréquence à laquelle on joue...

Le jeu des plaquettes qui réunit un meneur de jeu et des joueurs devant coopérer pour remplir une mission répond bien à la définition d'un jeu de rôles. Le meneur de jeu a un rôle décisif: il doit comprendre l'idée qui a été explicitée par les joueurs avant une manche, trouver soit les contre-exemples correspondants, soit les cas limites non repérés par les joueurs. Ainsi, il gagne la manche, mais surtout il fait progresser les joueurs, et relance l'intérêt de la partie. En tant que jeu de rôles, le jeu des plaquettes fait partie des jeux de simulacre dans la typologie de Caillois [Caillois 67a].

L'intérêt pédagogique d'un tel jeu est de pouvoir présenter en si peu de temps des concepts de haut niveau en programmation. Le jeu peut être utilisé au minimum en guise d'entrée en matière, permettant de présenter un certain nombre de choses significatives pour l'activité de programmation. L'enseignant pourra aussi beaucoup plus tard faire référence à ce qui a été vécu par les étudiants pendant le jeu. Car il s'agit bien de leur faire vivre une expérience, un travail en commun, puis d'analyser avec eux ce qui s'est produit et de leur indiquer l'analogie avec la programmation. Prendre les choses à l'envers détruirait l'intérêt du jeu et les bénéfices attendus.

Ce jeu a été souvent pratiqué ici dans notre entourage, mais aussi à l'étranger, face à des publics de culture et de nature diverses (étudiants, stagiaires en formation continue, élèves du secondaire). Dans tous les cas, le public s'est très vite pris au jeu, oubliant pour un moment le contexte du cours de programmation. Les stagiaires qui avaient déjà un vécu professionnel ont bien retrouvé dans ce jeu les éléments qui correspondaient à leur vision.

b) ChipWits:

Présentation générale

ChipWits est un logiciel de jeu [Sharp 84] fonctionnant sur Macintosh. Il s'agit de faire évoluer à l'écran un robot, ou plutôt sa représentation graphique, dans un environnement inconnu (et parfois hostile). Un environnement est composé de différentes salles; à l'intérieur de chaque salle, sont disposés des objets de nature très différente: tarte, burette d'huile, disquette, bombe....

Le robot est capable de regarder, de sentir, de toucher, de manger, de détruire et de se déplacer. Ses caractéristiques vitales sont son état de marche (il peut être plus ou moins endommagé), sa réserve d'essence, et le nombre de cycles qu'il lui reste à vivre.

Le robot doit agir avec discernement: prendre un café va lui donner de l'énergie (essence), prendre une burette d'huile va lui rapporter des points alors que tenter de prendre un "crabe" va l'endommager sérieusement... Il a d'ailleurs intérêt à se tenir à distance de ces petites bêtes qui peuvent l'attaquer, et à prendre les devants afin de les détruire, ce qui fera augmenter son score. Il devra aussi éviter de se taper la tête contre les murs ou de se cogner contre une bombe fatale. Le robot consomme de l'essence et des cycles de vie en fonction des actions qu'il entreprend.

Plusieurs environnements sont proposés, plus ou moins complexes, plus ou moins dangereux pour le robot. Le choix de l'environnement est laissé au joueur. Dans un environnement le joueur peut faire effectuer à son robot une mission ou bien une série de missions.

Le joueur détermine le comportement du robot en le programmant, et c'est sur ce plan que ce jeu nous intéresse. Il dispose pour cela d'un environnement de

travail où il crée son programme, et d'un environnement d'exécution qui permet de voir fonctionner le robot, et de mettre au point le programme grâce à la visualisation dynamique de la partie du programme exécutée, au choix de la vitesse d'exécution et à la visualisation du contenu de la mémoire du robot. Le joueur peut ainsi programmer plusieurs robots, chacun de ces programmes étant repérés par le nom du robot.

Le langage de programmation ne revêt pas la forme textuelle classique des langages traditionnels. Ce langage, nommé IBOL (Icon Based Operating Language) est basé sur un ensemble d'icônes. Un programme est construit en plaçant ces icônes sur un support à deux dimensions et en les interconnectant entre elles. Nous y reviendrons.

Le Jeu ChipWits

ChipWits est dès le premier abord très attrayant. Les qualités graphiques du jeu, la visualisation très plaisante qu'il donne de l'évolution du robot dans un environnement donnent envie de jouer. Même le langage de programmation a un air engageant et vous invite à ne plus attendre davantage. Tout semble léger, rapide, teinté d'humour, et en même temps fort riche. Il y a plein de choses à découvrir dans l'univers de ce sympathique robot qui nous fascine.

Face à ChipWits, le joueur doit faire la preuve qu'il maîtrise le comportement du robot et sait obtenir de bons scores. Pour parvenir à cette maîtrise, il va essayer, persévérer, s'entraîner. Il s'agit nettement ici d'un jeu de compétition selon la typologie de Caillois [Caillois 67a].

Analysons maintenant les ressorts que présente ce jeu en termes des éléments qui caractérisent selon Malone [Malone 80] les bons jeux sur ordinateur.

Le jeu consiste à programmer, et à ce titre propose les défis déjà cités pour l'activité de programmation. Le défi est encore plus marqué ici car le joueur doit faire un bon score, tenir compte de l'état du robot et de sa réserve de carburant, éviter ses ennemis (crabe, bombe,...), le tout pendant un nombre limité de cycles de vie.

Bon nombre d'indicateurs le tiennent au courant de sa performance tout au long d'une mission du robot, des statistiques plus générales concernant les diverses missions déjà effectuées par ce robot lui sont fournies à la fin d'une mission.

L'environnement suggère quantité de buts de difficulté très variable, selon l'endroit où l'on choisit de faire évoluer le robot (environnement plus ou moins hostile), selon que l'on choisit ou non de surveiller le niveau de carburant...

L'univers de ChipWits est très riche. Pénétrer dans cet univers éveille notre curiosité et nous invite à la découverte: découverte des différents environnements, découverte des objets présents et de leur nature pacifique, amicale ou hostile, découverte des actions du robot correspondant aux opérateurs fournis... Ces actions sont bien animées à l'écran, elles sont souvent bruitées, elles nous surprennent et nous ravissent. Ces découvertes que le joueur fait sous l'impulsion d'une curiosité à la fois sensorielle et cognitive s'échelonnent dans le temps: des effets de surprise sont toujours possibles. Vous savez faire avancer le robot depuis longtemps, mais tout d'un coup vous vous trouvez dans un cas particulier d'exécution: le robot se trouve devant un mur et veut avancer... le voici qui se cogne, et les murs de la salle tremblent !

Il s'agit aussi de découvrir des algorithmes efficaces en réponse à une curiosité purement cognitive cette fois. L'activité de mise au point de ces algorithmes éveille alors puis satisfait la curiosité du joueur de la façon déjà évoquée au paragraphe III.2.

Chipwits nous entraîne d'autant mieux dans sa **fantaisie** qu'elle est particulièrement réussie: comment résister à la tentation de programmer ce robot, de le baptiser et de l'envoyer en mission... Votre imagination peut vous entraîner encore plus loin: vous voici programmant "007 in BoomTown", avec le plaisir d'une exécution différente à chaque fois puisqu'un environnement change à chaque mission, tout en conservant des caractéristiques identiques. A vous d'être à la hauteur sinon les missions de votre robot risquent fort d'être de courte durée et explosives !

Le langage IBOL

Le langage de programmation, IBOL, à l'image de l'ensemble du jeu, possède une forme attrayante. Cette forme offre aussi l'avantage de permettre d'emblée une pratique. Décrivons plus précisément ce langage. IBOL propose 23 opérateurs, tous représentés par un symbole graphique. Les premiers opérateurs correspondent aux actions élémentaires du robot: l'opérateur "LOOK" représenté par un oeil, l'opérateur "SMELL" par un nez, "MOVE" par un véhicule... Viennent ensuite des opérateurs de contrôle: "SUB-PANEL" permet l'appel à un sous-programme, "BOOMERANG" permet le retour d'un sous-programme au programme principal juste après le point d'appel,... D'autres opérateurs permettent de travailler sur des piles. Trois piles sont ici proposées: l'une manipulant des mouvements, une autre des objets, la troisième des entiers. Signalons aussi un opérateur permettant d'avoir connaissance des entrées faites au clavier.

Ces opérateurs peuvent prendre des arguments, eux-mêmes représentés par des symboles. Par exemple les arguments disponibles pour l'opérateur "MOVE" sont quatre flèches indiquant les directions possibles du mouvement, plus le symbole représentant la pile de mouvements s'il s'agit de reproduire un mouvement mémorisé.

Pour "écrire" une instruction le programmeur doit tout d'abord choisir l'opérateur désiré dans l'ensemble des opérateurs proposés; seuls les opérateurs permis sont accessibles, par exemple l'opérateur "BOOMERANG" de retour au programme appelant n'est pas accessible depuis un programme principal. Une fois l'opérateur choisi, l'ensemble des arguments possibles pour cet opérateur apparaît; le programmeur doit y sélectionner l'argument voulu. Le composant ainsi constitué est alors placé à l'endroit choisi par le programmeur. Il est intéressant de noter qu'IBOL met ainsi le programmeur à l'abri de toute erreur syntaxique. Il ne peut sélectionner à un moment donné que ce qui est licite à ce moment-là.

La connexion entre les composants est réalisée par les flèches qui sortent des opérateurs. Certains opérateurs, comme "BOOMERANG" n'ont pas de point de sortie, car ils terminent une écriture. La plupart ont un point de sortie unique permettant de passer à l'instruction suivante. D'autres, c'est le cas de fonctions booléennes ont deux points de sortie, permettant d'indiquer quoi faire selon que la fonction prend la valeur vrai ou faux. Un opérateur spécial "JUNCTION" permet la connexion entre des opérateurs qui ne se situent pas côte à côte.

Analysons maintenant la mise en œuvre en IBOL, des principaux concepts de programmation:

- la notion de **séquence** s'exprime simplement par la connexion de deux composants.

- une **composition conditionnelle** peut être mise en œuvre autour de certains opérateurs qui sont en fait des fonctions booléennes. La forme "si... alors... sinon..." est construite de façon très naturelle; la forme "si... alors..." peut aussi être obtenue.

- le concept d'**itération** peut s'exprimer de façon explicite grâce à l'opérateur "LOOP" qui reprend l'exécution au début du programme (ou du sous-programme) concerné. Il peut aussi s'exprimer de façon implicite en construisant une boucle graphique d'opérateurs connectés entre eux.

- il n'est pas possible d'exprimer directement une **récurtivité**. Il faut en obtenir une traduction itérative et gérer les piles nécessaires.

- la notion de **types de données** existe, il y a en fait trois types de données prédéfinis (objet, direction de mouvement et entier).

- en ce qui concerne les **structures de données**, la seule structure proposée est la structure de pile.

- la notion de **sous-programme** existe, elle ne peut être utilisée qu'à un seul niveau.

- la notion de **paramètres** est présente. Indiquer l'argument d'un opérateur correspond à définir un paramètre "donnée"; il est défini par une valeur (constante ou lue en sommet de pile). Un paramètre "résultat" peut être reçu en sommet de pile; c'est le cas de l'opérateur "+" qui incrémente la valeur présente au sommet de pile. Il n'est pas question ici de manipuler de nombreux paramètres.

Ce langage permet donc d'aborder de nombreux concepts de base, même s'il a ses limites. Son niveau est comparable à celui de Basic. On peut écrire des programmes structurés avec IBOL, comme avec Basic; mais il est aussi possible dans un cas comme dans l'autre d'utiliser l'équivalent des "goto" à tort et à travers.

Analyse pédagogique

Chipwits est, selon la notice d'emploi (BrainPower, Inc.) destiné à des gens à partir de 12 ans. Ce jeu introduit à la programmation et ne requiert aucune connaissance préalable dans ce domaine. La société BrainPower qui le produit a nettement des objectifs pédagogiques. Elle utilise dans ce jeu une approche "apprentissage par la découverte". Au delà de ces intentions pédagogiques, essayons de voir plus précisément quels sont les apports de Chipwits dans l'apprentissage de la programmation.

Le premier avantage de ce jeu est de faire pratiquer une activité de programmation. Comme nous l'avons vu, le niveau du langage utilisé est comparable à celui de Basic. Sa forme permet d'éviter les erreurs syntaxiques, mais n'induit-elle pas chez les joueurs un certain "bricolage" ? C'est ce que nous craignons.

Il est en effet possible (et peut-être tentant) de placer des composants bout à bout, créant ainsi un programme syntaxiquement correct sans avoir une idée de ce que l'on souhaite faire. S'il est très facile de suivre pas à pas l'exécution d'un programme IBOL, il est très difficile d'avoir une vue sémantique du programme et donc de le relire: on aurait besoin de voir apparaître davantage une structure. De plus, il n'est pas possible de manipuler suffisamment des "blocs" d'instruction: si des contraintes d'organisation spatiale du programme nous obligent à déplacer un "bloc" logique, nous serons contraints de déplacer les composants un à un.

Une autre difficulté est due à la complexité du jeu. Il est difficile de programmer un robot en tenant compte de tous les éléments (essence, score, cycles de vie,...) qui rendent le jeu si attrayant. Programmer un robot qui sait faire un parcours efficace dans toutes ses missions, même dans un environnement dont les principales caractéristiques sont fixes n'est pas simple du tout.

Pour avoir ses chances, le joueur va devoir adapter le niveau du défi, en dépouillant progressivement le jeu d'un certain nombre de ses attributs, ce qui revient à simplifier le problème posé. Par exemple, une façon élégante de résoudre les problèmes de parcours aléatoires mais néanmoins habiles du robot dans les salles, est de programmer pour que les déplacements du robot se fassent en fonction des touches frappées au clavier. Le joueur se retrouvera alors à chaque exécution en train de tester ses réflexes et sa rapidité à réagir aux situations, ce qui constitue en soi une autre forme de jeu. Souhaitons que le joueur ne doive pas dépouiller le jeu au point qu'il ne prenne plus de plaisir à voir évoluer un robot trop sage, dans un univers trop calme !

Les erreurs d'exécution sont souvent visualisées de façon plaisante. Le joueur peut alors être tenté de provoquer volontairement ce type d'erreur plutôt que de chercher à l'éviter. Rappelons le danger évoqué par Malone [Malone 80] de voir le joueur tenter de perdre plutôt que de jouer de façon constructive et efficace d'un point de vue pédagogique, lorsque se déclenche quelque catastrophe spectaculaire en réponse à son échec. Nous ne nous trouvons pas ici dans un cas aussi extrême car l'effet spectaculaire d'une erreur n'est pas artificiel, il est lié à l'erreur elle-même; nous pensons que dans ce contexte (intrinsèque) la visualisation d'erreurs peut avoir un apport pédagogique intéressant.

Pour toutes les raisons précédemment exprimées, nous ne pensons pas que ce jeu puisse aider les joueurs et donc les étudiants à apprendre à programmer de façon satisfaisante, même s'il peut leur donner l'occasion d'aborder des concepts de programmation. Mais ne peut-on pas utiliser d'une autre manière ce remarquable jeu ? Nous pensons qu'il serait intéressant de proposer ChipWits aux étudiants avant le début du cours de programmation, et de laisser libre cours au penchant naturel de la plupart des débutants en programmation qui consiste à "bricoler" des programmes. Laissons-les jouer librement, et se rendre compte qu'ils ne s'en sortent pas ainsi (et là, nous mettons à notre service la complexité du jeu).

Intervenons alors pour mettre en évidence leurs difficultés, et pour donner, en démarrant le cours de programmation des réponses aux besoins qu'ils ont découverts. Il serait dangereux de tenter une telle pratique si le langage de programmation utilisé avait une forme plus classique: de mauvaises habitudes pourraient être prises. La forme du langage IBOL, ainsi que le contexte jeu permettent en revanche de bien séparer cette pratique d'une pratique professionnelle de la programmation. De plus, les étudiants seront motivés par un jeu aussi attrayant que ChipWits.

ChipWits est apparu dans notre entourage en 1985. Plusieurs enseignants s'y sont alors intéressés, dans le but d'en définir une utilisation pédagogique pour leurs étudiants. Mais le matériel nécessaire n'était pas disponible pour les étudiants et aucune expérimentation réelle n'a pu être menée dans notre environnement.

c) JAC, jeu d'aventure en conduite de projet:

JAC, Jeu d'Aventure pédagogique en Conduite de projet, est un jeu conçu, réalisé et utilisé dans le contexte de l'entreprise "SYSECA TEMPS REEL" [Crampes 87]. Ce jeu s'adresse à un public de chefs de projet débutants et a pour objectif de leur donner une expérience en conduite de projet.

Savoir conduire un projet informatique n'est pas simple. Certes diverses méthodes se proposent de venir en aide au chef de projet, mais ces méthodes ont leurs limites. En effet, chacune propose un schéma idéal d'application auquel risque de ne plus répondre le projet à mener, dès les premiers écarts (fatals) par rapport aux prévisions. De plus, une méthode ne doit constituer qu'une ligne de conduite, dont il faut savoir dévier dans certaines situations. Seule l'expérience permet de pallier à ces limites.

Comment transmettre aux chefs de projet débutants cette expérience qui leur fait défaut ? C'est pour répondre à ce besoin que le "jeu d'aventure en conduite de projet" présenté ici a été conçu. L'idée de base est de "faire vivre en raccourci et à moindre frais des projets au cours desquels le chef de projet débutant pourrait expérimenter son savoir et recueillir l'expérience des autres".

Il s'agit de placer le joueur dans un contexte de projet avec une application, un client, des fournisseurs, des contraintes, des délais... Dans ce contexte le joueur prend des initiatives, décide, organise, planifie des travaux. La simulation repose uniquement sur la conduite du projet: les tâches nécessaires à l'avancement du projet (élaboration du produit, rapports,...) restent fictives, elles sont évaluées par le dispositif de simulation en fonction du temps qui leur est consacré, des décisions prises pendant ce temps, de la consultation d'autres intervenants ou d'autres documents...

Le joueur peut se fixer deux types d'enjeu. Il peut mener effectivement l'ensemble du projet. Il peut aussi s'amuser à créer et à corriger des dérives au déroulement du projet. Dans le premier cas, le dispositif de simulation est capable de faire évoluer le projet en fonction des actions du joueur, soit dans le cadre des prévisions soit de manière perturbée en créant des incidents. Dans les deux cas, le dispositif de simulation peut aider le joueur à contrôler des situations difficiles.

L'objectif de cette simulation n'est pas de recréer toutes les situations possibles, ni de pouvoir tout corriger, mais de se situer dans un cadre suffisamment réaliste pour que l'aspirant chef de projet en tire une expérience profitable. Signalons à propos du réalisme visé, que tous les scénarios d'incidents intégrés dans le jeu sont issus d'incidents réels vécus par des chefs de projet de l'entreprise.

La version présentée au congrès EAO 87 est un dispositif de simulation sous forme manuelle faisant intervenir un groupe de 3 joueurs (tenant ensemble le rôle

de chef de projet) et un meneur de jeu. Le jeu d'aventures est ponctué de réunions avec le meneur de jeu. A la fin de chaque réunion, des faits sont mis à jour soit de manière objective, (à partir d'observations faites par le meneur de jeu au cours de la réunion), soit à la discrétion (constructive) du meneur de jeu.

Ce dispositif sous forme manuelle a été utilisé en formation, au cours de stages. Le jeu se déroule sur deux jours. Les joueurs paraissent très motivés au départ: ils s'approprient les règles du jeu (un manuel de 50 pages) en une soirée afin d'être prêts à jouer le lendemain ! A la fin du stage, les stagiaires se déclarent très satisfaits de la formation reçue. Des évaluations à plus long terme sont prévues.

Il est intéressant de signaler l'historique de ce jeu; il nous indique que le parcours conduisant de l'expression d'un besoin de formation jusqu'à la conception et à la réalisation d'un produit répondant à ce besoin n'est pas simple. Le besoin de transmission de l'expérience des chefs de projet a été repéré dans l'entreprise en 1983. Une première maquette exploratoire, assez éloignée du dispositif exposé plus haut a été réalisée en 1984. Le dispositif de simulation sous forme manuelle a été mis au point en 1986. Une informatisation du dispositif est alors envisagée pour faciliter la gestion des documents et apporter une aide au meneur de jeu. Le jeu informatisé, en version opérationnelle sera composé essentiellement de trois systèmes experts, le simulateur, le méthodologue et le pédagogue, orchestrés par un superviseur; le meneur de jeu toujours présent conserve au jeu la richesse d'une approche humaine. Le simulateur a été développé en 1987 donnant naissance à une première maquette. Le travail continue...

Ce jeu rentre bien dans la catégorie des jeux de "simulacre" [Caillois 67a], c'est un jeu de rôles: il faut tenir le rôle du chef de projet dans la simulation présentée. Notons ici qu'un seul rôle est proposé aux joueurs, les autres personnages intervenant ne sont pas joués. Toutefois, l'aspect coopération généralement présent dans un jeu de rôles se retrouve ici par le fait que trois personnes tiennent

conjointement le rôle du chef de projet: échanges d'idées, argumentations, discussions fructueuses, prise commune de décision sont nécessaires...

JAC nous montre bien l'apport décisif que peut avoir le jeu dans un apprentissage.

III.3.2.2 Des attentes:

Nous venons de voir que le jeu a un rôle à jouer dans la formation à la programmation. Certains enseignants ou formateurs n'hésitent pas à l'utiliser, voire à investir beaucoup de temps et d'effort pour concevoir et réaliser un jeu approprié.

Les jeux pédagogiques utilisés dans des formations à l'informatique présentés ci-dessus sont de type "compétition" ou de type "simulacre" selon la typologie de Caillois [Caillois 67a]. Nous n'avons pas rencontré de jeux de "hasard" ou de "vertige". Essayons maintenant d'analyser pourquoi certains types de jeu conviennent mieux à l'activité de programmation.

La classification des jeux proposée par Caillois repose sur le type de comportement pratiqué par le joueur. Réfléchissons donc sur le type de comportement que nous souhaitons demander à notre étudiant en programmation.

L'activité de programmation nous l'avons vu, exige une rigueur, une maîtrise de soi et de ses programmes. Ajoutons qu'elle est rarement pratiquée en solitaire: plusieurs personnes coopèrent dans l'élaboration d'un produit.

Rien de pire pour un programmeur qu'un programme qui semble avoir un comportement aléatoire. Le voilà désarmé, il ne dispose d'aucune piste pour débusquer le problème ! Il n'est pas non plus question de laisser un programme dans cet état ! Il ne paraît donc pas souhaitable d'encourager chez notre étudiant en

programmation un penchant pour le hasard et nous ne lui proposerons pas de jeux de hasard pur. Ceci ne nous interdira pas de faire un peu intervenir à petite dose, le hasard au côté d'un autre ressort du jeu, pour pimenter les choses ! (Tout au plus, nous pourrions proposer aux étudiants ayant un penchant naturel pour le hasard, des jeux de hasard pur visant à servir de défouloir à cet instinct, mais ce serait sans compter sur le loto, le tac o tac...)

Nous n'imaginons pas non plus d'encourager notre étudiant en programmation à s'abandonner à un quelconque vertige. Car s'il arrive au (mauvais) programmeur de ne plus maîtriser la situation (!), ce ne peut être en aucun cas son but. Si le vertige est au rendez-vous, il s'agit pour le programmeur, comme pour l'alpiniste, de le combattre non de le rechercher comme c'est le cas dans les jeux de vertige.

En revanche, puisqu'il s'agit de favoriser une maîtrise de soi et de ses actions, maîtrise nécessaire à l'activité de programmation, nous nous tournons naturellement vers des jeux de type **compétition** ou de type **simulacre**.

Choisir des jeux de compétition dans un contexte d'apprentissage semble assez naturel; l'étudiant exerce une compétence relativement précise, il ne peut compter que sur ses propres capacités qu'il affine au cours d'essais successifs. Ce type de jeu est d'ailleurs assez souvent utilisé, dans des disciplines diverses, lorsque les formateurs choisissent d'utiliser l'activité de jeu dans leur formation. Citons en programmation le jeu "Chipwits" [Sharp 84], en mathématiques les jeux "Darts" [Malone 80], "Navire" [Gers 84], "West" [Vivet 88], en sciences naturelles les jeux "Climatiseur de planètes", "Créer la vie" et "A la recherche du Big Bang" [Reinhorn 87]...

Les jeux de simulacre nous semblent aussi intéressants. Représenter un personnage demande une bonne maîtrise de soi. Tout comme celui qui dispute une compétition, le joueur est contraint d'être attentif et obligé à une présence d'esprit continue. Il doit improviser, agir, prendre des décisions, établir des stratégies efficaces... Nous avons bien perçu l'intérêt que peuvent présenter de tels jeux, notamment en informatique, avec le "Jeu d'aventure en conduite de projet" [Crampes 87]. Dans d'autres domaines, des jeux de simulacre ont aussi fait leurs preuves, notamment les jeux d'entreprise et les jeux de simulation (simulateur de vol,...) [Mucchielli 83].



Chapitre IV

**"Toi, Moi et Lui", un jeu de rôles
pour l'étude de la programmation**



Chapitre IV. "Toi, Moi et Lui" un jeu de rôles pour l'enseignement de la programmation

Ce chapitre présente le jeu de rôles que nous avons conçu et réalisé dans le cadre du laboratoire Arcade. Ce jeu repose sur une analogie entre la création d'un programme et la création d'une bande dessinée. Nous présentons les motivations pédagogiques qui ont été à l'origine de sa conception, ainsi que les raisons qui nous ont conduit à placer les étudiants dans un domaine autre que celui de la programmation. Le jeu réalisé est décrit; les analogies avec le travail de programmation sont précisées. Les diverses utilisations pédagogiques du jeu sont ensuite étudiées. Enfin nous recensons divers prolongements du jeu proposé.

IV.1. Motivations pédagogiques

Le thème abordé:

La programmation nécessite un comportement systématique, méthodique. Il ne s'agit pas ici de parler de la méthode X de programmation, mais plutôt de méthode comme ensemble de comportements de base entrant dans le processus de construction de programme. C'est pour éviter ce genre d'ambiguïtés que les auteurs d'Anna Gram [Gram 86], étudiant les principaux raisonnements qui sont à la base de la construction de logiciels, emploient les termes de stratégie, de tactique, et de paradigme plutôt que le mot méthode.

Nous nous référons aux composants ainsi repérés par Anna Gram, pour expliciter les éléments de méthodes que nous souhaitons faire pratiquer par nos étudiants. Il s'agit des tactiques de "décomposition" (notamment mise en œuvre par la stratégie de "structuration"), de "réutilisation" et de "spécification".

L'objectif de la décomposition est de diminuer globalement la complexité du problème en introduisant un ensemble de sous-problèmes intermédiaires plus simples, dont les solutions peuvent être combinées pour fournir des solutions au problème de départ. La décomposition est notamment appliquée dans la stratégie de structuration qui vise à mettre l'énoncé sous une forme arborescente de sous-énoncés exprimant l'héritage de propriétés.

La réutilisation a pour objectif d'exploiter au mieux tout travail déjà fait. On peut réutiliser tout ou partie d'un programme ou d'une analyse.

Spécifier consiste à modifier la forme d'un énoncé pour permettre le développement envisagé. Les défauts à éviter lors d'une spécification sont l'ambiguïté (une phrase peut être comprise de plusieurs façons), la contradiction (des définitions ou des demandes incompatibles entre elles), la sur-spécification (une phrase décrit plus une solution que le problème lui-même ou encore certaines informations sont inutiles pour décrire le résultat), la sous-spécification (certaines informations essentielles sont absentes).

L'énoncé résultat de la spécification doit permettre le point de vue, la compréhension et l'accord des personnes qui se situent en amont de ce travail de spécification et permettre à ceux qui se situent en aval, de dériver des solutions.

Le travail de spécification s'applique à un énoncé informel (ou pas assez formel) et nécessite donc une part importante d'invention. Pour exprimer une spécification, on utilise un langage naturel ou un langage formel qui n'offre toutefois pas toutes les garanties souhaitées.

Pour Anna Gram, spécifier consiste d'abord à désigner des entités, les choisir et les nommer. Il s'agit ensuite éventuellement de typer ces objets, c'est-à-dire d'en

décrire les propriétés utiles à ce niveau de l'analyse, grâce à un langage d'expression des propriétés. Représenter consiste alors à choisir, pour certaines entités, les types, les relations et le moyen d'expression adéquat.

Décomposer, réutiliser, spécifier,... quelle importance prennent ces éléments de méthode au cours du développement d'un logiciel ? Référons-nous au cycle de vie d'un logiciel. Il existe en fait divers modèles de cycle de vie dont Sommerville résume l'esprit en explicitant les 5 étapes suivantes [Sommerville 88]:

1. analyse et définition des besoins
2. conception du système et du logiciel
3. réalisation et tests unaires
4. tests du système
5. utilisation et maintenance

Décomposition, réutilisation et spécification n'apparaissent naturellement pas comme étapes dans le cycle de vie. Ce sont plutôt des mécanismes, mis en œuvre à différentes reprises, dans différentes étapes. Quelle importance attacher à la pratique de ces mécanismes dans le cycle de vie d'un logiciel ?

Boehm dans son article "Les facteurs du coût du logiciel" [Boehm 82] nous donne une réponse. Les tactiques de décomposition, de réutilisation et de spécification sont incluses dans ce qu'il nomme méthodes modernes de programmation: analyse et conception descendantes, représentation structurée de la conception, développement progressif et descendant, programmation structurée,... L'adoption de ces pratiques modernes de programmation permet, selon de nombreuses sources, des gains impressionnants de productivité. Boehm apporte toutefois une modération à ces chiffres en soulignant qu'il est difficile de faire nettement la part des gains dûs réellement à ces pratiques de ceux dûs à des facteurs qui leur sont liés (personnel plus qualifié, meilleurs outils logiciels,...).

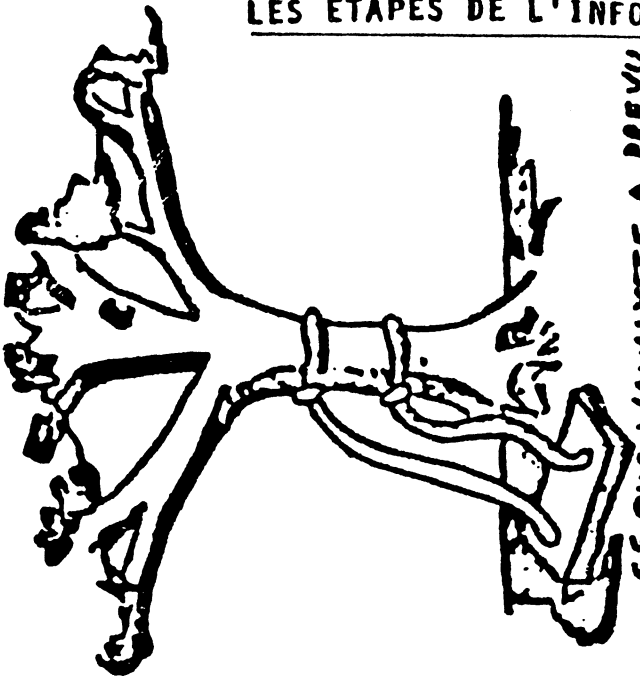
La spécification est utilisée notamment dans les étapes d'analyse et de définition des besoins avec l'établissement du cahier des charges et dans l'étape de conception. Le même genre de travail se reproduit lors de ces deux étapes, mais avec un niveau de détails plus grand. Les spécifications s'affinent jusqu'au produit.

Sigwart et Van Meer soulignent la difficulté de ces deux premières étapes que constituent l'analyse et la définition des besoins, et la conception. Que spécifier et comment, pour être sûr que le programme final répondra bien aux réels besoins des utilisateurs ? Pour eux, ce problème relève plus de l'art que de la science [Sigwart 86]. Toutes les recherches sur la preuve de programme reposent sur l'hypothèse d'une spécification correcte et tentent de montrer que le programme répond bien à la spécification. Mais comment s'assurer que le produit spécifié est bien celui qui répond aux besoins des utilisateurs ? Comment créer de telles spécifications ?

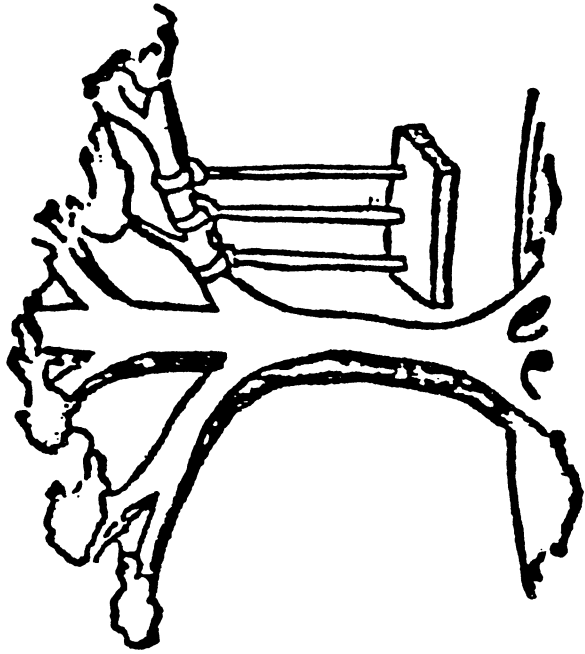
Ce problème qui nous apparaît logiquement, de par sa position-clé en amont de la production, comme un problème fondamental, est aussi reconnu comme tel, économiquement parlant. Boehm montre en particulier la croissance du coût de correction en fonction de l'avancement dans le cycle de vie [Boehm 82]. Dans le cas de gros projets, le coût d'une correction lors de la maintenance est de l'ordre de 100 fois celui de la même correction lors de la rédaction du cahier des charges: d'où le soin à apporter à la rédaction du cahier des charges et aux spécifications.

D'autant plus que les documents de spécification, de plus en plus précis au fur et à mesure qu'on avance dans la production, circulent de mains en mains. Ceci ne va pas sans difficultés, comme en témoigne le document ci-dessous, (dont la source nous est inconnue) qui donne de façon très percutante une vision des problèmes majeurs de spécification, de communication et de compréhension rencontrés lors des différentes étapes de l'informatisation.

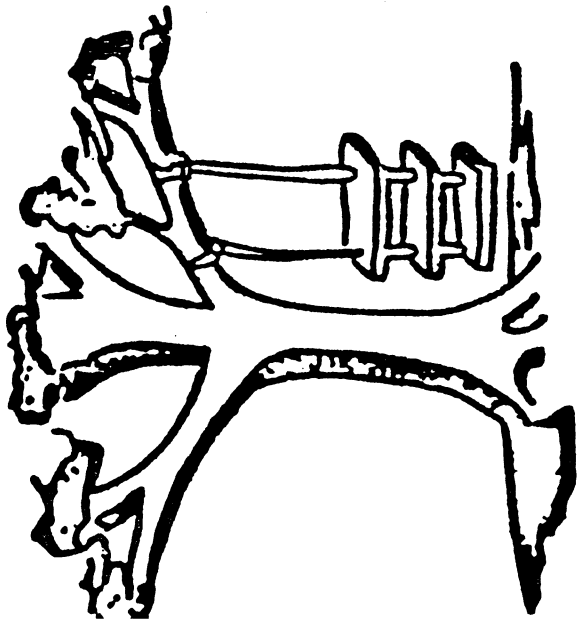
LES ETAPES DE L'INFORMATISATION



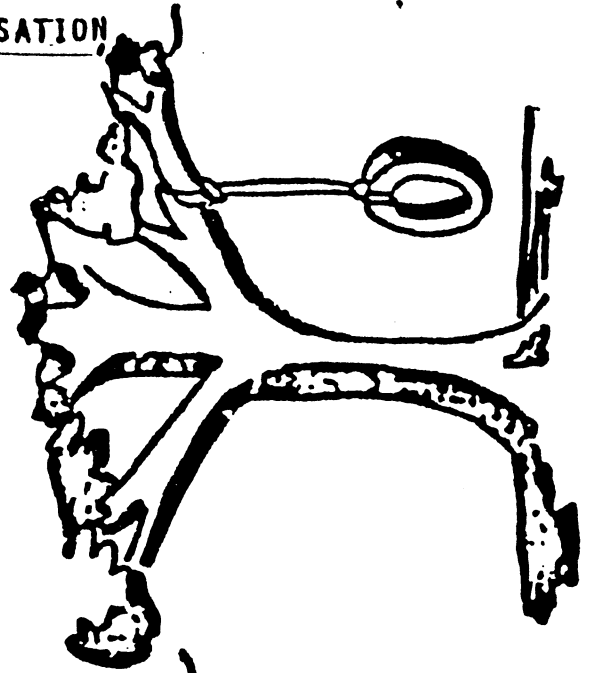
CE QUE L'ANALYSTE A PREVU



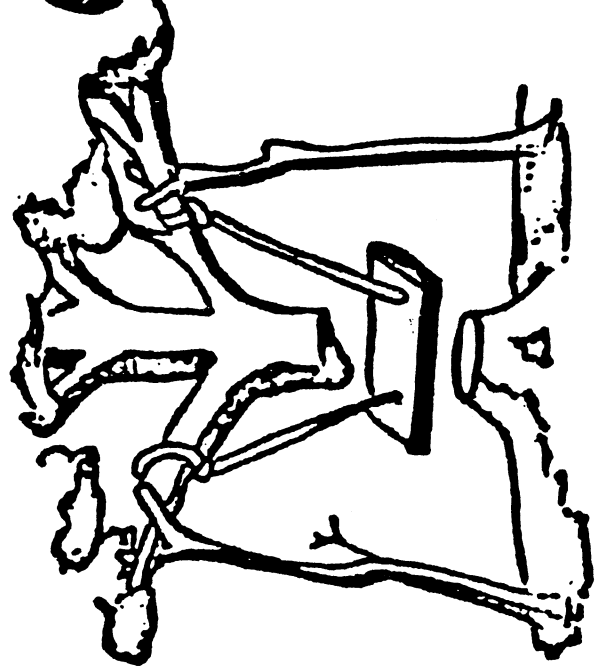
CE QUE TRÉVOIT LE CONTRAT



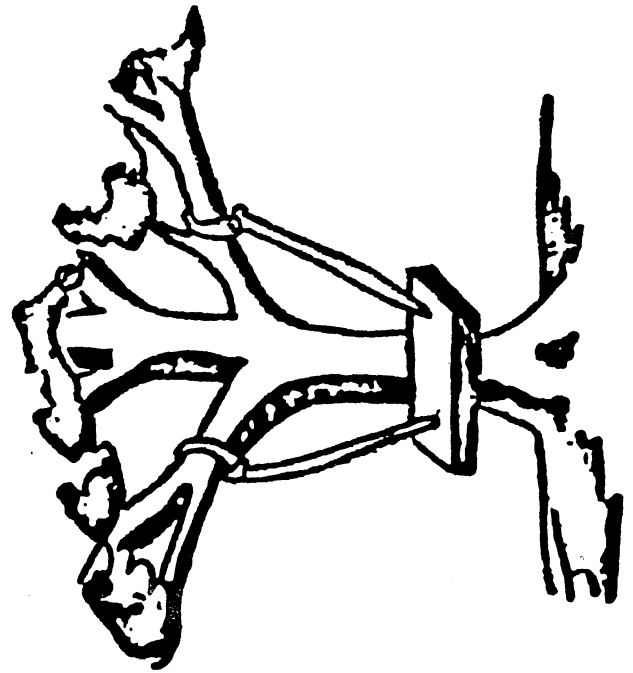
CE QUE DEMANDE L'UTILISATEUR



CE QU'IL FALLAIT



CE QUE LA MISE AU POINT A FAIT



CE QUE LE PROGRAMMEUR A ECRIT

Les difficultés de ce thème dans l'enseignement:

L'enseignement de ces éléments de méthode, qui ne peut être efficace que si les étudiants pratiquent les comportements visés, et en voient le bien-fondé, présente plus ou moins de difficultés selon le public auquel on s'adresse.

- A un niveau avancé:

Les étudiants sont mis en situation de réaliser, la plupart du temps en équipe, des projets de taille significative, qui nécessitent une certaine organisation de leur travail. Nous pensons donc qu'ils ont alors l'occasion à la fois de mettre en pratique les méthodes enseignées, d'en voir le bien-fondé, d'évaluer leur façon de travailler, et éventuellement de la modifier. Encore faut-il être réaliste et se demander si, dans le contexte pressant, voire stressant dans lequel se déroule le projet, les étudiants ont le temps de prendre un peu de recul et de tirer tous les enseignements possibles de leur expérience. En tous les cas, ils n'ont pas la possibilité de rejouer.

Une autre possibilité est de faire travailler les étudiants (non débutants) en cours, sur un "gros" problème. Les étudiants sont ainsi mis devant leurs limites, leurs incertitudes et leurs difficultés, que l'enseignant présent peut mettre en évidence. Celui-ci peut ensuite revenir à des problèmes "scolaires", tout en s'appuyant sur le vécu précédent des étudiants. Son discours est alors écouté et bénéfique. Les méthodes prennent tout leur poids.

Une difficulté particulière apparaît en ce qui concerne le travail de spécification. Les enseignants ont en général tendance à donner aux étudiants des choses bien définies et éliminent ainsi une partie même du travail. Ceci permet aux étudiants de gagner du temps et peut se justifier ainsi, à condition que cela ne soit pas systématique.

Ce problème de spécification initiale a été bien repéré par Sigwart et Van Meer [Sigwart 86]. L'approche qu'ils utilisent dans leur université consiste à donner aux étudiants un projet vaguement défini, et à leur donner la possibilité d'avoir des entrevues avec de futurs utilisateurs du produit à réaliser. Les principaux enseignements que tirent les étudiants d'une telle expérience sont les suivants:

- il est nécessaire de préparer l'entrevue en préparant au minimum une liste de questions.
- l'utilisateur, contrairement à l'enseignant, n'a pas tendance à limiter le problème mais peut se révéler gourmand (pourquoi ne le serait-il pas si les problèmes de délai et de coût ne sont pas abordés par les étudiants ?).
- il ne faut pas oublier d'accompagner chaque proposition de discussions sur le coût et le délai qu'elle nécessite.
- l'utilisateur n'est pas un informaticien, il est de la responsabilité de l'informaticien de parler un langage compréhensible par l'utilisateur.
- il est important de se faire une idée de l'utilisateur quant à sa compétence, son attitude face au projet (favorable, hostile, indifférent), de savoir que plusieurs utilisateurs peuvent avoir des demandes contradictoires, et qu'il s'agit de satisfaire la personne qui signe.
- des aspects subjectifs (aspects physique ou vestimentaire, apparence des documents présentés,...) peuvent avoir une influence sur l'acceptation du projet.

Cette approche paraît réaliste. Le rapport entre ce qui est appris et le temps passé pour l'apprendre semble très performant. Les erreurs faites en classe sont en général des erreurs qu'on ne fait qu'une fois. Les informaticiens ainsi formés s'en montrent satisfaits: ils savent mieux quoi attendre des utilisateurs.

- Au niveau débutant:

Il s'agit là aussi de faire pratiquer les étudiants sur des exemples significatifs, en mettant en évidence la démarche utilisée pour parvenir à une solution. Le premier souci n'est pas l'efficacité de l'algorithme obtenu mais la clarté de son

analysé ainsi que sa correction et sa lisibilité. Aussi, on note une même façon de procéder pour la plupart des "écoles": les notions et techniques enseignées sont abordées par des petits problèmes exemplaires bien choisis. Chacun est consacré à un petit nombre de sujets, de manière à sérier les difficultés rencontrées. Ceci a une incidence sur le choix de l'énoncé, qui doit conserver une certaine simplicité éventuellement au détriment de son réalisme [Scholl 88].

Cette idée convient tout à fait à la présentation des techniques puisque chacune d'elles est présentée dans un contexte d'application. Elle est aussi utilisée pour la présentation des méthodes. Par exemple, l'analyse descendante est introduite dans [Arsac 80] par l'exercice "Quel jour de la semaine tombe la fête de Noël ?", dans [Ducrin 84] par l'exercice "Calcul de la paie d'un ouvrier", dans [Lucas 83] par l'exercice "Nombre d'occurrences du couple 'LE' dans un texte". Mais la difficulté est de faire pratiquer ces démarches systématiques, de manière satisfaisante, sur des exercices aussi simples. La spécification et l'analyse semblent artificielles lorsqu'un énoncé est trop simple.

Le premier vrai problème en programmation, c'est la maîtrise de la complexité: il faut faire face à trop de détails à la fois [Arsac 80]. Or l'enseignant doit convaincre de cette complexité pour que le débutant puisse et veuille pratiquer dès le départ une conduite rigoureuse.

Une telle pratique est difficilement réalisable au niveau d'un public débutant. Les exemples et exercices proposés ne parviennent pas à le convaincre, puisqu'il peut parvenir à un résultat correct (qu'il estime correct !) sans avoir utilisé les méthodes enseignées. Elles lui apparaissent alors comme une rigidité de théoricien, comme des "marottes" d'enseignant. La tentation est trop grande de vite griffonner quelques instructions, de les rentrer en machine, et de voir la réponse de l'ordinateur. Car en informatique, c'est l'ordinateur qui domine dans l'esprit des étudiants [Pair 88b]. La référence pour le débutant, ce n'est pas ce que pense

l'enseignant de son travail, mais c'est bien l'ordinateur: "ça tourne !". Son but est d'aligner très vite, ligne après ligne, les instructions qui composeront son programme, en commençant naturellement par écrire le "début" ou "begin" fondamental ! Il n'a pas envie de prendre conscience de la démarche qu'il a utilisée, ni même de la provenance de ses erreurs. L'enseignant fait remarquer une erreur ? "Ah oui, mais si je mets «+1» ici, ça marche" !

Lorsque les projets consistants arrivent plus tard dans le cursus, le débutant qui n'a pas attaché assez d'importance au discours de l'enseignant, et qui a malgré cela programmé les premiers travaux pratiques, a perdu plusieurs mois et ne parvient souvent pas à combler son retard.

L'enseignant a donc deux préoccupations contradictoires lorsqu'il commence un cours de programmation: présenter les concepts de base à l'aide d'exercices presque simplistes mais accessibles, et concrétiser ces concepts à l'aide d'un problème réel mais quasiment infaisable par les débutants.

Nous pensons comme C. Pair qu'il est souhaitable de s'appuyer sur une expérience de l'apprenant et de ne rien présenter de l'extérieur qui ne réponde à un problème rencontré au cours de cette expérience [Pair 88b].

Nous voulons donc imaginer une activité permettant au débutant de comprendre la nécessité profonde d'une attitude méthodique et de transformer son comportement naturellement empirique en expérimentant les méthodes proposées sur des problèmes significatifs.

IV.2. Du changement de domaine au jeu de rôles

Il s'agit de proposer au débutant de mener à bien une sorte de projet, mais en omettant tous les détails de programmation. En fait nous voulons modéliser les

différentes attitudes d'un programmeur lors de la résolution d'un problème et en proposer une expérimentation.

Dans ce but nous voulons isoler des étapes dans le travail du programmeur, à savoir: l'étape de spécification initiale du problème, l'étape d'analyse, et enfin l'étape de programmation proprement dite. Pour formaliser ces différentes tâches, nous souhaitons les faire faire par trois personnes différentes, dialoguant et s'échangeant des documents formels.

Vers un changement de domaine

La façon la plus évidente d'offrir un travail sur la complexité d'une analyse sans aborder les techniques de la programmation est de remplacer ces techniques non encore enseignées par d'autres techniques connues, donc de changer de domaine.

Le débutant mettra en pratique, dans ce nouveau domaine, des attitudes similaires à celles que l'on demande à un programmeur. La réflexion sur ces comportements se fera au cours de cette pratique, et l'enseignant établira l'analogie avec la programmation au cours de son enseignement.

L'idée d'un changement de domaine n'est pas nouvelle: E. Dijkstra expliquait jadis les notions fondamentales de l'algorithmique à travers l'épluchage des pommes de terre [Dijkstra 71]; le premier proverbe de H. Ledgard est illustré par la cuisson d'un poulet [Ledgard 75]; J. Biondi et G. Clavel présentent l'analyse descendante à l'aide du problème de l'assemblage de documents [Biondi 84]...

Le domaine choisi doit être suffisamment connu du public visé, et c'est une question de culture. Choisir un domaine dont la connaissance serait insuffisante nous obligerait à enseigner les techniques nécessaires avant de pouvoir faire

pratiquer les comportements sur lesquels nous voulons faire réfléchir. Alors mieux vaudrait rester dans le domaine de la programmation !

Du changement de domaine au jeu

Un inconvénient du changement de domaine est de placer abruptement l'étudiant dans un univers différent, le coupant de la réalité immédiate. Cela risque de le dérouter. L'étudiant peut avoir alors du mal à se situer, et ne plus savoir sur quel pied danser. Il risque de s'interroger, d'hésiter, de rester perplexe, de se regarder agir... Pour accepter l'activité proposée, il va sans doute chercher trop vite l'analogie avec quelque chose qu'il maîtrise mal, au lieu de vivre d'abord l'expérience proposée. Il risque de vouloir trop vite intellectualiser les choses sans vraiment les vivre.

Or, nous voudrions d'emblée l'impliquer pleinement dans cet univers, lui faire vivre des expériences, selon sa propre personnalité. Ce n'est qu'au fur et à mesure des situations vécues qu'il pourra établir vraiment l'analogie et en tirer les enseignements.

Mais ne pourrait-on pas faire justement appel au jeu pour aller au-delà de cet obstacle ? Le jeu pour peu qu'il soit accepté permet justement cet engagement, cette confiance nécessaires. Il libère l'étudiant qui entre alors franchement dans l'univers séparé du jeu pour y vivre ses expériences. L'intérêt de mettre ici l'étudiant en situation de jeu est de faire apparaître ou de renforcer une motivation devant un type d'activité qui risque de le dérouter à première vue. Notons que l'aspect ludique est d'autant plus naturel que le changement de domaine introduit par lui-même une fantaisie dans le travail.

Le jeu va aussi permettre à celui qui est devenu "joueur" d'agir spontanément, selon sa propre personnalité, comme il le ferait plus tard en situation de projet et

non d'agir tel un robot, seulement en fonction des recommandations de l'enseignant.

Nous venons de voir que le jeu arrive à point nommé pour pallier aux inconvénients présentés par le changement de domaine proposé. Naturellement le jeu présente aussi d'autres avantages pour un apprentissage, avantages déjà précisés (chapitre 2) et sur lesquels nous ne revenons pas ici.

Du jeu au jeu de rôle

Nous voulons demander au débutant de vivre certains comportements, donc de tenir certains rôles. Les comportements du programmeur sont différents dans les phases d'analyse et dans les phases de programmation proprement dite. Nous souhaitons différencier nettement ces attitudes, même si elles appartiennent, à des moments différents, à une même personne. Pour cela, nous proposons plusieurs rôles. Enfin, soulignons que la programmation est un travail d'équipe qui nécessite une collaboration entre les différents intervenants.

Plusieurs personnes qui tiennent des rôles différents, et qui collaborent dans un jeu, cela conduit à la définition d'un **jeu de rôles**. Nous avons déjà vu qu'un jeu de rôles réunit un meneur de jeu et des joueurs. Il s'agit pour chacun de se mettre dans la peau d'un personnage, et de vivre à travers lui une aventure, dans le monde fantaisiste mais cohérent qui lui est proposé. L'objectif est collectif, il s'agit de mener à bien une mission précise [Paunin 86].

Le jeu de rôles fait partie de la catégorie "simulacre" dans la répartition des jeux de Caillois [Caillois 67a] déjà présentée. Or, vu la nature de l'activité de programmation, nous avons établi précédemment que les jeux de simulacre peuvent apporter une aide dans son apprentissage. Rien ne s'oppose donc maintenant à la définition de ce jeu de rôles reposant sur un changement de domaine.

IV.3. Un jeu de rôles dans l'enseignement de la programmation

Nous avons choisi le domaine de la bande dessinée. Le thème du jeu est la création d'une page de bande dessinée. Le jeu ne consiste pas en une simulation effective de l'activité de création d'une bande dessinée (et ne s'adresse pas a priori aux spécialistes) mais en la simulation de l'activité de programmation schématisée dans le domaine de la bande dessinée [Guéraud 88].

Le domaine choisi: la bande dessinée

Les étapes de la création d'une bande dessinée ([Duc 82], [Beaumont 84], [Faraut 88]) présentent des analogies avec les étapes de la construction d'un programme. Ce domaine particulier a été choisi d'une part en fonction de cette analogie avec l'activité de programmation, d'autre part en fonction du public visé. Nous faisons l'hypothèse que le monde de la bande dessinée est suffisamment banal et connu d'un jeune public français. Il ne s'agit pas de dire que chacun pourrait être un professionnel de bande dessinée, mais que les concepts et le vocabulaire de base sont connus: chacun sait ce qu'est une bulle, une vignette, un gros plan...

Mise à part cette analogie au niveau de la construction des produits (bande dessinée et programme), nous remarquons qu'une bande dessinée raconte une histoire qui se déroule dans le temps, de la même façon qu'un programme s'exécute dans le temps. De même, une bande dessinée raconte une interaction entre divers personnages, alors qu'un programme peut être considéré comme une interaction entre divers objets.

Principe du jeu:

Rappelons avant tout que notre objectif pédagogique est que l'étudiant s'approprie les éléments de méthode que sont la spécification, la décomposition et la réutilisation, qu'il en ressente la nécessité et les difficultés.

Il s'agit d'un jeu à trois joueurs se répartissant les rôles suivants :

- le client: il propose le synopsis, c'est à dire le résumé de l'histoire à raconter et présente les personnages principaux de l'histoire. Il propose donc le *cahier des charges*.
- le scénariste: il découpe l'histoire en différentes étapes correspondant chacune à une image de bande dessinée. Il définit ainsi, à partir du synopsis, le scénario et le découpage graphique de la page. Il fait donc l'*analyse* du synopsis et propose une *structuration* en images élémentaires.
- le dessinateur: il réalise la page de bande dessinée à partir du scénario et du découpage, en utilisant une bibliothèque de dessins. Il *programme* (dessine) l'*algorithme* (le scénario).

Les trois joueurs doivent coopérer dans le but de produire une page de bande dessinée qui réponde à la demande du client, tout en préservant leur intérêt propre (intérêt financier, intérêt de la création, rapport temps/prix...).

Chaque joueur dispose d'un micro-ordinateur; les trois ordinateurs sont connectés. La communication entre les joueurs s'effectue seulement via ce réseau avec la hiérarchie client-scénariste-dessinateur. Il y a deux types de communication. D'une part, les joueurs échangent des documents (synopsis, scénario...) et fixent des accords (délai, prix...). D'autre part, ils communiquent librement mais toujours

selon la hiérarchie et via le réseau, pour demander des précisions, négocier, demander un avis, consulter un "fournisseur"... N'oublions pas que si les paroles s'envolent, les écrits restent ! Personne n'a le droit de revenir sur un accord déjà conclu. Si l'un des acteurs change d'avis, il doit renégocier et de nouveaux accords doivent être formulés. A tout moment chacun peut poser une question, ou négocier une modification.

Déroulement du jeu:

Les joueurs assistent à une présentation du jeu et des fonctions de chaque personnage. Après cette présentation, les joueurs se mettent d'accord pour la répartition des rôles et le temps consacré au jeu. Chaque joueur définit les différents éléments qui caractérisent son rôle (niveau dans sa profession, spécialisation éventuelle dans un genre, ancienneté...). Il se retrouve alors dans le lieu d'exercice de sa fonction.

Chacun dispose sur son écran d'un environnement de jeu dont une grande partie est commune aux 3 joueurs: téléphone (pour contacter les autres joueurs), horloge, divers indicateurs, le petit manuel des règles d'or de la bande dessinée, un catalogue de produits réalisés... L'environnement de création est spécifique à chaque rôle; il est constitué d'outils permettant la création des documents dont le joueur a la responsabilité, et d'outils de consultation pour les documents fournis par les autres joueurs.

Et la partie peut commencer ! Le bilan en sera fait à la fin du jeu, en proposant un score global prenant en compte l'état d'achèvement de la page dessinée, le temps nécessaire pour parvenir à cet état, les bilans financiers des joueurs, la satisfaction de chacun par rapport au déroulement de la partie et par rapport au comportement des autres.

Au cœur du jeu:

Qui dit jeu de rôle, dit meneur de jeu. Sa présence est nécessaire pour donner les informations, arbitrer, animer la partie. Il doit intervenir, à bon escient, pour que le jeu se déroule bien; il a la charge de relancer la partie si elle manque d'intérêt, de débloquer la situation si un blocage se produit, d'imposer un juste milieu entre l'attitude la plus collective qui risque d'engendrer des parties un peu trop sages et le chacun pour soi absolu qui tournera au règlement de compte permanent [Paunin 86]...

Le meneur de jeu doit avoir la maîtrise du jeu. Si un joueur n'a pas un comportement efficace, le meneur de jeu doit intervenir pour que ce joueur soit mis devant ce manque, cette erreur. Il ne s'agit pas d'aller le voir et de le sermonner, mais bien de le mettre dans une situation concrète qui lui convainque de changer d'attitude. Le joueur sera alors pénalisé dans le jeu par une perte de temps ou d'argent, non pas sur une décision arbitraire du meneur de jeu, mais par la conséquence même de son erreur... Tel autre joueur est prévoyant, il sait s'organiser et fait preuve d'une efficacité remarquable dans l'organisation de sa tâche, le meneur de jeu va agir de telle sorte qu'il puisse en tirer tous les bénéfices.

Le meneur de jeu peut intervenir soit directement auprès du joueur concerné, soit auprès d'un de ses coéquipiers pour influencer sur le déroulement du jeu. Il doit toujours le faire en termes du jeu afin de ne pas briser la fiction. Avec l'intention de faire modifier au client les caractéristiques d'un personnage, il peut par exemple lui faire parvenir le message suivant: "Les éditions Dunat sont au courant de votre projet. Elles sont prêtes à vous intenter un procès parce que votre personnage Achille ressemble trop à celui dont elles ont l'exclusivité des aventures !". Il est probable que le client demande alors à ses coéquipiers une modification de l'allure d'Achille, ce qui va peut-être poser quelque problème au dessinateur s'il a travaillé constamment avec un niveau maximum de détails vignette après vignette, sans

structure sur les composants décors et personnages qui y interviennent... Le meneur de jeu doit avoir une vue claire de la partie qui se déroule, ainsi qu'une imagination prompte et fertile qui lui permette l'improvisation !

Pouce: ne bougeons plus !

Le jeu que nous venons de présenter a été conçu avec l'aide des concepts et résultats présentés au chapitre 2. Arrêtons-nous un instant et jetons un coup d'œil en arrière afin de voir si l'activité proposée relève bien du jeu.

Regardons tout d'abord si l'activité proposée satisfait aux caractéristiques des activités de jeu. L'activité est libre: l'étudiant pourra refuser de jouer, aucune sanction ne sera alors prise. Elle est nettement séparée de la vie réelle des étudiants, en particulier grâce au changement de domaine. Elle est improductive en terme de la vie réelle: ni notes, ni production de programmes,... la BD réalisée n'a aucune retombée sur leur année d'études, seules leurs compétences peuvent être modifiées au cours du jeu. L'activité est incertaine: va-t-on parvenir jusqu'au bout de la réalisation de la bande dessinée ? dans quelles conditions ? dans quel délai ? Et quelle va être la bande dessinée produite ?... Aucune réponse, aucune certitude n'est possible à l'avance. L'activité définie est de nature fictive: et si on était producteurs de BD... L'activité, pour fonctionner, a besoin aussi, comme tout jeu, à la fois de l'engagement des personnes (j'agis comme si j'étais scénariste ou dessinateur) et d'une certaine distance (mais dans la vie je ne fais pas de BD, je ne suis ni Tabary, ni Goscinny, ni Margerin...)

Mais où se situe le jeu défini dans la classification proposée par Caillois [Caillois 67a] ? De nature fictive, comme tout jeu de rôles, il appartient à la catégorie des jeux de simulacre. Il s'agit bien de devenir soi-même un personnage illusoire, d'agir en tant que tel, avec toute la maîtrise et l'improvisation nécessaires.

Vérifions enfin que le jeu conçu répond bien aux critères repérés par Malone, qui font les "bons" jeux sur ordinateur [Malone 80].

Le défi proposé est explicité clairement dans la mission définie aux joueurs: il faut créer une bande dessinée répondant à la demande du client. Le niveau de défi peut être variable selon qu'on décide ou non de prendre en compte l'aspect délai, coût, qualité de la production,... Des informations concernant l'avancement de la partie ("feedback" informatif) sont données à chacun des joueurs, précisant l'état de ses finances, indiquant le temps écoulé et schématisant l'avancement global de la BD.

La fantaisie apparaît avec le changement de domaine et la bande dessinée. Il s'agit selon la terminologie employée par Malone d'une fantaisie intrinsèque, c'est-à-dire liée à la compétence visée. En effet, réussir à faire une BD répondant à la demande du client dans le temps imparti est lié au fait, dans le domaine de la programmation, de pouvoir s'organiser pour réussir à proposer au client un programme répondant à son problème.

Enfin voyons en quoi le jeu proposé est capable d'éveiller puis de satisfaire la curiosité des joueurs. Divers évènements de type sensoriel peuvent attirer l'attention des joueurs. Ces évènements sont pris en charge sur l'ordinateur par une interface de type Macintosh, et par l'environnement de jeu lui-même. Enfin, la nature graphique de la bande dessinée a aussi un rôle à jouer au niveau de cette curiosité sensorielle.

Pour voir enfin finie la bande dessinée à la création de laquelle le joueur contribue, celui-ci s'efforce de découvrir, bout par bout, les méthodes, les stratégies efficaces pour son jeu. Sa curiosité cognitive est ainsi mise en éveil. Le meneur de jeu a pour rôle de maintenir cette curiosité en éveil en agissant en fonction des

découvertes de l'étudiant. Il agit donc dans le but de montrer à l'étudiant que les stratégies qu'il utilise sont plus ou moins efficaces, qu'elles ont éventuellement des limites, qu'elles ne sont pas toutes puissantes... Le joueur est donc informé de sa propre progression dans le jeu.

Nous pensons en conséquence que les conditions sont favorables pour que les étudiants jouent en pratiquant cette activité. Mais cela, eux seuls seront capables de nous le dire. Le public est seul juge en matière de jeu. Il est difficile, même pour les concepteurs professionnels de jeux, de prédire à l'avance si le jeu va être ou non un succès. Ce qui explique qu'il existe beaucoup de jeux du même genre car il est plus facile d'adapter une idée qui commercialement a fait ses preuves que d'innover avec un nouveau concept et prier pour que ce soit une réussite [Williams 82] ! Témoignent aussi de cette difficulté de grands créateurs de jouets qui n'hésitent pas à créer des écoles où les prototypes de jouets sont mis à la disposition des enfants. Les réactions de ceux-ci sont longuement et soigneusement étudiées avant de lancer une production... ou d'abandonner un projet.

Mais notre objectif n'est pas commercial, et nous ne raisonnons pas en termes de rentabilité économique du jeu. L'objectif n'est pas non plus purement ludique. L'important est que les étudiants pratiquent l'activité que nous leur proposons sous forme de jeu, et qu'ils en tirent les bénéfices que nous escomptons. Ce jeu ne s'intègre pas dans une arcade, mais bien dans un laboratoire d'enseignement de la programmation. Au minimum, envisager de proposer un jeu nous aura permis de nous libérer d'un certain académisme, d'oser un changement de domaine et d'aller jusqu'à choisir un domaine aussi peu "sérieux" que la bande dessinée... en bref de concevoir cette activité qui nous paraît riche d'un point de vue pédagogique, et en tous cas, d'étudier autrement l'activité de programmation. Alors, reprenons la partie, là où nous l'avions laissée...

IV.4. "Toi, Moi et Lui" ou l'analogie avec la programmation

Nous donnons ici une présentation du jeu destinée à donner aux étudiants une image vivante des différents rôles pendant la partie, à leur souffler des conseils et à leur suggérer l'analogie avec la programmation.

Toi

Scénariste, tu viens de recevoir un synopsis des mains du client. Garde-toi bien de foncer tête baissée ! Regarde bien ce document ! Sache y débusquer les imprécisions, les ambiguïtés. N'hésite pas à contacter ce client, à le cuisiner un peu. Il faut que tu en saches le plus possible avant de mettre les mains à la pâte. Après il risquerait d'être trop tard.

Ne te contente pas d'accords susurrés dans l'écouteur, ton client a peut-être la mémoire courte ! Ne t'engage pas dans n'importe quelle aventure, assure-toi d'avoir les ressources (personnelles, financières, humaines) et le temps nécessaire pour en venir à bout !

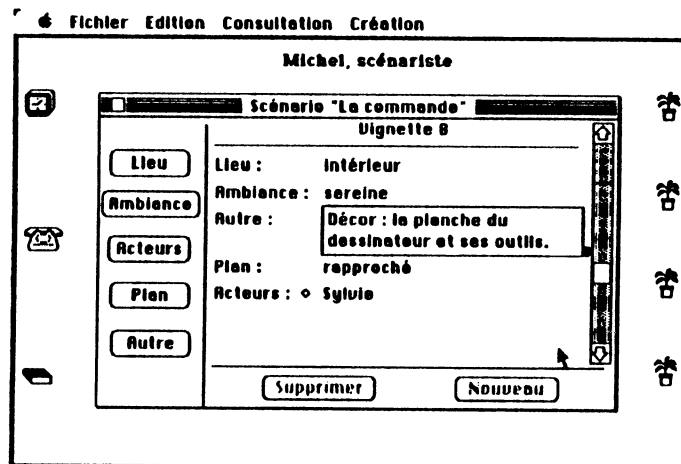
Lorsque tu commenceras à prendre l'histoire à bras le corps pour lui donner vie, ne va pas t'isoler avec le dessinateur dans la grande tour d'ivoire ! Votre création y serait très libre mais vous risqueriez de tomber de haut en retournant proposer votre œuvre au client qui n'y reconnaîtra sûrement pas ses petits !

Tu peux improviser une scène et demander au dessinateur de la mettre en image. Puis une autre, et encore une autre... tacher de recoller les morceaux tant bien que mal... et vite refiler le monstre au client. Tu peux aussi te faire un plan d'action, c'est à dire un scénario complet, demander l'accord du client puis passer le relai au dessinateur. A toi de choisir tes risques !

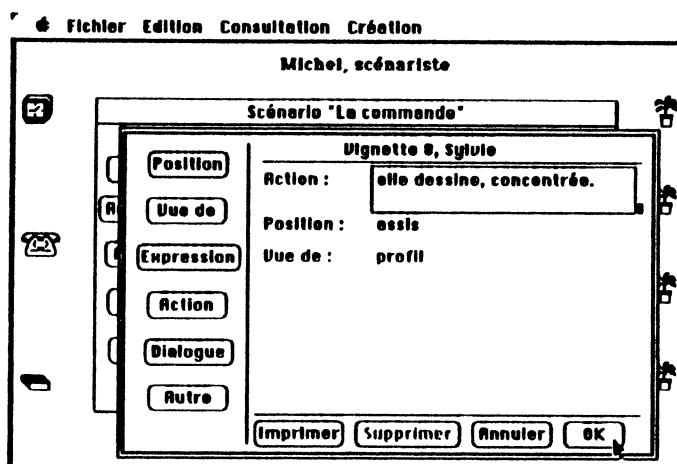
Tu as du mal à trouver du premier coup une décomposition de l'histoire qui soit juste, équilibrée, rythmée ? Mais c'est le contraire qui serait bien étonnant ! Pas de fée secourable ? Alors cent fois sur le métier, remets ton ouvrage !

A l'heure de décrire chaque scène pour le dessinateur, souviens-toi des ennemis qu'il a fallu débusquer dans le synopsis qui paraissait pourtant bien innocent. Essaie de ne pas laisser s'installer leurs complices dans ton propre scénario.

A cette étape, tu trouveras des outils. Ils t'aideront à ne pas vouloir exprimer tous les niveaux de détails à la fois. Ils te suggèrent de planter d'abord le décor général de la scène: où, qui...



Ensuite, ils te suggèrent de donner son rôle précis à chacun des acteurs de la scène.

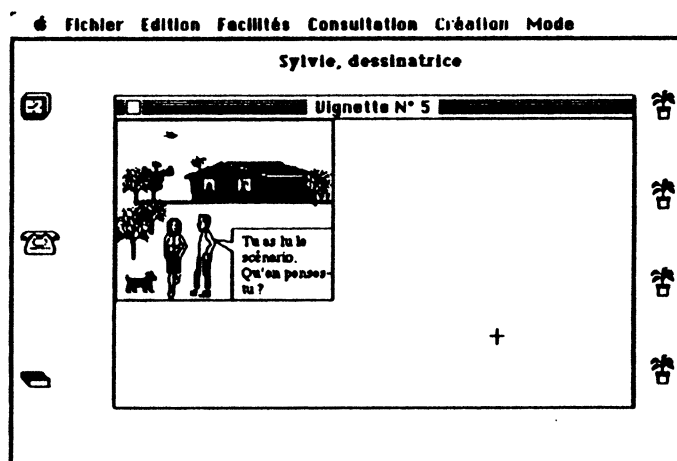


Ces outils sont en général spécialisés parce qu'ils répondent à un besoin habituel du scénariste. Résiste à l'envie de vouloir te servir de tous, ne choisis que ceux qui répondent à ton besoin particulier. Enfin, tout au fond de l'atelier, tu trouveras un dernier outil, très rustique; ne le méprise pas, il te permettra d'exprimer certaines choses non prévues mais essentielles pour ta scène !

Et maintenant tu peux ôter ton masque ami scénariste, nous t'avons vu à l'œuvre, nous avons reconnu tes soucis d'analyste !

Moi

Moi, je suis un dessinateur un peu particulier. Mes outils ne sont pas le crayon et la gomme, mais les ciseaux et la colle. Mes dessins sont tous constitués à partir de dessins de base que je trouve dans ma banque de dessins ! Cela perd en originalité, mais est assez efficace. Pourquoi toujours tout réinventer ?



Je suis un dessinateur malin ! Je ne l'ai pas toujours été mais depuis que je joue à ce jeu-là, j'ai beaucoup progressé ! J'ai appris à ne pas m'occuper de tous les niveaux de détails à la fois ! Par exemple, j'ai pris l'habitude de préparer à part certaines portions de dessins. C'est plus facile, plus clair et je gagne en souplesse car je sais bien qu'on finit toujours par me demander des modifications ! De plus, j'économise ainsi mes efforts, dans le cas de certains décors qui figurent plusieurs fois dans la page, même si je dois changer les dimensions de ces décors, car mes outils me le permettent !

Une petite chose encore ! Je n'oublie pas de prévoir la place des bulles au fur et à mesure de la construction de l'image, sinon je n'arrive plus à les insérer ! Il paraît que je ferais un bon programmeur ! C'est «lui» qui me l'a dit.

Lui

Sans lui pas de bande dessinée. C'est sur son idée, et grâce à la commande qu'il passe, qu'elle pourra naître !

Il doit savoir exprimer ses besoins le plus clairement possible: quels sont les personnages de l'histoire et que leur arrive-t-il ?

The screenshot shows a software window titled "Jean-Paul, client" with a menu bar containing "Fichier", "Edition", "Consultation", and "Création". Inside the window, there is a sub-window titled "Personnages". On the left side of the "Personnages" window, there are buttons for "Age", "Sexe", "Taille", "Corpulence", "Profession", and "Autre". The main area of the "Personnages" window contains the following fields and options:

- Nom :
- Age :
- Sexe : M F
- Taille : Petit Grand
- Corpulence : Mince Gros
- Profession :

At the bottom of the "Personnages" window, there are two buttons: "Supprimer" and "Nouveau".

Il a des idées mais aussi des exigences. Il sait selon les cas être souple ou intransigent.

Il se doit d'assurer un suivi du projet: il est bon de relancer de temps en temps ses fournisseurs, de savoir où ils en sont. Sans quoi, la réalisation risque de s'embourber, la motivation des fournisseurs risque de diminuer ou bien la BD finie risque de ne répondre en rien à la commande passée !

Les arrière-pensées de l'enseignant de programmation

Ah, chers étudiants, dites-moi comment vous jouez, je vous dirai comment vous programmez !... Voulez-vous connaître le fond de ma pensée ?

Le client a une histoire à faire raconter en bande dessinée, de même qu'en programmation, le client a un problème à faire résoudre par programme. Dans les

deux cas, il "spécifie" donc le travail à faire. Pour cela, en BD, le client nomme les personnages, c'est-à-dire les "désigne"; ensuite il décrit les caractéristiques de chacun, c'est-à-dire qu'il les "type". Il décrit alors brièvement l'histoire, c'est-à-dire l'interaction entre les personnages, autrement dit il "représente" l'histoire.

Le scénariste reçoit les documents établis par le client. Comme l'analyste recevant un cahier des charges, il doit se plonger dans les documents, en comprendre le sens, en repérer les défauts ("ambiguïté", "contradiction", "sur-spécification", "sous-spécification") afin de les faire éliminer par le client, vérifier la faisabilité du travail spécifié. Ceci étant fait, le scénariste doit imaginer la succession d'images qu'il va utiliser pour raconter l'histoire, comme l'analyste doit imaginer l'algorithme qu'il va utiliser pour résoudre son problème. Dans les deux cas, il s'agit de réduire le problème posé en le "décomposant": ce travail d'analyse requiert du temps, de l'imagination, et de la patience... Il doit aboutir à une solution, spécifiée dans le domaine de la BD par un découpage graphique de la page en vignettes, et par un scénario qui décrit le contenu de chaque vignette. En ce qui concerne la programmation, la solution est spécifiée par l'algorithme général répondant au problème, et par la spécification des procédures ou fonctions auxquelles l'algorithme général fait référence. Notons que ces nouvelles spécifications sont à la fois plus précises, et plus techniques que les premières (précision des plans employés (BD), ou du type d'algorithme de tri à utiliser, du type de représentation des données à employer (programmation)).

Le dessinateur (respectivement le programmeur), reçoit ces documents des mains du scénariste (respectivement l'analyste). A son tour de comprendre ces spécifications, de les clarifier. Comme le programmeur, le dessinateur de "Toi, Moi et Lui" met en œuvre la tactique de "réutilisation". Il utilise une bibliothèque de dessins, dont il peut paramétrer les dimensions, modifier les détails. Afin d'être plus efficace, il se crée sa propre bibliothèque à partir des décors et des personnages

qu'il compose; il peut alors par exemple réutiliser dans différentes vignettes, les décors ainsi constitués, changer la couleur du pantalon à un personnage ainsi préparé, sans devoir rajouter de nouveau au personnage original sa fausse barbe et ses bretelles !... Il procède avec méthode, comme l'a fait avant lui le scénariste, plantant cette fois concrètement le décor avant de placer les personnages...

IV.5. Toi, Moi, Lui ou un autre: quel rôle pour l'ordinateur ?

Notre postulat de départ dans le laboratoire Arcade est d'utiliser l'ordinateur pour proposer aux étudiants des activités venant en complément de l'enseignement traditionnel. Le jeu de rôles créé est donc naturellement proposé sur ordinateur. Nous allons voir qu'il ne s'agit pas seulement d'être conforme au postulat de départ, mais que l'ordinateur a réellement un rôle à jouer dans cette activité, au service de tous les intervenants.

Des outils simples et maniables: vers une création plus facile

L'ordinateur permet de fournir des outils logiciels simples et maniables pour la création des documents permettant de produire une BD. Le logiciel "Toi, Moi et Lui" est composé de trois applications: l'application "scénariste", l'application "dessinateur" et l'application "client". Chacune propose des outils facilitant la production.

Outre cet aspect purement pratique, ces outils permettent de "débloquer" le joueur, de libérer son imagination. Inventer une histoire ? Créer des personnages ? Diable ! Je ne sais pas faire...

Le crayon laisserait la feuille de papier désespérément blanche car ce mode d'écriture sous-entend (sous peine de gommage, de ratures) un choix définitif donc génial du premier coup. Par contre, l'outil logiciel proposé et l'ordinateur, de par

leur facilité d'emploi et leur interactivité, permettent de débloquent la situation. L'étudiant essaie, tâtonne, puis se prend au jeu, va jusqu'à donner un "look" à son personnage, des tics... Cela ne résoud pas tous les problèmes de créativité, mais cela la facilite, la libère...

Les outils proposés pour la création des personnages, la création du synopsis, ou celle du scénario procèdent d'une même idée. L'utilisateur dispose d'un jeu de "boutons" lui permettant de définir les caractéristiques qu'il juge nécessaires de définir à ce niveau. Pour définir la caractéristique désignée, un certain nombre de fenêtres de dialogue vont lui être proposées permettant de définir son choix, choix qui sera alors affiché au niveau de départ.

L'outil proposé pour la création du découpage permet de tracer très facilement les vignettes dans la page réduite proposée. Les vignettes sont alors automatiquement numérotées selon l'ordre habituel de lecture (haut-bas, gauche-droite). Là encore, il est facile d'essayer, d'effacer...

Les remarques précédentes quant à la libération de la créativité sont aussi vraies, et de manière encore plus cruciale, quand il s'agit de dessiner. Je ne sais pas dessiner mais l'outil plus les banques de dessins me permettent de faire des dessins corrects. J'accepte de dessiner à l'aide de logiciels de dessins sur ordinateur beaucoup plus facilement qu'avec un papier et un crayon. Et l'interactivité de ceux-là me permet de procéder par approximations successives de manière propre. L'application "dessinateur" propose quelques fonctionnalités simples: un outil permettant de sélectionner un dessin; les fonctions couper, copier, coller, effacer, déplacer, déformer, inverser, agissant sur un objet sélectionné; l'outil permettant de tracer des lignes pour les bulles ou de faire de petites retouches; l'outil permettant la frappe de texte. L'application met à la disposition de l'utilisateur l'accessoire de bureau Art Grabber (Macromind Inc.) permettant d'aller "piquer" des dessins dans les banques de dessins proposées, ainsi que l'accessoire Cheap Paint (Macromind

Inc.), reprenant les fonctionnalités principales de Mac Paint (Apple Computer Inc.), et permettant des retouches plus fines, donnant accès à une palette pour les coloriages, ainsi qu'à de nombreuses polices de caractères...

Une formalisation de la communication

L'ordinateur facilite la mise en place d'un protocole de communication. Les joueurs ne sont pas tentés de se passer quelques morceaux de papier griffonnés en guise de spécifications. Les outils facilitent la création, ils ont aussi l'avantage de donner un moule, une forme précise aux documents parce qu'ils sont créés avec une application et doivent être relus par une autre.

Naturellement, en dehors de ces échanges de documents, les joueurs peuvent communiquer par réseau de façon libre et informelle, pour demander une explication par exemple. Mais seuls les documents écrits font foi.

Une aide au meneur de jeu

Le meneur de jeu, nous l'avons vu, doit mener à bien des tâches d'arbitrage, d'animation et de diffusion de l'information. Nous allons voir en quoi l'ordinateur l'aide et le relaie dans ces différentes fonctions.

L'ordinateur a un rôle d'arbitre. Il contrôle certaines cohérences que ce soit à l'intérieur de la création de chacun des joueurs (par exemple, que le scénariste décrive autant de scènes dans le scénario qu'il découpe d'images sur la page graphique) ou entre les travaux des divers joueurs. Il enregistre les documents et accords intermédiaires des partenaires. Il contrôle les différents compteurs relatifs au jeu...

L'ordinateur a un rôle d'animateur. Il génère divers événements (par exemple, au dessinateur: "Vous venez de renverser de l'encre sur votre dessin", "Vous êtes obligé de vous absenter pour cause de maladie"...), de façon aléatoire, ou en fonction de certains aspects du déroulement du jeu. Il peut suggérer des analogies.

L'ordinateur a aussi un rôle d'informateur, tout d'abord par la mise à jour des différents indicateurs qu'il gère. Il est capable de donner des informations de base sur les différents métiers de la bande dessinée et de fournir des exemples de produit déjà réalisés (du synopsis à la page dessinée). Il fournit aussi des conseils sur l'utilisation des outils fournis.

En cela, l'ordinateur est meneur de jeu. Cependant il n'a pas tous les moyens d'un meneur de jeu humain. Par exemple, il ne contrôle pas totalement le respect des règles de communication (un client peut toujours aller discuter de vive voix avec un dessinateur), ni la cohérence totale entre l'histoire demandée au départ et la page finale de dessins (cohérences partielles seulement). Il n'a pas la clairvoyance nécessaire pour pouvoir toujours relancer par ses actions l'intérêt du jeu notamment d'un point de vue pédagogique. L'ordinateur ne remplace pas le meneur de jeu humain, il l'aide et le relaie seulement, ce qui n'est pas négligeable. Il peut ainsi permettre au meneur de jeu d'animer plusieurs parties à la fois.

Nous ne pensons pas confier complètement à l'ordinateur la lourde tâche de meneur de jeu. Pour nous l'approche humaine du jeu, de la partie et des intervenants reste importante. Nous n'imaginons pas que l'ordinateur remplace l'homme, dans sa double tâche de meneur de jeu et de pédagogue. D'autant moins que nous sommes à un stade d'expérimentation et que nous sommes loin d'avoir une expertise en ce domaine. Cette position s'inscrit d'ailleurs tout à fait dans la ligne de pensée du laboratoire Arcade: nous ne voulons pas non plus remplacer l'enseignement

traditionnel par notre laboratoire. L'ordinateur doit seulement aider enseignants et étudiants, dans les domaines où il se révèle commode, voire indispensable, et ce le plus loin possible.

Comment l'ordinateur aurait-il l'expertise et la science infuse quant à l'enseignement de l'informatique alors que l'enseignant d'informatique, si l'on en croit le colloque "Didactique de l'Informatique" [Didactique 88], ne prétend pas l'avoir ? Citons simplement C. Pair dont la communication a pour titre: "Je ne sais (toujours) pas enseigner la programmation", et pour conclusion: "Parfois, je me dis cependant que je pourrais (enfin) aider les élèves et les étudiants à l'apprendre" [Pair 88b].

Pour en revenir à la discussion sur l'ordinateur-meneur de jeu, nous aimerions proposer à la réflexion du lecteur des approches différentes de notre approche actuelle. Citons le jeu d'aventures en conduite de projet (JAC), déjà présenté (chapitre 3), visant à transférer une expérience aux jeunes chefs de projets. Ce jeu fonctionnait sous forme entièrement manuelle en 1986 [Crampes 87]. Un prototype informatisé est en cours de réalisation, intégrant plusieurs systèmes experts: le simulateur, le méthodologue, le pédagogue et les autres, utilisés pour orchestrer l'ensemble. M. Crampes et Y. Quemeneur sont donc allés plus loin que nous, en donnant à l'ordinateur une expertise et une responsabilité plus grandes. Toutefois, étant donné l'aspect psychologique et son importance à l'intérieur du développement d'un projet, il leur a semblé "plus riche de garder une approche purement humaine de ces aspects": le jeu d'aventures est ponctué de réunions menées par un meneur de jeu bien vivant et qui introduit toute sa subjectivité dans l'exercice. Le meneur de jeu constitue le complément informel d'une base de connaissance formelle. L'ordinateur le libère d'un certain nombre de tâches et lui permet de mener plusieurs parties de front.

La revue *Jeux & Stratégie*, en faisant le constat suivant "Les jeux de rôle sur micro existent bel et bien", nous indique une autre approche. L'auteur de cet article indique toutefois que les premiers jeux de rôles sur micro-ordinateur ne représentent que faiblement la qualité du jeu sur table, et que les puristes rechignent [Brassinne 88]. Constatons qu'ils permettent de jouer seul avec son ordinateur en guidant plusieurs personnages à la fois dans l'aventure. En tous les cas, ces jeux de rôles qui réussissent avec pour meneur de jeu l'ordinateur n'ont pas de souci pédagogique; seul le plan ludique les préoccupe. Ils ont aussi comme caractéristiques d'avoir été longuement testés par des joueurs et des meneurs de jeu; leur fonctionnement a été bien repéré avant de donner à l'ordinateur la responsabilité de meneur de jeu. Voilà peut-être autant d'explications à leur réussite, mais peut-être avons-nous aussi beaucoup à apprendre d'eux ?...

D'autres enseignements peuvent aussi nous être apportés par les recherches faites dans le domaine des tuteurs intelligents. L'un des axes qui intéresse ces recherches est centré sur l'expertise pédagogique. M. Vivet nous fait remarquer à ce propos que cette partie pédagogique est celle qui a été abordée le plus récemment dans le domaine des tuteurs intelligents, et qu'elle fait appel à une expertise complexe et difficile à formuler [Vivet 88]. Ceci constitue donc une recherche à suivre...

IV.6. Les différentes utilisations pédagogiques de "Toi, Moi et Lui"

IV.6.1. Quel meneur de jeu ?

Nous avons vu que le meneur de jeu n'a pas un rôle naïf. Il doit avoir la maîtrise du jeu. L'objectif du jeu peut être compris par le meneur de jeu en termes des pratiques nécessaires à la construction d'une BD par 3 acteurs client, scénariste et dessinateur. Il semble toutefois préférable que le meneur de jeu ne perde pas de vue l'objectif réel qui est la pratique de certaines méthodes et de certains

comportements nécessaires à la programmation. Cette double vision lui permettra de garantir que l'intérêt pédagogique de la partie sera maintenu au même titre que l'intérêt ludique.

Face à cette maîtrise nécessaire l'enseignant est donc susceptible de faire un bon meneur de jeu, il saura tirer pleinement profit de la situation qui est en train de se jouer. En outre, l'enseignant disposera de tous les éléments pour discuter ensuite avec ses élèves de ce qui s'est passé au cours de telle ou telle partie, et des leçons que l'on peut en tirer pour la programmation.

Nous pensons que la situation impliquée par le jeu de rôles est suffisamment dépaysante pour que la présence de l'enseignant ne perturbe pas le fonctionnement du jeu; si tel était le cas, il est possible de faire intervenir des enseignants non impliqués dans les filières des étudiants visés.

Le fait d'envisager que l'enseignant soit meneur de jeu n'exclut pas l'utilisation du jeu en libre service dans le laboratoire. Celle-ci sera dans cette éventualité simplement moins spontanée car elle nécessitera de réunir trois personnes dont un enseignant. Cette utilisation reste donc possible, d'autres apparaissent du fait de la présence de l'enseignant; nous en discuterons plus loin.

Mais nous pensons aussi pouvoir confier ce rôle de meneur de jeu à un étudiant, par exemple d'un niveau plus avancé, ayant déjà acquis une expérience significative en programmation, et ayant déjà pris un certain recul par rapport à cette expérience. Nous pouvons facilement imaginer un étudiant de deuxième année, motivé, ayant déjà expérimenté le jeu l'année précédente, prendre le rôle de meneur de jeu pour une partie avec des étudiants de première année. Nous pensons que la partie serait intéressante et profitable pour tous, y compris pour l'étudiant de deuxième année qui aurait à cette occasion une réflexion plus approfondie sur ces éléments de méthodes.

Nous ne voulons pas décourager d'autres éventuels meneurs de jeu. Il faut bien un jour se lancer. Mais il faut savoir qu'on ne s'improvise pas meneur de jeu. Cela se prépare. Il faut pour cela connaître le principe du jeu et son déroulement, comprendre l'analogie avec la programmation, et choisir le scénario qui va guider la partie.

IV.6.2. Meneurs de jeu, créez votre propre scénario !

La distribution des rôles

Commençons par celui du meneur de jeu, dont la première tâche est bien celle de conduire la partie. Dans ce but, et afin de mieux faire partie de la fiction le meneur de jeu peut s'attribuer un certain nombre de rôles secondaires: l'éditeur, un collègue dessinateur, un comptable, la secrétaire qui dépouille le courrier, la voisine avec son chat turbulent,... Il peut aussi, en plus de ces rôles s'attribuer un des rôles principaux: client, scénariste ou meneur de jeu.

Nous avons vu que le client se situe en amont de toute la production, et qu'il est amené naturellement à avoir un suivi de la réalisation. Tenir ce rôle de client faciliterait en fait la tâche du meneur de jeu. En effet, le meneur de jeu - client peut définir un synopsis suffisamment "riche" pour pouvoir donner lieu à des pratiques intéressantes. Il peut, en fonction de l'attitude de chacun, demander une modification du produit afin de faire apparaître le bien-fondé de tel comportement ou au contraire la nécessité de changer de façon de procéder, que ce soit pour le scénariste, ou pour le dessinateur, occupés par le souci de mieux jouer. Il peut ainsi agir facilement et très naturellement sur le déroulement de la partie.

La distribution des rôles que nous préconisons, au moins dans un stade d'expérimentation, est donc la suivante:

le meneur de jeu	=	client
1 étudiant	=	scénariste
1 étudiant	=	dessinateur

Naturellement d'autres combinaisons sont possibles, c'est au meneur de jeu de proposer une distribution des rôles dans laquelle il puisse se sentir efficace.

Il est important aussi que les étudiants tiennent différents rôles, afin d'expérimenter les points de vue et les comportements spécifiques à chacun et aussi d'en trouver les points communs. Il est donc tout à fait souhaitable qu'un étudiant fasse au moins 2 parties, l'une en tant que scénariste, l'autre en tant que dessinateur, d'autant plus que dans la réalité les travaux analogues d'analyste et de programmeur sont intimement mêlés et effectués le plus souvent par une seule et même personne qui se situe tantôt comme analyste, tantôt comme programmeur.

Choisir le lieu du jeu et la stratégie pédagogique applicable:

Suggérer de faire participer l'enseignant au jeu de rôles peut induire d'utiliser ce jeu en séance de **TD** ou **TP**. Là encore, il faudra vérifier à l'usage que l'utilisation du jeu dans ce contexte réel ne perturbe pas le fonctionnement du jeu.

Cette réserve étant faite, nous pouvons mettre en place la situation suivante. L'enseignant présente le jeu de rôles aux étudiants qui se répartissent en binômes où l'un prend le rôle de scénariste, l'autre celui de dessinateur.

L'enseignant-client, grâce au réseau qui le relie à chacun de ces groupes, peut mener à bien les différentes parties qui s'engagent. Notons qu'il peut donner aux différents groupes soit le même synopsis - et il serait alors intéressant de montrer ensuite à ces groupes la palette des résultats BD obtenus (de même que différentes équipes de programmation, face à un même problème, développent des solutions différentes) - soit des synopsis différents.

L'enseignant peut choisir à l'avance de mettre les étudiants devant un certain nombre de situations, identiques sur le principe pour tous les groupes; mais les réponses de chacun des groupes leur seront personnelles, les situations posant plus ou moins problème selon leur façon de faire... Cette technique a l'avantage de faire vivre à tous un même type d'expériences, de les faire réfléchir sur un même type de problèmes, et permet à l'enseignant d'observer leurs réactions et d'exploiter ensuite cela en cours.

L'enseignant connaissant les tendances, qualités ou défauts (devant un problème informatique) de chacun peut aussi choisir de personnaliser les situations qu'il propose. Il pourra ensuite, en situation de cours s'appuyer sur les expériences vécues par tel élève pour mieux lui montrer la façon dont celui-ci se comporte et en quoi cela pose problème au cours de la réalisation d'un projet.

Le jeu est aussi accessible en **libre service** dans le laboratoire. Celui qui a envie de jouer peut venir au laboratoire et y trouver (peut-être) des partenaires; ou bien fixer rendez-vous aux partenaires qu'il a choisis. Le courrier des lecteurs peut d'ailleurs servir à passer ce genre de petite annonce ! Nous pensons qu'une fréquentation importante du laboratoire facilitera les rencontres...

Si le meneur de jeu a une grande responsabilité dans la partie, les joueurs ont la responsabilité d'accepter ou de refuser tel ou tel meneur de jeu. On retrouve là

l'une des principales options pédagogiques mises en œuvre dans le laboratoire: offrons des outils supplémentaires aux étudiants et faisons leur confiance. Pour revenir au jeu de rôles, rappelons-nous que l'intérêt de la partie est intimement lié à l'efficacité du meneur de jeu, et que les joueurs auront donc intérêt (ludiquement parlant aussi) à trouver un meneur de jeu compétent. "Ne laissez pas un mauvais meneur de jeu vous dégouter définitivement. Essayez-en d'autres !" [Paunin 86].

Une partie peut se jouer en temps réel. Remarquons qu'une partie a un caractère séquentiel. Le client a un travail important à réaliser en début de partie, ses interventions sont ensuite plus ponctuelles (dialogues avec le scénariste, modifications du synopsis,...). De même, le scénariste a un travail essentiellement central dans la partie, alors que le dessinateur a un travail plutôt terminal. Chacun cependant doit rester accessible aux autres, doit pouvoir les consulter, doit pouvoir effectuer des modifications... Alors comment envisager ces temps disons "moins remplis" ?

Une première attitude peut constituer pour l'étudiant, à mettre à profit ce temps disponible pour explorer le domaine de la bande dessinée à travers le jeu: lire les règles d'or de la bande dessinée, regarder les différents exemples de synopsis, scénario et page finale BD fournis, explorer la banque d'images... Ceci peut ensuite faire gagner un certain temps au joueur ou lui éviter certaines erreurs.

Une autre façon d'envisager les choses est de mener deux parties croisées. Par exemple, le scénariste de la partie A est en même temps dessinateur de la partie B, tandis que le dessinateur de la partie A est scénariste de la partie B, le meneur de jeu - client pouvant être le même pour les deux parties. Ceci présente l'avantage de faire tenir par les étudiants les deux rôles de scénariste et dessinateur (analyste et programmeur) dans deux parties (projets) qui se déroulent en parallèle.

Un joueur peut aussi interrompre son activité dans le jeu, tout en restant dans le laboratoire et en explorant d'autres salles, d'autres logiciels. Il continue alors d'être accessible aux autres par le biais de l'accessoire Talkie Mac (Analyses Conseils Informations) grâce auquel ils peuvent communiquer informellement. Le joueur peut alors décider de revenir dans la partie, de consulter les documents qu'un autre lui envoie, de faire une modification dans ses spécifications...

L'étudiant peut aussi vouloir **jouer tout seul**, et créer la bande dessinée du début à la fin. Pourquoi pas ? Il semble que cette utilisation fasse perdre au jeu une partie de ses atouts: une spécification est toujours comprise par celui qui la crée, les défauts ne lui apparaissent pas, car il lit entre les lignes... les problèmes de communication disparaissent.

En revanche la structure du jeu met tout de même l'accent sur la question: quel travail relève de quelle personne ? En effet les rôles, et les documents intermédiaires ne peuvent pas être éliminés. Le joueur solitaire sera d'abord obligé de prendre la casquette du client pour créer personnages et synopsis (par l'application "client"); Ensuite il prendra le rôle du scénariste pour créer découpage et scénario (par l'application "scénariste"). Enfin, en tant que dessinateur il va dessiner la BD. Si le joueur-dessinateur par exemple, n'est pas obligé de prendre le téléphone pour poser une question au scénariste (lui-même en l'occurrence), il reste obligé de reprendre la casquette scénariste (application "scénariste") s'il veut par exemple que le découpage soit modifié avant de revenir poursuivre son travail de dessinateur (application dessinateur). Cette approche peut être intéressante aussi. Elle a également l'avantage de permettre aux étudiants de se familiariser en toute quiétude (sans la pression des autres) aux outils et techniques proposées.

Changer la mission confiée aux joueurs

Nous avons jusqu'à présent envisagé que le but proposé aux joueurs soit de créer une page de bande dessinée. C'est selon cette optique que le jeu a été conçu.

Cependant, dans le souci d'aller plus vite, on peut aussi proposer aux joueurs de **compléter** une page de bande dessinée, dont le synopsis, le découpage et le scénario ainsi que les vignettes, partiellement réalisés, seraient fournis. Ceci présente l'avantage de mettre les étudiants en situation de poursuivre un projet commencé par d'autres, avec les difficultés et les enseignements que cela comporte.

Une autre utilisation est de proposer aux joueurs de **modifier** une page dont l'ensemble des éléments serait fourni. Ceci met bien sûr les étudiants en situation de maintenance et/ou d'extension d'un produit réalisé par d'autres, ou par eux-mêmes précédemment d'ailleurs.

IV.6.3. Meneurs de jeu et enseignants: pour un transfert vers la programmation

Il est bien évident qu'il est nécessaire d'assurer le transfert de l'activité de création de BD vers l'activité de programmation. Il s'agit bien d'apprendre par analogie. Comme toute analogie, l'analogie considérée ne rend pas compte de toute l'activité de programmation; elle a ses limites. Si nous pouvons dire qu'il est possible de "paramétrer" une image par ses dimensions, nous n'avons pas d'analogie à proposer en ce qui concerne par exemple le passage d'une variable en paramètre par valeur ou par adresse. Peut-être en découvrirons-nous un jour, mais nous ne voulons en aucun cas en exhiber une à tout prix, au risque qu'elle soit trop artificielle. Encore une fois, le jeu et l'analogie sur laquelle il repose ne sont pas

censés enseigner toute la programmation. C'est un moyen de plus, avec un objectif délimité, proposé pour répondre à un problème repéré dans la formation classique que reçoivent les étudiants.

Nous avons vu que cette analogie permet à l'étudiant débutant d'obtenir dès le début du cours une expérience sur laquelle peut s'appuyer l'enseignant. Pour assurer ce transfert, l'analogie peut être mise en évidence à la fois par le meneur de jeu "du domaine BD vers la programmation", et par l'enseignant de la programmation "vers la BD" ou "depuis la BD". Le meneur de jeu peut intervenir dans ce sens pendant la partie. L'enseignant, lui, peut se servir de l'analogie pour présenter une méthode, pour faire comprendre à l'étudiant qui travaille en quoi son comportement n'est pas satisfaisant...

IV.7. Prolongements

Nous pensons que des expérimentations nous permettront d'améliorer la qualité du jeu de rôles, la manière de l'utiliser, et donc son efficacité. Au delà des frontières qu'il s'est fixées, ce jeu a été pour nous générateur d'un certain nombre d'idées, qu'il reste à explorer mais que nous souhaitons livrer ici à la réflexion du lecteur.

"Toi, Moi et Lui" face à d'autres formes de programmation:

Il serait intéressant de considérer le jeu de rôles par rapport aux autres types de programmation. Il est particulièrement tentant de faire un rapprochement avec la programmation orientée objet: spécifier des objets-acteurs, leur comportement...

"Toi, Moi et Lui" dans d'autres disciplines:

Les enseignants se plaignent souvent du fait que les élèves n'ont pas de méthode de travail. Ils devraient l'avoir appris avant, par imprégnation sans doute puisqu'aucune matière ne se donne vraiment les moyens de leur faire apprendre ces méthodes... comme le fait remarquer C. Pair, représentant le Ministre de l'Education Nationale au Colloque francophone sur la Didactique de l'informatique [Didactique 88]. Or il se trouve que ce problème de méthodes se pose de façon cruciale pour l'informatique et la programmation. Cet apprentissage de méthodes peut donc être un apport de l'enseignement de l'informatique.

"Toi, Moi et Lui" vise un apprentissage, une appropriation de méthodes dans le domaine de la programmation. Ce jeu apporte, à ce titre, des éléments de réflexion sur l'apprentissage de méthodes en général. De plus, il serait intéressant de savoir si le jeu tel quel pourrait être utile à l'enseignement de méthodes dans d'autres disciplines. En effet, l'analogie trouvée avec le domaine de la BD tend à montrer que de mêmes éléments de méthodes servent dans plusieurs disciplines... Il faudrait donc réfléchir aux méthodes utiles dans diverses disciplines, les comparer, voir les éléments communs...

"Toi, Moi et Lui" avec d'autres domaines d'analogie:

D'autres domaines ont attiré notre attention lors de la conception de "Toi, Moi et Lui", en particulier celui de la musique. Il serait intéressant de voir si nous pouvons effectivement déboucher dans cette voie. D'autres domaines peuvent-ils se prêter à l'analogie avec la programmation ? Si c'est le cas, le premier intérêt repose sur le fait qu'il serait alors possible pour l'étudiant de choisir le domaine, c'est-à-dire la fantaisie dans laquelle il veut jouer. Le second intérêt serait de

montrer qu'on peut généraliser les méthodes utilisées à divers domaines; ce qui viendrait renforcer la généralisation évoquée dans le paragraphe précédent, à diverses disciplines enseignées.

La bande dessinée vole de ses propres ailes dans le laboratoire Arcade:

Le logiciel de jeu a permis de faire circuler des bandes dessinées dans l'équipe. Par le biais de tests, d'expériences, des BD ont été créées sur des thèmes quelconques ou sur des thèmes informatiques qui nous sont chers. La bande dessinée n'était plus réservée aux seuls loisirs. Elle nous est tout d'abord réapparue comme moyen d'expression.

L'enseignant peut par la bande dessinée exprimer une idée, un algorithme, une approche. Il propose alors aux étudiants la **lecture d'une bande dessinée**. Un essai a été tenté sur le thème des Spécifications, thème qui nous concerne particulièrement dans le contexte du jeu "Toi, Moi et Lui". Il s'agit d'une BD en 2 planches au cours de laquelle un professeur amène progressivement ses étudiants à la découverte d'algorithmes de mieux en mieux adaptés à la résolution du problème posé: déterminer si 2 mots sont des anagrammes.

Anagrammes

Je veux un algorithme qui détermine si deux mots donnés sont des anagrammes.

Résultat de l'algorithme:

SAIN	ANIS	oui
CAFE	LAIT	non

C'est facile. Il suffit de regarder si chaque lettre d'un mot appartient à l'autre mot.

Ainsi, LE et ELLE sont des anagrammes.

?

?!

Monsieur, vous trichez! Dans vos exemples, les mots avaient le même nombre de lettres. Je le supposais aussi.

Ainsi LASSE et SALLE sont des anagrammes.

?!

Ffffff !

**JE VEUX
UNE
VRAIE
DEFINITION.**

Un temps après...

M1 est un anagramme de M2 si M1 est une permutation de M2.

Déduis-en un algorithme!

C'est facile; on regarde si un mot appartient aux permutations de l'autre.

On compare LAIT et

CAFE	CAEF	CAFE	CFAE	CEAF
CE	LAIT	et	CAFE	ne sont pas des anagrammes.
AE				AFEC
				FACE
				FAEC
				FECA
				EAFC
				EACF
				EFCF
				EFAC

Et si on veut savoir si ANTICONSTITUTIONNELLEMENT est un anagramme d'un autre mot de 25 lettres, il suffit de le comparer aux 25! permutations possibles; ce qu'un ordinateur fera en plusieurs milliards d'années.

La solution est donc parfaite théoriquement mais inacceptable pratiquement

Il faut trouver un algorithme basé sur le même principe, mais tenant compte du temps d'exécution et des contraintes de l'ordinateur.

Et si on représentait tous les éléments de l'ensemble par ordre alphabétique ?

On transforme...

SAIN en AINS	LASSE en AELSS
ANIS en AINS	SALLE en AELLS
CAFE en ACEF	
LAIT en AILT	

En une seule comparaison, on a la réponse!

Oui, mais il y a très peu de chances que 2 mots soient des anagrammes. Et là, il faut ordonner les lettres de chaque mot avant de les comparer.

Soit de l'ordre de n^2 comparaisons...

Regardez. Voici une autre méthode.

SAIN	ANIS
AIN	ANI
IN	NI
N	N

J'élimine S
J'élimine A
J'élimine I
J'élimine N
OK.

SALLE	LASSE
ALLE	LAS E
LLE	L S E
LE	S E

J'élimine S
J'élimine A
J'élimine L
J'élimine L
!!!

Ça fait un peu cuisine...

Oui, mais c'est rigoureux et efficace.

Et voici encore une autre méthode.

On associe à un mot les quantités de chacune de ses lettres.

	n _A = 1
SALLE	n _E = 1
	n _L = 2
	n _S = 1

Les compteurs des autres lettres sont à zéro.

Puis on passe en revue les lettres de l'autre mot, en soustrayant 1 aux compteurs du premier mot.

LASSE	L	n _L = 1
	A	n _A = 0
	S	n _S = 0
	S	n _S = -1

On s'arrête dès qu'un compteur devient négatif.

C'est efficace. Mais c'est très artificiel à côté de la solution mathématique de l'ensemble des permutations.

C'est tout le problème de la programmation. Trouver les compromis pour concilier les bonnes spécifications et les contraintes de l'ordinateur.

J.-P. F.

Le dialogue mis en scène permet de rendre vivante et plaisante (suspens... Quel sera l'algorithme retenu ?) une démarche rigoureuse de recherche d'algorithme présentée par ailleurs dans [Peyrin 88].

Une telle bande dessinée peut être faite sur ordinateur, avec éventuellement le logiciel "dessinateur", en utilisant la banque de dessins proposée dans le jeu (qui libère du je sais pas dessiner), en utilisant éventuellement des copies d'écran issues d'autres logiciels du laboratoire Arcade, et ... les méthodes évoquées auparavant !

La lecture peut se faire de façon traditionnelle, c'est-à-dire sur papier de format A4, mais l'ordinateur peut également servir de support (c'est le cas pour "Anagramme"). Dans ce cas l'étudiant ne voit que quelques vignettes à la fois en raison de la taille de l'écran du Macintosh. La focalisation sur une vignette est loin d'être un inconvénient, étant donné l'effort de réflexion demandé à l'étudiant (ce qui est rarement le cas dans les BD de loisirs !). Simplement, l'auteur de la BD fera un découpage différent selon le format de présentation.

Et la BD peut même être sonorisée...

ou articulée avec les séquences d'animation créées par ailleurs dans le laboratoire. Cette idée est née de la constatation qu'il est difficile de créer un scénario qui soit vivant, accrocheur, aisé à suivre et qui, dans le même temps, descende à un niveau de détail approfondi. D'où l'idée: on clique sur une vignette et une animation apparaît, approfondissant le problème. Puis la lecture reprend à la vignette suivante.

La bande dessinée sur un sujet informatique peut aussi servir de **support à un exercice**, sous la forme de vignettes à compléter, à créer ou à ordonner... "Courage, trions !" décrit l'exécution d'un algorithme de tri.

Courage, trions !

Une gare paisible dans le sud-est de la France. Deux cheminots s'apprêtent à trier les wagons d'un train, par ordre croissant.

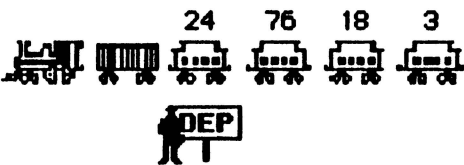


Bon. Je m'occupe de déplacer les wagons.

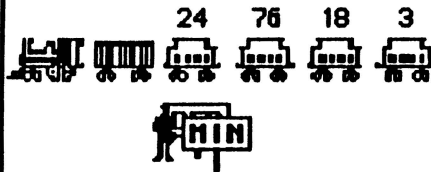
Et moi, je t'indique au fur et à mesure les wagons à déplacer.



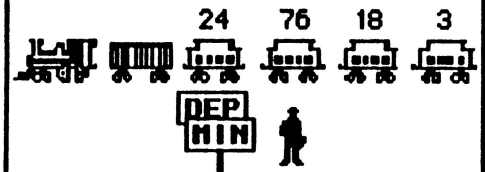
Regardez. Le chef commence le tri. Il plante le piquet DEP.



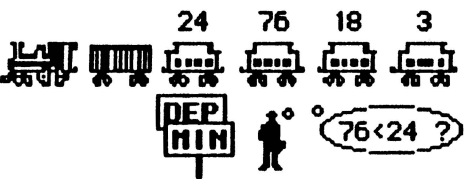
Il plante le piquet MIN près de DEP.



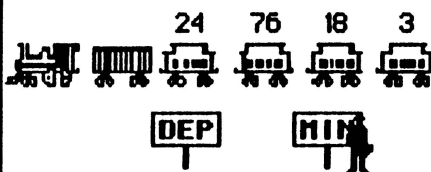
Il s'arrête devant le wagon suivant.



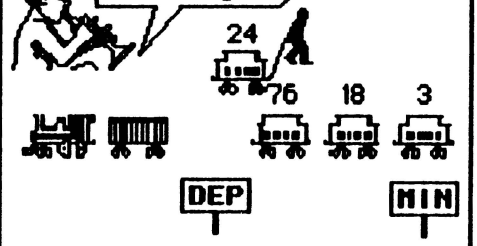
Il compare les numéros des wagons.



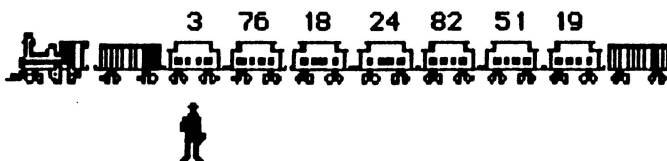
Devant le wagon 18 ($18 < 24$), il plante le piquet MIN.



Arrivé à la queue du train, il a appelé Henri pour l'échange.

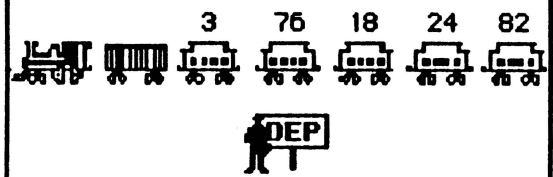


Et le chef retourne à la tête du train pour continuer le tri. Il va refaire les 6 actions qu'il vient d'effectuer devant nous. Suivons-le.



Il plante le piquet DEP.

A vous... Continuez l'histoire !



Nous proposons alors à l'étudiant une version incomplète de cette bande dessinée: "Tri en BD". Nous lui demandons de compléter les vignettes en respectant l'exécution de l'algorithme grâce au "prêt à coller" fourni (outils de dessin et mini-banque d'images). Il dispose aussi d'un corrigé, vignette par vignette ou global, il peut revenir à la vignette originale... L'étudiant peut ainsi tester sa compréhension de l'algorithme

Il est pertinent ici d'utiliser l'ordinateur, en proposant les outils pour compléter ou modifier la BD et pour vérifier le bien-fondé d'un choix.

La bande dessinée "Courage, trions !" est un exemple de BD à cheval sur ces 2 utilisations. Elle entre à la fois dans le cadre de la lecture d'une bande dessinée (l'étudiant assiste à l'exécution d'un algorithme) et dans celui de support à un exercice (l'étudiant lit le corrigé de sa propre création). De plus, elle a été réalisée sur ordinateur à partir des images du film La gare de triage et à l'aide de l'application Dessinateur. C'est la preuve que, dans l'équipe, le principe de "réutilisation" est mis en pratique! En effet, il serait paradoxal de constater qu'il en fût autrement...





Conclusion

Ce travail a permis de caractériser l'activité de jeu à proposer dans le cadre d'un enseignement de la programmation. Concrètement il a débouché sur la conception et la réalisation d'un jeu de rôles sur réseau d'ordinateur. Ce jeu porte sur la notion de méthodes mises en œuvre lors de l'activité de programmation.

Des logiciels de jeux portant sur d'autres thèmes de la programmation sont actuellement à l'étude. L'un reprend l'idée développée dans [Guéraud 85], concernant la découverte de l'algorithme répondant au jeu des tours de Hanoï. Un autre jeu repose sur le parcours de graphes.

D'autres logiciels du laboratoire présentent des aspects ludiques. Il est à noter par exemple que les concepts de jeu et de manipulation sont proches: manipuler et jouer évoquent une certaine liberté d'action, le besoin de concentration, une démarche de réflexion et de curiosité. De façon semblable, la frontière entre une activité d'exercice et une activité de jeu peut être floue.

Mais n'y a-t-il pas toujours une façon ludique d'aborder les choses ? Et notre laboratoire, au delà du sérieux profond de son contenu et de ses objectifs, n'induit-il pas, par la facilité d'accès offerte, par la fantaisie utilisée dans sa représentation, et par divers clins d'œil, une approche ludique ? Alors, voulez-vous jouer avec nous ?

Le laboratoire Arcade a été entièrement développé sur du matériel Macintosh (Apple). Ce choix de matériel a été réalisé en fonction du type d'interface offert par ce matériel (et nécessaire à la mise en œuvre du laboratoire), de son coût, et des logiciels disponibles.

Nous disposons maintenant d'une version opérationnelle du laboratoire, que nous souhaitons voir utilisée par les étudiants. Pour cela, il faut que les étudiants disposent de matériel Macintosh en nombre suffisant. Ceci n'est pas encore le cas dans notre environnement. Une salle devait être équipée de Macintosh à la rentrée 88, elle ne le sera que dans le courant de l'année scolaire.

Le laboratoire n'est donc pas actuellement disponible en accès libre service pour les étudiants. Seules des expérimentations partielles ont pu être faites, sous diverses formes. Nous avons pu donner accès au laboratoire:

- à quelques étudiants de diverses classes (Maîtrise d'Informatique Appliquée à la Gestion, MST Expert en Systèmes Informatiques) pendant des heures fixes de permanence, quelques semaines durant.
- à l'ensemble des étudiants d'une classe (DESS Informatique Double Compétence - Université des sciences sociales) pendant une demi-journée sur un nombre suffisant d'ordinateurs.
- à nos collègues informaticiens.

D'autre part certains logiciels du laboratoire ont été expérimentés par un enseignant auprès d'un public de professeurs du secondaire (Centre Informatique et Applications Pédagogiques) et d'un public d'élèves ingénieurs (Institut Technologique de Bandung, Indonésie).

Les réactions des étudiants, comme des collègues sont très encourageantes. Les étudiants sont en général très séduits par l'aspect du laboratoire et des activités; ils voient bien l'aide que cela peut représenter pour leur apprentissage de la programmation. Ils se sentent en général un peu frustrés de ne pas avoir accès au laboratoire de façon régulière tout au long de l'année. Les enseignants imaginent bien l'emploi qui peut être fait du laboratoire pour l'enseignement. Eux-même trouvent un intérêt dans la pratique des activités qui souvent leur permettent une certaine "redécouverte" des choses.

Nous avons aussi eu l'occasion de faire de nombreuses démonstrations du laboratoire à des publics ou à des personnalités d'origine diverses. Toutes les réactions ont été positives, voire totalement enthousiastes.

L'arrivée du matériel nécessaire va maintenant nous permettre une utilisation en vraie grandeur. Elle nous permettra de faire évoluer le laboratoire en fonction des réactions des étudiants et des enseignants. Elle nous permettra aussi de proposer aux étudiants des travaux pratiques ou des projets qui contribueront à l'agrandissement du laboratoire.



Références bibliographiques

[Aho 83] Aho A., Hopcroft J., Ullman J. "Data Structures and algorithms" Addison-Wesley (1983). Traduction française: "Structures de données et algorithmes" InterEditions (1987)

[Anderson 85] Anderson J.R., Reiser B.J. "The LISP Tutor" Byte Vol 10, n°4, (Avril 1985)

[Arsac 80] Arsac J. "Premières leçons de programmation" Cedic/Nathan (1980)

[Arsac 83] Arsac J. "Les bases de la programmation" Dunod (1983)

[Arsac 85] Arsac J. "Jeux et casse-tête à programmer" Dunod (1985)

[Aveline 61] Aveline C. "Le code des jeux" Hachette (1961)

[Avram 84] Avram D., Weidenfeld M. "Initiation à Logo" Cedic/Nathan (1984)

[Avram 84] Avram D., Savatier T., Weidenfeld M. "Logo, Manuel de référence" Cedic/Nathan (1984)

[Beaumont 84] Beaumont C. "Pour faire de la B.D." Editions Retz (1984)

[Berdonneau 84] Berdonneau C. "Recueil des pratiques pédagogiques autour de LOGO" I.N.R.P. (1984)

[Béville 86] Béville G. "Jeux de formation" Editions d'organisation (1986)

[Biondi 84] Biondi J., Clavel G. "Introduction à la programmation" Masson (1984)
(3ème édition révisée et commentée en 1987)

[Boehm 82] Boehm B.W. "Les facteurs de coût du logiciel" T.S.I. Vol 1, n°1, pp
5-24 (1982)

[Boussard 83] Boussard J.C., Mahl R. "Programmation avancée" Eyrolles (1983)

[Brassinne 88] Brassine M. "Les étoiles du ludiciel" Jeux & Stratégie n°53, pp 42-51
(Octobre-Novembre 88)

[Brown 85] Brown M., Sedgewick R. "Techniques for Algorithm Animation" IEEE
Software Vol 2, n°1, pp 28-39 (Janvier 85)

[Cagnat 86a] Cagnat J.M. "Présentation du projet Arcade" LGI, Projet Arcade,
Rapport n°1 (Avril 1986)

[Cagnat 86b] Cagnat J.M., Guéraud V., Painvin S., Peyrin J.P. "Vers un laboratoire
de programmation" LGI, Projet Arcade, Rapport n°3 (Juillet 1986)

[Cagnat 88a] Cagnat J.M. "Apprentissage guidé d'un algorithme: le tri par
minimums" LGI, Projet Arcade, Rapport n°10 (Août 1988)

[Cagnat 88b] Cagnat J.M. "Meccano de tri: une boîte d'outils pour trier des cubes"
LGI, Projet Arcade, Rapport n°11 (Août 1988)

[Cagnat 88c] Cagnat J.M., Guéraud V., Liem I., Painvin S., Peyrin J.P. "Un
laboratoire pour l'enseignement de la programmation" Colloque sur l'évolution de
l'outil informatique à l'université, Poitiers (Septembre 1988)

- [Caillois 67a] Caillois R. "Les jeux et les hommes" Gallimard (1967)
- [Caillois 67b] Caillois R. (sous la direction de) "Jeux et sports" Encyclopédie de la Pléiade, Gallimard (1967)
- [Caire-Jabinet 81] Caire-Jabinet M.P. "Jeux et jouets" Les Dossiers de l'Histoire n°34, pp.26-53 (Nov-Dec. 1981)
- [Chateau 54] Chateau J. "L'enfant et le jeu" Editions du Scarabée (1954)
- [Chazoule 84] Chazoule O. "Jouer avec son micro-ordinateur" Marabout (1984)
- [Choppy 85] Choppy C. "Techniques et aspects du prototypage" Actes des journées "Prototypage, maquettage et génie logiciel", Lyon, Janvier 1985 dans Bigre n°43-44 (Juillet 85)
- [Comar 87] Comar C. "Un environnement pour enseigner la programmation" Thèse de Doctorat de l'Université Paris 6 (Décembre 1987)
- [Clark 82] Clark P. "The Play's the Thing" Byte Vol 7, n°12, pp 6-10 (Décembre 1982)
- [Cloutier 88] Cloutier J.F. "Apports de différents paradigmes de programmation, comme autant d'outils de pensée" Colloque sur la Didactique de l'Informatique, Paris (Septembre 1988)
- [Coulette 87] Coulette B. "Iphigénie: un didacticiel expert en méthodologie de développement de projets logiciels" Congrès francophone sur l'EAO, EAO 87, Cap d'Agde (mars 1987)

[Crampes 87] Crampes M., Quemeneur J.Y. "Le transfert de l'expérience en conduite de projet informatique: jeu d'aventure et intelligence artificielle" Congrès francophone sur l'EAO, EAO 87, Cap d'Agde (mars 1987)

[Crawford 82] Crawford C. "Design Techniques and Ideals for Computer Games" Byte Vol 7, n°12, pp 96-108 (Décembre 1982)

[Dami 75] Dami C. "Stratégies cognitives dans des jeux compétitifs à deux" Thèse de psychologie, Université de Genève (1975)

[Didactique 88] Actes du Colloque francophone sur la "Didactique de l'Informatique", Université René Descartes, Paris (1-3 Septembre 1988)

[Dijkstra 71] Dijkstra E.W. "A short introduction to the art of programming" EWD 316, Department of Mathematics, Eindhoven, Pays-Bas (1971)

[Du Boulay 88] Du Boulay B. "Intelligent Systems for teaching programming" Artificial Intelligence Tolls in Education, IFIP (1988)

[Duc 82] Duc B. "L'art de la B.D." Tomes 1 et 2, Editions Glénat (1982)

[Ducrin 84] Ducrin A. "Programmation" Dunod (1984)

[Faraut 88] Faraut M.: Exposition centrée sur la bande dessinée "Les patriotes" (Giroud - Lacaf, Ed. Glénat). Cette exposition relie Grenoble d'aujourd'hui à Grenoble de 1788, et dévoile les coulisses de la conception de la BD, Glénat (1988)

[Freud 68] Freud S. "Psychopathologie de la vie quotidienne" Payot (1968)

[Gers 84] Gers J.N. "De la calculette au micro-ordinateur, un enseignement actif des mathématiques" Premier Colloque scientifique francophone sur l'EAO, EAO 84, Lyon (Septembre 1984)

[Gram 86] Gram A. "Raisonnement pour programmer" Dunod (1986)

[Grandbastien 88] Grandbastien M. "Une approche à bases de connaissances pour l'enseignement de la programmation" Thèse d'état, Nancy 1 (Juillet 88)

[Gros 83] Gros C., Permelet R. "Jeux vidéo" M.A. Editions (1983)

[Guéraud 85] Guéraud V. "Etude des fonctions EAO nécessaires pour un didacticiel de soutien à l'enseignement de la récursivité; réponses du système Macintosh à des besoins EAO" Rapport de DEA, INPG (Juin 1985)

[Guéraud 86a] Guéraud V., Painvin S. "La Cité des Sciences et de l'Industrie" LGI, Projet Arcade, Rapport n°2 (Juin 1986)

[Guéraud 86b] Guéraud V., Painvin S. "Voulez-vous jouer avec moi ?" LGI, Projet Arcade, Rapport n°5 (Octobre 1986)

[Guéraud 87] Guéraud V., Liem I. "Les salons EDUCATEC ET FORMATION 86, Le 6ème congrès de l'EAO" LGI, Projet Arcade, Rapport n°6 (Février 1987)

[Guéraud 88] Guéraud V., Peyrin J.P. "Un jeu de rôles pour l'enseignement de la programmation" Colloque sur la Didactique de l'Informatique, Paris (Septembre 1988)

[Henriot 76] Henriot J. "Le jeu" P.U.F. (1976)

[Jeanne 88] Jeanne F. "SYSECA: Apprendre la rigueur par la simulation" Dossier Les Hommes et l'Informatique, Le Monde Informatique N°347 (5 Décembre 1988)

[Johnson 85] Johnson W.L., Soloway E. "PROUST" Byte, Vol 10, n°4 (Avril 1985)

[Laurent 85] Laurent J.P. "Initialisation à l'analyse et à la programmation" Dunod (1985)

[Ledgard 75] Ledgard H.F. "Programming proverbs" Hayden Book Company (1975) traduit en français et annoté par J. Arzac, Dunod (1978)

[Leif 76] Leif J., Brunelle L. "Le jeu pour le jeu" Colin (1976)

[Liem 88a] Liem I., Peyrin J.P. "Le logiciel Tris Internes" LGI, Projet Arcade, Rapport n°7 (Février 1988)

[Liem 88b] Liem I., Peyrin J.P. "Le logiciel Dessins Récursifs" LGI, Projet Arcade, Rapport n°8 (Février 1988)

[Liem 89] Liem I. Thèse à paraître, Université J. Fourier, Grenoble I (1989)

[Longevialle 85] Longevialle F. "Pascal sur Macintosh" Edimicro (1985)

[Lucas 83] Lucas M., Peyrin J.P., Scholl P.C. "Algorithmique et représentation des données - Tome 1: "Files et automates d'états finis" Masson (1983)

[Lucci 89] Lucci A. "Production de logiciels d'enseignement: une expérience d'architecture de système construit sur un environnement Prolog" Thèse à paraître, Université J. Fourier, Grenoble I (1989)

[Malone 80] Malone T.W "What makes things fun to learn ? A study of intrinsically motivating computer games" XEROX Palo Alto Research Center (1980)

[Medici 69] Medici A. "L'éducation nouvelle" P.U.F. (1969)

[Meyer 78] Meyer B., Baudoin C. "Méthodes de programmation" Eyrolles (1978)

[Mucchielli 83] Mucchielli A. "Les jeux de rôles" P.U.F. (1983)

[Painvin 86] Painvin S. "Voulez-vous apprendre avec moi ?" LGI, Projet Arcade, Rapport n°4 (Octobre 1986)

[Painvin 88a] Painvin S. "La Gare de Triage: un dessin animé pour illustrer un algorithme" LGI, Projet Arcade, Rapport n°9 (Mars 1988)

[Painvin 88b] Painvin S, Cagnat J.M., Peyrin J.P., Liem I. "Pour quelques approches de plus" Bulletin de l'E.P.I. (Enseignement Public et Informatique) n°52 (Novembre 1988)

[Pair 88a] Pair C., Mohr R., Schott R. "Construire les algorithmes" Dunod (1988)

[Pair 88b] Pair C. "Je ne sais (toujours) pas enseigner la programmation" Colloque sur la Didactique de l'Informatique, Paris (Septembre 1988)

[Paunin 86] Paunin T., Rosenthal P., Sagot G. "N'ayez plus peur des jeux de rôle" Jeux et Stratégie n°38 (Avril-Mai 86)

[Peyrin 85] Peyrin J.P., Guéraud V. "Propositions pour la spécification d'un didacticiel de soutien à l'enseignement de la programmation" IMAG RR n° 586 (1985)

[Peyrin 88] Peyrin J.P. "Programmation, réalités et promesses" Premier Colloque sur l'Informatique, Bamako, Mali (Février 1988)

[Reboul 80] Reboul O. "Qu'est-ce qu'apprendre ?" P.U.F. (1980)

[Reinhorn 87] Reinhorn P. "Une méthode de conception de logiciels de jeux scientifiques" Congrès francophone sur l'EAO, EAO 87, Cap d'Agde (mars 1987)

[Reuchlin 78] Reuchlin M. "Psychologie" P.U.F. (1978)

[Rogers 70] Rogers C. "Le développement de la personne" Dunod (1970)

[Rosenthal 71] Rosenthal R. "Pygmalion à l'école" Casterman (1971)

[Scholl 88] Scholl P.C., Peyrin J.P. "Schémas algorithmiques fondamentaux: séquences et itération" Masson (1988)

[Sharp 84] Sharp D., Johnston M.D. "ChipWits, Robots you teach to think for themselves" BrainPower Inc. (1984)

[Sigwart 86] Sigwart C.D., Van Meer G.L. "Software Engineering: The art of the user interview" ACM SIGCSE Bulletin Vol 18, n°1 (Février 86)

[Sommerville 88] Sommerville I. "Le génie logiciel et ses applications" InterEditions (1988)

[Stowbridge 83] Stowbridge M.D., Kugel P. "Learning to learn by learning to play" Creative Computing Vol 9, pp. 180-188 (April 1983)

[Swenson 82] Swenson R.P., Anderson C. "The role of motivation in Computer-Assisted Instruction" Creative Computing Vol.8, pp.134-139 (Oct. 1982)

[Thomas 86] Thomas P. "La journée du concepteur médiatique: course d'obstacles" Conférences 7èmes journées micro-informatiques, Grenoble (Février 1986)

[Vajda 68] Vajda S. "Théorie des jeux et programmation linéaire" Dunod (1968)

[Verhoeff 83] Verhoeff J. "Recreational Informatics, a discipline of its own" Informatics Education for all students at University level, IFIP Conference, Delf (Juin 83)

[Vitale 84] Vitale D. "Jeux de simulation" M.A. Editions (1984)

[Vivet 88] Vivet M. "Les tuteurs intelligents: réalisations et tendances de recherches" T.S.I., Vol 7, n°1, pp 21-45 (1988)

[Williams 82] Williams G. "Ricochet" Byte Vol 7, n°12, pp 142-146 (Décembre 1982)

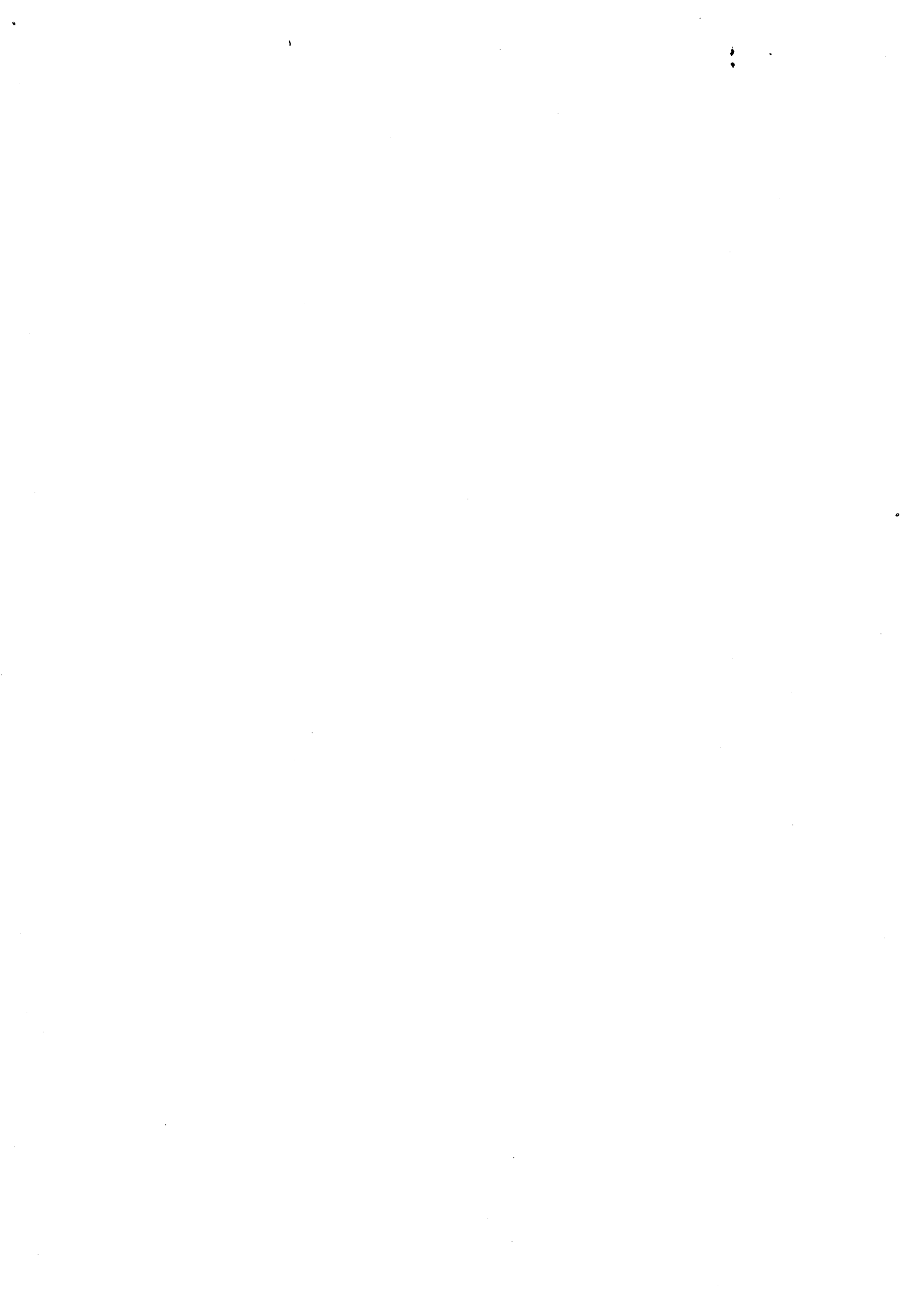
[Winnicott 71] Winnicott D.H "Jeu et réalité" Gallimard (1971)

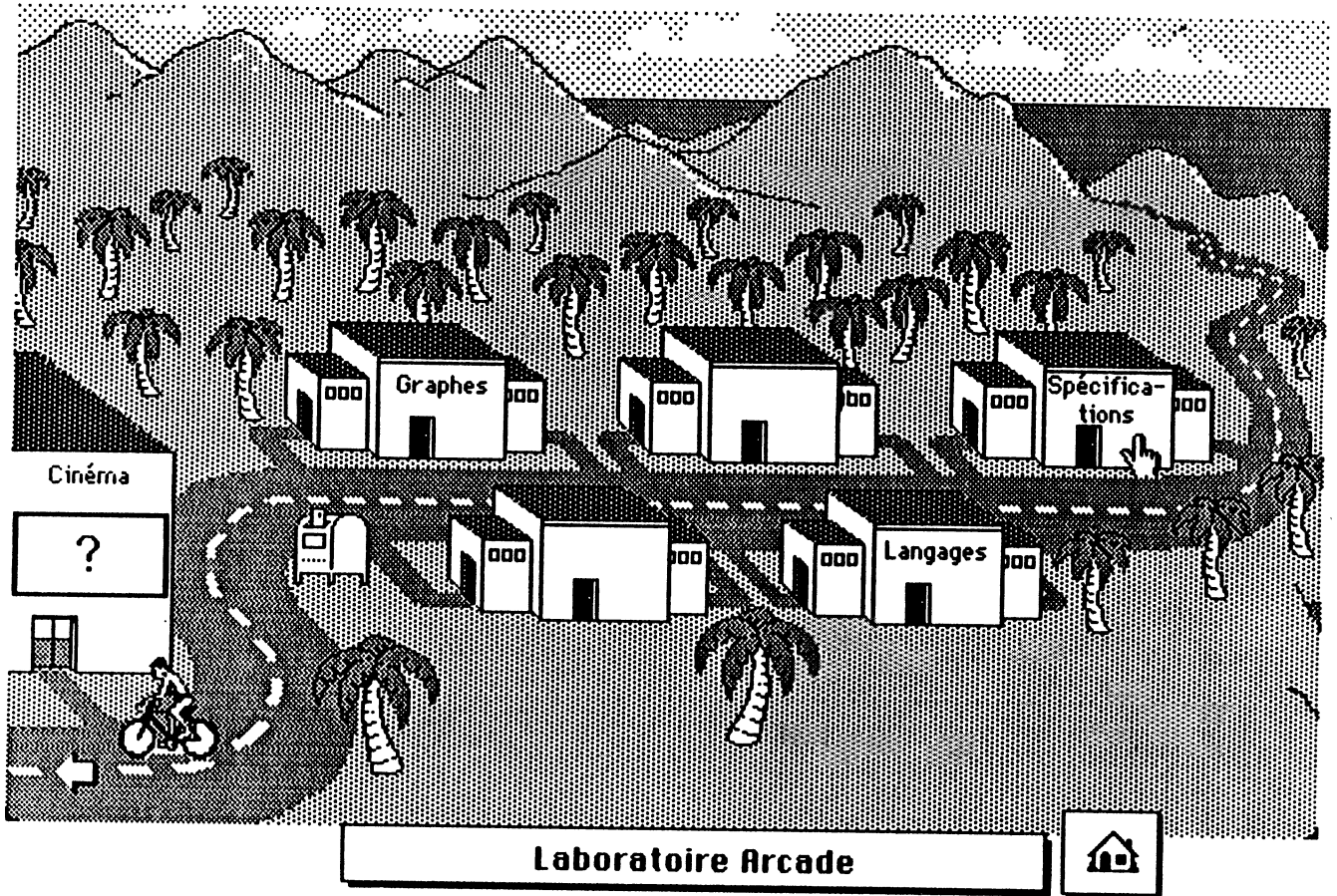
[Wirth 86] Wirth N. "Algorithms and Data Structures" Prentice-Hall (1986). Traduction française: "Algorithmes et structures de données" Eyrolles (1987)



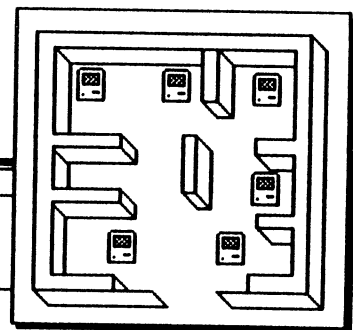
Annexe I :

**Jouons avec "Toi, Moi et Lui",
manuel d'utilisateur.**





Spécifications



Toi, moi et lui



Anagramme



Jeu de rôles à 3 joueurs, pour montrer

- la nécessité et la difficulté des spécifications
- l'intérêt du processus d'analyse descendante

Ce jeu utilise l'analogie entre les processus de création de bandes dessinées et de programmes, et entre les rôles des intervenants : client, scénariste, dessinateur = client, analyste, programmeur.

Client Scénariste Dessinateur

Le rôle du client est d'exprimer des besoins.

La réalisation sera le travail du scénariste et du dessinateur.

Le client évolue et critique le résultat et demande des modifications.

Le client a tous les droits, et surtout celui de changer d'avis.

Exemples de BD

Client

Client Notes sur l'utilisation

Le menu Création permet de créer le synopsis (les grandes lignes) de l'idée, et de définir les personnages (5 maxi).

- Dans les fenêtres Synopsis et Personnages, cliquer sur les boutons de gauche pour définir une caractéristique. Pour la supprimer, cliquer sur son nom (à droite) avant de cliquer sur Supprimer.
- Dans la fenêtre Synopsis, pour caractériser l'action d'un acteur faire un double-clic sur son nom.
- Les cases Profession, Dialogue et Autre sont extensibles. Cliquer sur le petit rectangle noir.

Le menu Consultation indique le travail fait par le scénariste et le dessinateur.

Le téléphone ou TalkieMac (dans ☎) permet de contacter les autres...

Fichier Edition Consultation Création

Jean-Paul, client

Personnages

Age: Sexe: M F

Taille: Petit Grand

Corpulence: Mince Gros

Profession:

Autre

Supprimer Nouveau

Fichier Edition Consultation Création

Jean-Paul, client

Synopsis "Le commande"

Moment 3

Lieu: Ambiance:

Autre:

Acteurs

Autre

Supprimer Nouveau

Fichier Edition Consultation Création

Jean-Paul, client

Synopsis "Le commande"

Moment 3

Acteurs

Jean-Paul

Michel

Sylvie

Annuler OK

Supprimer

Fichier Edition Consultation Création

Jean-Paul, client

Synopsis "Le commande"

Moment 3

Lieu: Ambiance:

Autre:

Acteurs: Jean-Paul
Michel
Sylvie

Supprimer Nouveau

Fichier Edition Consultation Création

Enregistrer

Imprimer

Quitter, ⌘Q

Jean-Paul, client

Synopsis "Le commande"

Moment 3

Lieu: Ambiance:

Autre:

Acteurs: Jean-Paul
Michel
Sylvie

Supprimer Nouveau

Jeu de rôles à 3 joueurs, pour montrer :

- la nécessité et la difficulté des spécifications
- l'intérêt du processus d'analyse descendante

Ce jeu utilise l'analogie entre les processus de création de bandes dessinées et de programmes, et entre les rôles des intervenants : client, scénariste, dessinateur = client, analyste, programmeur.

Client Scénariste Dessinateur

Le rôle du scénariste est de transformer les besoins (vagues) du client en des spécifications précises.

La réalisation détaillée est ensuite confiée au dessinateur.

Le scénariste évalue et critique le travail du dessinateur et demande des modifications.

Le scénariste transmet le résultat au client.

Exemples de BD


Scénariste


Scénariste *Notes sur l'utilisation*

Le menu Consultation indique le travail demandé par le client, et le travail fait par le dessinateur.

Le menu Création permet de créer le scénario (12 vignettes maxi) et sa structure visuelle (sur une page de format A4).

- Dans la fenêtre Scénario, cliquer sur les boutons de gauche pour définir une caractéristique. Pour la supprimer, cliquer sur son nom (à droite) avant de cliquer sur Supprimer.
- Pour caractériser l'action d'un acteur nommé, faire un double-clic sur son nom.
- Les cases Dialogue et Autre sont extensibles. Cliquer sur le petit rectangle noir.

Eventuellement, utiliser CheapPaint (dans ) pour faire des croquis.

Le téléphone ou TalkieMac (dans ) permet de contacter les autres...

Fichier Edition Consultation Création

Michel, scénariste

Découpage

1	2	3
4		

Supprimer

Fichier Edition Consultation Création

Michel, scénariste

Découpage

1	2	3
4	5	6
7	8	9
10	11	12

Supprimer

Fichier Edition Consultation Création

Michel, scénariste

Scénario "La commande"

Vignette 0

Lieu : Intérieur

Ambiance : serene

Autre : Bulle commentaire: VENDREDI.

Plan : rapproché

Acteurs : Sylvie

Supprimer Nouveau

Fichier Edition Consultation Création

Michel, scénariste

Scénario "La commande"

Vignette 0, Sylvie

Position : elle dessine, concentrée.

Position : assis

Vue de : profil

Imprimer Supprimer Annuler OK

Fichier Edition Consultation Création

Michel, scénariste

Scénario "La commande"

Vignette 0

Lieu : Intérieur

Ambiance : serene

Autre : Bulle commentaire: VENDREDI.

Plan : rapproché

Acteurs : Sylvie

Supprimer Nouveau

Enregistrer

Imprimer

Quitter [Command-]Q

Jeu de rôles à 3 joueurs, pour montrer :

- la nécessité et la difficulté des spécifications
- l'intérêt du processus d'analyse descendante

Ce jeu utilise l'analogie entre les processus de création de bandes dessinées et de programmes, et entre les rôles des intervenants : client, scénariste, dessinateur = client, analyste, programmeur.

Client Scénariste Dessinateur

Le rôle du dessinateur est de réaliser la BD à partir des spécifications fournies par le scénariste.

Exemples de BD


Dessinateur


Dessinateur Notes sur l'utilisation


Le menu Consultation indique le travail à faire.

Le menu Création permet de créer vignettes et dessins auxiliaires.

Utiliser les articles des menus Edition, Facilités et Mode pour créer dessins et bulles. Le menu Facilités permet aussi d'avoir une vue globale de la page BD.

Art Grabber+ (dans ) permet d'aller "chiper" des images. Ouvrir (Open) le document souhaité dans la bibliothèque d'images. Sélectionner un dessin, le prendre (Grab), et revenir le coller... dans une fenêtre Vignette ou Dessin ou dans CheapPoint.

CheapPoint (dans ) offre des fonctionnalités graphiques supplémentaires de type Macpoint (crayon, pinceau, loupe, contours, polices, motifs, miroir...)

Le téléphone ou TalkieMac (dans ) permet de contacter les autres...

Fichier Edition Facilités Consultation Création Mode

Sylvie, dessinatrice

Vignette N° 5

JEUOI

Tu as lu le scénario. Qu'en penses-tu ?

Fichier Edition Facilités Consultation Création Mode

Sylvie, dessinatrice

OK

Fichier Edition Facilités Consultation Création Mode

Enregistrer

Imprimer

Quitter *Q II

Sylvie, dessinatrice

Vignette N° 9

Voilà ce que j'ai fait.







Oui...


Le rôle du dessinateur est de réaliser la BD à partir des spécifications fournies par le scénariste

Exemples de BD

Dessinateur

Exemples disponibles

	Courage, trions !		Si Arcade...
	La commande		Le test
	Le séduction		Marco et Adrien

 Anagramme

La commande



LUNDI

Allo Michel ? Ici Jean-Paul... Il faut que tu me prépares un scénario sur l'histoire du projet Arcade.

MARDI

Arcade

MERCREDI

Il faut que je rappelle Jean-Paul pour savoir le nom des logiciels et discuter de...

Nous sommes d'accord sur le scénario et sur les modalités de travail. Je contacte Sylvie.

OK. Mais respectez bien les délais...

JEUDI

Tu as lu le scénario. Qu'en penses-tu ?

Je vais avoir du mal à dessiner l'arcade de jeux. Ma banque d'images n'est pas assez fournie.

Il faudrait aussi que je rencontre les membres du groupe Arcade pour faire leur portrait. Ce serait mieux que sur photo...

VENDREDI

SAMEDI

Voilà ce que j'ai fait.

Oui...

Mais pour terminer, il faut encore convaincre Eric de se laisser "croquer".

J'essaie de régler ce problème.

DIMANCHE

Eric est OK. Jean-Paul lui a offert un Mac3!

Evidemment, moi je ne lui avais proposé qu'une boîte de douze crayons couleur...

LUNDI

ça marche !

Si Arcade m'était costé...



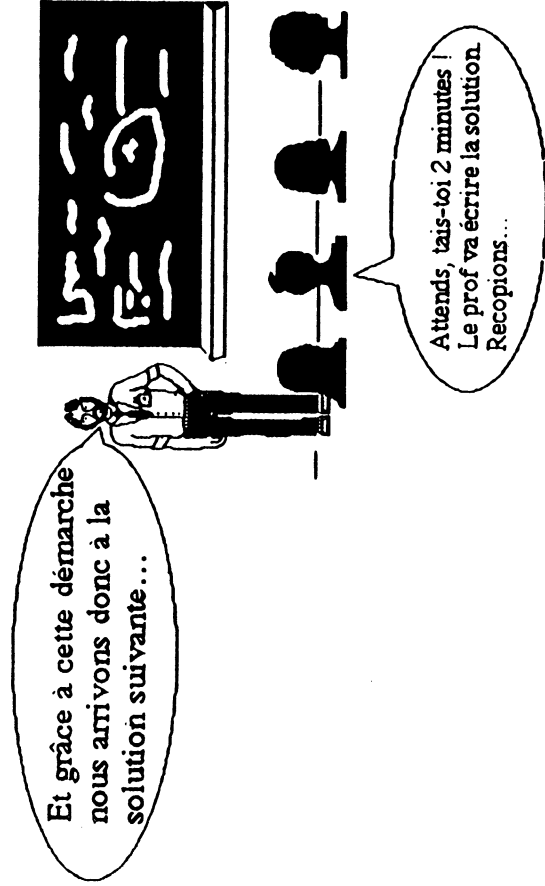
Annexe III :

**Jouons avec "Toi, Moi et Lui",
livret du meneur de jeu.**

Difficulté de mettre en évidence

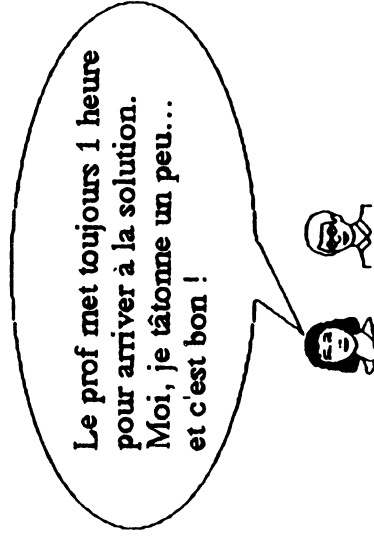
**la démarche
les méthodes de travail**

pour bien concevoir des programmes



Difficulté de les faire pratiquer

**de façon systématique,
dès le début du cours,
sur des exemples forcément simples.**



Pour mettre en évidence la démarche :

- ➔ isoler des comportements dans le travail global de programmation

Spécification du problème

Analyse

Programmation

- ➔ formaliser ces différentes étapes, en les faisant faire par 3 personnes différentes

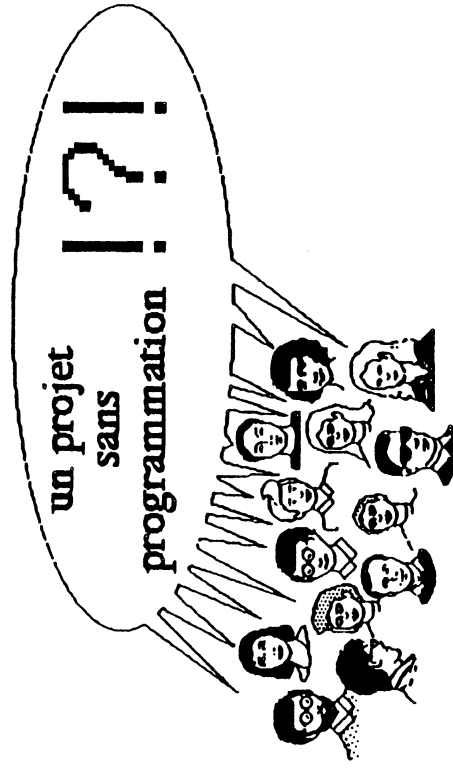
Le client

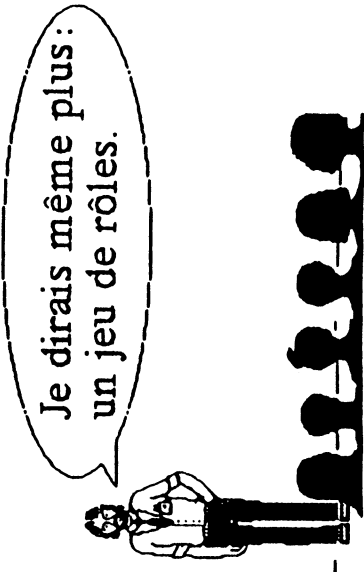
L'analyste

Le programmeur

Pour faire pratiquer ces démarches dès le début du cours :

- ➔ donner des problèmes difficiles ? *impossible*
- ➔ trouver un domaine différent de la programmation où l'on pratique des démarches analogues ?





Le jeu de rôles

CREER UNE B.D

L'analogie avec la programmation

CREER UN PROGRAMME

pour montrer :
 la nécessité et la difficulté des spécifications
 l'intérêt du processus d'analyse descendante

Le client

|

Le scénariste

|

Le dessinateur

Le client

|

L'analyste

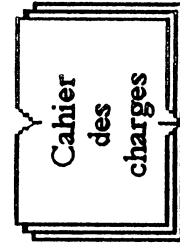
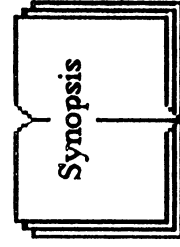
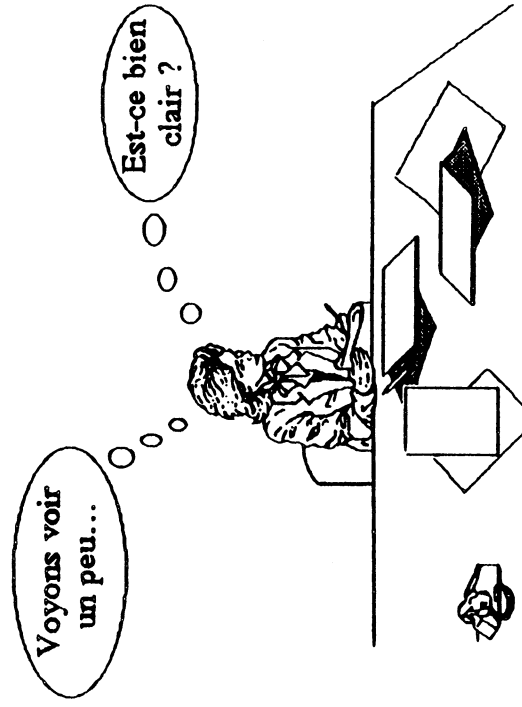
|

Le programmeur

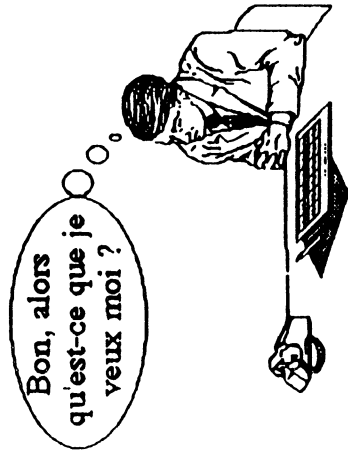
Le Scénariste

L'Analyste

étudient la faisabilité, lèvent les ambiguïtés...



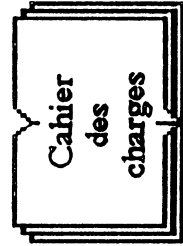
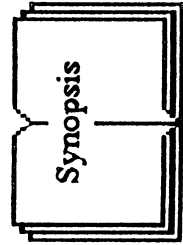
le Client



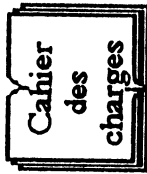
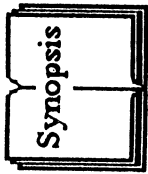
a une histoire à
faire raconter en
bande dessinée

a un problème
à faire résoudre
par **programme**

il spécifie



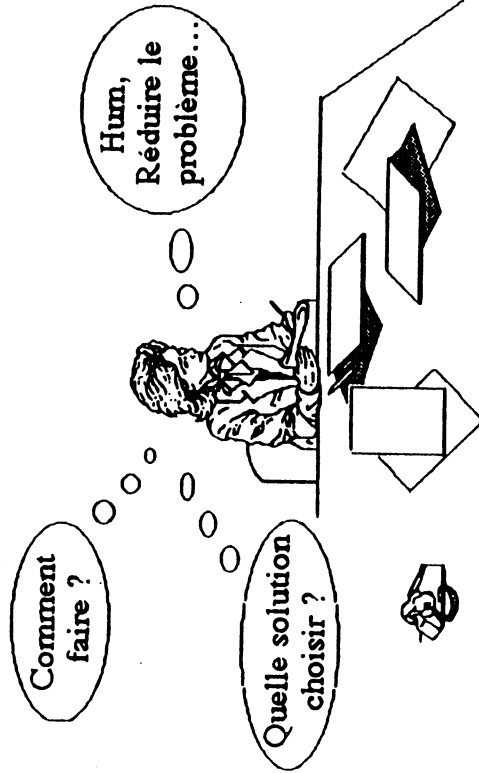
Alors...



Le Scénariste

L'Analyste

analysent le problème

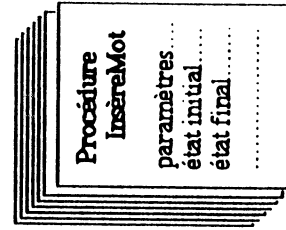
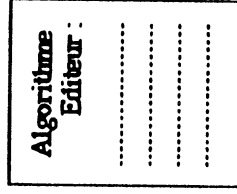
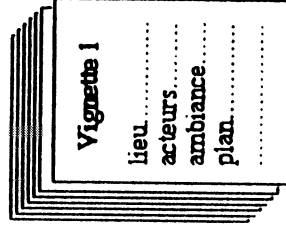
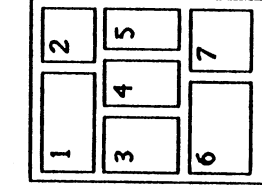
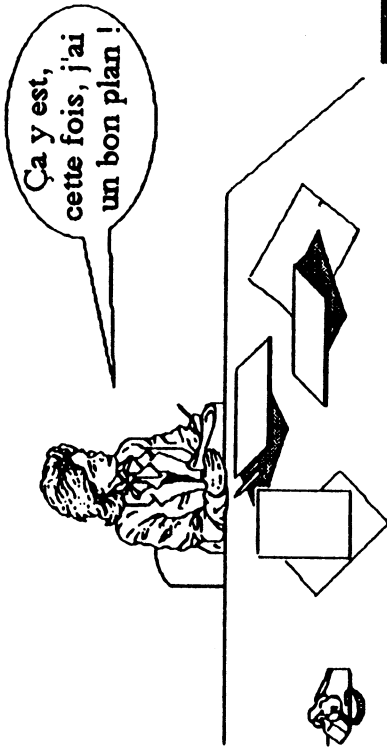


Et un peu plus tard...

Le Scénariste

L'Analyste

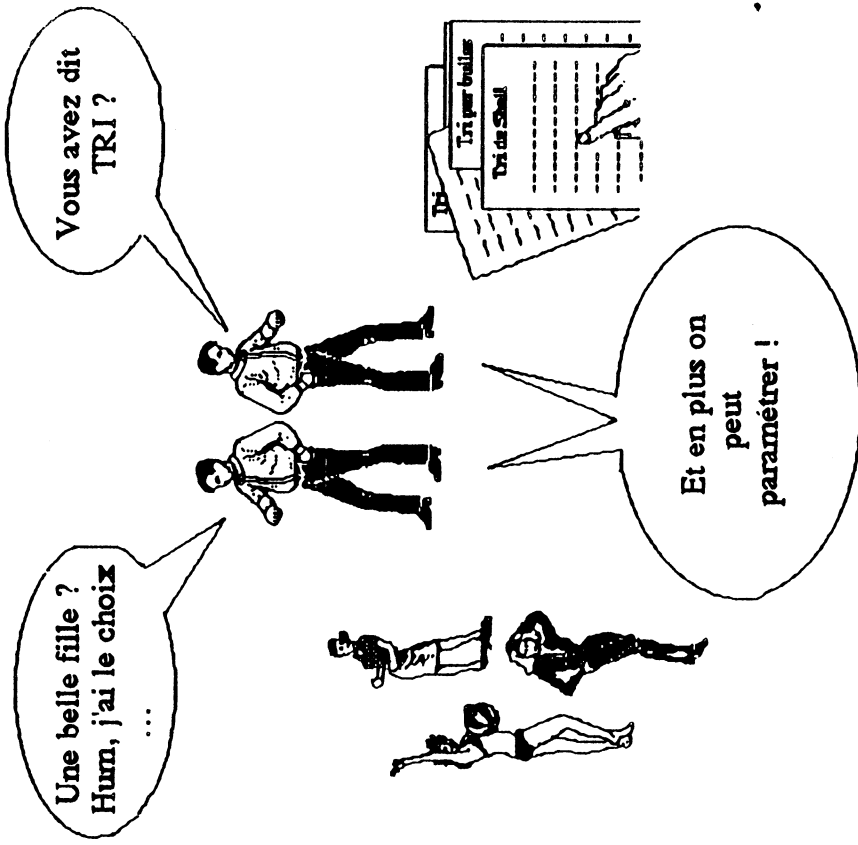
spécifient une solution



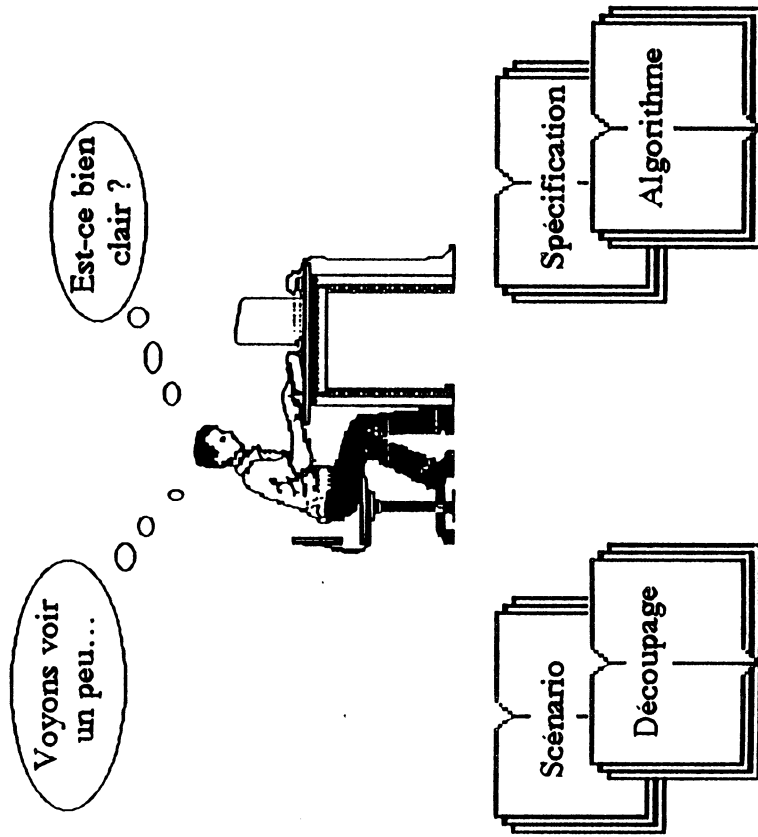
Découpage & Scénario

Algorithme général & Spécification des procédures

Le Dessinateur Le Programmeur
utilisent une bibliothèque



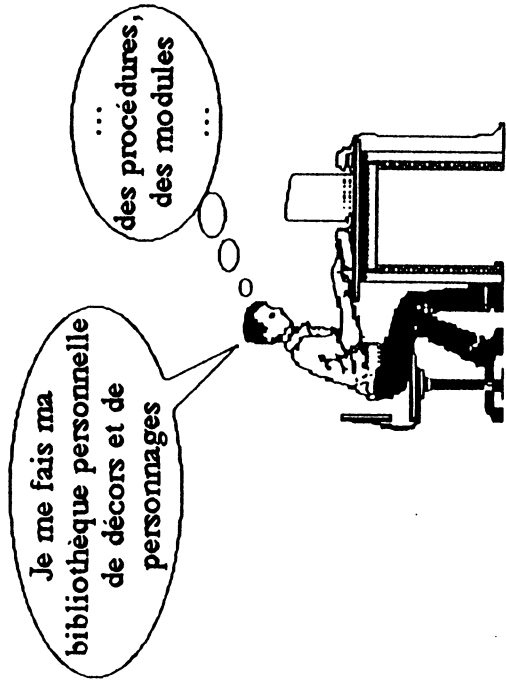
Le Dessinateur Le Programmeur
étudient la faisabilité, lèvent les ambiguïtés...



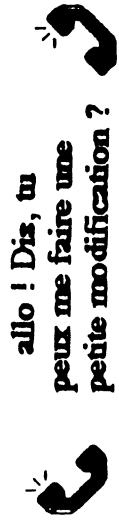
Le Dessinateur

Le Programmeur

**s'organisent pour être plus clairs,
plus sûrs, plus efficaces...**



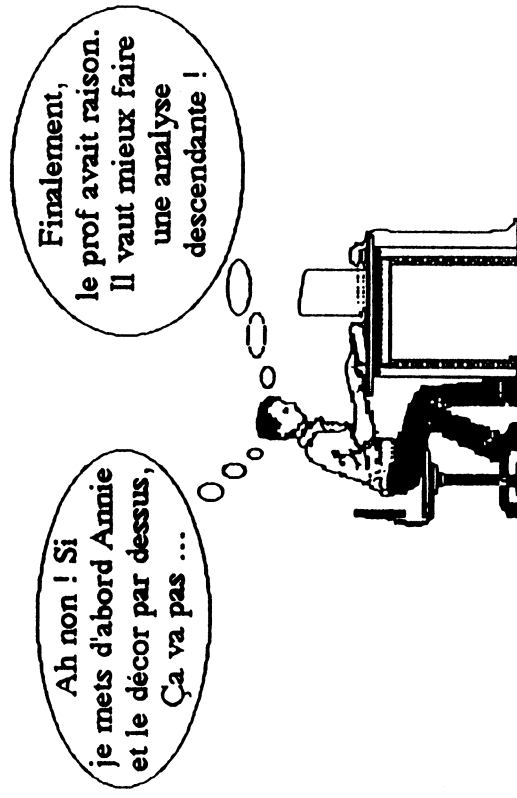
...et plus souples.



**allo ! Dis, tu
peux me faire une
petite modification ?**

Le Dessinateur **Le Programmeur**

travaillent avec méthode



Le jeu sur ordinateur

intérêt

L'ordinateur permet de fournir des outils simples et maniables

il facilite la mise en place d'un protocole de communication

il apporte une aide au meneur de jeu dans ses tâches d'arbitre, d'animateur et d'informateur

il peut permettre à un meneur de jeu d'animer plusieurs parties à la fois.

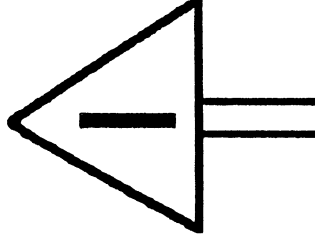
limite

il ne remplace pas un meneur de jeu humain

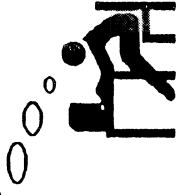
Réalisation

Le jeu fonctionne actuellement sur réseau de Apple Macintosh

Assurer le transfert vers la programmation



Bon, maintenant que le prof a fini de jouer, on va peut-être pouvoir travailler !



Pendant la partie :

le meneur de jeu

observe, suggère, débloque, relance, provoque,...

Avant, Après,

le professeur

observe l'étudiant en situation normale de travail,
fait des analogies avec des situations vécues dans le jeu...

UTILISATION DU JEU

QUI ?

le meneur de jeu = client

1 étudiant = scénariste

1 étudiant = dessinateur

ou bien

autres combinaisons,

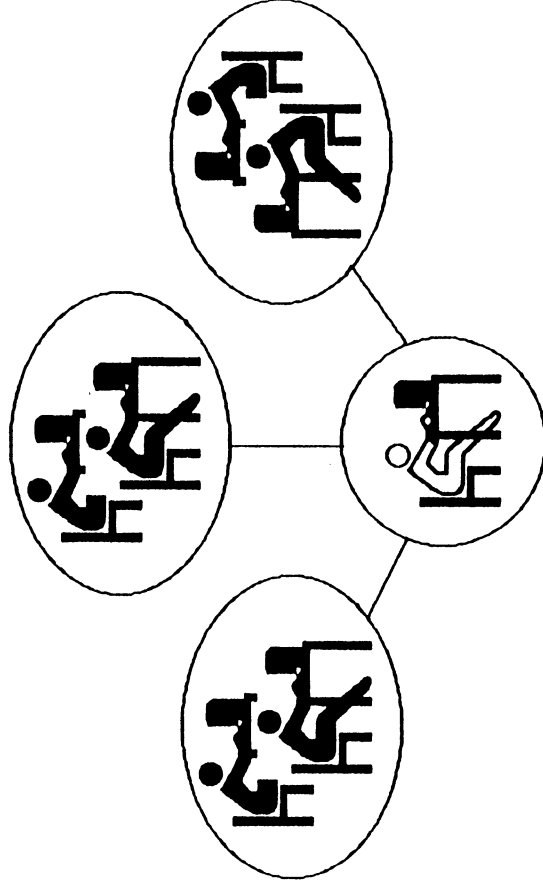
mais toujours avec un meneur de jeu

POUR QUOI FAIRE ?

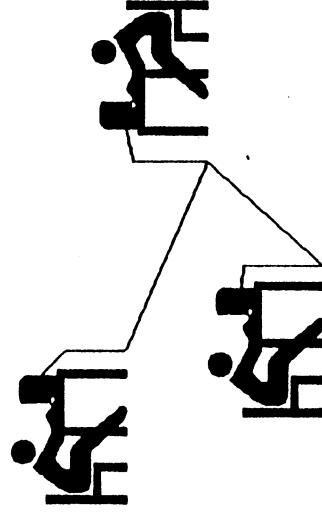
créer une page BD,
compléter une page BD proposée,
ou modifier une page BD proposée

MAIS OU ET QUAND ?

en séance TD, TP...



en libre service...





Annexe III :

Description technique du jeu

"Toi, Moi et Lui".



Description technique du jeu de rôles "Toi, Moi et Lui":

1. L'environnement de réalisation

Ce jeu de rôles a été développé sur **Apple Macintosh**.


Arrêtons-nous un instant sur cette famille de matériel, généralement très appréciée des utilisateurs. La philosophie du Macintosh tend à simplifier au maximum l'interface homme-machine, c'est-à-dire à rapprocher la machine des concepts familiers à l'homme plutôt que de forcer l'homme à s'adapter à la machine. Le résultat final est un produit remarquablement souple et qui nécessite très peu d'apprentissage pour l'utilisateur.

Si la complexité n'apparaît pas au niveau de l'utilisation des logiciels, elle n'en est pas moins présente du côté de la programmation des logiciels. Car, comme nous le fait remarquer Longevialle "Il n'y a pas de miracle. Plus le produit final est ergonomique, plus la programmation en est délicate" [Longevialle 85].

Les concepteurs du Macintosh ont cependant voulu faciliter le travail des programmeurs en mettant à leur disposition une "boîte à outils" regroupant un ensemble impressionnant de fonctions. C'est le bon usage de ces fonctions qui donne cet "air de famille" à tous les logiciels Macintosh. Ces fonctions sont disponibles dans la ROM, et représentent une aide très appréciable pour le programmeur.

Il est nécessaire pour programmer d'avoir une bonne connaissance de cette boîte à outils décrite en 5 volumes et 2200 pages dans *Inside Macintosh* (AppleComputer, Inc.). Les fonctions proposées sont regroupées en un certain nombre de "gestionnaires" (gestionnaire graphique, gestionnaire de ressources, de mémoire, de menus, de fenêtres, de dialogues...).

Cette boîte à outils permet notamment de séparer des informations "données" (taille et position initiales d'une fenêtre, messages d'alerte,...) du code exécutable. Celles-ci sont rangées grâce à des éditeurs spécialisés, dans un fichier "ressource"; elles sont accessibles depuis le code grâce au gestionnaire de ressources. Ces données peuvent ainsi évoluer facilement, sans que le programmeur doive retoucher le code.

De la même façon, les accessoires de bureau dont l'utilisateur peut disposer à tout moment dans le menu  permettent de télécharger l'application d'un certain travail qui ne lui serait pas spécifique. Un accessoire de bureau peut ainsi être mis au service de diverses applications. De plus, si un accessoire de bureau évolue ou si un nouvel accessoire offrant des fonctionnalités de même type apparaît plus performant, l'application peut bénéficier immédiatement et sans modification de cette évolution.

Notons aussi que la philosophie de l'interface Macintosh, si précieuse à l'utilisateur, impose une programmation de gestion d'évènements qui est plus proche d'une programmation "système" que d'une programmation classique, avec tous les problèmes que cela implique.

Nous avons utilisé l'environnement de programmation **Lightspeed Pascal** (THINK Technologies Inc.). Cet environnement permet une programmation modulaire en Pascal. Son compilateur présente certains des avantages d'un interpréteur et facilite ainsi la mise au point des programmes.

2. Le jeu de rôles "Toi, Moi et Lui"

Le jeu est constitué de **trois programmes** "client", "scénariste" et "dessinateur", écrits en Pascal. L'ensemble représente un code de l'ordre de 20 000 lignes.

Ces programmes utilisent:

- l'accessoire de bureau **Talkie Mac** (Analyses Conseil Informations),
- l'accessoire de bureau **Art Grabber⁺** (Macromind Inc.),
- l'accessoire de bureau **CheapPaint** (Macromind Inc.),
- une **bibliothèque de dessins** constituée à partir de dessins issus du logiciel VidéoWorks (MacroMind Inc.), du logiciel HyperCard (Apple Computer Inc.) et de la banque de dessins Art à la Mac (Springboard Software Inc.)

Pour créer une bande dessinée, ces programmes manipulent différents documents:

Document	Contenu	Fichier
"Personnages"	Description des personnages	"Client.FPerso"
"Synopsis"	Description du synopsis	"Client.FSyn"
"Découpage"	Description du découpage graphique	"Scen.FDecoup"
"Scénario"	Description du scénario	"Scenariste.FScen"
"Vignette 1", "Vignette 2", ...	Dessin de la vignette 1 Dessin de la vignette 2 ...	"Dess.FVign1", "Dess.FVign2", ...
"Dessin 1", "Dessin 2", ...	Dessin supplémentaire 1 Dessin supplémentaire 2 ...	"Dess.FDess1", "Dess.FDess2", ...

Le **volume global** occupé par l'ensemble des applications, de la bibliothèque de dessins et des fichiers produits est de l'ordre de 1600 KOctets. L'importance de ce volume s'explique par le nombre élevé des dessins traités. Il semble donc préférable de disposer d'un disque dur pour utiliser de façon confortable le jeu "Toi, Moi et Lui".

La version actuelle du jeu est une maquette qui nous a tout d'abord permis de montrer sa viabilité. Cette version est opérationnelle et doit maintenant être expérimentée de façon systématique auprès des étudiants. Des extensions en cours vont permettre d'une part de renforcer l'aspect jeu et d'autre part d'améliorer le confort des joueurs.

Table des matières

Plan	7
Introduction	11
Chapitre I. Une aide pour l'étude de la programmation: le laboratoire Arcade ...	17
I.1. Le laboratoire Arcade tel que nous le rêvions	21
I.2. Apprendre et enseigner dans le laboratoire Arcade	23
I.3. Approche de la programmation dans le laboratoire Arcade	34
I.4. Description du laboratoire	35
I.5. Richesse du laboratoire	46
I.6. Extensibilité du laboratoire	49
Chapitre II. Une activité particulière dans le laboratoire Arcade: le jeu	53
II.1. L'activité de jeu	57
II.1.1. Définitions et caractéristiques	57
II.1.2. Typologie des jeux	62
II.1.3. Les jeux de rôles	64
II.1.4. Les jeux sur ordinateur	67
II.2. Le rôle pédagogique du jeu	71
II.2.1. Une vue bibliographique de la question	71
II.2.2. Notre point de vue	76

Chapitre III. Jeu et programmation	83
III.1. La programmation et son univers	85
III.2. La programmation est-elle un jeu ?	87
III.3. Jeu et étude de la programmation	95
III.3.1. Le jeu sujet d'études	96
III.3.2. Le jeu outil d'apprentissage	98
III.3.2.1. Des réalités: le jeu des plaquettes, ChipWits, JAC	98
III.3.2.2. Des attentes	115
 Chapitre IV. "Toi, Moi et Lui", un jeu de rôles pour l'étude de la programmation ..	119
IV.1. Motivations pédagogiques	121
IV.2. Du changement de domaine au jeu de rôles	129
IV.3. Un jeu de rôles dans l'enseignement de la programmation	133
IV.4. "Toi, Moi et Lui" ou l'analogie avec la programmation	140
IV.5. Toi, Moi, Lui ou un autre: quel rôle pour l'ordinateur ?	146
IV.6. Les différentes utilisations pédagogiques de "Toi, Moi et Lui"	151
IV.6.1. Quel meneur de jeu ?	151
IV.6.2. Meneurs de jeu, créez votre propre scénario !	153
IV.6.3. Meneurs de jeu et enseignants: pour un transfert vers la programmation ..	158
IV.7. Prolongements	159
 Conclusion	169
 Références bibliographiques	173
 Annexes	
I. Jouons avec "Toi, Moi et Lui", manuel d'utilisateur	183
II. Jouons avec "Toi, Moi et Lui", livret du meneur de jeu	195
III. Description technique du jeu "Toi, Moi et Lui"	207

AUTORISATION DE SOUTENANCE

DOCTORAT 3ème CYCLE, DOCTORAT INGENIEUR,
DOCTORAT DE L'UNIVERSITE JOSEPH FOURIER - GRENOBLE 1

Vu les dispositions de l'Arrêté du 16 avril 1974,

Vu les dispositions de l'Arrêté du 5 juillet 1984,

Vu les rapports de Mr. BOUSSARD *Jean-Claude, professeur*.....
Mr. VIVET *Martial, professeur*
Mr. PERROT *Jean, professeur*.....

Mlle *GUERAUD Viviane*.....est autorisé(e)
à présenter une thèse en vue de l'obtention du *DOCTORAT DE*.....
L'UNIVERSITE JOSEPH FOURIER (GRENOBLE I).....

Grenoble, le **25 JAN. 1989**.....

Le Président de l'Université
Joseph Fourier - Grenoble 1


J.J. PAYAN



UN JEU DE RÔLES DANS LE LABORATOIRE ARCADE : UNE AUTRE FAÇON D'ENSEIGNER LA PROGRAMMATION

RESUME :

Le **laboratoire ARCADE** est un lieu d'expériences où les étudiants en programmation complètent, par la manipulation, l'observation et le jeu, la formation qui leur est donnée par ailleurs sous la forme classique. Ce laboratoire est une collection de logiciels organisés selon la métaphore d'un ensemble de salles dans lesquelles on se rend pour accéder librement aux diverses activités proposées.

Après la description générale de ce laboratoire, nous présentons un ensemble de réflexions et de propositions centrées sur l'activité de **jeu**. Nous axons ces réflexions sur l'intérêt des jeux pédagogiques dans l'enseignement de la programmation et nous déterminons quels types de jeux peuvent favoriser un tel enseignement.

Enfin, nous concrétisons ce travail par la proposition d'un **jeu de rôles** pratiqué par trois joueurs sur réseau d'ordinateurs. Ce jeu a pour objectif principal de faire vivre aux étudiants une expérience mettant en évidence les comportements nécessaires au programmeur, dans le travail qui le mène de l'énoncé du problème à l'une de ses solutions.

MOTS CLES :

Methodologie de la programmation, Didactique de la programmation, Enseignement assisté par ordinateur, Laboratoire d'enseignement, Jeu, Jeu de rôles.