

# THESE

présentée au

**Conservatoire National des Arts et Métiers**

pour obtenir le titre de

**Docteur en Informatique**

par

**Liliana VEGA-ZAZUETA**

## **Modélisation et analyse spatiale et temporelle des jeux vidéo basées sur les réseaux de Pétri**

Thèse dirigée par **M. Stéphane NATKIN**

Soutenue le 23 novembre 2004

Membres du jury :

Président

**M. Eric GRESSIER**      **CNAM-Paris, Laboratoire CEDRIC**

Rapporteurs

**M. Pascal ESTRAILLIER**      **Université de La Rochelle, Laboratoire L3i**  
**M. Imad SALEH**      **Université Paris 8, Laboratoire Paragraphe**

Examineurs

**M. Pierre COMBES**      **France Télécom R&D / DTL**  
**M. Norman GOUGH**      **Université de Wolverhamptom, Laboratoire RIATec's**  
**M. Stefan GRÜNVOGEL**      **Nomads Lab**  
**M. Qasim MEDHI**      **Université de Wolverhamptom, Laboratoire RIATec's**  
**M. Jean-François PEYRE**      **CNAM-Paris, Laboratoire CEDRIC**

*À ma mère, à mon père, à mes sœurs et frère,  
pour leur affection et leur compréhension.*

*À ma fille,  
qui, avant de naître, m'a accordé le temps nécessaire pour  
terminer cette quête.*

*Tout spécialement à Chema,  
pour avoir partagé ce projet avec moi et pour son amour.*

## *Remerciements*

Tout au long de mon séjour en France et pendant la durée de mon travail de recherche, j'ai eu la chance de rencontrer des personnes exceptionnelles. Une grande partie de ce travail a été achevé grâce à elles. Pour cette raison, je leur en suis très reconnaissante.

Je souhaite remercier tous les membres du jury. Particulièrement, M. Éric Gressier pour m'avoir fait l'honneur de présider ce jury, M. Pascal Estrailier et M. Imad Saleh d'avoir accepté la laborieuse tâche d'être rapporteurs de ma thèse, M. Pierre Combes, pour sa réponse si rapide à notre invitation, M. Norman Gough et M. Qasim Medhi pour leur disponibilité et leur intérêt. Je suis très reconnaissante à M. Jean-François Peyre, pour ses commentaires et ses corrections quant à la formalisation et la rédaction des concepts du chapitre 3, ainsi qu'à M. Stefan Grünvogel, pour son aide à la concrétisation du chapitre 4.

Je tiens à remercier de manière très spéciale mon tuteur, M. Stéphane Natkin, pour m'avoir accordé son temps et m'avoir donné un véritable exemple de l'esprit du chercheur, pour sa patience, pour m'avoir accueilli dans le laboratoire CEDRIC et m'avoir intégrée à ses projets.

J'ai apprécié de tous leurs conseils, réflexions et leur généreuse invitation à continuer mon travail de recherche.

Je remercie les chercheurs du 55 pour leur soutien. À Gérard Florin et Daniel Enselme pour maintenir une atmosphère agréable et propice à la recherche. À Viviane Gal, Alexandre Topol et Jérôme Dupire pour leur générosité, leur énergie, et leur enthousiasme contagieux. Ils m'ont montré que la recherche et les jeux vidéo font une chose sérieuse, passionnante et surtout, amusante. À chacun d'eux je dois des heures de patience et de dévouement vers mon travail. Un grand merci aussi à Nicolas Bouillot, Emmanuelle Doucet et Sami Evangelista. Merci pour leur aide lors des corrections d'orthographe et de grammaire de cette thèse.

Mes remerciements vont aussi vers Stéphanie Poux pour ses mots d'encouragement, sa chaleureuse compagnie, sa patience et son intérêt pour moi, mon travail et ma famille.

Merci à tous pour leur temps, leurs conseils et leur amitié.

Finalement, je souhaite exprimer ma reconnaissance à l'organisme mexicain CONACyT pour le support financier pendant mes travaux de recherche.

# Table de Matières

CHAPITRE 1 .....	9
<i>Introduction</i> .....	9
Méthodes d'analyse et de spécification dans les jeux vidéo.....	11
Plan de la thèse.....	12
CHAPITRE 2 .....	13
<i>L'Etat de l'art</i> .....	13
2.1. Principes de l'analyse de jeux vidéo.....	14
2.1.1 Introduction du chapitre.....	14
2.1.2 Concepts de base.....	14
Jeux, Jeux vidéo et Jeux d'ordinateur .....	14
But, choix et règles .....	17
2.1.3 De l'analyse de film à l'analyse des jeux vidéo.....	17
2.1.4 De l'analyse littéraire à l'analyse des jeux vidéo.....	19
2.2. Game Design et Level Design .....	21
2.2.1 Introduction à l'écriture des jeux .....	21
2.2.2 Game Design.....	21
Objets .....	22
2.2.3 Level Design .....	23
2.2.4 La notion de labyrinthe .....	24
2.2.5 Le gameplay .....	25
2.3. Différentes approches dans l'analyse de jeux vidéo .....	26
2.3.1 Approches Narratives.....	27
Théâtre .....	27
La notion de Quête.....	27
La narration interactive .....	29
2.3.2 Approches sociologiques .....	29
Les Schèmes de Frintz .....	29
Modèles socio comportementaux .....	30
2.3.3 Approches Ludologiques .....	31
L'analyse esthétique, poétique et d'écriture dans l'espace d'Espen Aarseth ...	31
La thèse de Jesper Juul.....	32
L'Approche centrée dans le jeu de Celia Pearce .....	33
2.3.4 Approches formalistes .....	34
La méthode d'analyse de jeux de Lars Konzack .....	34
L'approche MDA (mécanique, dynamique et esthétique ) .....	35
2.3.5 Approches fonctionnelles.....	36
Les techniques de génie logiciel .....	36
Les systèmes basés sur des règles.....	39
2.4. Conclusion .....	41

<b>CHAPITRE 3</b> .....	43
<i>Modélisation avec les réseaux de Pétri</i> .....	43
3.1. Syntaxe des réseaux de transactions .....	44
3.1.1. Retour sur les besoins d'expression .....	44
3.1.2. Segmentation d'un jeu vidéo .....	44
Niveaux .....	45
Quêtes .....	45
Transactions .....	46
3.1.3. Réseau de transactions .....	47
Description formelle de la syntaxe .....	47
Description informelle de la sémantique d'exécution .....	48
3.2. Introduction aux réseaux de Pétri .....	49
3.3. Interprétation des réseaux de transactions en terme de réseaux de Pétri .....	52
3.3.1. Modèle d'une Transaction .....	52
3.4. Réseau de Pétri associé à un Réseau de Transactions.....	54
3.4.1. Modèle d'un réseau de Transactions.....	54
3.4.2. Opérateurs d'ordre .....	55
3.4.3. Opérateurs Logiques .....	58
3.4.4. Réseau de Pétri associé à un réseau de Transactions.....	58
3.4.5. Abréviations .....	60
3.4.6. Propriétés d'un réseau de Pétri associé à un réseau de transactions .....	62
3.5. Quêtes .....	73
3.6. Niveaux .....	74
3.7. Exemple .....	75
3.7.1. Description du concept du jeu Myst .....	75
Solution du niveau 1 .....	76
3.7.2. Traduction de la structure en Réseaux de Pétri.....	77
3.8. Conclusion .....	78
<b>CHAPITRE 4</b> .....	79
<i>Intégration de la topologie dans la modélisation</i> .....	79
4.1. Problématique de l'espace dans l'écriture des jeux .....	80
4.1.1. La représentation de l'espace dans les mondes virtuels.....	81
Niveau géométrique .....	83
Niveau cartographique .....	83
Niveau topologique .....	84
Niveau sémantique .....	84
4.1.2. La notion de structuration de l'espace du jeu selon Rollings .....	85
4.2. Modélisation de l'espace d'un jeu .....	87
4.2.1. Un modèle permettant de modéliser l'évolution de la topologie en fonction des transactions .....	87
4.2.2. Un modèle permettant de faire évoluer le réseau de Pétri en fonction de la topologie .....	91
Les réseaux de Pétri colorés .....	91
4.2.3. Un modèle d'un jeu avec localisation et dépendance réciproque des transactions et de la topologie.....	95
4.2.4. Mise en œuvre du modèle précédent .....	96
4.2.5. Implantation dans les outils de conception d'un jeu .....	98
4.3. Conclusion .....	102

<b>CHAPITRE 5</b> .....	103
<i>Analyse du jeu</i> .....	103
5.1. Analyse du jeu DOOM3 .....	104
5.1.1. Introduction.....	104
5.2. Game design et level design .....	105
5.2.1. Game design.....	105
Scénario du niveau 1 .....	105
Scénario du niveau 9 .....	105
5.2.2. Level design.....	106
Description des quêtes du niveau 1 .....	106
Modélisation de l'espace du jeu du niveau 1 .....	107
Description des quêtes du niveau 9.....	111
Modélisation de l'espace du jeu du niveau 9.....	114
5.3. Conclusions.....	118
5.3.1. Aspects techniques.....	118
5.3.2. Analyse du jeu.....	119

<b>CHAPITRE 6</b> .....	121
<i>Conclusion Générale</i> .....	121
Travaux futurs .....	122
<i>Publications</i> .....	125
<i>Références</i> .....	127
<i>Annexes</i> .....	133



# CHAPITRE 1

## *Introduction*

---

L'industrie des jeux représente une part de plus en plus importante du marché des loisirs au niveau mondial. En 2003, ses ventes ont atteint le record de 18.2 milliards de dollars<sup>1</sup>, ce qui la place comme l'industrie culturelle ayant la croissance la plus rapide actuellement. Une telle croissance s'explique en partie grâce à l'évolution de son public. Tout d'abord, le panel des utilisateurs s'étend des enfants aux jeunes adultes et adultes, des « *hardcore gamers* » à de multiples profils de joueurs et d'une population majoritairement masculine à un ensemble mixte ( de 25 à 69% des femmes selon le pays<sup>2</sup>).

Un autre indicateur de l'évolution de ce marché est la multiplication et la spécialisation, récentes et rapides, de ses acteurs. Les premiers programmeurs de jeux, isolés et passionnés, se sont progressivement transformés en petits studios indépendants d'édition, de production et de distribution (avec un ou deux employés - programmeurs et graphistes) jusqu'à devenir de grandes entreprises spécialisées (avec au moins une vingtaine d'employés avec des métiers très spécifiques). Les organismes représentatifs, qui n'existaient pratiquement pas il y a 10 ans, sont aujourd'hui présents à différents niveaux dans les domaines industriel, politique et académique. Parmi les plus importants, on trouve l'APOM<sup>3</sup> (Association des Producteurs d'Oeuvres Multimédia), le SELL (Syndicat des Editeurs de Logiciels de Loisirs) en France, l'ESA<sup>4</sup> (*Entertainment Software Association*) aux Etats-Unis et l'IGDA (*International Game Developer Association*) au niveau international.

L'industrie des jeux vidéo ne peut cependant pas être évaluée seulement par rapport à son chiffre d'affaire. Elle possède en effet des caractéristiques socio-culturelle et technologique<sup>5</sup> fortes. Sous son aspect socio-culturel, l'étude des jeux a traditionnellement occupé une place importante dans les sciences humaines (ludologie). Les jeux vidéo représentent un nouveau défi. Cependant, leur structure particulière fait qu'ils ne peuvent

---

1 Source : Lettre d'informations No 68 – 4 octobre 2004. Agence Française pour le Jeu Vidéo.

2 *Essential Facts About the Video and Computer Game Industry*. Enquête effectuée en 2002 par l' IDSA.

3 Syndicat professionnel créé en Février 2002, regroupant les principaux créateurs français indépendants de jeux vidéo. Adresse électronique : <http://www.apom-france.org/>

4 Avant IDSA (Interactive Digital Software Association) est l'équivalent de la SELL en France. Adresse électronique : <http://www.theesa.com/>

5 Yves GUILLEMOT, Jean-Claude LARUE et Antoine VILLETTE. Lettre Ouverte APOM - SELL - UBISOFT : A vous de jouer, M. le Premier Ministre !, 11 May 2004. Disponible à partir du site internet de l' APOM.

se restreindre à cette approche uni-disciplinaire, les propulsant comme un domaine des plus attrayants pour les chercheurs et les académiciens d'horizons divers. Les modèles de communication utilisés font émerger un discours social particulier. Basés sur de nouveaux modèles narratifs et esthétiques, les relations interpersonnelles (joueur-joueur, joueur-ordinateur) sont construites dans un espace-temps privilégié. Les technologies et les dispositifs employés (ordinateurs, consoles, PDAs, portables, etc.) favorisent l'apparition de nouveaux canaux de communication. Les jeux massivement multijoueurs en ligne sont un exemple clair de ces nouvelles formes de socialisation et convergence technologique.

Rappelons que les jeux vidéo sont nés au moment où les ordinateurs ont commencé à être utilisés à d'autres fins que le calcul scientifique et la gestion, donnant naissance au multimédia numérique. Selon Stéphane Natkin, l'industrie du jeu vidéo est à l'avant garde des médias du XXI siècle [Natkin04], les jeux solo pouvant être vu comme une évolution du cinéma, les jeux en ligne comme une anticipation de la fusion de la télévision et du Web. Si ce pronostique s'avère exact, il se peut que d'ici quelques années la plus grande partie de la production informatique relève de l'audio visuel interactif. Ceci concerne le domaine des loisirs mais aussi tous les aspects qui relèveront d'une interface évoluée plaçant les utilisateurs à la frontière d'un univers virtuel de collaboration.

Face à ce développement d'univers d'un nouveau style les développeurs de jeux ont conçu une méthode de développement originale et efficace. Elle trouve son origine autant dans l'informatique que l'audio visuel. Elle n'est pas formalisée, mais elle démontre que les techniques d'analyse informatique d'une part et de développement de l'audio visuel d'autre part ne sont pas suffisantes à couvrir les besoins. Depuis quelques années, différents chercheurs ont commencé à proposer des méthodes d'analyse et de conception des jeux. Elle est à la confluence de trois domaines : la narration audio visuelle, le génie logiciel et l'analyse cognitive et sociale des comportements du joueur. Un jeu est d'abord destiné à faire plaisir à ses utilisateurs. Ses seuls critères de réussites ne peuvent provenir que d'une analyse perceptive. C'est d'abord une œuvre audio visuelle qui doit chercher des mécanismes narratifs et perceptifs originaux. Enfin c'est un logiciel interactif dont, on le sait dans d'autres applications, la conception et la validation sont très complexes.

Un jeu est un système complexe, qu'on le considère du point de vue du joueur ou du concepteur. Les éléments à prendre en considération par rapport à cette complexité sont nombreux [Mallender99, Le Diberder02, Kreimeier03] :

- La nature des jeux en tant qu'œuvres collectives et interactives, l'augmentation du nombre de personnes participantes, leurs différents profils et connaissances.
- L'utilisation d'un langage "spécialisé" pour chaque société, mais encore non normalisé dans l'industrie.
- L'absence de méthodes formelles pour la planification et la définition du jeu.

Tous ces éléments peuvent entraver le développement. Il existe cependant quelques tentatives et projets intéressants en ce qui concerne ces problématiques.

Par exemple, une des difficultés majeures reconnues dans cette industrie est le manque ou la mauvaise communication entre les différents membres de l'équipe de création d'un jeu, principalement dans les rapports entre concepteurs et programmeurs. Les premiers efforts pour décrire et normaliser les méthodes de conception et de développement des jeux proviennent des protagonistes eux mêmes (*game designer*, *level designers* et

programmeurs). Des projets comme *FADT* et *400 rules* figurent comme les premières initiatives sérieuses pour établir un langage commun pour communiquer les concepts de base et échanger des idées pendant le processus de création d'un jeu. *Patterns* et *MDA* sont autres initiatives dirigées vers la planification, la spécification et la formalisation de ce processus de création. La série de conférences et de tables rondes organisées par l'IGDA ou la GDC retracent ces efforts [Kreimeier03].

Cette complexité à la fois de structure et d'utilisation, associée à la jeunesse du phénomène, met en évidence un manque cruel d'outils d'analyse adaptés, tant pour les professionnels du jeu vidéo que pour les universitaires. Comment qualifier cette structure ? Est-elle trop simple ou complexe ? Quels effets produits-elle chez le joueur ? Comment faire pour provoquer des nouveaux comportements dans le joueur ?

Les outils théoriques envisagés posent trois types de problèmes devant être formalisés :

Comment concevoir un jeu intéressant en terme de *gameplay* et de narration,  
Comment concevoir un univers, audio visuel interactif attractif et immersif,  
Comment décrire les parcours dans le temps et dans l'espace qui sont donnés au joueur ?

Avant d'établir des outils théoriques et pratiques de synthèse, il faut, dans ces trois domaines disposer de principes d'analyse qui permettent de comprendre et de comparer les réponses qui ont été données à ces questions dans les jeux existants. Dans cette thèse nous faisons un état de l'art relatif à ces trois questions et nous nous concentrons sur la troisième qui relève le plus d'une approche de formalisation au sens mathématique du terme.

## **Méthodes d'analyse et de spécification dans les jeux vidéo**

Le sujet qui nous occupe dans cette thèse est l'analyse et la spécification de jeux vidéo. Notre objectif consiste à formaliser la description de la scène en modélisant les actions que le joueur peut effectuer, les objets dont il y a besoin pour effectuer ces actions et leur localisation dans l'univers du jeu. La correspondance entre ces trois éléments fait partie du mécanisme central du jeu, le *gameplay*, et détermine en grande partie la conception générale.

Pour atteindre cet objectif, nous avons considéré l'utilisation de certaines techniques d'analyse et de spécification informatique comme les réseaux de Pétri et les hypergraphes. Ces techniques s'appliquent au travers d'une méthode qui permet de résoudre de manière plus fluide et efficace les problèmes qui apparaissent dans les différentes étapes du cycle de vie des logiciels. Bien qu'il s'agisse d'abord d'employer ces techniques pour l'analyse de jeux vidéo déjà existants, dans une perspective d'étude et de critique de ceux-ci, nous considérons la possibilité de les utiliser pour la spécification et la conception de nouveaux jeux.

Pourquoi nous intéressons-nous à l'analyse formelle et à la critique de jeux vidéo ? Plusieurs réponses sont possibles. Tout d'abord, acquérir une meilleure compréhension des éléments constitutifs d'un jeu et des relations existantes entre eux : hiérarchies, dépendances et échanges des données (communication). Ces connaissances permettent de produire une théorie de jeux qui explique comment fonctionnent ces éléments, l'importance de chacun et la manière dans laquelle ils peuvent ou doivent être combinés pour produire

un type de jeu particulier. D'autre part, pouvoir établir une méthodologie critique formelle comme support de base à l'étude, à la compréhension et à l'appréciation des jeux, pour la constitution d'une mémoire culturelle de ce nouveau, mais déjà important, type de media.

Toutefois il n'existe pas encore de méthode formelle décrivant les techniques de création utilisées dans l'industrie. Notre travail de thèse part dans cette direction, comme une contribution à la formalisation de certains aspects de l'élaboration d'un jeu. La méthodologie que nous proposons s'insère dans les pratiques actuelles de l'industrie, point que nous considérons essentiel pour l'acceptation et l'utilisation éventuelle de notre méthode.

## **Plan de la thèse**

Cette thèse présente d'abord les travaux effectués sur les techniques et les méthodes utilisées dans l'analyse et la spécification de jeux vidéo. Cette compilation, présentée dans le chapitre 2, n'est pas exhaustive mais représentative des différents courants identifiés jusqu'à présent. La classification que nous proposons pour la présentation de telles analyses part d'une vision humaniste, en passant par quelques courants semi-formels, jusqu'à arriver à une analyse scientifique, basée sur quelques techniques de génie logiciel. Certains de ces travaux, initialement opposés, commencent, avec le cours des années et les nouvelles appréciations sur les jeux vidéo, à adopter des points de vue en communs et à converger par plusieurs aspects.

Dans le chapitre 3, nous proposons un modèle capable de décrire formellement les propriétés de l'ordre des séquences d'actions dans un jeu. Grâce à la modélisation avec des Réseaux de Pétri, certaines propriétés, comme la cohérence scénaristique peuvent être évaluées et la sémantique d'exécution expliquée. Le type d'analyse que nous proposons est pertinent dans le cas particulier de jeux vidéo avec les caractéristiques suivantes : le besoin d'avoir une histoire à raconter, un objectif à accomplir prédéterminé et un processus analytique ou des choix d'où le joueur peut réaliser les actions. Les jeux d'aventure et de stratégie sont des bons exemples de ce genre.

D'autres aspects importants dans l'analyse d'un jeu sont la description, la distribution et la localisation des objets concernés dans la réalisation d'une action. Cette notion d'espace, distinguant le processus d'écriture d'un jeu vidéo d'autres formes de production multimédia [Natkin04, Aaestern03], est prise en compte dans notre proposition d'analyse grâce à la description topologique du niveau de jeu. Cette définition a plusieurs implications importantes: la cohérence narrative, la cohérence topologique et enfin, la cohérence entre la narration et la topologie. Pour la représentation de la description topologique nous utilisons des hypergraphes. Ensuite, pour vérifier la cohérence du niveau, nous mettons en relation les modèles des séquences des actions en réseaux de Pétri avec les modèles de description topologique des hypergraphes.

Dans le chapitre 5, nous illustrons notre propos par l'application de notre méthode à l'analyse du jeu DOOM 3.

# CHAPITRE 2

## *L'Etat de l'art*

---

### Contenu du Chapitre

Dans ce chapitre nous présentons l'état actuel du développement dans les méthodes d'analyse des jeux vidéo. Nous introduisons les premières techniques utilisées pour l'analyse et la spécification d'un jeu empruntés à d'autres domaines comme le film et la littérature. Lors de la présentation du processus d'écriture d'un jeu, nous montrons les différentes approches sur l'analyse de jeux selon notre propre classement.

### Organisation du chapitre :

Section 2.1. Principes de l'analyse de jeux vidéo – p. 14 : Introduction aux concepts de base et aux premières analyses de jeux vidéo.

Section 2.2. Le *Game Design* et le *Level Design* – p. 21 : Présentation du processus d'écriture des jeux vidéo. Définition des étapes principales (Game et Level Design) et de la notion du *gameplay*.

Section 2.3. Différentes approches dans l'analyse de jeux vidéo – p. 26 : Classement des différentes approches et discussion sur leurs différents points de vue.

Section 2.4. Conclusion – p. 41 : Résumé de points à prendre en considération par rapport à l'analyse de jeux vidéo.

## 2.1. Principes de l'analyse de jeux vidéo

### 2.1.1 Introduction du chapitre

Les jeux vidéo semblent être les plus avancés des médias interactifs. Contrairement à de nombreux sites Web, un jeu est un ouvrage réalisé pour un public bien défini. Ceci permet à l'industrie des jeux de définir ses méthodes de conception et de production. Même si la culture de jeu est encore jeune, la capacité de créer un jeu utilisé de par le monde est la preuve d'une certaine maturité. Le futur des jeux vidéo, à partir des jeux MMOG (jeux massivement multijoueurs en ligne) et des jeux proactifs, est un paradigme pour le développement de médias interactifs en ligne [Natkin03].

En trois décennies, les jeux vidéo sont devenus des œuvres audio visuelles d'un nouveau genre culturel. La capacité d'établir une analyse critique des jeux, pour comprendre et comparer leur différentes structures, décomposer et classer ces éléments constitutifs, la mise en scène, le rapport entre le son et l'image... sont des moyens essentiels au développement de cette nouvelle forme de médias [Le Diberder02], [Natkin04]. Le développement d'une théorie analytique des jeux est donc le thème central de cette thèse. Plusieurs chercheurs ont, depuis quelques années travaillé dans ce sens [Aarseth03]. Les différentes propositions faites dérivent des méthodologies utilisées dans d'autres domaines comme l'analyse littéraire ou la théorie cinématographique d'une part et les méthodes d'analyse et de conception informatique d'autre part. Ce chapitre fait une analyse critique des principaux travaux réalisés. Il débute par un paragraphe destiné à définir les principaux termes utilisés et à préciser les pratiques de production du domaine des jeux vidéo.

### 2.1.2 Concepts de base

#### Jeux, Jeux vidéo et Jeux d'ordinateur

Répondre à la question « Qu'est ce qu'un jeu vidéo ? » n'est pas simple. De manière générale un jeu peut être défini comme une activité ludique qui implique un ou plusieurs joueurs, un but à atteindre par ces joueurs et un ensemble de règles déterminant les activités que les joueurs peuvent ou pas réaliser [Dico01]. Un jeu vidéo est donc d'abord un jeu qui se joue seul ou à plusieurs. La nature de ce jeu peut être très diverse allant de jeux proche d'un sport d'adresse ou d'équipe jusqu'à des versions informatisées du jeu d'échec. Dans tous les cas ce qui caractérise un jeu vidéo est l'intervention d'un ordinateur, d'un écran et de haut-parleurs.

Dans cette thèse nous appelons jeu vidéo une œuvre audio visuelle interactive à caractère ludique qui utilise pour sa reproduction un appareil basé sur une technologie informatique.

L'ordinateur peut avoir deux rôles distincts. Il est toujours la machine qui simule l'univers du jeu : Il fabrique en temps réel un univers visuel et sonore dont l'évolution dépend de la conception du jeu et des actions du joueur. L'ordinateur contrôle également l'observation des règles du jeu. Dans certains cas, et en particulier pour tous les jeux à un joueur, dits jeux solos, l'ordinateur est également l'adversaire du joueur.

Lorsque l'ordinateur est un système dédié en environnement privé (e.g. jeux joués sur des consoles connectées à un téléviseur ou sur des mini consoles portables, téléphones mobiles, PDA's) on parle de jeu sur console (*video games*). Lorsque ce dispositif est utilisé dans une salle publique de jeu, on parle de jeu d'arcade. Les jeux joués sur ordinateurs généraux sont appelés en anglais *computer games*.

Il existe plusieurs classifications des jeux vidéo selon des critères soit historiques, éditoriaux ou narratifs. La presse spécialisée distingue, en général, les jeux d'action, d'aventure, de stratégie, de sport. Nous utilisons une classification qui repose sur les mécanismes de jeu [Natkin 04, Gal 02].

Considérons d'abord les jeux "solo", pour lesquels le joueur est face à l'ordinateur. Nous distinguons quatre grandes classes de jeux : les puzzles, les jeux de stratégie, les jeux d'action et les jeux d'aventure.

Les jeux de puzzle (terminologie empruntée à Rollings [Rollings00]) sont la reprise sur ordinateur des jeux "classiques" à plusieurs joueurs dans lequel l'ordinateur joue le rôle d'un des joueurs: jeu de go, échecs, Monopoly, ... Ce qui caractérise ces jeux, c'est la position symétrique du joueur et de l'ordinateur : tous les deux connaissent au départ les règles qui leur sont imposées et ont un rôle symétrique.

Les jeux de stratégie donnent au joueur la sensation de gérer un univers. L'intérêt du jeu réside ici dans la complexité apparente des règles qui régissent l'univers virtuel et donc la subtilité des stratégies de jeu.

Un jeu d'action est un jeu dont le principal moteur ludique est la dextérité du joueur. « Un tas de boutons et de manettes à agiter frénétiquement », selon Rollings. Les jeux de combat : arts martiaux, jeux de tirs à diverses armes (FPS : *First Person Shooter*), ... sont des prototypes du genre.

Les jeux d'aventure sont les plus scénarisés. Un tel jeu est une histoire dont le joueur est le héros. Il a toujours une trame narrative et des règles qui ramènent nécessairement le joueur dans cette trame.

Dans la littérature on trouve de nombreuses autres classes de jeux solo, mais du point de vue du *gameplay*, ce sont des combinaisons des trois dernières.

Les jeux multi joueurs relèvent des mêmes types d'univers que les jeux solo. Un jeu à souvent un mode solo et un mode multi joueurs. On peut donc également parler dans ce cas, de jeux de stratégie, de sport, d'aventure... Mais les motivations des joueurs, et par conséquent le *gameplay*, sont souvent très différents.

Un jeu multi joueur peut être soit coopératif, les joueurs jouent ensemble contre l'ordinateur, soit compétitif, les joueurs jouent, seuls ou en équipe, les uns contre les autres.

Les jeux coopératifs sont, en terme de *gameplay*, une extension des jeux solo, ajoutant la dimension de l'effort collectif pour vaincre la machine. Les jeux compétitifs reposent sur de tout autres principes, qui les rapprochent des jeux de sociétés ou des sports. La différence essentielle entre les deux types de jeu tient à la connaissance des règles. Dans les jeux contre l'ordinateur, le joueur ne connaît pas exactement les règles du jeu. Le contrôle de

l'apprentissage fait partie du *gameplay*. La situation entre le joueur (les joueurs dans un jeu coopératif) et l'ordinateur n'est pas symétrique. Dans les jeux compétitifs, tous les joueurs doivent avoir le même niveau de connaissance des règles, même si, dans certains cas, ils acceptent de ne pas jouer le même rôle dans le jeu.

Enfin il est nécessaire de distinguer les jeux en communauté ouverte et fermée. Dans le premier cas un joueur n'a dans ce cas pas de contrôle sur l'identité des personnes avec qui il joue. En général, les joueurs ne se connaissent pas physiquement et se rencontrent via leurs avatars dans l'univers du jeu. Ces jeux se jouent sur Internet. Les jeux en communautés fermées mettent en compétition des joueurs ou des équipes qui se sont identifiées et qui acceptent de jouer ensemble. De tels jeux peuvent se jouer en salle de jeu (*LAN partie*), dans ce cas les joueurs se connaissent physiquement ou sur Internet. Dans ce dernier cas, les joueurs peuvent ne se connaître que via le réseau. Les communautés sont limitées à une dizaine de joueurs.

La figure suivante, extraite de [Natkin04] représente une classification des jeux reposant sur les bases précédentes.

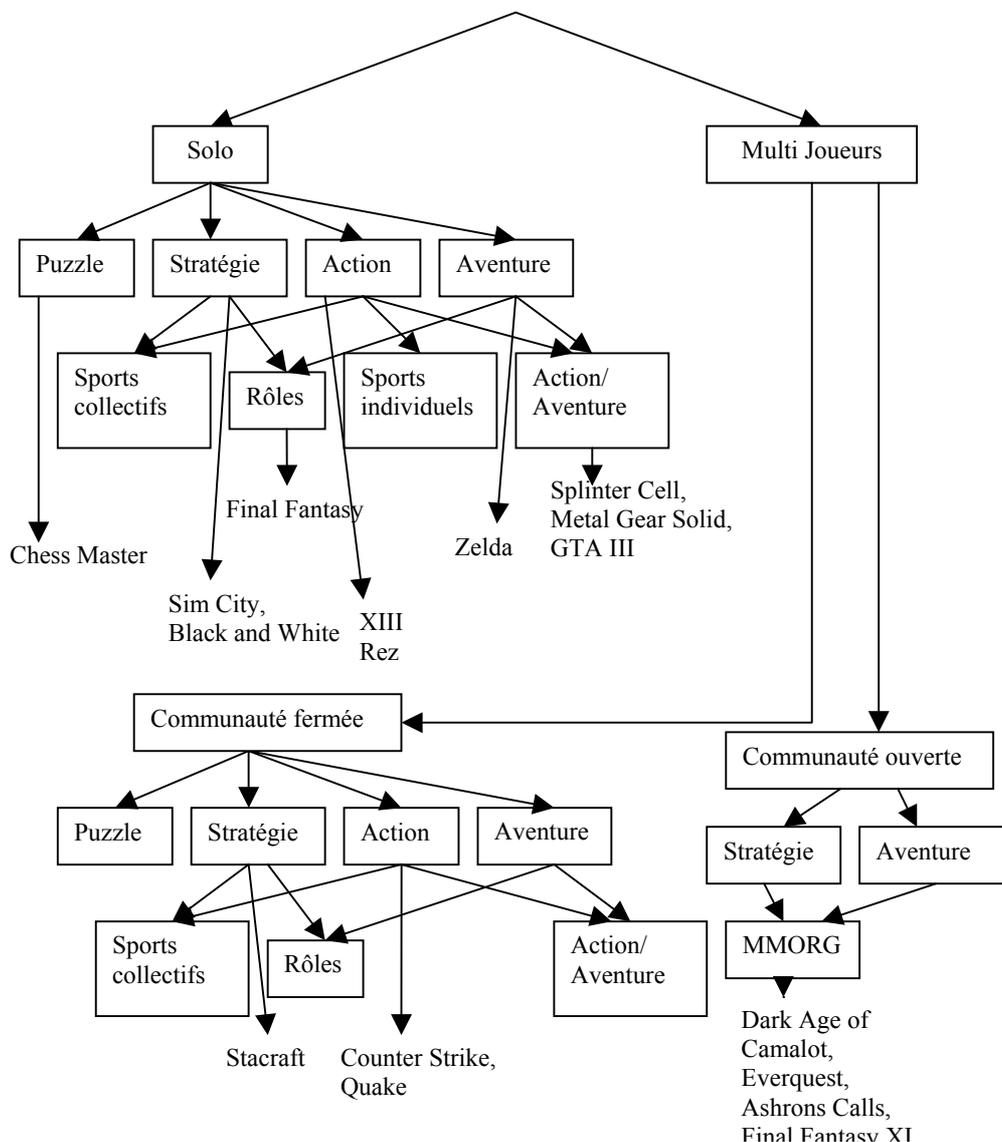


Figure 2.1 : Classification des jeux vidéo.

## Joueur

Le joueur est l'utilisateur du système de jeu qui accepte le défi et les règles de l'univers virtuel. Dans cet univers, sa présence est représentée soit par un icône soit par un personnage appelé avatar du joueur. L'avatar peut être visuellement identifiable (e.g. perspective en troisième personne), ou le jeu est vu au travers du regard de l'avatar (e.g. perspective en première personne).

Dans les jeux en solo (un seul joueur) le personnage rencontre des obstacles qui s'opposent à la réussite d'un objectif. Pour jouer, progresser et éventuellement gagner, le joueur doit s'engager, accepter les règles du jeu, prendre des décisions et gérer des ressources en fonction des informations acquises [Guardiola00]. Dans les jeux multijoueurs, ce type d'objectif est un moyen pour développer des activités sociales avec les autres participants [Pearce01, Thorhause03].

### **But, choix et règles**

Le but général d'un jeu définit le critère de victoire du joueur. Il peut être présenté au joueur de plusieurs manières. Il peut être implicitement ou explicitement introduit directement au début du jeu ou dévoilé au fur et à mesure que le joueur progresse à partir d'une hiérarchie de buts [Guardiola00]. Un ensemble de sous buts peuvent être accomplis dans différents ordres, mais, en général dans un niveau donné, les mêmes buts doivent être réalisés [Lindley02].

Indépendamment du type de jeu, il existe toujours un système de règles définissant en grande partie le *gameplay* (voir notion de *gameplay* dans §2.3 Level Design). Dans les jeux de progression les règles sont introduites au joueur « progressivement » selon son cheminement dans le jeu. La difficulté du jeu est basée sur le degré des connaissances du joueur sur ces règles. Dans les jeux émergent le système de règles est plus élaboré et le joueur possède plus des informations pour maîtriser le jeu [Natkin04]. Les choix proposés au joueur doivent maintenir une cohérence dramatique. Les options sélectionnées par le joueur dépendent en grande partie des informations qu'il possède [Guardiola00].

Maintenant que les éléments de base d'un jeu ont été caractérisés nous pouvons introduire deux théories très utilisées initialement dans l'analyse de jeux vidéo : l'analyse cinématographique et l'analyse narrative. Nous présentons ainsi une critique sur certains des points dont l'application de ces analyses semble inadéquate et exige la définition d'une théorie et d'une méthode d'analyse propre à la nature des jeux.

### **2.1.3 De l'analyse de film à l'analyse des jeux vidéo**

Si nous considérons une approche d'analyse filmique classique [Vanoye01], les principaux éléments dans une description de film sont :

- 1) Le synopsis du film.
- 2) La segmentation du film dans les principales parties (actes), selon plusieurs critères : l'espace, le temps, les signes de ponctuation (*fades*), et la structure narrative.
- 3) La division de chaque acte du film en séquences et plans. Une séquence est une unité narrative qui conserve l'unité de temps et de lieu. Un plan est une unité de prise de vue.

4) L'analyse des séquences :

- La structure narrative de la séquence.
- L'analyse plan par plan : durée, principaux dispositifs visuels, caractéristiques de l'appareil photo, déplacements de caméra et *zooms*, transitions entre les plans consécutifs, conception sonore, rapport images-sons.

À partir de cette description semi-formelle de nombreuses méthodes d'analyse ont été développées selon divers buts et points de vue esthétiques ou historiques.

Si nous considérons la possible transposition de cette méthode vers les jeux vidéo de nombreux aspects fondamentaux doivent être adaptés :

D'abord, l'analyse de film ne peut décrire aucun aspect du *gameplay*. Par conséquent, une autre méthodologie doit être adoptée pour décrire les règles du jeu. Elle peut être basée sur la théorie des jeux par exemple [Rollins00, Natkin02]. Ainsi pour les jeux "non narratifs" (jeux d'action, jeux stratégiques, ...), seulement les deux premiers aspects de l'analyse de film peuvent être pris en considération.

Le synopsis d'un jeu ne peut pas être considérée seulement comme une description narrative. Un jeu est d'abord et essentiellement un univers imaginaire. Par conséquent, la première étape dans la spécification d'un jeu consiste à définir les aspects principaux de cet univers : Le contexte du jeu, le scénario global (topologie, carte de navigation globale, caractères principaux, nature et hiérarchie des niveaux), les caractéristiques principales du jeu, les principes du *gameplay* (modalités, buts, règles, choix stratégiques principaux, cahiers de charge image et son, principes ergonomiques : Interface, apprentissage du jeu, options de sauvegarde et de chargement...). La première étape dans la spécification du jeu devrait être aussi la première étape dans la description du jeu.

La segmentation du jeu ne peut pas être considérée uniquement par rapport à la structure narrative ; elle devrait aussi concerner les niveaux de jeu qui sont généralement basés sur une analyse topologique (voir §1.3 pour plus de explications concernant cet aspect).

La notion de séquences et de plan doit également être modifiée. Le pendant d'une séquence modifiée peut être une quête [Guardiola00], constituée d'un but, des obstacles et des solutions.

Les plans ne peuvent pas être définis seulement par la continuité du point de vue : dans de nombreux jeux le joueur peut passer d'une caméra virtuelle à une autre. Seulement le changement obligatoires de caméras doit être pris en considération, car c'est le joueur qui dirige les caméras. Par contre cette direction est limitée et d'une certaine façon dirigée par la conception du jeu. Ceci permet de limiter l'espace de perception du joueur ou d'attirer son attention sur un objet particulier. Ceci doit être traduit dans une description formelle de la scène (comme dans VRML – *Virtual Reality Modeling Language*) et des positions, déplacements, et cadrages possibles des caméras. Ceci pourrait mener à une théorie de Mise en scène pour les jeux.

Les mêmes problèmes surgissent pour la conception sonore et les rapports image-son. Par exemple, la classification du son en tant que *in*, *out* et hors champs [Chion00] ne peut pas être employée sans adaptation.

### 2.1.4 De l'analyse littéraire à l'analyse des jeux vidéo

En partant de l'analyse traditionnelle de la narration, les principaux éléments à considérer sont:

- 1) La structure
  - À partir de la théorie de la structure du discours littéraire [Todorov73], elle est décrite comme une structure composée de trois parties principales, indispensables à toute narration : Equilibre initial, conflit et résolution (retour à l'équilibre).
- 2) Les codes
  - Verbaux : utilisation du langage.
  - Symboliques : utilisation des images et sons qui en accord avec les antécédents sociaux et culturels de l'audience (lecteur, spectateur) et du contexte de l'histoire évoquent certaines images.
  - Techniques : emploi de techniques (illumination, sons, mouvements, composition, ...) pour donner une atmosphère, provoquer une sensation dans l'audience.
- 3) Les personnages
  - Définition de la personnalité, des qualités, des actions et du rôle dans la trame de chaque personnage, archétypes clairement définis dans la théorie de Propp [Propp70].
- 4) Le conflit
  - Identification des forces opposées qui créent le conflit dans l'histoire. Élément analysé à partir de la Théorie d'Opposition Binaire [Levi-Strauss58].

Bien qu'elle constitue une base théorique incontournable, l'analyse théorique narrative classique nécessite une adaptation considérable pour s'appliquer à l'ontologie de la narration des jeux de vidéo [Juul99, Guardiola00, Pearce01], même si l'on ne considère que les jeux à narration prépondérante.

Comme nous l'avons remarqué précédemment, un jeu n'est pas seulement ni principalement une histoire. Cependant, indépendamment du type du jeu et de la part de narrativité et d'interactivité que celui-ci contient, la structure en trois actes est en général présente. Même lorsque l'importance accordée à l'histoire dans un jeu est insignifiante, il existe toujours un conflit à résoudre qui se transforme en objectif pour le protagoniste (joueur). Toutefois, la structure narrative du jeu doit prendre en considération la segmentation en niveaux et sa relation avec la trame. La trame définit l'action principale de chaque niveau et l'ordre dans lequel ceux-ci se succèdent en créant ainsi la logique narrative du jeu [Guardiola00].

*« Le jeu amène une nouvelle manière de raconter une histoire et tenter d'en décortiquer les mécanismes est une aventure qui se poursuit de jour en jour. »*

[Guardiola00]

L'audience, le narrateur et la relation de communication qui existe entre eux sont aussi différents dans un jeu. L'audience, dans notre cas le joueur, n'a pas un rôle passif. Le concepteur d'un jeu ne crée pas de personnages pour qu'ils racontent une histoire, il crée une trame et dispose les éléments nécessaires pour que le joueur construise l'histoire à partir de son personnage. Les mécanismes de tension narrative normalement créés et exploités pour rendre l'histoire plus intéressante au lecteur ou spectateur, doivent être remplacés par d'autres

mécanismes dans le jeu [Juul99]. Qui est le narrateur dans un jeu? S'il n'existe pas explicitement quelqu'un qui raconte l'origine de l'histoire et qui motive le joueur, certains éléments du jeu doivent être à l'origine de ces motivations pour donner au joueur un objectif. Les interactions du joueur, outre la motivation et de l'immersion dans l'univers, doivent avoir des répercussions dans l'histoire qui est en train de se construire.

D'autre part, l'importance accordée aux personnages n'est pas la même dans tous les jeux. Dans un jeu, il peut ne pas exister des personnages (e.g. jeux abstraits comme *Pong*, *Tetris* ou les échecs).

L'utilisation adéquate du langage et des symboles non verbaux doivent aboutir à une cohérence générale dans l'univers du jeu et servir à des propos bien précis. Prenons par exemple le langage original utilisé pour les personnages dans le SIMS, il est incompréhensible au premier regard, mais sert à exprimer les émotions des personnages. Le concepteur a réussi à faire communiquer les personnages sans rentrer dans la complexité du traitement automatique de la langue et de la gestion des mécanismes intelligents qui permettent de ne pas répéter constamment les mêmes dialogues.

Les différents temps dans la narration servent à distinguer l'histoire du discours. L'histoire est l'ordre chronologique des faits, tandis que le discours est la manière dans laquelle l'histoire est reconstituée par le narrateur, c'est à dire, la séquence dans laquelle l'histoire est présentée à l'audience. Dans un jeu, ces temps cessent d'exister séparément [Juul99]. C'est au moment de l'exécution des actions du joueur que la séquence est produite et que l'histoire est ainsi reconstituée [Guardiola00], à l'exception des cinématiques, où le rôle du joueur devient passif pour introduire une partie de l'histoire ou des informations supplémentaires. Elle peut être présentée dans un autre temps (passé ou futur). Contrairement aux effets positifs des mécanismes de tension narrative que la distinction de ces temps provoque dans l'audience, dans les jeux vidéo, les cinématiques interrompent l'immersion du joueur dans le monde virtuel et l'expérience du jeu devient frustrante si le concepteur abuse de son utilisation.

Les jeux accordent plus ou moins d'importance à la structure narrative selon leur genre. Guardiola considère deux contextes comme facteurs coexistants au sein d'un même jeu : La structure narrative prépondérante et les règles prépondérantes. Dans le premier cas, le joueur commence au début d'une trame qu'il devra découvrir au fur et à mesure qu'il avance dans son jeu, en fonction de ses explorations et interactions. Même si plusieurs parcours narratifs sont disponibles, la trame reste invariable (e.g. les jeux d'aventure et de rôle). Pour les jeux aux systèmes de règles prépondérantes, l'univers est régi par des systèmes de ressources et de règles extrêmement complexes (e.g. les jeux de gestion et de stratégie). Dans ce type de jeux, le joueur a une liberté d'action plus grande qui lui permet d'atteindre ses objectifs de différentes manières. Il s'agit d'atteindre l'objectif du jeu dans un espace et un temps définis (obtenir une victoire).

Dans cette section nous avons introduit la fonction narrative dans les jeux vidéo et nous avons montré comment elle est différente de la narration traditionnelle. Les premières techniques d'analyses utilisées provenant de la théorie littéraire et cinématographique sont un premier pas vers la compréhension du nouveau média. Cependant, de telles techniques se sont montrées toutefois insuffisantes étant donné les caractéristiques interactives des jeux. Le paragraphe suivant décrit les particularités du processus d'écriture dans les jeux vidéo et explique les éléments constitutifs du nouveau récit.

## 2.2. Game Design et Level Design

### 2.2.1 Introduction à l'écriture des jeux

La conception d'un jeu vidéo est une tâche difficile car elle résulte de la combinaison d'un processus artistique et d'un processus technique. Compte tenu de la nature interactive de ces applications et de leurs buts surtout ludiques, l'auteur doit laisser une liberté de contrôle au joueur. Le joueur doit donc résoudre des problèmes non triviaux mais pas trop complexes, menant à une succession de buts dans un délai raisonnable. Il doit avoir ainsi le sentiment de liberté dans ce monde interactif, même si en réalité il est guidé vers la solution d'une façon inconsciente. Pour résoudre ce paradoxe, l'industrie des jeux a adapté plusieurs techniques dérivées de la théorie des jeux, de la littérature classique et de l'informatique.

Dans ce paragraphe nous analysons ce processus d'écriture à partir de la description des phases généralement acceptées dans l'industrie : Le *Game Design*, le *Level Design* et les notions principales comme le *gameplay*.

La compréhension de ce processus d'écriture nous semble très importante pour plusieurs raisons :

- Il met en évidence la différence entre les éléments d'un jeu vidéo et les éléments apparemment semblables d'autres médias.
- Il présente de manière générale la méthode d'écriture employée dans l'industrie.
- Il permet d'analyser l'originalité de cette nouvelle technique d'écriture basée sur des relations espace-temps.
- Il fournit les concepts permettant de construire une pensée critique sur les jeux vidéo.

La création d'un jeu commence par l'ébauche d'une idée et l'élaboration de la scénarisation. Un jeu est dans un premier temps un univers imaginaire. La première étape consiste alors en la définition des aspects principaux de cet univers [Gal02]: Époque et style, contexte du jeu, but à atteindre, principaux types d'objets impliqués et perception du jeu entre autres. Cette partie de la définition du jeu est appelée étape de conception ou *game design*.

L'étape de conception ultérieure est connue comme la conception de niveaux ou *level design*. Un niveau de jeu est constitué d'un espace virtuel, d'un ensemble de puzzles à résoudre dans cet espace, et d'un ensemble d'actions principales à exécuter par le joueur pour achever un but donné.

### 2.2.2 Game Design

Le processus de *game design* consiste à imaginer le jeu, définir son fonctionnement, décrire les éléments qui le compose et transmettre ces informations aux différents corps de métiers intervenants [Rollings03].

Nous nous sommes fortement inspirés du « document de conception »<sup>6</sup>, la méthode de conception la plus utilisée dans l'industrie [Kreimeier03a, Kreimeier03b et Rollings03], pour expliquer cette notion. Ce document permet de définir le *game design* et d'illustrer le concept du jeu : scénario, mission générale, mission par niveau, description des personnages, décors, etc. [Bates00, Rollings03]. Il devient ensuite le document de référence des membres de l'équipe de production.

Les composants principaux dans la conception d'un jeu sont :

- Le contexte du jeu : époque, style, références historiques ou mythiques.
- Le scénario global : topologie de l'espace, charte globale de navigation, personnages principaux, nature et hiérarchie des niveaux.
- Les caractéristiques principales du jeu, c'est-à-dire ce qui le rend unique. Les caractéristiques doivent être classées en tant qu'aspects fondamentaux de la conception.
- Les principes du *gameplay* : mécanismes du jeu, modalités, buts, règles, principaux choix stratégiques.
- Les caractéristiques perceptives de l'univers (images, sons et retour de force)<sup>7</sup>.
- Les principes ergonomiques : Interface, apprentissage du jeu, sauvegarde et chargement.
- Les classes d'objets dans le jeu : les constituants de l'univers, les objets d'ambiance et les outils de mise en scène [Natkin03].

## Objets

Nous comprenons la notion d'objet dans un contexte informatique. Un objet est une entité capable d'avoir un comportement qui peut être modifié selon certaines informations. L'objet est donc une structure qui conserve à la fois une description de lui-même et des opérations permettant d'accéder ou de modifier cette description. Les objets qui ont dans leur description des caractéristiques perceptives et actives similaires appartiennent à une même classe. Les caractéristiques perceptives sont définies par la représentation visuelle de l'objet. Les caractéristiques actives sont constituées de deux types d'opérations : celles produites par l'objet et celles qui ont un effet sur l'objet. Dans le jeu vidéo, « *Quelque chose a le statut d'objet parce qu'il peut produire ou subir une action qui a une influence sur le déroulement du jeu. Une porte dans un décor peut ne pas être ou être un objet du jeu. S'il est possible de l'ouvrir, elle est un objet du jeu, si elle est uniquement un élément du décor, elle n'existe pas en tant que tel, elle sera une espèce de trompe l'œil peinte sur un mur (une texture au sens des infographistes).* » [Natkin04].

D'autres considérations liées à la définition des caractéristiques perceptives des objets doivent être prises en compte. Comme l'explique Natkin, pendant l'écriture d'un jeu, les différents objets qui donnent forme et « vie » à l'univers en train d'être créé doivent être définis au début de la chaîne de conception. Contrairement aux processus traditionnels de création littéraire, ils doivent être dévoilés avant la narration, en dehors du récit.

Tout au long de la chaîne de production d'un jeu, les objets sont conçus, définis, créés et utilisés par les différents professionnels impliqués (concepteurs du jeu, infographistes,

---

<sup>6</sup> *Design Document* en anglais. Un contra exemple intéressante est proposé par Marc Cerny [Cerny02].

<sup>7</sup> Souvent références dans la littérature américaine comme le *Look & Feel*.

programmeurs, concepteurs de niveaux, ...). Ils sont ensuite employés par l'utilisateur. Ceci suppose que le mode de conception d'un objet lui permette d'être utilisé dans de multiples applications non déterminées a priori. Cependant, ils doivent être cohérents pour garantir la crédibilité et l'immersion dans le jeu. Par exemple, les caractéristiques visuelles des objets sont conçues en 3D pour pouvoir être aperçus de tous les angles, indépendamment du positionnement et de l'orientation d'une camera active en particulier.

### 2.2.3 Level Design

Un niveau peut être vu comme un bloc principal du jeu. Un jeu « miniature » qui contient en lui-même tous les éléments qui le rendent fonctionnel mais faisant partie d'une progression globale. « Le tout est nécessaire pour la compréhension de ses parties, ainsi que les parties sont nécessaires pour la compréhension du tout ».

Chaque niveau a un objectif ou une raison d'être qui rend cohérent le jeu dans sa totalité. L'objectif d'un niveau doit être l'élément central qui unifie le sujet de ce niveau [Bates00]. Avec cet objectif à l'esprit, le concepteur doit équilibrer les objets et règles régissant le niveau, justifier l'existence de tels objets et motiver le joueur. Il s'agit d'un équilibre qui n'est pas toujours facile à obtenir. Pour cela, les concepteurs doivent prendre en compte à chaque étape de la description de l'espace deux problématiques principales : le positionnement correct des objets et les règles logiques mettant ceux-ci en relation.

Ces deux problématiques sont fortement liées. Comme nous pouvons le constater dans beaucoup de jeux, une façon récurrente de contrôler le parcours du joueur [Bates00] consiste à délimiter les chemins possibles à parcourir en plaçant des objets qui obligent la convergence vers certaines zones géographiques. La notion de progression peut aussi être transmise en combinant des règles et des objets de telle sorte que le joueur puisse être retenu dans une zone le temps de résoudre la quête ou bien d'éviter qu'il puisse retourner à des endroits où l'objectif a été déjà atteint. Ces techniques d'écriture donnent lieu à d'autres avantages comme une meilleure gestion de la mémoire, de la mise à jour des données, des sauvegardes du jeu, et de l'augmentation de tension dramatique.

Les principales tâches dans la conception d'un niveau sont [Watt03]:

- Le *mapping*, c'est-à-dire la construction de la carte du niveau.
- La description détaillée et la construction de l'environnement dans lequel le niveau est joué.
- Le positionnement des objets interactifs et des ressources dans cet environnement.
- L'écriture des scripts qui précisent le scénario sous forme d'un programme utilisant les méthodes des objets. Ce programme est souvent (dans les jeux d'aventure) lié aux différents espaces géographiques.

La plupart des jeux modernes sont basés sur des images en 3D. Le concepteur de niveaux emploie des outils modélisateurs en 3D ou des éditeurs de mondes afin de créer et éditer les niveaux. De plus en plus souvent les concepteurs d'un jeu mettent à disposition du jeu des éditeurs de niveau qui permettent d'introduire une nouvelle aventure et même de modifier l'apparence des objets. Ceci donne naissance à de nouveaux jeux, les MODS (pour modification).

## 2.2.4 La notion de labyrinthe

Il est certain que l'écriture dans l'espace est un élément essentiel du Game Design et du Level Design. Toutefois, ce n'est plus uniquement avec des contraintes topologiques (comme dans les jeux 2D à scrolling horizontal), que le concepteur tente de guider le joueur dans ses déplacements.

Généralement pour la construction d'un niveau, la géométrie de l'espace est définie en premier : Une carte, une arène, un labyrinthe, un circuit de course donné, etc. Le concepteur de niveau choisit alors les emplacements et les actions liées aux objets dans ce niveau [Gal02 et Natkin04]. Ceci définit la suite des quêtes qui constituera le niveau ( voir notion de quête dans chapitre 2). Pour garder une sensation de liberté, le concepteur du niveau s'appuie sur l'existence d'un ensemble de quêtes parallèles. Dans des niveaux plus complexes, il inclut l'existence des quêtes facultatives.

Les puzzles du niveau doivent être conçus de telle manière que le joueur ne se sente pas perdu ou emprisonné dans une boucle. La difficulté des énigmes augmente graduellement. Par conséquent, les niveaux sont généralement conçus et reliés entre eux par un degré de complexité croissante.

La structure du niveau est déterminée par une hiérarchie de buts implicites ou explicites. Dans les deux cas le joueur est conduit par un scénario implicite, qui limite le nombre de quêtes pertinentes possibles.

Natkin appelle cette méthode d'écriture, la technique du labyrinthe [Natkin04] :

*« Le concepteur de niveau part, en général, d'une représentation de l'espace dans lequel se déroule le niveau. Il place un ensemble d'objets qui vont définir la suite des quêtes que doit mener le joueur pour terminer le niveau.*

*[...]*

*Le point d'entrée dans le niveau, la topologie du niveau et l'enchaînement des quêtes, dont l'ordre n'est pas totalement fixé mais qui est très contraint par la logique des énigmes, détermine un parcours qui maintient la tension dramatique et se termine en "happy end" [...]. Bien entendu le joueur ne connaît pas les bons chemins. La recherche de ce chemin et des clefs des énigmes qui le parsèment est son espace de liberté. Mais comme il doit finir le niveau dans un temps raisonnable, le concepteur lui construit un fil d'Ariane qui limite les errements. Il s'agit d'indications orales, textuelles, visuelles ou sonores qui précisent les quêtes à résoudre ou les moyens pour se procurer un objet utile pour la suite.*

*Cette aide est souvent matérialisée par la carte, accessoire indispensable que tout héros de jeu de rôle possède dans sa besace. »*

Les caractéristiques soulevées par Stéphane Natkin par rapport à cette méthode d'écriture mettent en évidence la nature interactive et la place centrale accordée à l'utilisateur des jeux :

*« Ce qui est donc très innovant dans cette écriture est le fait que le récit soit basé sur l'espace et non pas sur le temps et que la linéarisation du récit tient compte de la volonté du joueur de résoudre le niveau et donc de suivre un parcours « efficace » qui lui est suggéré. »*

Donc si le parcours d'un joueur est contraint par la topologie, ce n'est pas une contrainte absolue. Il existe d'autres mécanismes qui servent à piloter le joueur dans l'espace et dans le temps. Natkin donne trois exemples très différents :

*« Dans le jeu d'aventure Great Threat Auto, le joueur incarne une petite frappe qui doit faire carrière dans une ville tenue par différents gangs. Dans chaque niveau il obéit à un employeur (la mafia, puis les triades...) et termine le niveau en éliminant son chef ou en retournant sa veste. Chaque niveau est constitué d'une suite de quêtes (aller chercher la copine du boss, tuer un gêneur, faire sauter un bâtiment) qui est exposé par une courte cinématique. Pour accomplir sa mission le joueur doit voler des automobiles qui lui permettent de se déplacer librement dans la ville. La plus grande partie des missions n'a pas de durée limitée, donc le joueur a la possibilité de se promener et d'accomplir sa tâche tout à loisir. Mais dans la pratique il dispose d'une carte dynamique (une sorte de GSM) qui lui montre le plus court chemin vers les obstacles à franchir successivement. Chemin qu'il suit, en général.*

*Homeworld est un jeu de stratégie qui se déroule dans le futur et l'espace. L'intérêt principal du jeu réside dans la qualité des graphiques 3D temps réel. Comme dans la plus part des jeux de stratégie, le joueur doit développer une économie et une structure militaire qui lui permet de vaincre une race adverse. Dans Homeworld (2), le jeu est découpé en batailles successives d'une complexité croissante qui supposent une suite adaptée de choix d'investissements et de placement des unités combattantes. L'espace est ouvert mais (dans les parties solo), l'ordinateur (donc le concepteur du jeu), a l'initiative du placement des unités ennemies, ce qui positionne les étapes du jeu. Les jeux de stratégie sont essentiellement scénarisés en fonction d'un processus d'apprentissage, le joueur ayant une grande liberté de décision sur la gestion des ressources et des unités. Dans les premiers niveaux de Homeworld, une suite de décisions qui permet de vaincre l'adversaire est fortement suggérée par l'aide en ligne.*

*Notre dernier exemple traite d'un genre de jeu qui peut sembler, à priori, ne faire l'objet d'aucun scénario : les jeux de simulation automobile. Pourtant, au sens du Level Design, ce sont les jeux les plus contraints. La plupart de ces jeux laissent au joueur le choix de son avatar, de son équipement, du circuit, du type de course qu'il veut réaliser et des options de stratégie de courses. En fait une fois que le niveau est lancé, le joueur n'a plus qu'à suivre le circuit, le plus vite possible et en évitant les obstacles. »*

### **2.2.5 Le gameplay**

Il n'existe pas une définition claire et formelle du *gameplay*. Ce concept, au centre du processus de conception d'un jeu, reste flou. La notion est présentée dans le meilleur des cas par une série d'éléments constitutifs ou des principes simples à ne pas négliger. Par exemple, un choix doit être intéressant, le jeu doit être juste, le jouer doit pouvoir gagner, ... [Rollings00]. Chaque principe est le plus souvent du temps illustré par des exemples.

Sid Meier, réfléchit au *gameplay* comme étant « une série de choix intéressants ». Rollings et Adams complètent cette idée en la définissant comme « une ou plusieurs séries de challenges, mis en relation par hasard, dans un environnement simulé » [Rollings03]. L'explication donnée par les auteurs de cette définition est intéressante et convient à notre propos :

- Le mot série sous-entend implicitement l'existence d'une relation d'ordre quelconque des éléments qui la constituent.
- Le mot challenge fait référence aux éléments présentés au joueur pour le motiver à s'engager dans le jeu.
- Enfin, les termes « la relation par hasard » renvoie à un indéterminisme, clef de la sensation de liberté du joueur.

Nous utiliserons la définition suivante :

*« Le gameplay comprend l'objectif du jeu, les principales phases, le type de quêtes que le joueur devra réaliser et les mécanismes ludiques utilisés (révélation des objectifs partiels, obstacles, techniques de résolution). »* [Natkin04]

En ce qui concerne la construction d'un niveau, la disposition physique de la carte du jeu est fortement influencée par le type de *gameplay* [Bates00]. Bates donne deux exemples qui illustrent de façon incontournable ce principe :

1. Les niveaux d'un jeu en solo (un seul joueur, ang. *single-player*) sont linéaires. Si le monde est ouvert et permet l'exploration, comme dans la plupart des jeux d'aventure, le niveau doit être conçu pour inclure des éléments qui limitent les errements du joueur.
2. Les niveaux de combat sont circulaires (i.e. arènes). L'architecture doit rester simple et facile à naviguer. Le joueur doit être capable de repérer la carte rapidement. Il ne doit pas y avoir des territoires sécurisés pour se cacher, ...

## 2.3. Différentes approches dans l'analyse de jeux vidéo

Dans cette section nous présentons les travaux les plus représentatifs des réflexions au tour de l'analyse des jeux vidéo. Nous avons organisé ces travaux selon les cinq différents groupes de pensées critiques que nous avons identifiées : Les théories narratives, les approches sociologiques et les ludologiques, les théories fonctionnalistes qui dérivent une approche scientifique axée sur l'informatique et les formalistes qui se situent entre humanistes et scientifiques. Ce classement est flou, comme nous verrons au cours du prochain paragraphe, certains propositions se situent à la frontière de plusieurs classes.

Des travaux comme la thèse de Brenda Laurel [Laurel91] ou la notion de quête [Guardiola00] présentent des réflexions intéressantes sur la fonction narrative dans les jeux vidéo. Chacun de ces travaux introduit une analyse sur certains des éléments de la littérature classique qui peuvent être comparés à certains aspects des jeux. Les auteurs reconnaissent les limites de telles approches et prennent conscience de l'interactivité comme élément central de ces divergences. Le point de vue des ludologistes, développés par Espen Aarseth et Jesper Juul, invitent à une réflexion plus centrée sur les composants originaux d'un jeu. L'étude des jeux vidéo d'un point de vue fonctionnel est basée sur des techniques de spécification du

logiciel. Elle est en plein essor dans les communautés de concepteurs et développeurs de jeux.

Selon les éléments mis en valeur par chacune des approches, les critères pour construire une critique convergent sur plusieurs points (e.g. fonctionnalistes et sociologistes) ou s'opposent complément sur d'autres (e.g. narrativistes versus ludologistes).

### 2.3.1 Approches Narratives

#### Théâtre

Dans « *Computers as Theatre* » [Laurel91] l'auteur présente une thèse basée sur la théorie Aristotélicienne appliquée aux arts littéraires. Cette théorie définit la forme et la structure de la littérature dramatique et narrative et fournit une compréhension des éléments structurels et de la façon de les combiner pour créer des êtres "organiques complets".<sup>8</sup> À partir des éléments qualitatifs dramatiques identifiés par Aristote, et des relations causales existantes entre eux, Laurel présente un modèle servant comme outil de création, d'analyse et de validation applicable tant au théâtre qu'aux systèmes interactifs. Applications interactives que l'auteur interprète comme représentations des activités et qu'il appelle "activités homme-machine".

L'auteur considère les études aristotéliens comme la base permettant d'effectuer une analyse des activités homme-machine (jeux de vidéo inclus). Une telle étude a pour but d'offrir un cadre de travail théorique-dramatique qui peut être appliqué à la tâche de création d'interactions homme-machine.

En appliquant les causes identifiées par Aristote, Laurel les définit et les emploie comme des forces opérant de manière concourante et interactive pendant le processus de création.

Le travail de Laurel est intéressant car il est une des premières tentatives pour formaliser une théorie dramatique de la conception des jeux vidéo. D'autres auteurs, principalement dans le domaine de la narration interactive, ont continué à explorer les éléments dramatiques nécessaires qui expliquent la génération d'une narration significative dans les applications multimedia (voir dans narration interactive).

#### La notion de Quête

Guardiola identifie et définit la quête comme l'élément de base d'une structure de jeu [Guardiola00]. C'est à partir de cet élément que l'ensemble des actions du joueur sera observé. L'auteur compare la quête dans sa forme de base avec celle d'une structure linéaire dramatique et la présente de la manière suivante :

---

<sup>8</sup> Comme explique l'auteur, Aristote utilise le terme « organique » pour évoquer une analogie des êtres vivants, où un organisme est plus que la somme de ses parties, toutes les parties sont nécessaires pour la vie, et les parties ont certaines relations nécessaires entre elles.

Formule dramatique classique:  
Personnage – objectif – obstacle – réponse dramatique

Cette formule résume les éléments constitutifs d'une histoire : Un personnage a un objectif et pourra ou non surmonter des obstacles pour l'atteindre.

Dans les jeux, la transformation de cette structure en une quête implique que la réponse narrative soit positive pour permettre au joueur de progresser dans la trame du jeu. Guardiola présente la quête comme l'engagement du joueur vers l'objectif. La confrontation aux obstacles est le mécanisme narratif qui conduit à une solution.

La quête :  
Objectif – obstacle – solution

Dans le tableau suivant, l'auteur présente de manière résumée tous les éléments qu'une quête peut contenir :

	Exploration	Optionnel : le joueur recherche son prochain objectif et en prend connaissance.
<b>Epreuve</b>	<b>Objectif</b>	Engagement du joueur dans la quête, il commence à agir dans le but de l'atteindre.
	Exploration	Optionnel : le joueur a décidé d'atteindre un objectif mais n'a pas encore été confronté à l'obstacle.
	<b>Obstacle</b>	Confrontation à ou aux obstacles.
<b>Enigme</b>	Exploration	Recherche des éléments de la solution et de la bonne mécanique de résolution.
	<b>Solution</b>	Le joueur applique la bonne solution.
	Exploration	Optionnel : si le joueur n'est pas récompensé immédiatement après la résolution de l'énigme.
	Récompense	Le joueur acquiert la récompense.

**Tableau 2.1 : Éléments d'une quête selon Guardiola.**

Guardiola distingue deux éléments additionnels qui se révèlent au travers des composants de la quête : l'épreuve et l'énigme. Il définit l'épreuve comme le défi proposé au joueur, tandis que l'énigme désigne le problème à résoudre.

Pour Guardiola comme pour Laurel, la structure de trois actes dans les scénarios (texte, film, théâtre ou jeux) définit la structure de la trame narrative. Pour Guardiola, ceci impose à son tour l'ordre des niveaux du jeu. L'ordonnement des quêtes doit toujours se plier aux exigences d'une trame dramatique.

## La narration interactive

Interactive drama, cyberdrama, narration interactive, fiction interactive... autant de termes qui expriment le même objectif de recherche. Cette approche propose l'analyse de jeux d'un point de vue littéraire et détermine les formes de ce nouveau type de narration.

Nicolas Szilas présente une classification intéressante qui résume les différentes approches pour la conception du drame interactif et présente sa propre théorie narrative [Szilas03].

L'auteur classe les différentes théories narratives dans trois groupes principaux :

1. Celles donnant une importance capitale au personnage. Cette approche consiste à fournir des personnages riches en caractéristiques. Il s'agit de développer les personnages dans le monde virtuel.
2. Celles structurant le drame en macro éléments ou scènes. Les scènes peuvent être organisées par rapport à un ordre préétabli par l'auteur, dynamiquement ou bien en fonction du modèle de Propp (trois actes).
3. Celles se focalisant sur les propriétés narratives :
  - Les relations entre *actants* et les événements : obligation, désir, capacité.
  - Le calcul des actions successives pour créer le conflit dans l'histoire.
  - L'utilisation du suspense pour donner forme aux actions dans le drame.

Comme exemples représentatifs de chacun de ces groupes nous pouvons citer les travaux de Ruth Aylett sur la caractérisation de personnages [Villette04], les projets PHACADE de Michael Mateas et Andrew Stern sur le l'intégration d'agents intelligents dans la construction de la structure du drame interactif [Mateas03], le projet Realtime Drama de Joe Andrieu et Jeff Rawling sur la structuration narrative [Andrieu03] et finalement le projet IDtension de Nicolas Zsilas03 basé sur les propriétés narratives [Zsilas03].

Le travail de Ruth Aylett est focalisé vers la partie interne et émotive de personnages [Villette04]. Elle propose la création de personnages physiquement et visuellement crédibles à partir des agents intelligents dotés d'un système émotionnel interne qui produit et communique des comportements expressifs.

Le projet PHACADE et Realtime Drama sont basés sur la théorie Aristotélicienne pour la création de drame interactif (introduit dans § 2.3.1. Approches narratives). Les deux projets segmentent l'histoire en scènes ou *beats* et offrent au joueur l'opportunité de changer le cours de l'histoire. Les des deux systèmes évaluent l'expérience dramatique en fonction de la réaction du joueur. Realtime Drama qui se trouve encore dans sa phase de conception, s'intéresse aux jeux massivement multi joueurs.

### 2.3.2 Approches sociologiques

#### Les Schèmes de Frintz

Cette méthode propose l'utilisation des schémas humains, culturels et sociaux pour l'analyse des jeux. Après l'introduction des schémas et de leur fonctionnement, l'auteur explique

comment le jeu recrée les mécanismes de l'esprit humain et comment les joueurs s'y adaptent [Frintz98].

Les schémas sont des structures influençant la perception et contrôlant l'activité humaine. Déclenchés par un stimulus, des schémas créent du sens et réduisent la complexité des événements. Ils sont une aide à l'extraction et à l'abstraction de règles pour réaliser des tâches récurrentes.

Les schémas dans les jeux vidéo sont le résultat très conventionnel des habitudes de l'industrie du jeu. Pendant plus de vingt ans d'histoire des jeux vidéo, certains schémas généraux ont émergé. Nous les identifions comme des genres de jeux (action, aventure, RPG, stratégie, ...). A l'origine, ces schémas sont l'invention des concepteurs, s'ils sont réussis, ils deviennent la base pour la conception de nouvelles idées et éventuellement de nouveaux schémas. Selon l'auteur, un schéma doit accomplir deux fonctions : Utiliser les habitudes et pratiques du joueur et lui offrir un nouveau terrain d'activité pour appliquer les aptitudes et connaissances acquises précédemment.

Le profil d'un jeu est défini selon les différentes combinaisons des schémas. Frintz classe les schémas dans une hiérarchie composée de quatre couches qui regroupe chacune les différents aspects du jeu (schémas généraux, particulières au genre, liés aux règles des actions du joueur, et en relation aux règles des objets de l'univers).

L'application des résultats d'une telle analyse est une aide à l'amélioration de l'expérience du jeu perçue par le joueur et à l'identification des modèles de base réussis afin de concevoir des jeux plus performants.

### **Modèles socio comportementaux**

De manière similaire Stefan Grünvogel et Craig Lindley considèrent l'intégration des comportements sociaux pour l'analyse des jeux [Villette04]. Grünvogel identifie et décrit plusieurs schémas sociaux que nous pouvons trouver dans les différents genres de jeux (conflit, combat, puissance, ...). Il propose de se servir de tels schémas pour évoquer des comportements sociaux (règles implicites) dans le joueur pour analyser les jeux en déterminant les types de mécanismes qui les font fonctionner et en identifiant leurs possibles relations existantes. Enfin, il prend en compte la possibilité de formaliser ces différents schémas pour décrire une méthode basée sur les comportements sociaux.

Lindley définit le *gameplay* à partir de l'utilisation de mécanismes cognitifs et physiques déjà connus du joueur [Lindley02]. L'analyse des schémas d'interaction perceptifs, cognitifs et moteurs qui peuvent être utilisés dans un jeu permet de décrire, selon l'auteur, les éléments du *gameplay* d'une manière plus précise. Un schéma particulier est appelé *gameplay* gestalt. Par exemple tous les jeux de batailles mettant en jeu des unités ayant diverses capacités mettent en œuvre un schéma de décision intransitif dont l'exemple le plus simple est le jeu papier, ciseau, caillou. Un *gameplay* gestalt peut être mis en œuvre sous de nombreuses formes, selon les différents styles de jeu et même dans un jeu donné en utilisant des mécanismes variés. Le joueur apprend le mécanisme qui instancie le *gameplay* gestalt pour pouvoir après progresser dans le jeu.

### 2.3.3 Approches Ludologiques

#### L'analyse esthétique, poétique et d'écriture dans l'espace d'Espen Aarseth

L'analyse de jeux proposée par Aarseth est basée sur les aspects esthétiques, poétiques et spatiales des jeux de vidéo. Cette approche comprend trois catégories de base :

- 1) Action, notion qui présente la rhétorique et la structure d'événements définissant un jeu vidéo.
- 2) Temps, notion qui définit la représentation et la mise en oeuvre de la temporalité dans un jeu.
- 3) Espace, notion qui définit la représentation spatiale du monde virtuel recréé à partir de trois aspects essentiels : modélisation du personnage et du reste des éléments du monde, le type de représentation spatiale et l'influence du joueur dans l'univers.

Selon l'auteur la représentation spatiale est l'aspect central et le motif d'être d'un jeu. Cet élément est plus important que le temps (qui peut s'arrêter), que les actions, événements et objectifs (qui peuvent être ennuyeux et répétitifs) et sans aucun doute, plus que la caractérisation qui peut couramment ne pas exister). Cette dernière notion constitue par elle-même une analyse de jeux basée sur la perspective spatiale dénommée par l'auteur "allégories de l'espace" [Aarseth00].

Aarseth étudie deux conceptions d'espace différentes basées sur les analyses de Henri Lefebvre et d'Anita Leirfall<sup>9</sup> pour ensuite construire et présenter sa propre définition.

La perspective d'Aarseth reprend et intègre soigneusement deux aspects de la théorie spatiale de Lefebvre : Les jeux vidéo sont à la fois une représentation de l'espace (un système formel de relations) et des espaces représentables (images symboliques avec des buts principalement esthétiques).

Aarseth reprend l'explication de Leirfall sur le phénomène du cyberspace postulant que les espaces ne sont pas réels, mais des objets et des lieux qui doivent être observés comme un système de signes, raison pour laquelle nous pouvons les apercevoir.

À partir de ces théories, Aarseth propose la représentation spatiale des jeux vidéo comme une opération réductrice, conduisant à une représentation de l'espace qui n'est pas en elle-même spatiale mais symbolique et basée sur des règles. Ces réductions représentent le moyen pour créer le *gameplay*.

Pour nous, la représentation de l'espace d'Aarseth équivaut à la description topologique, tandis que les espaces représentables constituent la description sémantique de l'univers du jeu.

L'auteur distingue trois aspects essentiels dans l'analyse de la spatialisation des jeux :

- La distinction entre la représentation de l'univers et celle du joueur.

---

<sup>9</sup> Leirfall, Anita 1997. "Space, place and dimensionality," paper présenté à la conférence The digital challenge: New information technology, media and communication, Université de Trondheim, département d'Arts et Media.

- La distinction entre les modes de représentation spatiale : terrains ouverts ou labyrinthes ("outdoors" et "indoor").
- La distinction des niveaux d'influence du joueur sur l'univers.

En ce qui concerne les deux derniers aspects, Aarseth considère le terrain plus qu'une simple carte, comme un ordonnancement. Ceci s'applique autant aux espaces intérieurs qu'aux extérieurs. À travers l'utilisation d'obstacles dans le terrain (e.g. falaises, rivières,...) le concepteur détermine les mouvements disponibles, limite l'influence du joueur sur l'univers et construit une topologie "allégorique" représentant de façon figurative le monde réel qu'il prétend simuler.

La notion d'allégories de l'espace est en rapport avec nos concepts de topologie dynamique et cohérence avec l'ordre d'actions. L'influence du joueur est déterminée par les mécanismes de *gameplay* et le système de règles que le concepteur intègre pour créer une tension narrative. Il place donc les obstacles et difficultés dans le plan de façon à ce que le joueur, dans son parcours, rencontre des obstacles et vive des événements selon une trame qui gère la tension narrative (en contrôlant les espaces accessibles et inaccessibles).

Deux observations intéressantes de l'auteur concernent la représentation spatiale :

- L'univers 3D est trop détaillé pour représenter la symbolique conceptuelle de l'univers du jeu, c'est pourquoi il est encore nécessaire de retourner à une représentation planaire lors de la phase de conception.
- Les dernières innovations du domaine ne concernent pas ni le *gameplay*, ni la structure dramatique ni les thèmes des jeux, mais la représentation spatiale.

Selon Aarseth, même l'évolution d'autres aspects d'un jeu (e.g. moteurs physiques) résultent de l'accroissement de la complexité de l'espace du jeu, Ceci doit changer si nous voulons voir apparaître des jeux innovants qui enrichissent la culture des jeux vidéo [Jenkins02, Rolling03].

### **La thèse de Jesper Juul**

Dans ce travail, l'auteur adopte un point de vue analytique et comparatif. Basé sur les principes de la théorie littéraire, Juul [Juul99]détermine les éléments de différence entre les jeux vidéo et la littérature, pour ensuite expliquer ces différences à travers son analyse.

L'auteur analyse la narration dans les jeux à partir de l'adaptation des moyens d'expression classiques (oraux, littéraire, cinématographiques).

Juul observe dans son analyse plusieurs relations intéressantes entre les éléments de la narration :

- *Temps, jeu et narration*, en identifiant les différents temps utilisés dans la littérature pour distancier le narrateur et l'audience (le passé utilisé traditionnellement, le présent pour le roman moderne, et le futur pour la fiction ou les prophéties), l'auteur déclare l'incapacité de créer les mêmes mécanismes de tension pour les jeux étant donné la non-existence du narrateur et la fusion de tous les temps dans le présent imposé par l'interactivité.

- *Séquence*, la séquence dans un jeu n'est pas focalisée dans la relation histoire-discours comme dans la narration traditionnelle. La nature interactive du jeu est constituée des fonctions qui contrôlent et combinent l'accès du joueur au contenu (images, sons, textes, etc.)
- *La narration* du jeu se focalise dans les aspects temporels et l'ordre pour expliquer ce que le joueur doit faire. La narration donne une signification aux actions cependant, comme elle n'est pas un aspect essentiel du jeu, elle peut être remplacée facilement (e.g. remplacer l'apparence des objets visuels d'un jeu par d'autres en utilisant les mêmes règles mais dans un contextes différents).
- *Joueur et jeu*, cette relation est observée à partir de plusieurs éléments: L'empathie du joueur avec son personnage, la motivation pour jouer, les connaissances et la dextérités acquises par le joueur à travers les différents tentatives infructueuses d'une même quête et le fait de jouer plusieurs fois le même jeu. L'auteur signale les particularités des jeux permettant l'immersion du joueur même s'il n'existe pas un personnage auquel s'identifier, l'avatar étant alors un symbole.

L'auteur propose l'utilisation de ce cadre théorique comme méthode d'analyse pour les jeux vidéo. L'application de cette méthode est présentée à travers l'analyse de cinq jeux appartenant aux différents genres (*Myst* comme jeu d'aventure, *Doom* et *Unreal* comme jeux d'action FPS – *First Person Shooter*–, *Witness* et *LastExpress* comme des jeux plutôt orientés vers la narration interactive ).

La thèse conclut que sur les différences fondamentales entre la narration classique et la structure des jeux vidéo et l'attente tout aussi différente du lecteur/spectateur et du joueur.

### **L'Approche centrée dans le jeu de Celia Pearce**

Pour l'auteur un point fondamental pour distinguer les jeux de vidéo des autres médias est la définition de son centre d'intérêt : Les jeux vidéo sont axés sur le jeu et l'expérience du joueur et non sur l'histoire comme dans la littérature ou le cinéma [Pearce01].

Pearce explique comment la narration fonctionne à un niveau différent dans les jeux par rapport aux autres médias. Un jeu vidéo est le plus simplement décrit par une structure permettant de jouer. Dans la plupart des cas, cette structure est constituée par un type de but, des obstacles empêchant d'atteindre ce but, des ressources pour aider le joueur à réaliser le but, aussi bien que des conséquences (sous forme de pénalités et de récompenses) pouvant se traduire en obstacles et ressources. De manière élémentaire, ces éléments constituent une structure narrative générique décomposée dans un ensemble de tension suivies d'issues. Finalement cette structure sert à recréer à partir de l'expérience du joueur la courbe émotionnelle de la dramaturgie.

Par ailleurs, le récit dans un jeu est par définition non complètement déterminé, afin de laisser des choix au joueur. Dans les jeux persistants, ce non-déterminisme peut s'étendre à une forme d'incomplétude : le joueur doit personnaliser son avatar, peut utiliser objets, lieux et situations de façon non prévue par le concepteur. L'idée centrale de la thèse de Celia Pearce est de faire reposer sur « l'incomplétude » des personnages la création d'une histoire et à la qualité de l'expérience du jeu. Par exemple dans les jeux de stratégie comme *Age of Empires* et *Civilization*, l'auteur explique comment une structure asymétrique (joueurs ayant des ressources différentes ) permet d'augmenter le drame et le potentiel du récit émergeant,

en personnalisant la position de chaque joueur. L'exemple le plus significatif est la possibilité de créer une histoire émergente à partir du développement des personnalités des avatars en particulier dans les jeux de rôle massivement multijoueurs (MMORG).

Comme nous avons constaté, le point de vue des ludologistes est axé principalement sur le jeu, compris comme un ensemble de règles et une interface. Ceci s'oppose à la position de la narration interactive qui structure le jeu selon l'exposition d'une histoire ou d'un univers.

### 2.3.4 Approches formalistes

#### La méthode d'analyse de jeux de Lars Konzack

L'auteur décrit une méthode d'analyse des jeux vidéo basée sur une description en sept couches hiérarchiques: Matérielle, code, fonctionnelle, *gameplay*, signification, référentielle, et contexte socioculturel [Konzack02]. Il reprend en cela les principes de spécifications utilisés dans de nombreux domaines de l'informatique. Selon Konzack, une phase préliminaire contenant la description du jeu semble nécessaire pour garantir une vue d'ensemble avant d'aborder les composants traités de manière particulière dans chaque couche.

Chacune de ces couches peut être traitée individuellement, mais l'auteur signale que pour une analyse complète du jeu il faut comprendre chaque angle. Pour cette raison, il propose de prendre en compte tant les perspectives techniques que les aspects esthétiques et socioculturels.

Tout d'abord, l'auteur distingue deux perspectives : l'espace virtuel et « la cour du jeu » (*playground*), qui se placent à deux niveaux différents de description du jeu. Dans l'espace virtuel l'attention porte sur l'esthétique et l'imaginaire tandis que dans *la cour du jeu* l'observation se focalise sur les conditions et mécanismes physiques d'interaction. Ces deux points de vue doivent être pris en compte dans chaque niveau de l'analyse mais avec une importance relative variable.

Il décrit ensuite les sept couches :

- Couche matérielle : Contient la description technique du matériel et du type d'équipement pour déterminer la nature physique de la cour du jeu.
- Couche code : Comprend le code du programme du jeu. Ces informations, n'étant pas la plupart de temps accessibles, peuvent être partiellement trouvées à partir de la couche fonctionnelle.
- Couche fonctionnelle : Cette partie de l'analyse comprend le comportement et les réactions de l'interface par rapport aux interactions du joueur.
- Couche *gameplay* : Ce niveau traite la structure du jeu d'un point de vue ludologique (positions, ressources, espace, temps, but, obstacles, connaissances, récompenses ou pénalités).
- Couche signification : Comprend la sémantique du jeu déduite des signes ornementaux et du récit (images, sons ou texte utilisés).

- Couche référentielle : Met en relation certaines caractéristiques du jeu par rapport au genre ou aux autres médias. Ceci facilite la relation du joueur avec l'espace virtuel à travers la reconnaissance de certains signes (concept similaire à la notion de schémas de Frintz).
- Couche contexte socioculturel : Prends en compte les différents types d'interaction : joueur-ordinateur ou entre joueurs.

Tous les niveaux sont nécessaires pour analyser exhaustivement un jeu et arriver à comprendre leur complexité. La méthode peut être appliquée en utilisant seulement une ou plusieurs couches nécessaires à la compréhension d'un jeu selon un point de vue donné.

L'objectif de l'auteur de recherche d'une méthode exhaustive d'analyse entraîne un manque d'homogénéité et même de cohérence dans la description des techniques d'analyse. Quelques concepts sont définis de manière détaillées (e.g. couches signification, *gameplay* et référentielle) tandis que d'autres sont présentés de manière générale (e.g. couches matérielle et code). Certes qu'idéalement plusieurs aspects, niveaux ou concepts doivent être employés pour produire une analyse juste et appropriée, sa vaste diversité et généralité mettent en danger l'efficacité de l'analyse.

### **L'approche MDA (mécanique, dynamique et esthétique )**

Hunicke, Le Blanc et Zubek présentent un cadre de travail formel pour une conception cohérente et systématique des jeux vidéo [Hunicke04]. L'approche s'adresse à un public vaste (concepteurs, développeurs, chercheurs) des jeux et tente de concilier les différents niveaux d'abstraction de chacun de ces acteurs. Il s'agit de décomposer, de comprendre et de créer de la cohérence à partir des liens entre les différents niveaux d'abstraction : du système et du code jusqu'au contenu et à l'expérience du jeu. Les auteurs postulent que le contenu d'un jeu est essentiellement comportemental plus que médiatique. Ce point de vue permet de penser les jeux comme des systèmes construisant du sens via l'interaction.

Le modèle est constitué de trois éléments :

- Les mécaniques décrivent les règles du jeu et leurs descriptions algorithmiques.
- Les dynamiques qui décrivent les comportements liés aux interactions avec le joueur.
- L'esthétique définit la réponse émotionnelle que le joueur éprouve. Pour la caractériser, l'auteur suggère une taxonomie à huit composants de base écrivant l'expérience du joueur (sensation, fantaisie, narration, défi, enjeu social, découverte auto-expression et soumission).

Ce dernier élément décrit différents types de *gameplay* comme des combinaisons, selon des proportions variables des composants esthétiques. Une dynamique révèle une expérience esthétique et ses comportements sont des conséquences directes du système de règles.

Selon l'auteur, le concepteur et le joueur ont des vues opposées du modèle. Pour le concepteur les mécaniques donnent lieu aux comportements du système dynamique qui par conséquent donneront des expériences esthétiques. En revanche, pour le joueur c'est l'esthétique qui est permet de comprendre la dynamique et éventuellement des mécaniques.

Les auteurs suggèrent d'utiliser différentes méthodes dans chaque composant : UML pour le modèle mécanique, machines à états finis aléatoires et systèmes de pré-conditions pour le modèle dynamique et la taxonomie à huit composants pour le modèle esthétique.

Nous retirons deux concepts de cette approche :

- La caractérisation de la notion de jouabilité (*gameplay*) par les éléments qui la composent.
- La proposition d'un modèle transverse qui met en relation les différents intervenants (corps de métier et joueur) de la chaîne de production jusqu'aux consommateurs d'un jeu.

### 2.3.5 Approches fonctionnelles

#### Les techniques de génie logiciel

Après avoir utilisé très souvent des processus artisanaux pour développer le code, les industriels du jeu s'intéressent de plus en plus au génie logiciel, donnant lieu à de nombreuses références (serie Gems, articles gamasutra et game developer par exemple).

Il est important de noter deux points : d'une part un jeu n'est pas qu'un programme, d'autre part, comme la plus grande partie des interfaces, son cycle de développement ne suit pas un processus en V [Print02], mais un processus en spirale [Durard97] :

*« il est impossible de mesurer la qualité d'un jeu et l'efficacité de ses règles sans l'essayer. Une analyse papier, aussi fouillée qu'elle soit, ne permettra pas de détecter une faiblesse de l'interface ou un mauvais réglage des paramètres des règles du jeu qui le rend trop simple ou difficile. La qualité de l'image et du son ne peut être appréciée que par la pratique. »* [Natkin04].

Dans le prochain paragraphe nous présentons quelques propositions de méthodes semiformelles basées sur l'utilisation d'UML (*Unified Modeling Language*), *reverse-engineering* et *design patterns*. Nous incluons aussi d'autres modèles formels employés en informatique que nous considérons comme candidats possibles pour l'analyse ou la spécification de certains aspects dans les jeux vidéo comme les Réseaux de Pétri et les Hypergraphes.

#### *Utilisation d'UML dans le processus d'analyse et de conception des jeux*

UML (*Unified Modelling Language*) est un langage de modélisation normalisé par l'OMG (*Object Management Group*) issue de plusieurs méthodes d'analyse et de conception orientées objet, principalement celles de Booch, Rumbaugh et Jacobson [Booch99]. Il est constitué de 12 diagrammes, groupés dans trois catégories [OMG04]:

- Les diagrammes structurels (diagrammes de classes, d'objets, de composants et de déploiement).

- Les diagrammes comportementaux (diagrammes de cas d'utilisation, de séquences, d'activités, de collaborations et d'états-transitions).
- Les diagrammes de gestion du modèle (packages, sous systèmes et modèles).

Grâce au fait d'être un langage indépendant des méthodologies, UML est devenu la référence dans l'industrie du logiciel et plusieurs outils sont disponibles pour son utilisation. Selon le type de développement, de méthode de travail, de composition des équipes, ... un processus particulier pour modéliser avec UML sera plus approprié aux besoins du projet. Nous présentons par la suite trois méthodes de modélisation avec UML orientées vers la conception de jeux vidéo [Demanchy02, Bethke03 et Siang04].

La méthode de [Bethke03] mentionne l'utilisation des techniques de génie logiciel pour trouver l'équilibre entre les objectifs commerciaux et le concept du jeu, premier pas vers la réconciliation des deux contextes. L'auteur propose une méthode de développement pour les jeux vidéo en utilisant le langage UML, particulièrement les cas d'utilisation (*Use Case*). Il introduit leur application de manière simple à travers deux exemples : l'analyse des cas d'utilisation des jeux *Diablo* [Blizzard01] et *Grand Turismo* [Polyphony98].

La procédure présentée par Bethke pour définir les cas d'utilisation consiste en trois phases :

- 1) Définir les activités « clé » du jeu, en décrivant toutes les activités principales que le joueur effectue dans le jeu. Chaque activité principale est composée de plusieurs actions unitaires. Chaque action est représentée par un cas d'utilisation.
- 2) Grouper ces activités selon leurs similitudes fonctionnelles.
- 3) Esquisser le diagramme de cas d'utilisation complet en construisant d'abord un cas générique pour ensuite décrire les particularités des différentes utilisations.

L'auteur propose de créer des diagrammes pour chaque groupe d'activité. Ces cas d'utilisation seront la base d'un processus récursif d'amélioration des étapes de conception et de développement du jeu.

Cependant, d'autres approches basées aussi sur l'utilisation d'UML proposent une méthode plus complète :

Siang et Radha [Siang04] reconnaissent comme [Pearce02] le rôle capital du *gameplay* dans la conception interactive, cependant leur point de vue est plus orienté vers une conception technique. Ils étudient l'utilisation d'UML (diagrammes de cas d'utilisation et de classes) pour la modélisation de la conception interactive à partir des techniques de retroconception (*reverse-engineering*).

La méthode proposée par les auteurs est illustrée par l'analyse des jeux *Pacman* et *Spaceship shooter*. Elle consiste en 4 phases :

- 1) Décomposition du jeu en avatars et jetons (comme proposé par [Rollings00]). Les jetons représentent les objets interactifs du jeu à l'exception des avatars.
- 2) Définition des activités que le joueur réalise pour construire les diagrammes de cas d'utilisation (même processus que [Bethke03]). Ces cas d'utilisation modélisent les interactions joueur-objet uniquement.
- 3) Identification des interactions objet-objet (comportement entre jetons).

- 4) Construction des diagrammes de classe à partir de la compilation des informations des interactions. Les méthodes de classes décrivent les comportements joueur-objet et objet-objet.

La méthode présentée par [Demanchy02] inclut les phases de [Bethke03] dans la définition des cas d'utilisation plus la création des diagrammes de séquences et de collaboration. Dans un diagramme de séquences, chaque séquence représente le déroulement dans le temps d'un scénario possible du cas d'utilisation. Dans un diagramme de collaboration le déroulement d'un scénario est décrit en fonction des relations d'appels entre objets. Ainsi les deux aspects dynamiques (comportements) et statiques (structure) du scénario sont considérés. Les cas d'utilisation servent aussi à définir les diagrammes de classes et la hiérarchie des objets.

Selon l'auteur, d'autres diagrammes d'UML peuvent être appliqués selon le type de programmation choisi (programmation des niveaux par scripts (scripting) ou programmation en dur - *hard coding*). Il donne l'exemple du diagramme d'états-transitions, adaptable aux comportements et à l'intelligence artificielle quand ils sont implémentés dans un langage de script orienté objet comme Python.

Un point commun entre ces approches est la reconnaissance des problèmes de communication suscités entre concepteurs et développeurs lors de la création d'un jeu. Le fait d'utiliser un langage commun aux deux équipes permettant du travail collaboratif est une solution logique. L'utilisation d'UML est très éloignée de la culture actuelle des Game Designer.

### ***Les diagrammes de navigation et les Réseaux de Petri dans la conception multimédia***

Castellani et Jiang [Castellani01] proposent l'analyse et la conception des scénarios multimédia à partir des diagrammes de navigation et des réseaux de Petri. Un scénario multimédia est une orchestration dans le temps et l'espace de plusieurs médias présentée aux utilisateurs finaux. Un scénario interactif d'une application multimédia permet aux utilisateurs finaux d'interagir dans cette orchestration. Les auteurs proposent un moyen de construire un réseau de Pétri qui décrit un scénario selon une méthode qui repose sur les liens d'appels entre objets. La méthode est fortement orientée vers l'analyse hyper textuelle. L'analyse des scénarios s'appuie sur la création des diagrammes de navigation. Il s'agit d'un graphe dont les sommets sont les objets interagissant avec les utilisateurs et les arcs sont les relations d'activation entre objets, formalisant les liens d'un hypermédia. Les arcs précisent également des contraintes d'activation entre les différents objets (synchronisation, partage de ressources, ...).

Dans un premier temps, le diagramme est décomposé en composants indépendants, c'est à dire n'ayant pas de liens de navigation communs. Ces composants sont des graphes biparties dont les sommets sont soit des nœuds destination soit de nœuds sources, tels que:

- Tous les objets du sous-ensemble source peuvent activer tous les objets du sous-ensemble destination.
- Tous les objets du sous-ensemble destination peuvent être activés par tous les objets du sous-ensemble source.

Parmi toutes les décompositions possibles, une décomposition en composant maximaux (ayant le maximum de liens) est retenue. Chaque composant peut être traduit en un réseau de Pétri qui modélise la séquence possible des activations. Finalement, ces réseaux sont fusionnés en un seul qui tient compte des relations de navigation et des contraintes des composants de navigation.

Cette méthode suppose connue tous les objets d'un système multimédia et leurs relations d'appel. Elle permet de déduire d'un hyper texte ou d'un schéma de classe un modèle dynamique de façon partiellement automatisable. Elle suppose une connaissance de la structure interne du système et peut s'avérer impraticable pour des systèmes trop complexes et en particulier ceux qui ne sont pas uniquement des hypermédia.

### ***Machine à états- finis***

Rollings et Morris [Rollings00] proposent un modèle pour la spécification du logiciel d'un jeu. Ils partent d'une identification des objets du jeu appelés jetons. Les interactions entre jetons peuvent être décrites par une matrice qui fournit le plus haut niveau de spécification. Le comportement de chaque objet est ensuite décrit par un automate dont le langage d'entrée est constitué par les interactions avec les autres jetons, sauf pour les avatars qui dépendent également des actions du joueur. Les jeux *Pong* et *Pacman* sont utilisés comme exemples pour illustrer ce processus, appelés "tokenization". Utilisées de manière appropriée, ces techniques peuvent servir à la documentation et à la conception de l'architecture d'un jeu.

Il s'agit ici d'une présentation d'une méthode assez classique de spécification objet adaptée à la conception du code d'un jeu. Elle se situe en aval du Game Design et ne concerne que des aspects de programmation.

### **Les systèmes basés sur des règles**

#### **L'Analyse semi-automatique du *gameplay* de Southey et Holte [Southey04].**

Les auteurs présentent une vue d'ensemble sur la problématique de la gestion du *gameplay*, plus particulièrement, sur la partie dédiée à l'analyse de cette tâche. En proposant une méthode basée sur l'apprentissage actif (technique de l'Intelligence Artificielle), les auteurs ont réalisé la mise en oeuvre d'un outil logiciel permettant d'automatiser partiellement l'analyse comportementale du jeu afin d'augmenter la productivité des concepteurs.

Les auteurs proposent de déterminer un certain nombre de critères de jouabilité : la probabilité de gagner, la variance dans les résultats, la localisation des « espaces doux » à l'origine des quêtes trop faciles, et à contrario, la détection des « espaces durs » où les quêtes deviennent insolubles. Cette idée est développée dans un cadre très spécifique : un jeu de football et la recherche des positions de penalty idéales.

Le principe développe une technique basée sur l'évaluation des règles du *gameplay*. Un premier ensemble de règles est initialement fourni. Elles sont évaluées expérimentalement en utilisant les critères de jouabilité. En fonction des résultats, la machine propose de



## 2.4. Conclusion

Comme nous l'avons constaté, les méthodes d'analyse et de spécification dans les jeux vidéo sont très récentes. Celles-ci commencent à apparaître comme des outils pour la compréhension du processus de création d'un jeu vidéo partant de techniques issues d'autres domaines de recherche (e.g. Laurel91).

À travers la présentation des différentes analyses proposées par les observateurs du domaine des jeux, nous avons constaté l'intérêt croissant de chercheurs pour théoriser leur étude. Comme nous l'avons vu, la simple adaptation de techniques existantes n'est pas suffisante.

Nous retiendrons de cette étude les points suivants :

La nature d'un jeu ne peut être limitée à un scénario. En particulier la nature du *gameplay* est un élément essentiel.

Un jeu ne peut être considéré du même point de vue par le joueur (l'analyste) et le concepteur.

Les facteurs perceptifs et immersifs d'un jeu relèvent difficilement d'une approche formelle, au sens des modèles informatique. Tous les modèles effectifs que nous avons considérés partent soit d'un point de vue perceptif, soit d'un point de vue narratif, soit d'un point de vue sociologique. On ne peut imaginer et spécifier un bon jeu selon une approche purement formelle : ses principales qualités ne se révèlent qu'expérimentalement.

Plusieurs approche de types automates ou systèmes d'état transition ont été proposées. Elles se placent essentiellement dans une optique de description des interactions homme machine. Par contre les approches de scénarisation non linéaire, comme celle de Guardiola et le fait qu'un jeu s'écrive autant dans l'espace que dans le temps est peu exploité.

C'est cette voie qui est étudiée dans cette thèse. Elle fonde son modèle d'analyse sur les principes utilisés par les professionnels et permet d'envisager une méthode de synthèse et de validation partielle. Mais, poursuivant dans ce sens nous avons conscience que nous ne traitons ni les aspects classiques de mise en scène, ni les aspects essentiels qui caractérisent le plaisir de jouer.



# CHAPITRE 3

## *Modélisation avec les réseaux de Pétri*

---

### Contenu du Chapitre

Dans ce chapitre, nous présentons un modèle permettant une description formelle des propriétés de séquençement dans un jeu. Nous partons d'une description sous forme d'un langage permettant d'exprimer les relations d'ordre entre les actions élémentaires du joueur. Nous interprétons ensuite la syntaxe de ce langage en termes de réseaux de Pétri place-transition. Cette démarche présente trois avantages. Elle permet tout d'abord de définir une sémantique d'exécution du jeu qui est celle du réseau de Pétri. D'autre part, en utilisant les propriétés de dépliage de ces réseaux, nous montrons la généralité de notre modèle. Enfin, nous sommes à même d'utiliser les propriétés mathématiques des réseaux de Pétri pour formaliser certains concepts liés aux quêtes et aux niveaux d'un jeu. Nous établissons par ailleurs quelques propriétés qui permettent de vérifier la cohérence scénaristique.

### Organisation du chapitre :

Section 3.1. Syntaxe des réseaux de transactions – p. 44 : Introduction aux concepts de base de la structure d'un jeu.

Section 3.2. Introduction aux réseaux de Pétri – p. 49 : Définitions et propriétés générales des réseaux de Pétri et des réseaux construits par composition de transactions.

Section 3.3. Interprétation des réseaux de transactions en termes de réseaux de Pétri – p. 52 : Introduction du modèle de base de notre méthode d'analyse.

Section 3.4. Réseau de Pétri associé à un Réseau de Transactions – p. 54 : Introduction des opérateurs d'ordre et des relations logiques pour la construction d'un réseau de transactions.

Section 3.5. Quêtes – p. 73 : Introduction du modèle de quête qui constitue, du point de vue de la sémantique du jeu, la brique de base du *gameplay*.

Section 3.6. Niveaux – p. 74 : Présentation du modèle de niveaux, l'élément de décomposition le plus haut dans notre méthode d'analyse.

Section 3.7. Exemple – p. 75 : Illustration de l'application de la méthode d'analyse à partir de l'analyse du premier niveau du jeu *Myst*.

Section 3.8. Conclusion – p. 78 : Résumé des aspects du jeu considérés dans l'analyse et des limitations de notre méthode.

## 3.1. Syntaxe des réseaux de transactions

### 3.1.1. Retour sur les besoins d'expression

Depuis leur apparition, les jeux vidéo ont progressivement évolué d'interfaces textuelles ou graphiquement rudimentaires vers des mondes virtuels complexes en 3D et en temps réel. Ils ne sont plus seulement un divertissement destiné à une population spécifique mais s'adressent aujourd'hui au grand public. Son évolution inclut des aspects techniques, sociaux et culturels qui commencent à intéresser les communautés de chercheurs et d'enseignants. Dans l'étude des jeux vidéo, l'analyse des caractéristiques et la comparaison des structures fondamentales qui les constituent sont des aspects essentiels pour leur compréhension. C'est dans cette perspective que notre travail offre une contribution formelle en fournissant une méthode d'analyse pour l'étude critique des jeux.

Dans ce chapitre, nous présentons un modèle basé sur l'étude de la structure narrative du jeu. La méthode d'analyse proposée est limitée à un certain type de jeu dont les caractéristiques essentielles sont :

- La place essentielle de l'histoire dans le jeu.
- L'atteinte d'un objectif prédéterminé.
- Les actions exécutées sont des choix ou des processus analytiques réalisés par le joueur.

Comme le montrent différents auteurs [Adams03, Bates00], le jeu d'aventure contient de manière plus évidente ce type de structure, bien qu'elle ne se limite pas uniquement à ce genre de jeu.

Les modèles proposés présentent la décomposition de la logique du jeu dans cette structure narrative. Celle-ci est constituée par des séries d'actions possibles à effectuer par le joueur pendant le jeu.

Dans un premier temps, nous présentons les concepts de base et ceux liés aux Réseaux de Pétri nécessaires à la compréhension du modèle proposé. Ensuite, nous abordons les différents modèles utilisés pour la construction de notre méthode d'analyse. Ces modèles sont organisés selon une approche ascendante « *bottom-up* » de la méthode. Nous commençons par l'explication du fonctionnement du modèle de base : le modèle de transaction. Nous discutons ensuite des opérateurs employés pour construire le modèle de réseau de transactions. Nous aboutissons à l'assemblage du modèle de quête, qui permet à son tour la constitution du modèle de niveau. Nous accordons un intérêt particulier au modèle de quête dans l'analyse de la structure du jeu. Enfin, des cas pratiques sont présentés pour illustrer la méthode.

### 3.1.2. Segmentation d'un jeu vidéo

D'un point de vue macroscopique, les principaux aspects dans la structure d'un jeu sont :

- La carte de l'univers.
- Les niveaux du jeu.

## Niveaux

Un niveau peut être défini comme la partie linéaire principale de la structure du jeu. Chaque niveau est associé à un objectif [Bates00, Guardiola00]. L'ordonnement possible des niveaux est déterminé par la structure logique du puzzle du jeu. Chaque niveau a un objectif, soit narratif, soit perceptif, qui justifie son existence et rend le jeu cohérent [Bates00]. De plus, nous remarquons que, de façon générale, les niveaux sont associés à une région de la carte.

Nous pouvons permettre au joueur d'aller "visiter" un niveau particulier avant qu'il n'ait fini les niveaux précédents. Mais il découvrira rapidement qu'il ne peut pas progresser dans celui-ci. Soit les buts du niveau ne seront pas explicités, soit son avatar devra gagner de nouveaux attributs et trouver des objets seulement disponibles dans les niveaux précédents afin de pouvoir surmonter des obstacles.

Pour illustrer ceci, prenons comme exemple le jeu *Myst* [Myst94]. Dans ce jeu, le joueur doit visiter plusieurs îles pour trouver et récupérer certaines feuilles appartenant à un livre magique et ainsi compléter le volume. Si le joueur n'apporte pas les feuilles de l'île correspondante, le niveau est considéré comme inachevé. Toutes les îles peuvent être visitées dans n'importe quel ordre. Le jeu a une topologie en étoile, dont l'île *Myst*, située au centre, sert de portail pour visiter les autres îles.

## Quêtes

Un niveau peut être divisé en quêtes<sup>1</sup>. La structure du jeu se compose de la description des quêtes et de leurs relations [Guardiola00]. Elle décrit une hiérarchie de buts. La quête constitue une partie centrale de la structure d'un jeu. Selon Guardiola, la structure d'un jeu est composée d'éléments semblables reliés entre eux de manière cohérente. Une autre approche de la notion de quête dans les jeux vidéo et la difficulté de sa conception est proposée par Susana Tosca [Tosca03].

D'après Guardiola, une quête « [...] permet de mieux comprendre la conception d'un ensemble d'épreuves et d'énigmes, et ce, de la trame générale du jeu au détail d'une séquence. Elle est composée des éléments qui vont être dissociés pour que le joueur les assemble. »

Selon cette même terminologie, une quête est définie par trois caractéristiques principales :

- Un but, par exemple : trouver un code secret.
- Des obstacles qui sont opposés à l'accomplissement du but, tel que l'existence d'un passage secret pour accéder aux codes.
- Une solution ou une méthode de résolution permettant de surmonter les obstacles, par exemple : l'activation d'un mécanisme qui ouvre un passage secret.

---

<sup>1</sup> Le terme quête est traditionnellement employé pour distinguer l'objectif principal à atteindre dans les jeux de rôle. Pour nous, la notion de quête n'est pas limitée à un genre de jeu en particulier. Nous l'utiliserons dans un sens plus large (i.e. elle fait appel aux missions dans le jeu d'aventure aussi).

Guardiola a classifié le genre de rapport entre quêtes selon trois facteurs :

- 1) obligatoire ou facultatif
- 2) linéaire ou non linéaire
- 3) successif ou parallèle

Le premier facteur lie la notion de quête à une décomposition en niveaux. Un niveau se compose d'un ensemble de quêtes. Une quête est obligatoire si son accomplissement est nécessaire pour terminer le niveau et facultative sinon. Le second facteur fait référence à la structure interne de la quête. Il introduit la décomposition en transactions. Une quête est linéaire si elle doit être nécessairement exécutée du début jusqu'à la fin, et non linéaire si son exécution peut être interrompue par l'exécution de tout ou partie d'une autre quête. Le troisième facteur concerne la présentation des quêtes au joueur. Deux quêtes sont successives si elles sont présentées au joueur l'une après l'autre et parallèle si le joueur, connaissant les deux quêtes, peut choisir la prochaine à exécuter.

La hiérarchie de buts du jeu détermine le type de relations entre les quêtes. Cette hiérarchie n'est pas toujours dévoilée au joueur de façon immédiate. Par exemple, au début du jeu, le joueur peut être confronté à la résolution d'un but secondaire ou de bas niveau. Le joueur découvre des buts principaux et des buts secondaires à réaliser pendant son exploration et progression dans le jeu.

Par exemple, le jeu *Black & White* [B&W03] commence par une phase de formation. Le joueur doit exécuter de nombreuses quêtes dédiées à l'évaluation de la dextérité du joueur. Par exemple, il doit trouver un troupeau de moutons perdus dans les montagnes ou aider des marins à construire un bateau. Le joueur peut trouver un premier mouton, puis apporter du bois pour le bateau, et enfin trouver un deuxième mouton...

Donc, une quête n'est pas une unité atomique. Le joueur peut être impliqué simultanément dans plusieurs quêtes. Ainsi, pour décrire l'ordonnancement possible des actions dans un jeu, nous avons besoin d'une description plus fine. Ce nouveau niveau de description nous amène à la décomposition des quêtes en actions de base que le joueur doit réaliser. Ces actions sont définies comme les interactions possibles ou les actions atomiques du joueur dans l'univers du jeu. Nous les appelons « transactions ».

## **Transactions**

Une quête est composée de transactions. Nous définissons une transaction comme une action atomique du joueur. Une transaction devrait être définie en fonction de la mémoire du jeu. En ce qui concerne cette mémoire, le gestionnaire du jeu peut décider qu'une transaction n'a pas commencé ou qu'elle est finie.

Une transaction peut être : « trouver un mouton et ramenez-le à la bergerie ». Naturellement le joueur peut trouver un mouton puis, ayant choisi de faire autre chose, l'abandonner sur la route de la bergerie. Du point de vue de la mémoire de jeu cette action n'a pas été commencée.

Les différents concepts présentés dans cette introduction ont un aspect informel qui rend difficile, sinon impossible, une véritable analyse comparative de jeux et, à fortiori, la

définition d'un modèle constructif. Ce chapitre est consacré à la formalisation des relations d'ordre dans un jeu. Nous nous appuyons pour cela sur les Réseaux de Pétri.

### 3.1.3. Réseau de transactions

Dans ce paragraphe nous donnons la syntaxe d'un langage permettant de décrire tout ou partie des relations d'ordre dans un jeu. Nous appelons une telle description un réseau de transactions.

Un réseau de transactions est décrit par :

- Un ensemble fini de transactions TR.
- Pour chaque transaction A, une propriété Precond(A) qui détermine les conditions d'ordre sur l'exécution des autres transactions qui doivent être vérifiées pour que A soit exécutable. Ces conditions sont exprimées en utilisant deux opérateurs de base : avant(B) et après(B). Ces opérateurs expriment respectivement que A ne peut être exécutée qu'avant ou après l'exécution de B.

#### Description formelle de la syntaxe

Formellement la syntaxe d'un réseau de transactions est définie par :

- Un ensemble fini de transactions TR.
- Deux opérateurs temporels sur TR, avant et après.
- Deux opérateurs logiques,  $\wedge$  et  $\vee$ , vérifiant les propriétés usuelles de commutativité, d'idempotence et de distributivité. Nous notons  $\emptyset$  l'élément neutre de  $\vee$  et l'élément absorbant de  $\wedge$  et 1 son complémentaire.

Pour tout A dans TR, une formule Precondition(A) est définie de la façon suivante :

$$\forall B \in TR, B \neq A$$

1,  $\emptyset$ , avant(B) et après(B) sont des formules A

Si  $\Phi$  et  $\Theta$  sont des formules ( $\Phi \wedge \Theta$ ) et ( $\Phi \vee \Theta$ ) sont des formules

Notons que Precondition(A) peut être écrite sous forme canonique d'une somme de minterms :

$$\text{Precondition}(A) = \bigvee_i m_i$$

$$m_i = \bigwedge_{B \in TR, B \neq A} h(B)$$

avec

$$h(B) = 1 \text{ ou } h(B) = \emptyset \text{ ou } h(B) = \text{avant}(B) \text{ ou } h(B) = \text{après}(B) \text{ ou } h(B) = \text{avant}(B) \wedge \text{après}(B)$$

Exemple : Supposons qu'une quête ait pour objectif d'accéder à un lieu L qui se trouve derrière une porte fermée. Pour ce faire, le joueur doit « trouver la porte ». Cette transaction implique :

- Soit de trouver une clef ou une hache,
- S'il a trouvé la clef, il ouvre la porte,
- S'il a trouvé la hache, il casse la porte,
- Puis il franchit la porte.

Décrivons cette quête par un réseau de transactions :

$TR = \{ \text{trouver\_clef}, \text{trouver\_hache}, \text{ouvrir\_porte}, \text{casser\_porte}, \text{accéder\_L} \}$ , où

$Precond(\text{trouver\_clef}) = 1$

$Precond(\text{trouver\_hache}) = 1$

$Precond(\text{ouvrir\_porte}) = \text{après}(\text{trouver\_clef}) \wedge \text{avant}(\text{casser\_porte})$

$Precond(\text{casser\_porte}) = \text{après}(\text{trouver\_hache})$

$Precond(\text{accéder\_L}) = \text{après}(\text{ouvrir\_porte}) \vee \text{après}(\text{casser\_porte})$

### Description informelle de la sémantique d'exécution

Pour donner une interprétation informelle à la description précédente, il faut se donner une représentation de l'évolution du réseau en termes de logique temporelle :

Une transaction A peut avoir trois états :

1. Non exécutée noté PreA.
2. En cours noté A.
3. Exécutée noté PostA.

L'espace d'état est le produit cartésien des états de toutes les transactions du réseau. Nous utilisons un temps discret ayant un instant initial noté 0, le temps s'incrémente de un à chaque étape (changement d'état). Dans l'état initial, toutes les transactions sont non exécutées.

Pour traduire le caractère atomique des transactions, nous supposons qu'une transaction peut changer d'état en une étape et qu'au plus une transaction peut être en cours à un instant donné. Une transaction peut être exécutée au plus une fois.

Le lecteur familier aura reconnu le modèle de logique temporelle arborescente, les préconditions étant des formules simples dans cette logique. Pour formaliser cette sémantique, nous allons donner une interprétation des réseaux de transaction en termes de réseaux de Pétri.

## 3.2. Introduction aux réseaux de Pétri

Un Réseau de Pétri (RdP) est un cas particulier de formalisation d'un système place-transition. Un RdP est un outil de modélisation graphique et mathématique. Il est composé de *places*, de *transitions* et des *arcs* qui les relient.

Au niveau graphique, les places sont représentées par des cercles, les transitions par des barres et les arcs par des lignes reliant les places et les transitions. Les places peuvent contenir des *jetons*. Un jeton est modélisé par un point noir. Nous dirons qu'une place est marquée lorsqu'elle contient un ou plusieurs jetons. L'état actuel du système, que nous appelons marquage, est donné par le nombre de jetons à chaque place.

Les transitions représentent des composants actifs. Elles modélisent les actions qui peuvent se produire. Une transition est franchissable (tirable) si toutes les conditions préalables sont satisfaites, e.g. si les places d'entrée ont suffisamment de ressources (jetons) pour que l'action soit exécutée.

Quand une transition est franchie, des jetons sont retirés de toutes les places d'entrée et ajoutés dans toutes les places de sortie associées à celle-ci. Le nombre de jetons retirés correspond à la valeur de chacun des arcs reliant les places aux transitions. De la même façon, le nombre de jetons ajoutés correspond à la valeur de chacun des arcs reliant les transitions aux places.

Dans la figure 3.1, tous les éléments d'un RdP sont montrés. La transition T a deux places d'entrée P1 et P2. Les valeurs des arcs reliant les places d'entrée et la transition définissent la condition préalable, qui doit être remplie pour franchir la transition. Selon les arcs (P1, T) et (P2, T), dont les valeurs sont respectivement 1 et 2, la transition T est franchissable si la place P1 contient au moins un jeton et si la place P2 contient au moins deux jetons. Cet ensemble de valeurs s'appelle une pré-condition. Quand T est franchi, un jeton est retiré de P1 et deux jetons sont retirés de P2. La valeur de chaque arc entre T et les places de sortie : (T, P3), (T, P4), (T, P5) définit le nombre de jetons qui sont ajoutés à chaque place de sortie quand T est franchi. Cet ensemble de valeurs s'appelle post-condition. Lorsqu'un arc n'est pas étiqueté, sa valeur par défaut est 1.

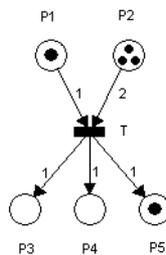


Figure 3.1 : Modèle de Base d'un Réseau de Pétri

C=		t
	p1	-1
	p2	-2
	p3	1
	p4	1
	p5	1

Figure 3.2 : Matrice d'incidence

### Définition 1. (Réseau de Pétri)

De manière plus formelle, nous dirons qu'un Réseau de Pétri  $R$  est défini par

$$R = (P, T, C^+, C^-) \text{ où}$$

$P$  est un ensemble fini de places, non vide, noté  $\{p_1, \dots, p_n\}$ , où  $n = |P|$ .

$T$  est un ensemble fini de transitions, non vide, noté  $\{t_1, \dots, t_m\}$ , où  $m = |T|$ .

$C^-(PXT)$  est la fonction d'incidence des pré-conditions.

$C^+(TXP)$  est la fonction d'incidence des post-conditions.

Ces fonctions peuvent être fusionnées dans une seule matrice, appelée matrice d'incidence  $C$ , où

$$C = C^+ - C^-$$

Cependant, certaines informations seront perdues parce que  $C$  n'est pas toujours équivalent à  $C^+ - C^-$ , en particulier dans le cas d'une boucle. La figure 3.2 montre la matrice d'incidence correspondant au RdP de la figure 3.1.

Chacun des éléments de chaque matrice contient une valeur entière indiquant les relations existantes entre places et transitions. Plus précisément, l'effet que chaque franchissement des différentes transitions aura sur la marque de chaque place :

- Pour  $C^-(PXT)$ , les éléments expriment les marques consommées et nécessaires par le tir de la transition  $t_i$ .
- Dans  $C^+(TXP)$ , les éléments indiquent les marques produites par le tir de la transition  $t_i$ .

### Définition 2. (Marquage)

Le marquage du réseau,  $M_k$ , représente la distribution des jetons (marques) dans les places du réseau. Il peut être représenté par un vecteur de  $n$  composants où  $n = |P|$ , dans lequel le composant  $i$  est le nombre courant de marques contenues dans la place  $P_i$ .

$$M_k = ( M(p_1), \dots, M(p_i), \dots, M(p_n) ).$$

Pour le RdP dans figure 3.1, le marquage  $M_0$  est (1,3,0,0,1).

### Définition 3. (Franchissement d'une transition)

Une transition est franchissable s'il y existe un marquage du réseau suffisamment grand pour effectuer son tir. Si cette condition est validée alors le tir de  $t_i$  en  $M$  conduit à un marquage conséquent,  $M'$ . Nous utilisons l'équation d'état [Vidal92, Silva85] pour vérifier cette possibilité. Nous dirons que  $M'$  est un marquage conséquent de  $M$  si, à partir de  $t_i$ ,

1.  $M \geq C^-(i)$  ( $i^{\text{ème}}$  colonne de  $C^-$ ).
- et
2.  $M' = M + C(i)$

Ceci est noté  $M \xrightarrow{t_i} M'$

**Definition 4. (Franchissement d'une séquence de transitions)**

Par extension, nous pourrons appliquer la notion de franchissement d'une transition à une séquence de transitions [Vidal92]. Considérons  $s$  comme une suite de transitions franchissables dans  $M$  noté par  $s_i = \{t_{i1} \cdot t_{i2} \cdot \dots \cdot t_{in}\}$  tel qu'il existe une série de marquages  $M \xrightarrow{s_{i1}} M_{i1}, M_{i1} \xrightarrow{s_{i2}} M_{i2}, \dots, M_{i_{n-1}} \xrightarrow{s_{in}} M'$ .

Nous disons que  $M'$  est accessible à partir de  $M$  après le tir de  $s_i$  et nous notons ceci  $M \xrightarrow{s_i} M'$ .

Si nous notons  $\hat{s}_i$  le vecteur colonne de  $s_i$  indiquant dans la  $i^{eme}$  composante le nombre d'occurrence de  $t_i$  dans  $s_i$ , alors  $M' = M + C \hat{s}_i$

Dans la figure 3.1, si le marquage  $M$  est tel que  $(1,3,0,0,1)$ , la transition  $T$  est franchissable. Le franchissement de  $T$  mène au marquage conséquent  $M' = (0,1,1,1,2)$ .

**Définition 5. (Graphe de marquage)**

Soit  $M_0$  un marquage, appelé marquage initial. Nous appelons graphe du marquage conséquent le graphe  $G_{M_0} (X_{M_0}, U_{M_0})$ , où

$$X_{M_0} = \{ M \mid \text{tel qu'il existe } s \ M_0 \xrightarrow{s} M \}$$

$$U_{M_0} = \{ (M, M') \mid \text{tel qu'il existe } t_i \ M \xrightarrow{t_i} M' \} \text{ avec } M \in X_{M_0} \text{ et } M' \in X_{M_0}$$

Les arcs sont labelés avec les noms des transitions correspondantes.

Un réseau est dit sain pour le marquage initial  $M_0$  si  $\forall M \in X_{M_0}$  et  $\forall p_i \in P \ M(p_i) \leq 1$ . Dans ce cas, nous notons souvent un marquage par la liste des places marquées (comportant un jeton).

**Définition 6. (Invariant des places ou Flot positif)**

Considérons  $s$  comme une suite de transitions franchissable dans  $M$ , dont  $M'$  est un marquage conséquent et  $\hat{s}$  est le vecteur caractéristique. S'il existe un vecteur  $Y$  indexé par  $P$ , non nul, capable d'annuler la matrice d'incidence  $C$ ,  $Y^T C = 0$  alors,

$$M' = M + C \hat{s}$$

et

$$Y^T M' = Y^T M + Y^T C \hat{s}$$

Nous avons donc,

$$Y^T M' = Y^T M$$

Si  $Y$  est un vecteur positif, il est appelé flot positif. Dans ce cas, lorsque la  $i^{eme}$  composante de  $Y$  est non nulle, la place  $P_i$  est bornée.

$$\forall M' \in \text{Acc}(\mathfrak{R}, M_0), Y^T M' = Y^T M$$

Un flot  $Y$  définit l'invariant  $\forall M' \in \text{Acc}(\mathfrak{R}, M_0), \sum_{p \in P} Y(p) \cdot M(p) = \sum_{p \in P} Y(p) \cdot M_0(p)$

Un flot sur les transitions est appelé un rythme.

### Définition 7. (Invariant de transitions ou rythme)

S'il existe une séquence  $s$  telle que  $M s \rightarrow M'$ , alors  $C \hat{s} = 0$  et  $\hat{s}$  est appelé un rythme.

## 3.3. Interprétation des réseaux de transactions en terme de réseaux de Pétri

### 3.3.1. Modèle d'une Transaction

Dans cette section, nous présentons le modèle de l'opération atomique dans un jeu, la transaction. Nous considérons le jeu comme un séquenceur non déterministe géré par le joueur : à chaque instant le joueur a le choix d'exécuter une transaction parmi plusieurs. Il en choisit une, l'exécute et retourne à la phase de choix. Le concepteur du jeu va, en utilisant différentes techniques de *gameplay*, délimiter les choix possibles à chaque étape.

Notre modèle distingue deux types de places:

1. Les places définissant le statut d'une transaction, où la marque peut exprimer :
  - Le nombre de fois qu'une transaction *peut être* exécutée (franchise).
  - Le nombre de fois que la transaction *a été* exécutée.
  - Le fait que la transaction soit *en cours* d'exécution.
2. Une place modélisant le joueur.
  - Elle est marquée quand le joueur n'est pas impliqué dans l'exécution d'une transaction.

Toutes les transitions potentiellement franchissables définissent l'ensemble des transactions possibles que le joueur peut exécuter. Notons que dans la description d'une transaction simple, donnée par la figure 3.3, la place du joueur, *Player*, est à la fois :

- Place d'entrée de la transition de début de transaction, *BeginA*.
- Place de sortie de la transition de fin de transaction, *EndA*.

La figure 3.3 représente le modèle de base d'une transaction  $A$ . L'exécution d'une transaction commence par le franchissement de *BeginA*. Quand la transaction est en cours d'exécution, la place  $A$  est marquée et les places *PreA* et *Player* sont vides. L'exécution de la transaction finit par le franchissement d'*EndA*. Par conséquent, la marque de *PostA* est

alors égale à 1. Nous appelons un tel réseau une transaction répétitive simple. Ce modèle est étendu à d'autres types de transaction dans le paragraphe 3.4.5.

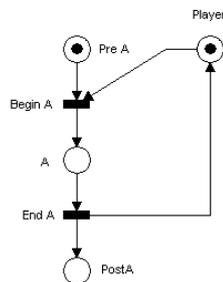


Figure 3.3 : Modèle de Base d'une Transaction Simple

Nous aurions pu modéliser une transaction par une simple transition, en fusionnant les transitions BeginA et EndA. Ceci est possible parce les interprétations sémantiques des Réseaux de Pétri ne permettent pas en général le franchissement simultané de transitions. Cependant, notre modèle a un niveau de description plus fin. Il nous amène à une sémantique claire du temps : les transitions sont franchies instantanément. La durée de la transaction est associée au temps de séjour dans la place A [Vidal92]. Le modèle indique explicitement qu'une transaction est en cours d'exécution.

### Définition 8. (un réseau de Pétri associé à une Transaction)

Nous définissons une Transaction A par

$$A = (P, T, C^+, C^-) \text{ où}$$

- P est un ensemble fini de places,  $|P| = 4$ , noté par  $\{PreA, A, PostA, Player\}$
- T est un ensemble fini de transitions,  $|T| = 2$ , noté par  $\{BeginA, EndA\}$ , où  
BeginA est la transition de début de transaction.  
EndA est la transition de fin de transaction.

- $C^-(PXT)$  est la fonction d'incidence des pré-conditions.
- $C^+(TXP)$  est la fonction d'incidence des post-conditions.

Le marquage initial est  $M(PreA)=M(Player)=1$ , la marque de toutes les autres places étant nulle.

La capacité d'annuler une transaction peut être intégrée à ce modèle de façon simple. En partant de la figure 3.3, une transition AbortA est ajoutée au modèle de base, avec A comme place d'entrée et PreA et Player comme places de sortie. Ceci donne le modèle décrit par le réseau dans la figure 3.4. Cette extension ne modifie pas les propriétés du modèle de jeu et est omise dans le reste du document.

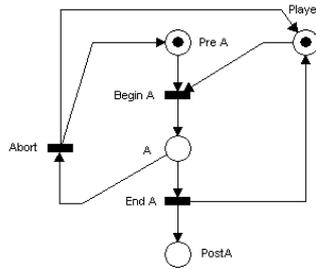


Figure 3.4 : Model de Base d'une Transaction Annulable.

## 3.4. Réseau de Pétri associé à un Réseau de Transactions

### 3.4.1. Modèle d'un réseau de Transactions

Dans cette section nous présentons le modèle d'un Réseau de Transactions. Nous appelons Réseau de Transactions un réseau R obtenu à partir d'un ensemble de transactions en :

1. Fusionnant toutes les places joueur en une seule place, Player.
2. Modifiant le réseau pour modéliser les pré-conditions associées à chaque transaction.

Considérons  $\mathfrak{T}$  comme un ensemble de transactions. Le Réseau de Transactions le plus simple que nous puissions construire est obtenu en fusionnant toutes les places joueur en une seule. Pour deux transactions A et B, cela donne le réseau de la figure 5.

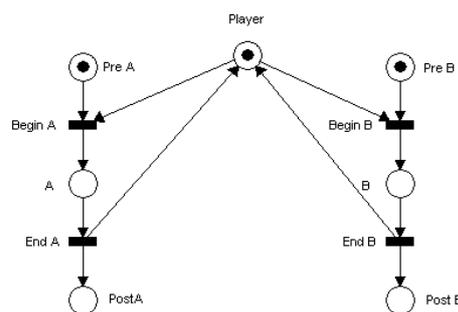


Figure 3.5 : Un modèle de RdT avec deux transactions indépendantes.

Il existe un conflit dans ce réseau de Pétri : deux transitions partagent une même place Player (la place du joueur) comme pré-condition. Il spécifie une exclusion mutuelle pour les places PreA et PreB. Le réseau illustre le rôle de contrôle de la place joueur : il doit s'engager vis à vis d'une des deux transactions. Le joueur doit obligatoirement finir une des deux transactions pour être à nouveau disponible et commencer l'autre transaction.

Le graphe de marquage de ce réseau est donné par la figure suivante :

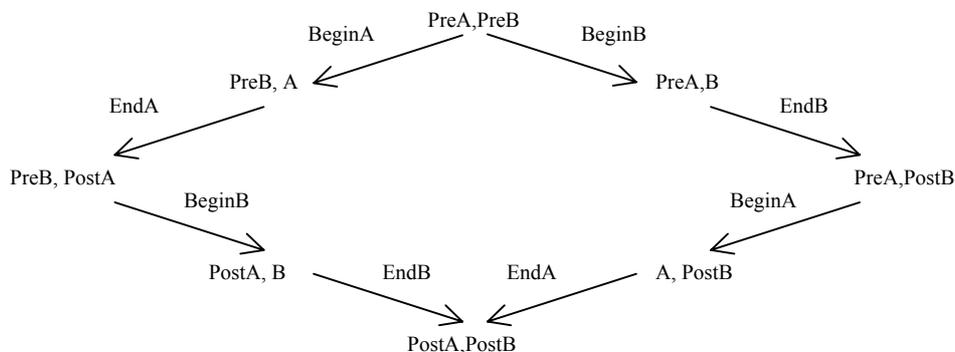


Figure 3.6 : Graphe de marquage du RdT R.

### 3.4.2. Opérateurs d'ordre.

Partant du réseau de la figure 3.5 nous devons modéliser les opérateurs d'ordre avant et après. Une relation d'ordre traduit le fait que le début de l'exécution d'une transaction A (BeginA) dépend de l'état d'avancement d'une autre transaction B : Soit l'exécution de B ne doit pas être commencée, soit elle doit être finie. La relation de dépendance de l'exécution de A par rapport à l'état d'avancement de B se traduit par une condition de franchissement de BeginA relatif à l'état de B, c'est à dire à la marque PreB ou PostB.

Soit A une transaction. Considérons la précondition de A, cette expression est écrite sous forme d'une réunion de minterms :

$$\text{Precondition}(A) = \bigvee_i m_i$$

$$m_i = \bigwedge_{B \in TR, B \neq A} h(B)$$

avec

$$h(B) = 1 \text{ ou } h(B) = \emptyset \text{ ou } h(B) = \text{avant}(B) \text{ ou } h(B) = \text{après}(B) \text{ ou } h(B) = \text{avant}(B) \wedge \text{après}(B)$$

Considérons un minterm m et le fait que la transaction B apparaisse dans ce minterm.

1. B n'apparaît pas dans le minterm m. Il n'y a pas de condition d'ordre direct liant l'exécution de A à l'état de B dans ce minterm. Nous notons ce type de relation ( A || B ). Elle n'induit aucun arc entre les deux sous réseaux constitués par les deux transactions.

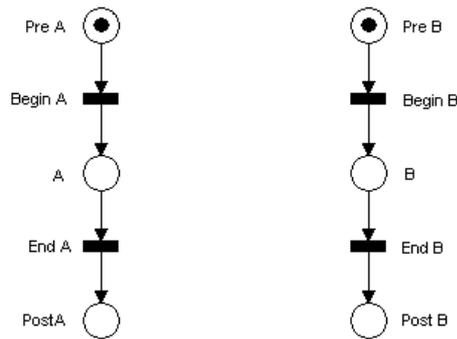


Figure 3.7

Note : *Pour faciliter la lisibilité des réseaux, nous omettons la place Player dans les dessins, mais elle reste toujours incluse dans la définition formelle des modèles.*

Dans le réseau de la figure 3.7, les transactions A et B n'ont aucune contrainte d'exécution entre elles. A ou B peuvent commencer à n'importe quel instant, indépendamment de l'autre transaction. Cette condition est représentée par l'absence d'arcs de synchronisation entre les transactions A et B.

2. B apparaît dans le minterm m sous la forme avant(B)  
Cet opérateur d'ordre se modélise par deux arcs:  
PostB, BeginA et BeginA, PostB

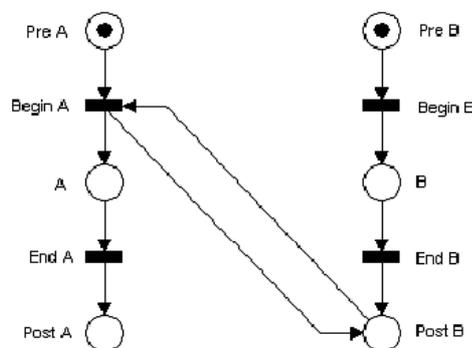


Figure 3.8 : avant(B).

Dans le réseau avant(B) la transaction A peut commencer uniquement si la transaction B a déjà été exécutée. Les arcs reliant PostB à BeginA et BeginA à PostB représentent cette condition.

3. B apparaît sous la forme après(B)  
Cette relation se modélise par deux arcs:  
PreB, BeginA et BeginA, PreB

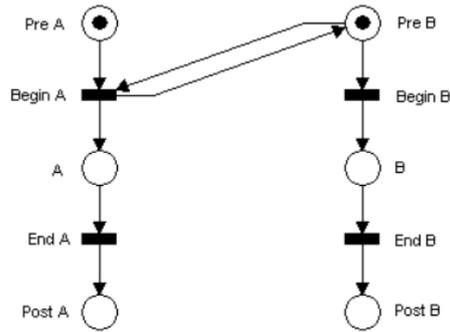


Figure 3.9 : après(B).

La transaction A peut être exécutée uniquement si la transaction B n'a pas été exécutée. Les arcs reliant la place PreB à la transition BeginA représentent cette condition. Dans l'état initial, les deux transactions peuvent être exécutées. Si la transaction A est exécutée d'abord, la transaction B pourra être exécutée après. Mais si la transaction B est exécutée en premier, PreB ne sera plus marquée et BeginA ne sera jamais exécutée.

4. B apparaît sous la forme  $\text{avant}(B) \wedge \text{après}(B)$

A doit être à la fois avant et après B et donc ne peut pas être exécutée.

Cet opérateur de négation se modélise par quatre arcs:

PreB, BeginA      et      BeginA, PreB

PostB, BeginA      et      BeginA, PostB

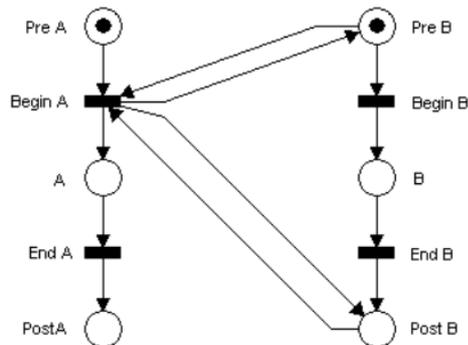


Figure 3.10 :  $\text{avant}(B) \wedge \text{après}(B)$ .

Dans le réseau de la figure 3.10, la transaction A est bloquée. Il n'y existe aucune possibilité d'exécution. Les arcs reliant les places PreB et PostB à la transition BeginA représentent cette condition. Etant des transactions monomarquées, la possibilité d'avoir PreB et PostB marquées simultanément est nulle. Par conséquent, BeginA ne sera jamais exécutable.

### 3.4.3. Opérateurs Logiques.

Les quatre opérateurs précédents modélisent la relation de précédence imposée par l'apparition de B dans un minterm. La modélisation d'un minterm comportant le nom de plusieurs transactions est simple : il suffit de créer tous les arcs suivant les règles précédentes.

La modélisation de l'opérateur de disjonction  $\vee$  entre les minterms est plus complexe :

Le modèle d'une transaction A comporte alors autant de transitions labellées Begin A que de minterms. Notons  $\text{BeginA}_i$  la transition associée au minterm  $m_i$ . Chaque transition  $\text{BeginA}_i$  est liée aux places  $\text{PreX}$  et  $\text{PostX}$  des transactions X apparaissant dans le minterm  $m_i$  selon les règles du paragraphe précédent. Il existe un arc  $\text{PreA}$ ,  $\text{BeginA}_i$  et  $\text{Player}$ ,  $\text{BeginA}_i$  pour tout  $m_i$ .

- La figure 3.11 montre comment ces relations sont modélisées dans le cas où B et C apparaissent dans  $\text{Precondition}(A)$  sous la forme  $F(B)$  et  $G(C)$  dans un minterm ou deux minterms respectivement.

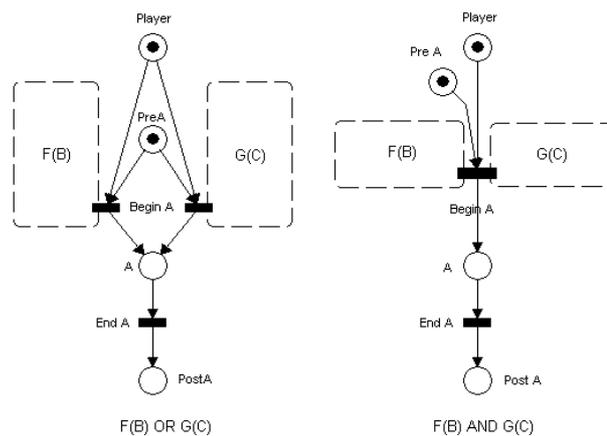


Figure 3.11 : Relations logiques entre RdT.

### 3.4.4. Réseau de Pétri associé à un réseau de Transactions

Il résulte des définitions précédentes qu'un réseau de Pétri associé à un réseau de transactions a la forme suivante :

Soit  $\text{RT} (\text{TR}, \mathfrak{R})$  un Réseau de Transactions où

$\text{TR}$  est un ensemble fini de transactions, non vide, noté par  $\{a_1, \dots, a_k, \dots, a_n\}$ ,  
où

$\mathfrak{R}$  est un ensemble de  $n$  Préconditions, chaque Précondition est une application de TR dans l'ensemble des formules définies en §1.3. Réseau de transactions : Description formelle de la syntaxe (page 6).

Précondition( $a_k$ ) est écrite sous forme canonique d'une somme de minterms :

$$\text{Precondition}(a_k) = \bigvee_i m_i^k$$

$$m_i^k = \bigwedge_{j \in [1..n], j \neq i} h(a_j)$$

avec

$$h(a_j) = 1 \text{ ou } h(a_j) = \emptyset \text{ ou } h(a_j) = \text{avant}(a_j) \text{ ou } h(a_j) = \text{après}(a_j) \text{ ou } h(a_j) = \text{avant}(a_j) \wedge \text{après}(a_j)$$

Nous notons  $n(k)$  le nombre de minterms apparaissant dans Précondition( $a_k$ ).

RT est modélisé par réseau de Pétri  $R(P, T, C^+, C^-, V, M_0)$ , où

$P$  est un ensemble fini de places, non vide, avec  $|P| = 3n+1$  et à chaque transaction  $a_i$ , est associé trois places  $\{\text{Pre}_i, P_i, \text{Post}_i\}$  plus une seule place  $\{\text{Player}\}$ .

$T$  est un ensemble fini de transitions, non vide  $T = \bigcup_k T_k$   
 $T_k$  est un ensemble de  $n(k)+1$  transitions associées à la transaction  $a_k$   
 $T_k = \{\text{Begin}_k^1 \dots \text{Begin}_k^{n(k)}, \text{End}_k\}$   
 $\text{Begin}_k^i$  est associée au minterm  $m_i^k$

$C^+, C^-$  sont les fonctions de valuation de post-conditions et pré-conditions du réseau de transactions définies par :

- Les transitions, ensemble comportant tous les arcs des transactions de la forme :
  - Pre<sub>i</sub>, Begin<sub>i</sub>   valué à 1
  - Begin<sub>i</sub>, P<sub>i</sub>     valué à 1
  - P<sub>i</sub>, End<sub>i</sub>       valué à 1
  - End<sub>i</sub>, Post<sub>i</sub>   valué à 1
  - Player, Begin<sub>i</sub> valué à 1
  - End<sub>i</sub>, Player   valué à 1

Notons qu'en fusionnant la place du joueur, Player est à la fois place d'entrée pour toute transition de début de transaction et place de sortie pour toute transition de fin de transaction.

- Les synchronisations induites par les opérateurs d'ordre, ensemble d'arcs de la forme :
  - Post<sub>i</sub>, Begin<sub>j</sub> et Begin<sub>j</sub>, Post<sub>i</sub>   valué à 1
  - Pre<sub>i</sub>, Begin<sub>j</sub> et Begin<sub>j</sub>, Pre<sub>i</sub>   valué à 1

$Pre_j, Begin_i$	et	$Begin_i, Pre_j$	valué à 1
$Post_i, Begin_j$	et	$Begin_j, Post_i$	valué à 1
$Post_j, Begin_i$	et	$Begin_i, Post_j$	valué à 1

$M_0$  est le marquage initial du réseau, où

$$\forall a_i \in R, \quad M_0(\text{Player}) = 1,$$

$$M_0(Pre_i) \leq 1 \text{ et}$$

$$M_0(P_i) = M_0(Post_i) = 0$$

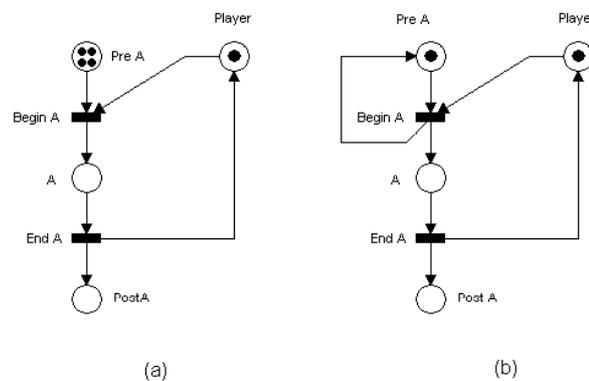
Quand aucune confusion n'est possible, dans la suite de ce texte, nous appelons Réseau de transactions le réseau de Pétri associé à un réseau de transactions.

### 3.4.5. Abréviations

Dans ce paragraphe nous utilisons les propriétés des réseaux de Pétri pour abréger la représentation de certains modèles. Une abréviation est un modèle plus concis qui préserve les propriétés essentielles du réseau initial.

#### Transactions multiples et transactions répétitives

Supposons que, dans un jeu, il soit possible de réaliser  $m$  fois la même transaction  $A$ . Dans le réseau de transactions elle apparaît  $m$  fois avec des noms différents,  $A_1, A_2, \dots, A_m$ , mais pour toutes ces occurrences les préconditions sont identiques. Il lui correspond  $k$  sous réseaux mono marqués dans le modèle. Une abréviation classique des réseaux de Pétri consiste à utiliser un seul sous réseau initialement marqué avec  $k$  jetons. Tant que  $k$  reste fini, cette abréviation ne modifie pas les propriétés du réseau, mis à part la capacité à distinguer les différentes occurrences d'une même transaction. Par contre s'il est possible de répéter infiniment souvent la transaction, les propriétés sont modifiées [Diaz01, Vidal92]. Nous appelons le modèle de telles transactions une transaction répétitive. Les réseaux de Pétri correspondant sont modélisés par la figure suivante :



**Figure 3.12 : Modèle de Base d'une Transaction Multiple (a) et une Transaction Répétitive (b).**

## Ordre généralisé

Considérons toujours que, dans un jeu, il soit possible de réaliser  $m$  fois la même transaction  $A$ . Supposons maintenant que l'on veuille modéliser le fait qu'une transaction  $B$  ne puisse s'exécuter que si  $A$  s'est exécutée au moins  $r$  fois et au plus  $s$  fois avec  $0 < r \leq s \leq m$ . Ceci généralise la précondition  $\text{avant}(A) \wedge \text{après}(A)$ . Un réseau de transactions permet de modéliser un tel comportement.  $A$  apparaît à nouveau  $m$  fois avec des noms différents,  $A_1, A_2, \dots, A_m$ . Il suffit d'introduire dans la précondition de  $B$  un terme de la forme :

$$\text{après}(A_1) \wedge \text{après}(A_2) \dots \wedge \text{après}(A_r) \wedge \text{avant}(A_{s+1}) \wedge \text{avant}(A_{s+2}) \dots \wedge \text{avant}(A_m)$$

On en déduit le réseau de Pétri correspondant. Tant que  $m$  est fini il existe une abréviation de cette condition en utilisant une transaction multiple  $A$ . Les deux arcs liant  $\text{Post}A$  à  $\text{Begin}B$  sont valués à  $r$  et ceux liant  $\text{Pre}A$  et  $\text{Begin}B$  sont valués à  $m-s$ . Par conséquent, tant que  $\text{Post}A$  ne contient pas  $r$  jetons, donc que  $A$  n'a pas été exécutée  $r$  fois,  $B$  n'est pas exécutable. Quand  $A$  a été exécutée plus de  $s$  fois,  $\text{Pre}A$  contient moins de  $m-s$  jetons.  $B$  n'est plus exécutable.

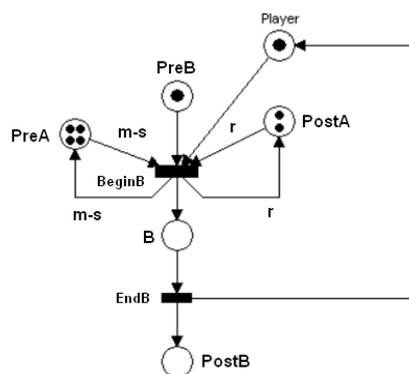


Figure 3.13 : Modèle généralisé de la précondition  $\text{avant}(A) \wedge \text{après}(A)$ .

Dans ce cas également, tant que  $m$  est fini, ceci ne modifie pas les propriétés du réseau. Dans le cas où  $A$  est une transaction répétitive, il faut utiliser sur le réseau de Pétri des arcs inhibiteurs testant des places non bornées. Ceci change complètement la puissance des réseaux et donc leurs propriétés [Vidal92, Silva85].

Dans le chapitre 4 nous utiliserons un autre type d'abréviation, les réseaux colorés qui permettent de conserver toutes les propriétés, y compris la capacité de distinguer plusieurs occurrences d'une même transaction.

Notons enfin que, pour pouvoir utiliser ces abréviations dans le langage de spécification d'un réseau de transactions il faut introduire :

- Dans la définition d'une transaction, son nombre maximal d'occurrences.
- Dans la définition des opérateurs avant et après les nombres d'occurrences  $r$  et  $s$ .

### 3.4.6. Propriétés d'un réseau de Pétri associé à un réseau de transactions

Dans ce paragraphe, nous démontrons que le réseau de Pétri associé à un réseau de transactions correspond bien (dans sa sémantique d'exécution) aux propriétés que l'on peut attendre des opérateurs qui ont servi à le construire.

Nous considérons que la sémantique d'exécution d'un réseau de Pétri correspond aux règles d'évolution du marquage et est traduite par le graphe des marquages accessible. Pour simplifier les notations, nous introduisons la notion de graphe réduit.

#### Graphe Réduit

Soit  $M$  et  $M'$  deux marquages consécutifs du marquage initial.

Si  $M \xrightarrow{\text{Begin } t_i} M'$ , alors la seule transition tirable en  $M'$  est  $\text{End } t_i$ . Alors, pour simplifier les notations dans le graphe de marquage, nous remplaçons  $M \xrightarrow{\text{Begin } t_i} M' \xrightarrow{\text{End } t_i} M''$  par  $M \xrightarrow{i} M''$ .

Le graphe obtenu par cette transformation est appelé graphe réduit. Nous allons analyser l'effet des opérateurs définis précédemment sur ces graphes réduits.

Dans la figure 3.14, nous illustrons la réduction du graphe du marquage introduit par le RdT de la figure 3.6. Le graphe réduit est montré en noir tandis que le graphe du marquage reste en arrière plan, en gris.

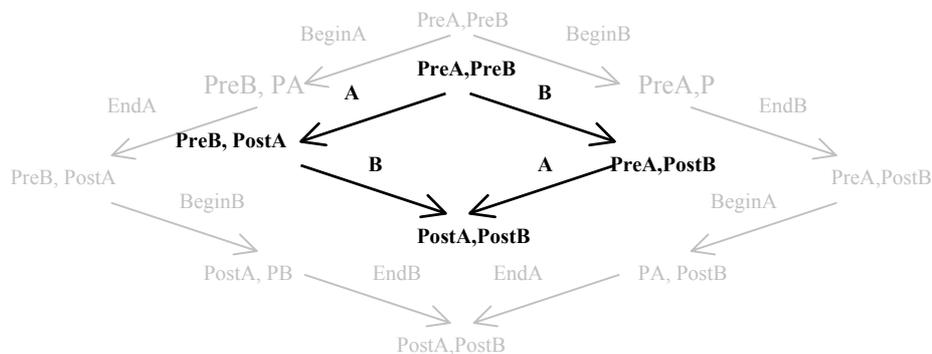


Figure 3.14 : Graphe réduit du RdT (figure 4.6, §4.1).

Soit un réseau de transactions, notons  $TR$  l'ensemble des transactions et  $TRM$  les transactions multiples

#### Théorème 1.

Dans un réseau de transactions une base de flots positifs est donnée par les invariants

$$\forall A \in TRM \quad M(PreA) + M(A) + M(PostA) = M_0(PreA)$$

$$\forall A \in TR \quad M(Player) + M(A) = M_0(Player)$$

Le fait que ces deux types de relations définissent des flots positifs se vérifie immédiatement sur C et par construction ils sont indépendants. Ils couvrent toutes les places du réseau à l'exception des places PreA et PostA des transactions répétitives. Or, par construction, ces places sont non bornées. Elles ne peuvent donc être couvertes par un flot positif. L'ensemble des invariants proposés définissent donc une base de P flot.

### Corollaire

Toutes les places d'un réseau de transactions, à l'exception des place PreA et PostA des transactions répétitives, sont bornées. Plus précisément

$$M(Player) \leq 1$$

$$\forall A \in TR$$

$$M(A) \leq 1$$

$$\forall A \in TRM$$

$$M(PreA) \leq M_0(PreA)$$

$$M(PostA) \leq M_0(PreA)$$

### Théorème 2.

Soit A une transaction répétitive d'un réseau de transactions, alors les transitions BeginA et EndA sont franchissables au plus  $M_0(PreA)$  fois.

Montrons que tout vecteur entier et positif X (2n, 1) tel que :  $M_0 + C.X \geq 0$  est tel que

$$\forall A \in TRM \quad X(BeginA) \leq M_0(PreA) \quad \text{et} \quad X(EndA) \leq M_0(PreA)$$

En effet supposons que x soit une séquence franchissable depuis le marquage initial et X son vecteur caractéristique. L'équation  $M = M_0 + C.X$  relativement à la transaction A s'écrit :

$$M(PreA) = M_0(PreA) - X(BeginA)$$

$$M(A) = X(BeginA) - X(EndA)$$

$$M(PostA) = X(EndA)$$

$$\text{Comme } M(PreA) \geq 0$$

$$M_0(PreA) \geq X(BeginA)$$

$$\text{Comme } M(A) \geq 0$$

$$X(BeginA) \geq X(EndA)$$

$$\text{donc } M_0(PreA) \geq X(BeginA) \geq X(EndA)$$

## Corollaire

Soit  $A$  une transaction répétitive d'un réseau de transactions, alors la transition apparaît au plus  $M_0(\text{Pre}A)$  fois dans le graphe réduit. En particulier, si le réseau ne comporte que des transactions simples (i.e.  $\forall A \quad M_0(\text{Pre}A) = 1$ ), alors le graphe réduit comporte au plus  $n$  arcs, chaque arc étant labellé au plus une fois par le nom d'une transaction.

## Théorème 3.

Le graphe des marquages conséquents d'un réseau de transactions est sans circuit.

Nous allons montrer que le nombre

$$w(M) = \sum_{A \in TR} (3M(\text{Pre}A) + 2M(A) + M(\text{Post}A))$$

décroît strictement avec le nombre de transitions tirées depuis le marquage initial (c'est un P sous flot positif). Remarquons d'abord que ce nombre est toujours strictement positif et inférieur ou égal à  $\sum_{A \in TR} 3M_0(\text{Pre}A)$ . Si  $w(M)$  est un sous P flot positif, ceci implique le

caractère non cyclique du graphe car on ne peut avoir deux marquages  $M$  et  $M'$  tel que  $M$  soit accessible de  $M'$  et  $M'$  de  $M$  par deux séquences non vides. En effet nous aurions à la fois :

$$w(M) \geq w(M') \text{ et } w(M') \leq w(M) \text{ avec } w(M) > 0 \text{ et } w(M') > 0.$$

Etudions l'effet du franchissement de chaque transition sur  $w(M)$ .

Si  $M'$  est atteint à partir de  $M$  par le tir de Begin  $A$ , seules les marques de  $\text{Pre}A$  et  $A$  sont affectées.

$$M'(\text{Pre}A) = M(\text{Pre}A) - 1 \text{ et } M'(A) = M(A) + 1. \text{ Donc } w(M') = w(M) - 1.$$

Si  $M'$  est atteint à partir de  $M$  par le tir de End  $A$ , seules les marques de  $\text{Post}A$  et  $A$  sont affectées.

$$M'(A) = M(A) - 1 \text{ et } M'(\text{Post}A) = M(\text{Post}A) + 1. \text{ Donc } w(M') = w(M) - 1.$$

## Corollaire

Les relations d'atteignabilité dans le graphe des marquages conséquents et le graphe réduit d'un réseau de transactions sont des relations d'ordre partiel strict.

Dans la suite de ce paragraphe, nous ne considérons que des réseaux ne comprenant que des transactions simples. Tant qu'un réseau ne comprend pas de transactions multiple, ce n'est pas une restriction.

Nous commençons par montrer une relation fondamentale entre les préconditions d'un réseau de transactions et le graphe des marquages du réseau de Pétri correspondant.

#### Théorème 4.

Soit RT et RT' deux réseaux de transactions ayant même ensemble de transactions et qui ne se distinguent que par la condition suivante :

Il existe une transaction  $a_k$  et un minterm  $m_s^k$  tel que dans RT

$$\text{Precondition}(a_k) = \bigvee_i m_i^k$$

et dans RT'

$$\text{Precondition}(a_k) = \bigvee_i m_i^k$$

$$\forall i \neq s \ m_k^i = m_k^i$$

$$m_k^s = m_k^s \wedge g(X)$$

ou X est une transaction et  $g(X) = \text{avant}(X)$  ou  $\text{après}(X)$  ou  $\text{avant}(X) \wedge \text{après}(X)$

Alors le graphe des marquages conséquents de RT' est inclus dans le graphe des marquages conséquents de RT.

La propriété est liée au fait que les deux réseaux de Pétri ne diffèrent que par l'existence dans RT' d'arcs en boucle qui relie la transition  $\text{Begin}_k^i$  et PreX ou PostX. Ces arcs n'existent pas dans RT. Par conséquent, les matrices d'incidence de RT' et RT sont liées par les relations  $C'^+ \geq C^+$  et  $C' = C$ . Si  $\text{Begin}_k^i$  est franchissable dans RT' à partir de M elle l'est dans RT et le franchissement conduit aux mêmes marquages. La réciproque est fautive, ce qui implique la propriété demandée.

#### Théorème 5.

Soit RT et RT' deux réseaux de transactions ayant même ensemble de transactions et qui ne se distinguent que par la condition suivante :

Il existe une transaction  $a_k$  tel que dans RT

$$\text{Precondition}(a_k) = \bigvee_i m_i^k$$

et dans RT'

$$\text{Precondition}(a_k) = \bigvee_i m_i^k \wedge m'_{n(k)+1}$$

ou  $m'_{n(k)+1}$  est un minterm. Alors le graphe des marquages conséquents de RT' est inclus dans le graphe des marquages conséquents de RT.

La propriété est liée au fait que les deux réseaux de Pétri ne diffèrent que par l'existence, dans RT', d'arcs d'une transition  $\text{Begin}_{n(k)+1}^k$  qui n'existe pas dans RT. Toute transition franchissable dans RT à partir de M l'est dans RT' et le franchissement conduit aux mêmes marquages. De plus il existe un ensemble de marquages (éventuellement vide) tel que  $\text{Begin}_{n(k)+1}^k$  est franchissable alors que les transitions  $\text{Begin}_i^k$   $i \neq n(k)+1$  ne sont pas franchissables. Ceci qui implique la propriété demandée.

Dans le paragraphe consacré aux opérateurs d'ordre, nous avons modélisé l'influence que peut avoir une transaction B sur la précondition d'une transaction A. Considérons

maintenant la combinaison de cette relation et de la relation inverse. Nous considérons dans un premier temps que les Préconditions de A et B comportent qu'un seul minterm. Nous obtenons 10 cas possibles :

1. B n'apparaît pas dans aucun minterm de A et réciproquement.

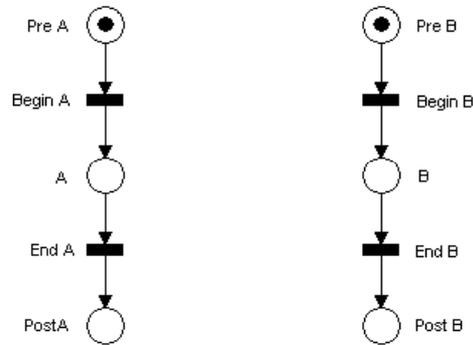


Figure 3.15 : A || B

2. B apparaît sous la forme avant(B), A n'apparaît pas dans la précondition de B.

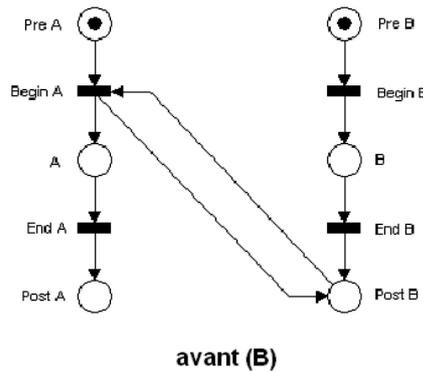


Figure 3.16 : avant (B).

3. A apparaît sous la forme avant(A), B n'apparaît pas dans la précondition de A.

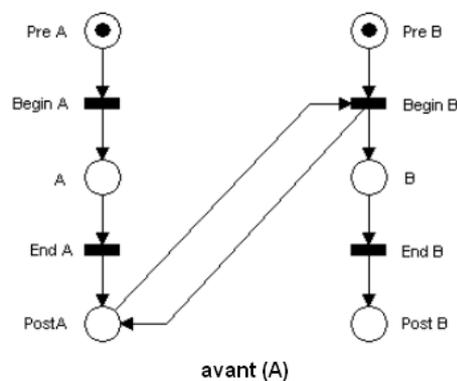


Figure 3.17 : avant (A).

4. B apparaît sous la forme après(B), A n'apparaît pas dans la précondition de B.

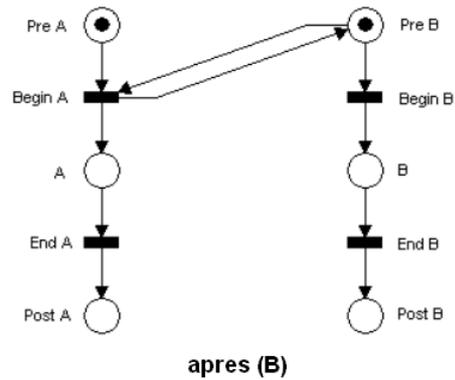


Figure 3.18 : après (B).

5. A apparaît sous la forme après(A), B n'apparaît pas dans la précondition de A.

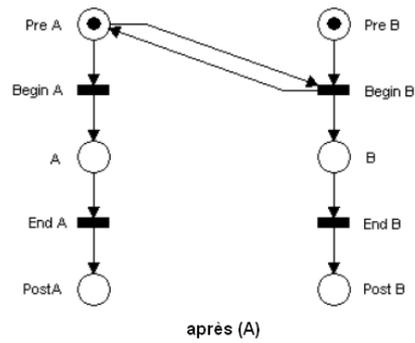


Figure 3.19 : après (A).

6. B apparaît sous la forme avant(B)  $\wedge$  après(B), A n'apparaît pas dans la précondition de B. Nous notons cette relation  $\lceil A$ .

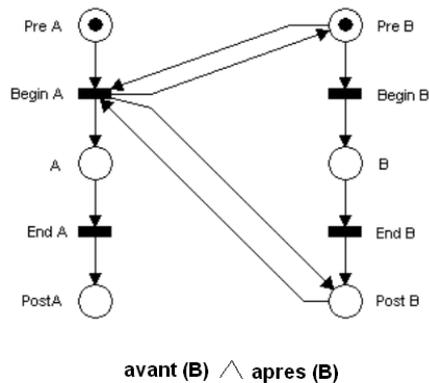


Figure 3.20 :  $\lceil A$ .

7. A apparaît sous la forme  $\text{avant}(A) \wedge \text{après}(A)$ , B n'apparaît pas dans la précondition de A. Nous notons cette relation  $\lceil B$ .

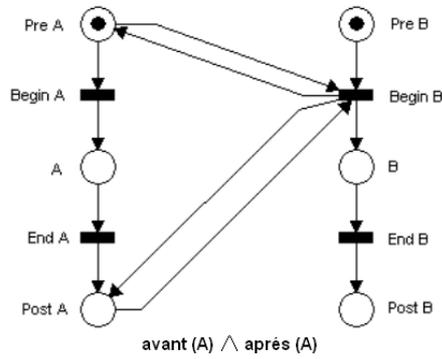


Figure 3.21 :  $\lceil B$ .

8. B apparaît sous la forme  $\text{après}(B)$ , A apparaît sous la forme  $\text{après}(A)$ . Notons cette relation exclusion mutuelle  $(A \oplus B)$ .

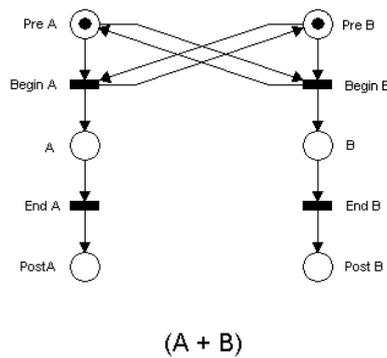


Figure 3.22 :  $(A \oplus B)$ .

9. B apparaît sous la forme  $\text{après}(B)$ , A apparaît sous la forme  $\text{après}(A)$ .

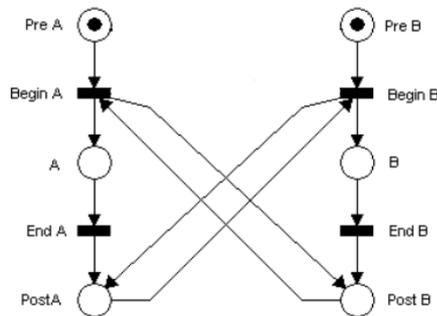
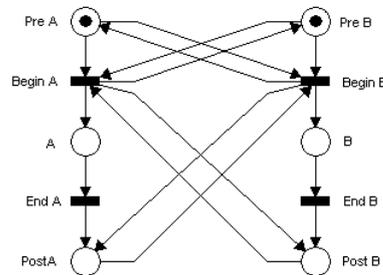


Figure 3.23 :  $(A \oplus B)$ .

10. B apparaît sous la forme  $\text{avant}(B) \wedge \text{après}(B)$ , A apparaît sous la forme  $\text{avant}(A) \wedge \text{après}(A)$ .



(  $\lceil$  A  $\rceil$  B )

Figure 3.24

Nous montrons dans la suite que, dans les deux cas précédents, ni A ni B ne peuvent pas être exécutés. Nous notons ces relations  $\lceil$ A $\rceil$ B.

Nous allons démontrer que ces relations entre A et B constituent un treillis ordonné par l'inclusion des graphes de marquages correspondants.

**Théorème 6.**

Si B, dans la précondition de A, apparaît sous la forme  $\text{avant}(B)$ , alors, dans le graphe réduit, il n'existe aucun chemin de la forme :

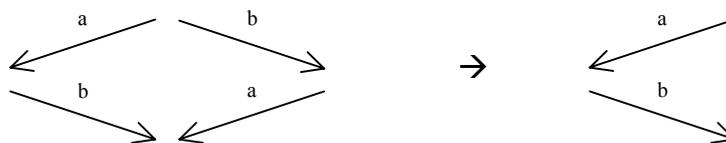
$$(t_R^1, t_R^2, B, \dots, A, t_R^n). \text{ Aucun arc de label B ne précède aucun arc de label A.}$$

D'après l'équation  $M=M_0+C.X$  on a

$$M(\text{Post}A) = X(\text{End}A) \leq 1 \text{ et } X(\text{Begin}B) \leq M(\text{Pre}B) \leq 1.$$

D'autre part, une des conditions nécessaires de franchissement de BeginB est  $M(\text{Post}A) \geq 1$ , soit encore  $X(\text{Begin}B) \leq M(\text{Post}A)$ , d'où  $X(\text{Begin}B) \leq X(\text{End}A)$ . En conséquence, la transaction B ne peut être exécutée que lorsque A est terminée.

Lemme. Les chemins comportant des labels A ou B, s'ils existent, sont soit de la forme (s1, A, s2), soit de la forme (s1, A, s2, B, s3).



**Théorème 7.**

Si B, dans la précondition de A, apparaît sous la forme après(B), alors dans le graphe réduit il n'existe aucun chemin de la forme :  $(t_R^1, t_R^2, A, \dots, B, t_R^n)$ . Aucun arc du label A ne précède aucun arc du label B.

D'après l'équation  $M=M_0+C.X$ , on a  $X(BeginA) \leq M(PreA) \leq 1$ .

D'autre part, une des conditions nécessaires de franchissement de BeginB est  $M(PreA) = 1$  soit encore  $X(BeginB) \leq M(PreA)$  d'où  $X(BeginB) \leq X(BeginA)$ . En conséquence, la transaction B est exécutée nécessairement avant A. De plus, si BeginA est franchi dans un marquage M, pour tout marquage M' conséquent de M,  $M'(PreA)=0$  donc B ne peut plus être exécutée.

Lemme. Les chemins comportant des labels A, s'ils existent, sont de la forme  $(s1, A, s2)$  avec  $B \notin s2$ .



**Théorème 8.**

Si B, dans la précondition de A, apparaît sous la forme  $avant(B) \wedge après(B)$  alors dans le graphe réduit il n'existe aucun chemin de la forme :  $(s,A, r)$ . Ceci résulte des théorèmes 4, 6 et 7.



On montre de la même façon :

### Théorème 9.

Si A et B sont en relation exclusion mutuelle ( $A \oplus B$ ) alors, dans le graphe réduit, il n'existe aucun chemin de la forme  $(s, A, r)$  avec  $B \in r$  ou  $B \in s$ , ni de chemin de la forme  $(s, B, r)$  avec  $A \in r$  ou  $A \in s$ .



### Théorème 10.

Si A et B sont dans la relation  $\lceil A \rceil B$  si dans Precondition (A) B apparaît sous la forme après(B) ou avant(B)  $\wedge$  après(B) et A apparaît dans la Precondition de A sous la forme après(A) ou avant(A)  $\wedge$  après(A), alors aucun arc labellé par A ou B n'apparaît dans le graphe réduit.



Les théorèmes précédents permettent d'ordonner les graphes réduits des réseaux de transactions se distinguant uniquement par les opérateurs d'ordre liant uniquement deux transactions A et B.

La relation d'inclusion des graphes correspondant constitue le treillis dans la figure 3.25. Ce treillis peut être généré à partir de toutes les combinaisons possibles entre deux opérateurs avant et après. Ce résultat donne une autre interprétation intuitive de notre construction. Considérons les deux contraintes que l'exécution de B peut imposer à A : B est avant A, B est après A. Ceci revient à conditionner BeginA soit à la marque de PostB soit à celle de PreB. En combinant ces deux contraintes et en permutant le rôle de A et B, nous obtenons les 9 relations de contrainte entre A et B. Avant et après sont donc suffisantes pour définir toutes les structures d'ordre d'exécution entre deux transactions.

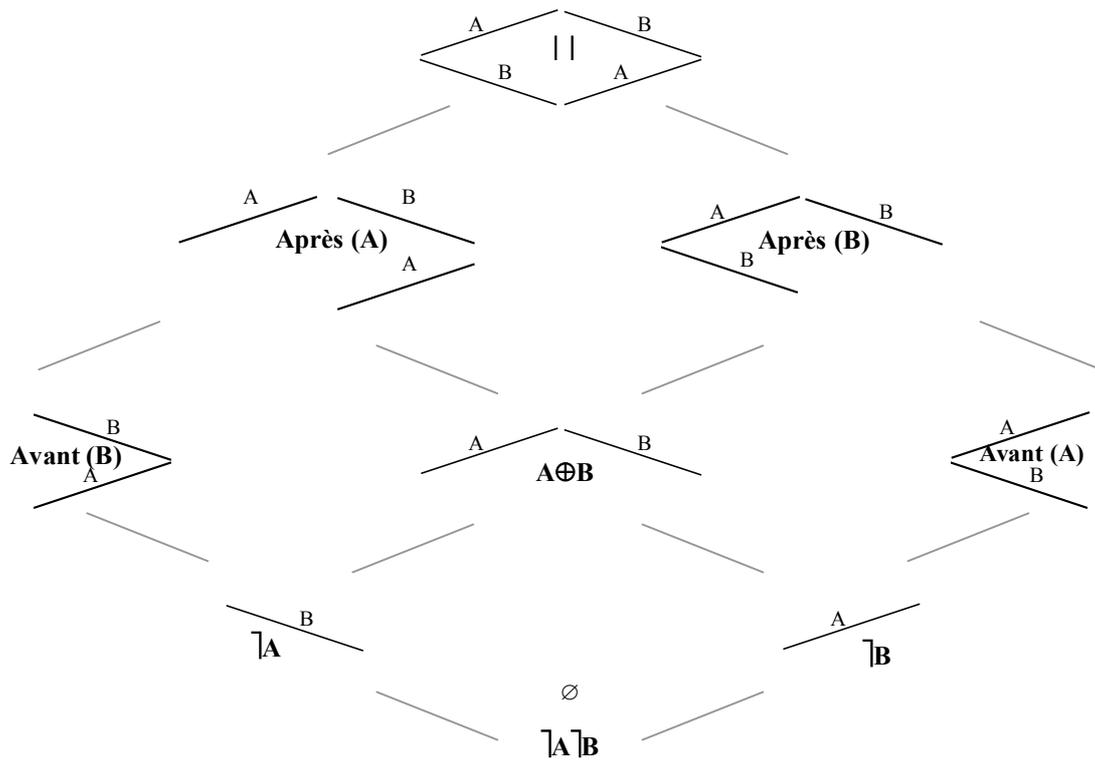


Figure 3.25 : Treillis des graphes réduits.

Les notions classiques de vivacité n'ont pas, dans le cas des modèles ludiques, une application directe : il est possible de terminer un jeu sans avoir exécuté toutes les transactions. Il est même possible de quitter un niveau sans avoir exécuté toutes les transactions et sans pouvoir les exécuter dans la suite du jeu. Par contre, quand le concepteur a créé une transaction, il doit être possible de l'exécuter. Nous donnons une condition nécessaire pour que cette propriété soit vérifiée.

Considérons un réseau de quête et définissons la relation entre transactions  $\xrightarrow{\text{toujours}}$  définie par  $A \xrightarrow{\text{toujours}} B$  si, soit  $\text{Précondition}(A)$  peut s'écrire sous la forme  $\text{Précondition}(B) = \text{avant}(A) \wedge f$  ou  $f$  est une formule non vide, soit  $\text{Précondition}(A) = \text{après}(B) \wedge g$  ou  $g$  est une formule non vide.

### Théorème 11.

Si la relation  $\xrightarrow{\text{toujours}}$  n'est pas un ordre strict, alors il existe une transaction  $A$ , qui n'est jamais exécutée (qui n'est le label d'aucun arc du graphe réduit)

Si  $\xrightarrow{\text{toujours}}$  n'est pas un ordre strict alors il existe un ensemble de transaction  $A, B, \dots, X$  tel que

$A \xrightarrow{\text{toujours}} B \xrightarrow{\text{toujours}} \dots X \xrightarrow{\text{toujours}} A$ . En conséquence des théorèmes 6 et 7, dans aucun chemin du graphe réduit on ne trouve de chemin de la forme  $A\dots B$  ni  $B\dots A$ . Mais pour les mêmes raisons  $B$  ne peut apparaître que dans un chemin de la forme  $A\dots B\dots$ . Donc  $B$ , ni aucune des transactions de la séquence ne sont exécutées.

### 3.5. Quêtes

Dans ce paragraphe nous introduisons le modèle de quête, qui constitue du point de vue de la sémantique du jeu, la brique de base du *gameplay*. Une quête est composée, d'un ensemble de transactions, d'un début et d'une fin. Si le jeu est cohérent, il est possible de terminer la quête en exécutant tout ou partie de ses transactions. Nous formalisons ceci par la définition suivante.

Une Quête est un Réseau de Transactions possédant :

- Une seule transaction initiale, notée  $B$  telle que la précondition de  $B$  soit 1. Donc dans l'état initial la transition  $\text{Begin}B$  est franchissable.
- Une transaction finale, transaction notée  $E$  telle que  $E$  n'apparaisse dans aucune précondition des autres transactions du réseau de quête.

Il peut exister plusieurs transactions ayant cette propriété. Dans ce cas, une d'entre elles est choisie comme transaction finale. Lorsque  $E$  est exécutée, nous dirons que la quête est terminée.

- Soit la quête est réduite à une transaction  $B=E$ , soit il existe une suite de transactions  $a_1, a_2, \dots, a_j$  telle que  $B \xrightarrow{\text{toujours}} a_1 \xrightarrow{\text{toujours}} a_2 \xrightarrow{\text{toujours}} \dots a_j \xrightarrow{\text{toujours}} E$

Ceci veut dire qu'il existe une suite de transactions qui doit obligatoirement être exécutée pour terminer la quête. Par contre une quête peut être exécutée sans que toutes ses transactions soient exécutées.

Sans perte de généralité, nous supposons que les transactions initiale et finale peuvent être exécutées seulement une fois. Si ce n'est pas le cas, il est toujours possible d'ajouter ces transactions fictives au modèle.

Par conséquent, le modèle de Réseau de Quêtes a une place et une transition de début ( $\text{Pre}B$  et  $\text{Begin}B$ ) et une place et une transition de fin ( $\text{Post}E$  et  $\text{End}E$ ).

Dans une quête :

- La transaction finale ne peut pas être exécutée si la transaction initiale n'est pas exécutée.
- Si la transaction initiale est exécutée, la transaction finale sera éventuellement exécutée.

## 3.6. Niveaux

Dans ce paragraphe, nous présentons le modèle de réseau de niveaux. Nous considérons que chaque niveau est constitué d'un ensemble de quêtes cohérentes nécessaires pour que le joueur réalise un objectif. Un niveau représente ainsi un chapitre de l'histoire, un bloc principal de la structure du jeu. Les modèles de construction présentés jusqu'à présent (transactions et quêtes) définissent les différents degrés de relation d'ordre des actions dans un jeu, où l'élément de décomposition le plus haut est le *niveau*. Nous supposons qu'une quête ne peut pas être exécutée plusieurs fois dans un niveau donné et qu'un niveau a toujours une quête de début et une quête de fin.

Autrement dit, un niveau est un réseau constitué d'un recouvrement de quêtes et qui possède au moins une quête initiale, pouvant être commencée avant toutes les autres, et au moins une quête finale, dont l'achèvement ne conditionne aucune autre quête.

### Définition 12.

Nous appelons Réseau de niveaux, un réseau de Pétri  $R(P, T, C^+, C^-, M_0)$  associé à un réseau de transactions tel qu'il existe un ensemble de réseaux de Pétri  $R_1(P_1, T_1), R_2(P_2, T_2), \dots, R_n(P_n, T_n)$  avec :

- Les réseaux  $R_k$  sont des quêtes
- $\cup P_i = P, \cup T_i = T$
- Si une place  $p_{i,k} \in P_k$  ou une transition  $t_{j,k} \in T_k$  alors toute la transaction correspondante est dans  $R_k$
- Si une place  $p_{i,k} \in P_k$  et une transition  $t_{j,k} \in T_k$  alors tout arc de la forme  $(p_{i,k}, t_{j,k})$  ou  $(t_{j,k}, p_{i,k})$  est dans  $R_k$  avec la même valuation que dans  $R$ .
- Il existe au moins une transaction  $R_1$  telle que, la Precondition dans  $R$  de la transaction initiale de  $R_1$  dans  $R$  soit la même que dans  $R_1$  considéré isolément, soit 1.
- Il existe au moins une transaction  $R_k$  telle que, dans  $R$ , la transaction finale de  $R_k$  n'apparaisse dans aucune précondition d'aucune transaction de  $R$ .

Dans un niveau donné les quêtes peuvent avoir des relations d'ordre résultant au moins d'une des relations structurelles suivantes. Deux quêtes peuvent :

- ne pas avoir de relation (aucune transaction des quêtes est reliée aux autres),
- avoir des relations d'ordonnancement (au moins une transaction d'une quête est reliée à une transaction d'une autre quête),
- avoir des transactions communes.

Ceci nous amène à une classification plus raffinée. Considérons deux quêtes, Q1 et Q2. Notons D1, D2 et F1 et F2 respectivement les transactions initiales et finales de Q1 et Q2. Q1 est avant Q2 si, pour l'ordre  $f \rightarrow$  toujours F1 précède D2. En conséquence aucune transaction de Q2 ne peut être exécutée tant que la transaction de fin de Q1 n'est pas exécutée. Ceci

définit un ordre partiel  $R$  entre les quêtes d'un niveau. Si  $Q1$  et  $Q2$  ne sont pas ordonnées nous dirons que ces transactions sont parallèles.

Considérons un niveau comme un ensemble de quêtes ordonnées par  $R$ .  $R$  peut être représenté par un graphe orienté dans lequel les sommets sont des quêtes.  $Q1$  est en relation avec  $Q2$  si  $Q1$  est avant  $Q2$ .

Les quêtes d'entrée du niveau sont celles qui n'ont pas de prédécesseurs tandis que les quêtes de sortie du niveau n'ont pas de successeurs. Une solution possible du niveau sera un cheminement entre une quête d'entrée et une quête de sortie. Si une quête appartient à tous les cheminements du graphe, elle est obligatoire, sinon elle est facultative.

Dans la figure 3.26, la quête initiale du niveau est  $Q1$ , la quête finale est  $Q5$ .  $Q1$ ,  $Q3$  et  $Q5$  sont obligatoires dans le niveau.  $Q2$  et  $Q4$  sont facultatives.  $Q4$  et  $Q5$ , par exemple, sont parallèles.  $Q1, Q3, Q5$  est une solution du niveau,  $Q1, Q3, Q2, Q5$  en est une autre.

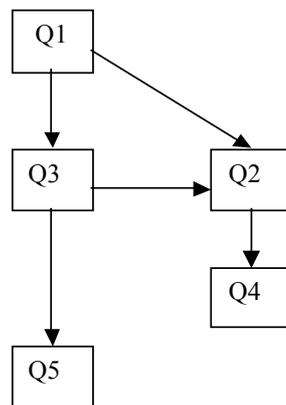


Figure 3.26 : Solution d'un Niveau

## 3.7. Exemple

### 3.7.1. Description du concept du jeu Myst

Nous présentons l'analyse du premier niveau du jeu d'aventure Myst. Nous effectuons ce travail en partant des modèles de RdP de base définis précédemment.

Ce jeu d'aventure est devenu un classique du genre. Toutefois, à la différence de la plupart de ces jeux, vous ne rencontrez pas de personnages non joueurs (PNJ) pour interagir ou vous renseigner. Vous êtes seul dans l'île pour découvrir l'énigme.

## Synopsis<sup>10</sup>

L'histoire tourne autour de certains livres mystérieux qui fonctionnent comme des portails pour des mondes fantastiques. Au début de cette aventure, le joueur se trouve dans l'île de Myst après avoir trébuché sur un de ces volumes. Myst, est la maison d'Atrus et de sa famille. Atrus a dominé l'art hérité de la civilisation D'ni de faire et d'utiliser les livres.

Une fois que le joueur commence à explorer l'île et à découvrir progressivement ses secrets (messages dispersés : lettres, journaux, et vidéos), il découvrira que quelque chose de très grave est arrivé à la famille d'Atrus. Sa mission consistera à trouver d'autres livres magiques pour effectuer le parcours dans les mondes qu'Atrus avait colonisé.

## Solution<sup>11</sup> du niveau 1

Au début le joueur est sur le quai de Myst. Il devra se promener sur toute l'île. Il devra trouver une lettre (FL) laissée à côté du chemin en bois. Après, il doit trouver (FLV1) et activer le premier levier (ALV1) pour trouver (FSP1) et accéder (ASP1) au passage secret sur le quai. Il devra descendre les escaliers pour trouver un code (FC1). A ce stade-là, le joueur pourra trouver un dispositif (FD1) et l'activer (AD1) avec des informations récupérés dans la lettre et le code. Le dispositif lui permettra d'accéder à une cinématique qui aide à la compréhension du jeu. Les images seront disponibles à travers le système d'affichage localisé dans le socle du bassin.

Le joueur devra repérer (FLVi, i=1..n) et activer le reste des leviers (ALVi, i=1..n) qui sont accessibles partout sur l'île. Le huitième levier, celui se trouvant à côté du phare, restera inaccessible dans ce niveau. Les leviers, LVi, sont des interrupteurs qui servent à activer certaines zones sur la carte accrochée à un des murs de la bibliothèque.

A partir de la carte, le joueur pourra orienter la tour d'observation (AMi, i=1..n) qui donne accès aux différents codes. Pour accéder à la tour, il devra trouver un deuxième passage secret (FSP2) qui est dissimulé derrière les étagères. Pour accéder aux codes (FCOi, i=1..n), il devra monter les escaliers en haut de la tour. Uniquement k leviers (4 dans Myst) parmi les 7 actifs donnent accès aux codes. Les autres leviers n'ont pas d'autres application. Les codes seront utiles pour les voyages dans les autres mondes.

Le joueur doit choisir au moins un des deux livres (rose ou bleu) à lire. Pour lire le livre (ou livres) il faut insérer la page qui se trouve à côté. Un message de Sirrus ou de Achenar, les fils de Atrus, (une deuxième ou troisième cinématique se déclenche selon le livre choisi). Le message qui au début n'est pas très clair s'améliorera au fur et à mesure que le joueur rapportera des feuilles correspondantes.

## Association des actions et constitution des quêtes (logique et géographique)

Au début du jeu, le joueur peut choisir d'activer le levier du quai, trouver et lire la lettre ou rentrer dans le passage. Toutefois, pour pouvoir visualiser la première cinématique, le joueur doit « entrer » dans la salle de projection à laquelle conduit le passage et effectuer les actions nécessaires pour activer le mécanisme de projection. Le joueur ne peut avancer au fond du passage que s'il a préalablement activé le levier du quai.

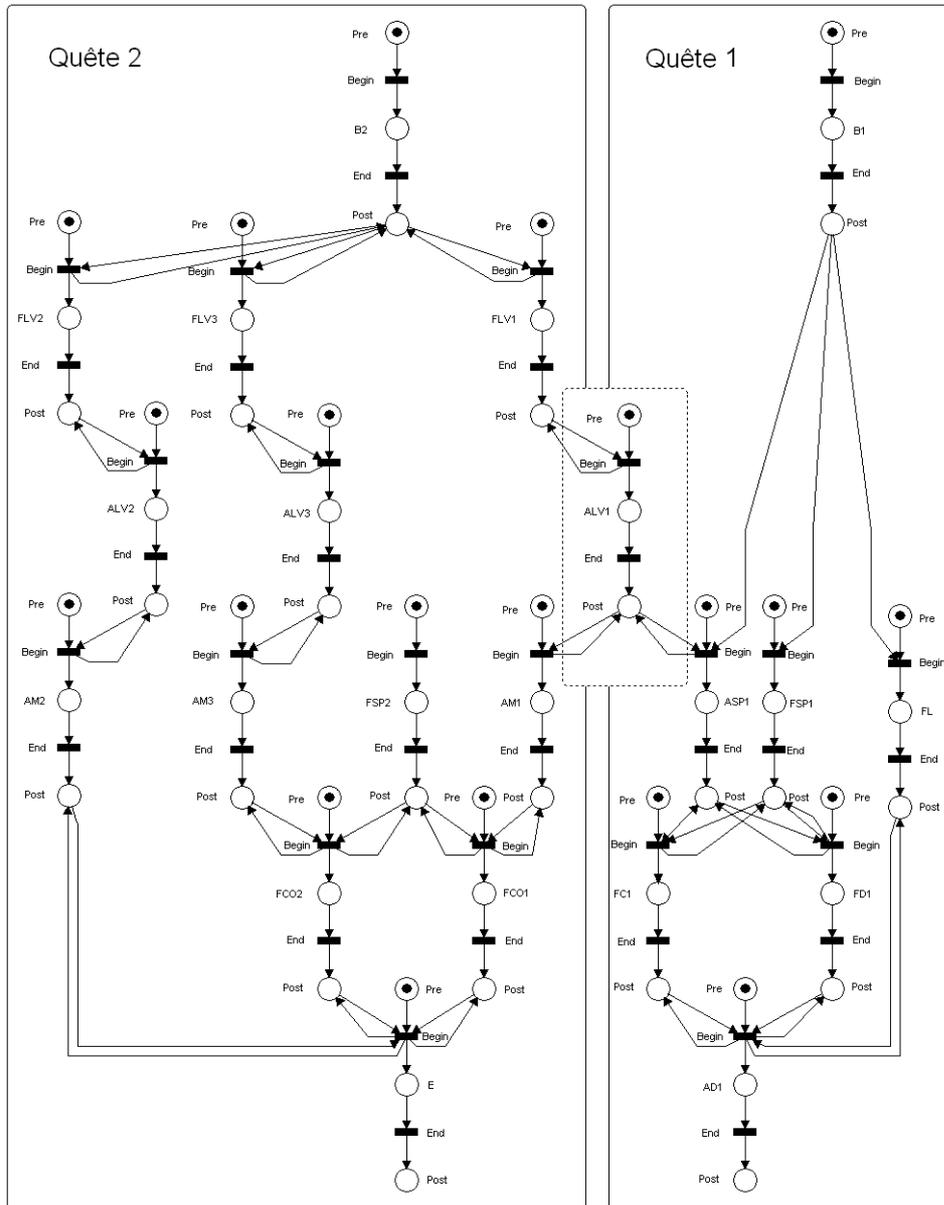
---

<sup>10</sup>Synopsis basé sur celle de [Wojciech Kotas](http://www.maclede.com/Reviews00/mystME/mystME.html), <http://www.maclede.com/Reviews00/mystME/mystME.html>

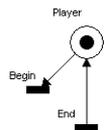
<sup>11</sup> Myst soluces, <http://213.186.38.133/~jeuxvide/soluces/consult.php3?idX=261>

### 3.7.2. Traduction de la structure en Réseaux de Pétri

Le Réseau de Quêtes suivant modélise la première partie du premier niveau du jeu. Notons que seul trois leviers sur sept et deux codes sur quatre sont modélisés pour que le réseau reste compréhensible.



Note:



Toutes les transitions partagent la place Player:

- Pour les transitions Begin elle est une pré condition.
- Pour les transitions End elle est une post condition.

Figure 3.27 : Réseau de quêtes de *Myst*

L'analyse du RQ à partir d'un point de vue sémantique nous amène à plusieurs conclusions. Cette première partie du niveau est composée par deux quêtes : la première correspond au niveau de connaissances du jeu du joueur, la seconde concerne les mécaniques du jeu et la progression du joueur. La quête 1 est facultative pendant que la quête 2 est obligatoire. De plus, la quête 2 est en même temps la quête initiale est la quête finale du niveau.

### 3.8. Conclusion

Le contenu de ce chapitre traite principalement de la formalisation des relations d'ordre des actions dans un jeu. Nous considérons que ces relations d'ordre peuvent contribuer selon un certain point de vue à l'analyse objective d'un jeu. Dans cet esprit, nous avons utilisé la description de la structure de la narration du jeu pour expliquer sa segmentation.

En considérant des structures narratives non déterministes, nous limitons l'application des modèles proposés aux jeux dont les caractéristiques fondamentales sont:

- Le jeu possède une histoire importante à raconter.
- Le joueur raconte l'histoire à travers les actions qu'il exécute.
- Les actions exécutées sont des choix non triviaux sélectionnés par le joueur. À d'autres occasions, ils sont le résultat d'un processus analytique.
- Les choix sont motivés par la réalisation d'un objectif prédéterminé.

Pour expliquer formellement la constitution de la structure d'un jeu, nous nous sommes basés sur les Réseaux de Pétri. Les modèles présentés expliquent les différents niveaux de relation identifiés entre les actions qui constituent cette structure. Notre analyse conduit à une définition de la structure dynamique d'un jeu.

Le modèle ne prend pas en considération quelques aspects importants comme :

- La possibilité de représenter des préconditions liées au temps. Ce problème peut être résolu avec des Réseaux de Pétri temporisés.
- La représentation du *game over*, du succès du joueur et d'autres statistiques.
- La représentation de la notion de localité, relation entre les séquences des actions et l'univers du jeu.

L'organisation des modèles présentés pour l'analyse induit à une méthode constructive qui pourrait être utilisée pour valider la conception du jeu.

# CHAPITRE 4

## *Intégration de la topologie dans la modélisation*

---

### **Contenu du Chapitre**

Nous présentons dans ce chapitre la notion d'espace dans la conception et l'écriture d'un jeu vidéo. Elle est en relation avec la description de l'ordre des séquences des transactions défini par la quête. En tant que mécanisme d'écriture, l'implémentation de cette notion définit l'espace physique de l'univers, les éléments qui le composent (objets) et leurs emplacements dans cet espace.

### **Organisation du chapitre :**

Section 4.1. Problématique de l'espace dans l'écriture des jeux – p. 80 : Introduction à la représentation de l'espace dans les mondes virtuels et sa relation avec la structure du jeu.

Section 4.2. Modélisation de l'espace d'un jeu – p. 87 : Présentation de trois modèles d'écriture spatiale. Présentation du processus pour l'insertion du modèle le plus pertinent dans la pratique industrielle.

Section 4.3. Conclusion – p. 102 : Explication de l'importance de la méthode proposée dans l'analyse du processus de conception d'un jeu.

# 4.1. Problématique de l'espace dans l'écriture des jeux

## Introduction

Dans ce chapitre, nous étudions et formalisons le concept de spatialisation comme méthode d'écriture d'un niveau de jeu. Plusieurs points de vue représentatifs sont évoqués comme points de départ pour établir la cadre de travail de ce principe.

Nous partons de la méthode utilisée dans l'industrie pour définir les éléments de base nécessaires à la conception. Rappelons que, durant la phase de *Game Design*, sont définis les niveaux (*levels*) du jeu et la navigation entre ces phases. Dans la phase de *level design*, les quêtes sont déterminées par le positionnement des objets qui constituent aides et obstacles à la résolution de la quête. Ces quêtes induisent un ensemble d'actions partiellement ordonnées que le joueur doit exécuter pour terminer le niveau.

Notre approche est particulièrement axée sur les relations spatio-temporelles dans les jeux. Les relations temporelles sont représentables par des réseaux de quêtes, présentés dans le chapitre précédent, qui définit les concepts de base et les opérateurs d'ordre nécessaires pour une modélisation par les réseaux de Pétri (RdP). En ce qui concerne la représentation des relations spatiales, nous utilisons un sur-ensemble de graphes, les hypergraphes, pour modéliser naturellement les modèles spatial et temporel.

Nous traitons deux problèmes concernant cette conception :

1. La description de l'espace en terme de logique de contraintes de cheminements.
2. L'emplacement des objets dans cet espace.

Ceci permet de construire un modèle apte à valider la cohérence des niveaux créés par rapport aux relations existantes entre l'ordre des transactions, l'emplacement des objets, la localisation du joueur et l'accessibilité aux lieux.

Le chapitre est organisé de la façon suivante : tout d'abord, nous commençons par définir les principaux concepts de la représentation de l'espace des mondes virtuels. Ensuite, nous présentons un ensemble de notions de base du modèle d'écriture pour les jeux vidéo. Dans un troisième temps, nous présentons formellement les différents modèles évolutifs que nous avons conçu. Enfin, la conclusion donne un aperçu général des possibilités et contraintes du modèle proposé.

#### 4.1.1. La représentation de l'espace dans les mondes virtuels

La description de l'espace dans un monde virtuel peut être traitée à différents niveaux : géométrique, cartographique, topologique ou sémantique. La représentation géométrique est utilisée dans la réalisation des jeux, par le biais des outils de modélisation 3D. Elle prend une importance essentielle quand l'espace du jeu doit être mis en correspondance avec un espace réel. C'est le cas des jeux mobiles utilisant la localisation (GPS - *global positioning system*). La cartographie du jeu est la vision de l'univers que pilote le joueur. Presque tous les jeux ont une carte dans leur interface. Mais cette carte est le résultat d'un processus d'écriture qui repose sur un niveau topologique lié à la sémantique du jeu en général et aux mécanismes de *gameplay* en particulier. Ce paragraphe précise de façon informelle cette approche.

Avant de traiter l'écriture dans l'espace, nous donnons une définition plus exacte de chacun des niveaux de représentation :

Le **niveau géométrique** renvoie à un positionnement précis dans l'espace. Dans ce niveau, la représentation de l'espace est une description géométrique. L'information traitée est décrite par rapport à un système de coordonnées. Les modeleurs 3D, comme 3ds Max ou Maya servent à la création de l'espace du jeu et de la géométrie des objets que contient cet espace. Ils utilisent des systèmes des coordonnées cartésiennes dans l'espace [Kerlow00]. L'intersection des trois axes définit l'origine ou point de référence (voir figure 4.5). Ce système sert de référence commune à tous les éléments de la composition. Il définit le système de coordonnées globales du monde à partir de l'origine. Ce système de coordonnées après transformation est la référence qui est utilisé dans l'algorithmique du jeu. Par exemple, les techniques de *pathfinding* ou de détection de collisions sont traitées à ce niveau de description de l'espace.

Le **niveau cartographique** fait référence à l'« *ensemble des techniques et des arts graphiques ayant pour objet la conception, la préparation, la rédaction et la réalisation de tous les types de plans ou de cartes.* » [Dico03]. La carte est une représentation symbolique de l'espace permettant, de façon visuelle, d'en déduire les relations topologiques et de proportions avec, en général, une relation d'échelle (isométrie). La conception d'un niveau de jeu repose sur une carte dont une vision simplifiée ou partielle est fournie au joueur pour lui permettre de s'orienter.

Le **niveau topologique** représente « [...] *des relations de voisinage qui s'établissent entre des figures géométriques, relations qui ne sont pas altérées par la déformation des figures.* » [Dico03]. Ce niveau détermine la possibilité de cheminement (connexité) dans l'espace du jeu à un instant donné.

Enfin, le **niveau sémantique** définit la géométrie par référence à un contexte sémantique.

Exemple : Supposons que la totalité de l'espace d'un niveau du jeu correspond au modèle 3D<sup>12</sup> montré dans la figure 4.1. Dans ce monde, la mission du personnage principal (magicien indiqué par le petit cercle) consiste à trouver le chemin qui mène au milieu du lac (indiqué

---

<sup>12</sup> Composition Virtools [Virtools04] disponible dans les fichiers des exemples des tutoriaux du logiciel (fichier : GridPathfindingDEMO.cmo).

par le grand cercle). Les figures 4.2 et 4.3 montrent respectivement la position initiale du magicien au début du niveau et le point d'arrivée à la fin du niveau.

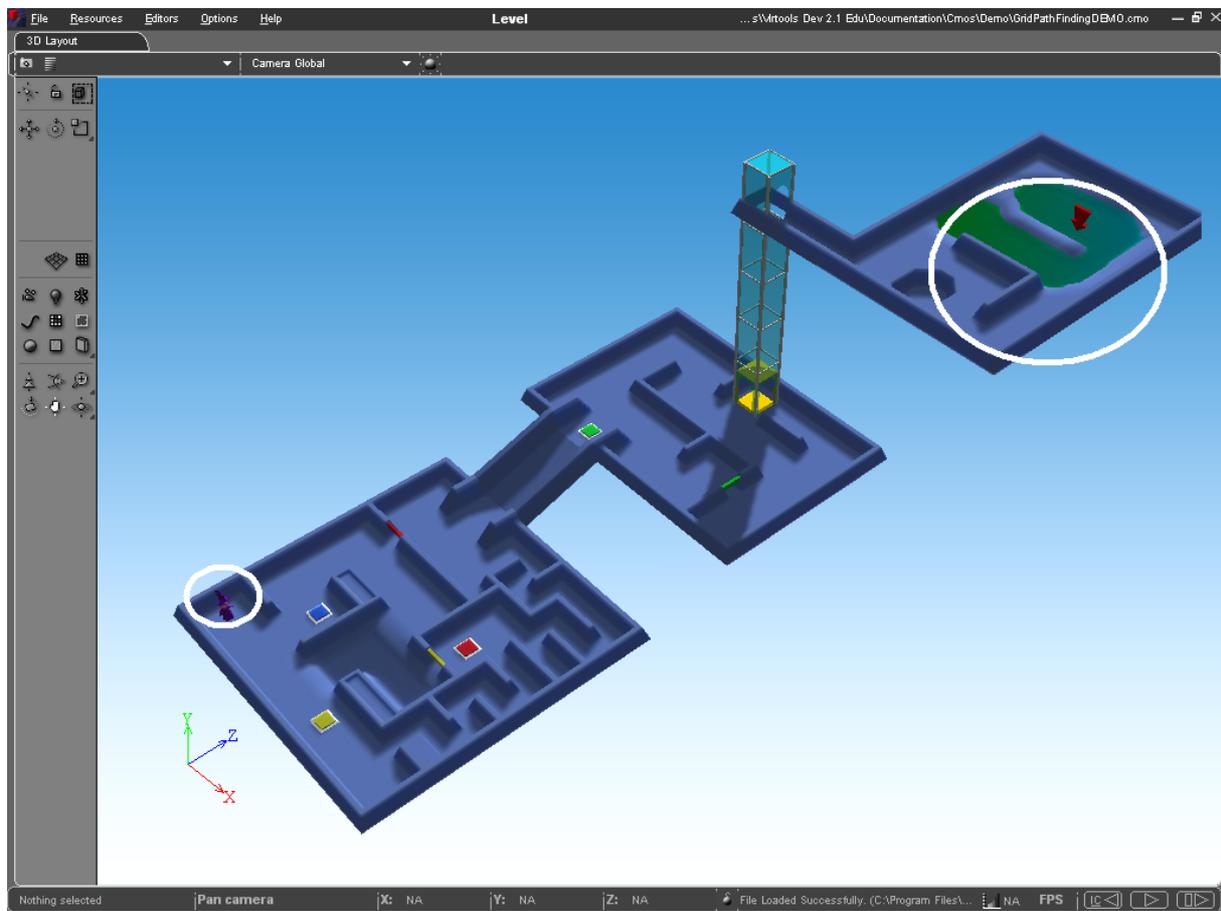


Figure 4.1 Composition Virtools disponible pour démonstration.

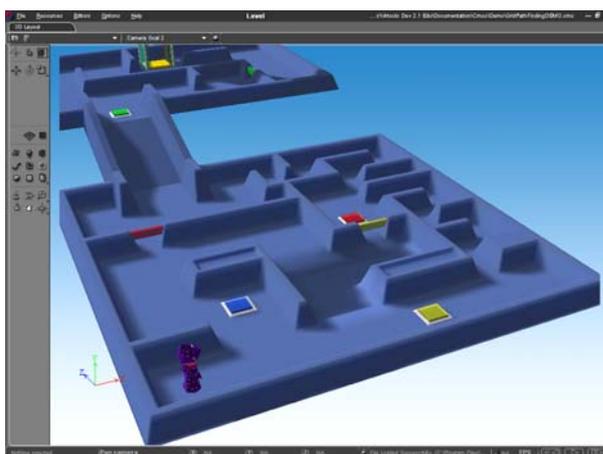


Figure 4.2 : Début du niveau.

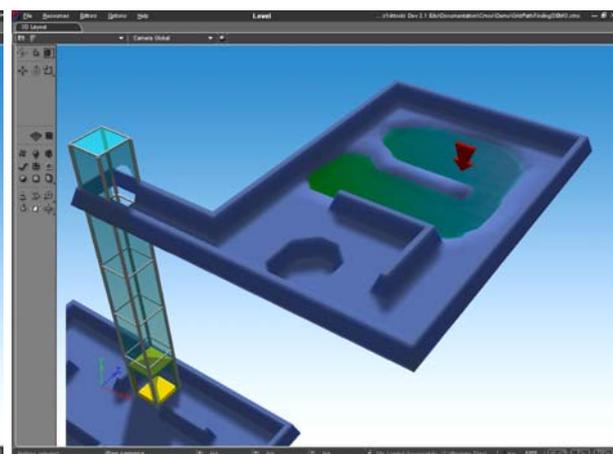


Figure 4.3 : Fin du niveau.

## Niveau géométrique

La description de l'univers à ce niveau correspond à la courbe  $f(x,y,z)$  de la figure 4.4. Chacune des coordonnées représente un point dans le système cartésien (voir figure 4.5).

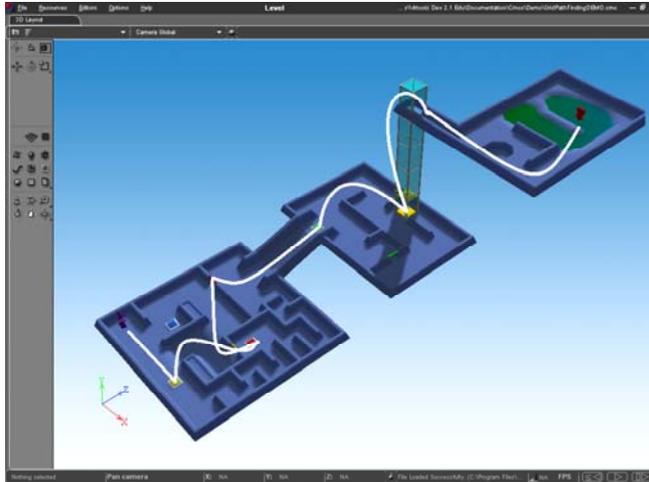


Figure 4.4 : Courbe  $f(x,y,z)$  décrivant le parcours le plus court du magicien.

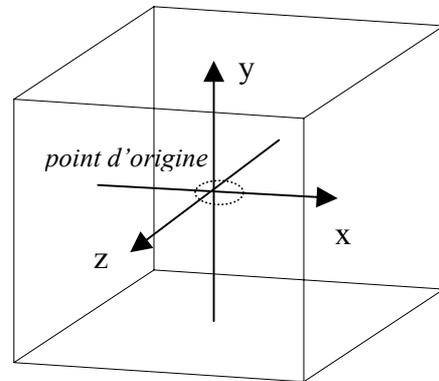


Figure 4.5 : Système cartésien (repère main droite).

## Niveau cartographique

La figure 4.6 montre le monde d'un point de vue isométrique (vue du dessus). Une représentation simplifiée du monde est donnée par la carte de la figure 4.7. Cette description cartographique est accompagné par une interface montrant la progression du joueur (barre colorée à droite de la carte).

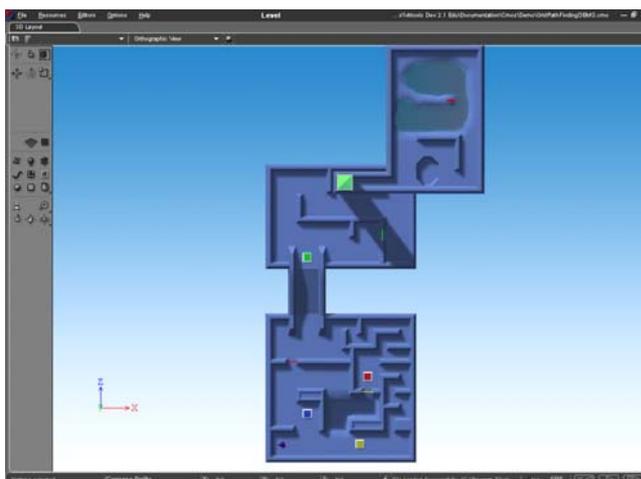


Figure 4.6 : Point de vue isométrique du niveau (top view)

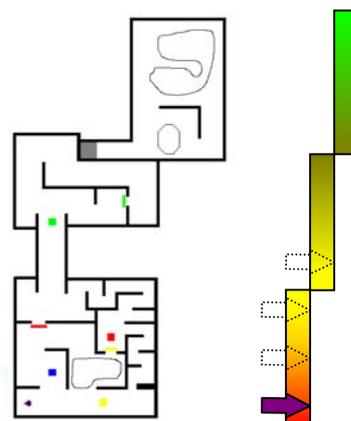


Figure 4.7 : Carte du niveau indiquant la progression du joueur

## Niveau topologique

Le niveau du jeu est constitué de trois zones principales (A, B, C) accessibles séquentiellement (figure 4.8).

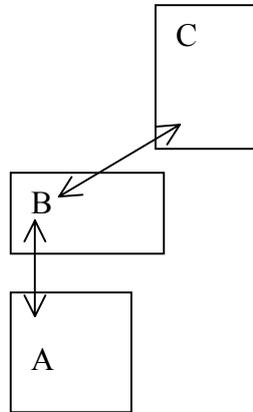


Figure 4.8 : Description topologique

## Niveau sémantique

La représentation de l'univers correspond à une modélisation en 3D du style « jeu de plate-forme ». A la sémantique du *gameplay* d'un jeu de ce type comprend une abstraction des éléments constituant le jeu : zones libres, obstacles, mécanismes d'activation et but du niveau.

L'utilisateur familiarisé aux jeux de plate-forme sait ce qu'il doit faire dans le niveau. Normalement, il existe plusieurs façons de progresser dans le niveau. La première consiste à activer tous les mécanismes possibles pour avancer rapidement et gagner le maximum de points. La seconde permet au joueur de progresser plus facilement (moins d'obstacles et de mécanismes avec lesquels interagir) mais propose un parcours plus long et moins amusant.

Une interprétation sémantique possible pour parcourir la zone C de cet exemple et finir le niveau est :

En sortant de l'ascenseur, continuer tout droit, passer à côté du puit et après être arrivé au mur, prendre à gauche jusqu'au lac. Traverser ensuite par le côté le moins profond pour regagner l'autre rive.

En ce qui concerne le modèle que nous proposons, l'information spatiale nécessaire doit exprimer de manière globale la distribution des zones qui constituent la carte et la manière dans laquelle celles-ci sont mises en rapport. De cette façon, nous pouvons établir les accès possibles dans la carte à un moment donné du jeu. C'est pourquoi nous avons besoin d'utiliser une représentation topologique dans notre modèle.

#### 4.1.2. La notion de structuration de l'espace du jeu selon Rollings

La structure narrative du jeu établit une relation entre les différents lieux de l'univers du jeu et les différents épisodes de l'histoire [Rollings03]. Le déroulement du jeu est lié à une progression dans l'espace. C'est la topologie de la carte du jeu qui définit les parcours possibles du joueur et détermine, en partie, les histoires qu'il peut expérimenter.

Dans les figures qui suivent, 4.9, 4.10 et 4.11, les graphes représentent les structures de différents jeux d'aventure [Rollings03]. Les sommets représentent des zones de l'univers du jeu et les arcs, les passages possibles entre ces zones. *D* est la zone d'entrée au début de la partie et *F* la zone de sortie qui marque la fin du jeu.

La figure 4.9 montre la topologie d'un jeu d'aventure privilégiant l'exploration du monde sur la narration. Les premiers jeux d'aventure, essentiellement textuels, utilisaient ce mode de fonctionnement. Le joueur pouvait passer son temps à explorer et découvrir le monde sans contrainte de temps qui l'obligeait à progresser dans la trame narrative.

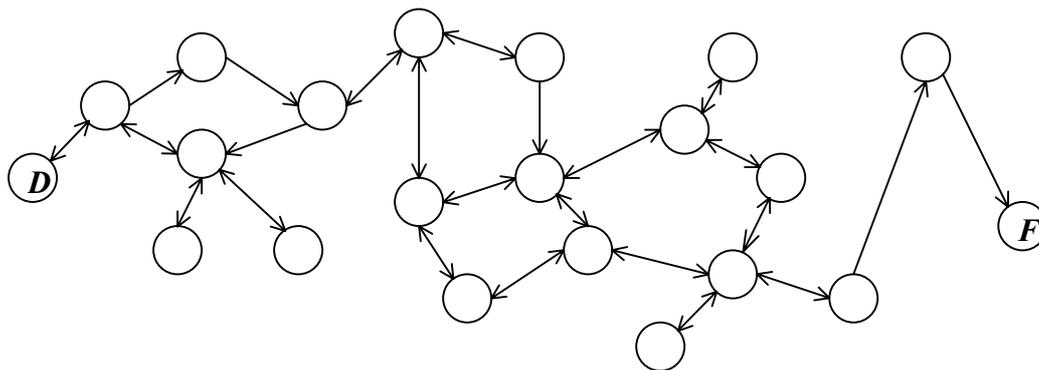
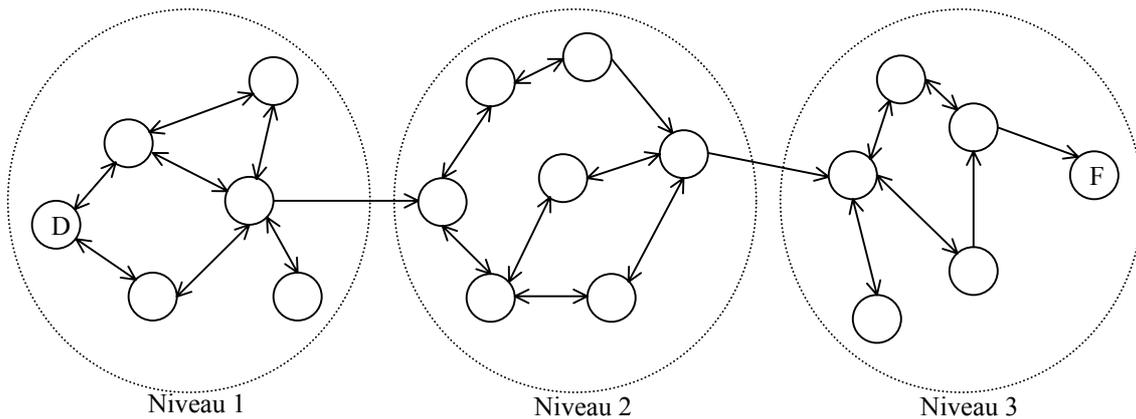


Figure 4.9 : Structure des premiers jeux d'aventure ( ang. *text adventure games*).

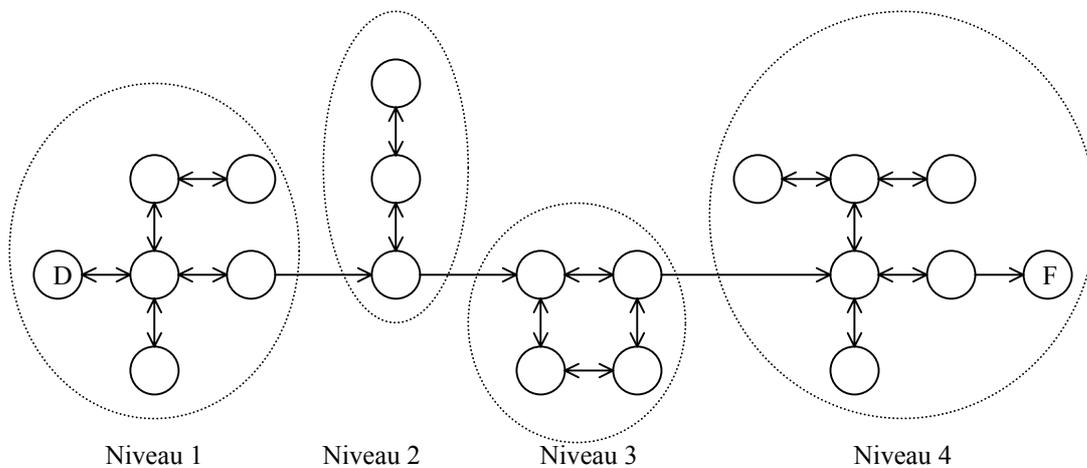
Selon Rollins, plus la structure du jeu est liée à une histoire, plus elle contraint le déroulement du jeu. Un découpage en chapitres détermine des points de convergence dans l'histoire. Ceux-ci deviennent des passages obligés pour progresser dans le jeu et constituent, au sens donné dans cette thèse, des niveaux.

Les figures 4.10 et 4.11 sont extraits de Rollings qui considère que la topologie est le mode directeur essentiel de la narration dans les jeux [Rollings03]. Donc, les composantes fortement connexes des graphes constituent les niveaux. A l'intérieur du niveau, le joueur a plusieurs possibilités pour réaliser les parcours définis par les passages possibles entre les zones. Un joueur ne peut quitter le niveau que s'il est terminé. Le joueur ne peut plus y revenir, passe au suivant et ainsi de suite jusqu'au dernier niveau du jeu.



**Figure 4.10 : Structure de jeux d'aventure conduits par l'histoire.**

Selon Rollins, dans les jeux d'action, la réactivité devient un facteur essentiel. Le monde (ses décors et parcours) et l'histoire restent simples pour que le joueur puisse instinctivement effectuer ses déplacements, attaques, fuites ou autres actions nécessaires à sa survie. La Figure 4.11 présente un exemple d'une structure des jeux d'aventure-action ou des FPS.



**Figure 4.11 : Structure des jeux d'aventure**

Dans le chapitre 2, nous avons présenté les techniques de construction basées sur l'espace : notion de labyrinthe de Natkin et écriture dans l'espace de Aarseth. Ensuite, dans le chapitre 3, nous avons présenté une méthode d'analyse fondée sur la structure du jeu et la description des quêtes. L'aboutissement de ces quêtes dépend de l'utilisation de certains objets du jeu. Par exemple, dans le réseau de quête initial, le joueur doit trouver un bouclier et une épée dans un endroit quelconque pour ensuite affronter le dragon. La localisation de ces objets doit rester dans les zones accessibles pour le joueur, dans le cas contraire le jeu sera bloqué. Si l'espace du jeu est tel qu'il faut d'abord passer par la salle où se trouve le bouclier puis, par le couloir où se trouve l'épée, la topologie contraint l'ordre des transactions. D'un autre point

de vue, nous pouvons considérer que certaines transactions doivent être accomplies ou non pour pouvoir ouvrir une porte qui mène d'un lieu à un autre. En ce sens, le graphe de niveau altère la topologie.

Dans les prochains paragraphes, nous formalisons cette méthode d'écriture dans l'espace. Nous combinons les mécanismes d'ordre induits par la structure des quêtes, présentés au chapitre précédent et une description dynamique de la topologie. En outre, nous montrons un modèle permettant la mise en correspondance entre la structure du jeu, la topologie du niveau et les objets requis par les quêtes.

## 4.2. Modélisation de l'espace d'un jeu

### 4.2.1. Un modèle permettant de modéliser l'évolution de la topologie en fonction des transactions

La conception de la plupart des niveaux est basée, comme nous l'avons vu, sur une écriture spatiale. Nous supposons que le niveau est représenté, soit par un hypergraphe soit par une carte que l'on peut déduire de cet hypergraphe.

Un hypergraphe est un couple  $H(V,E)$ ,  $V$  est un ensemble de sommets et  $E = (e_1, e_2, \dots, e_m)$  un ensemble de parties de  $V$ . Les éléments de  $V$  de ce graphe sont appelés les sommets et ceux de  $E$ , les arcs. Un hypergraphe est orienté si les éléments de  $E$  sont eux recouvert par deux sous ensembles,  $e_i = (h_i \cup t_i)$ ,  $h_i$  est appelé origine de l'arc (*head*) et  $t_i$  l'extrémité de l'arc (*tail*). Par abus de notation, nous noterons un arc  $e_i = (h_i, t_i)$  et nous supposerons que les ensembles  $X_i$  et  $Y_i$  comprennent les éléments  $h_i$  et  $t_i$  respectivement.

Dans le cadre d'un jeu et en reprenant les conventions de Rollings, l'espace du jeu peut être représenté par un graphe topologique dont les sommets sont des zones sans obstacles et les arcs indiquent la possibilité, à certaines étapes du jeu, de cheminer d'un sommet vers un autre. Les sommets de l'hypergraphe sont constitués par les sous-graphes du graphe topologique qui sont fortement connexes à toute étape du jeu. Lorsque le joueur se trouve sur une des composantes du graphe, il peut, quelle que soient les quêtes qu'il a ou n'a pas réalisées, atteindre tous les sommets du sous-graphe topologique qui est un des sommets de l'hypergraphe. Selon la terminologie de Flury [Flury03], un sommet de l'hypergraphe est un des locus du jeu. L'ensemble des locii, donc des sommets de l'hypergraphe, est statique. Notons  $H(V)$ , l'ensemble des hypergraphes ayant un ensemble de sommets  $V$  donné.

L'accessibilité dans un jeu étant une relation dynamique, un hypergraphe va représenter les cheminements possibles à une étape du jeu. Un arc  $E_i = (X_i, Y_i)$  détermine un ensemble de locii finaux  $Y_i$  accessibles à partir des locii initiaux  $X_i$ . L'évolution dynamique est construite en utilisant la notion de remplacement qui, à un instant donné, substitue à un arc de l'hypergraphe un autre arc. Un remplacement est donc une application de  $H(V)$  dans  $H(V)$  qui modifie les arcs de l'hypergraphe. Un remplacement  $r$  est noté  $r = \{(a|b), (c|d), \dots\}$   $a$  et  $c$  sont

les arcs avant remplacement. Ils sont respectivement remplacés par  $b$  et  $d$ . Les arcs peuvent être éventuellement vides, ce qui permet de créer et de détruire dynamiquement les chemins.

Formellement, les règles de remplacement sont conditionnées par un réseau de transactions qui modélisent le *gameplay*. A un instant donné, la suite des transactions effectuées par le joueur se traduit par un chemin dans le graphe d'accessibilité réduit du réseau de transaction, qui est également un mot fini d'un langage dont les lettres sont les noms des transactions.

Une connexion associée à chaque marquage d'un réseau de transactions un remplacement éventuellement vide. Par exemple, à une place du joueur marquée, il existe une transaction  $i$  exécutable (i.e.  $\text{Begin}_i$  est tirable)

Nous définissons donc un modèle de jeu par :

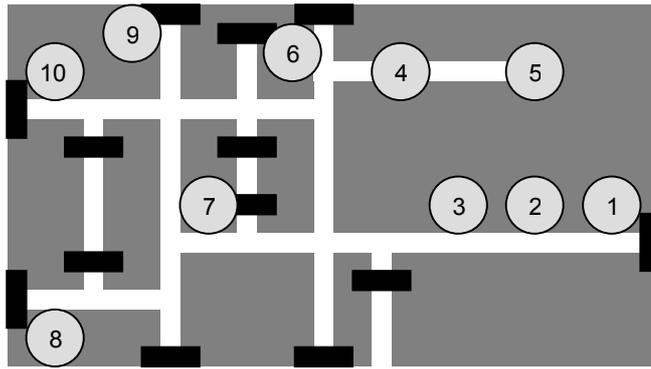
- Un réseau de transactions  $R$ ,
- Un hypergraphe initial  $H_0$  dont les sommets sont notés  $V$ ,
- Un ensemble de remplacements RMP défini sur  $H(V)$
- Un ensemble de connexions  $n_{i,M}$  où  $M$  est un marquage accessible de  $R$  tel que la place du joueur soit marquée et  $i$  une transaction tel que  $\text{Begin}_i$  soit tirable en  $M$ ,  $a$  et  $b$  étant deux sous ensembles éventuellement vides de  $V$ .

La sémantique de simulation du jeu que nous adoptons est la suivante : à une étape donnée l'hypergraphe courant est  $H$ , le marquage est  $M$ . Une transaction  $i$  de  $R$  est exécutée. Si  $a$  est un hyperarc de  $H$  et  $b$  n'en est pas un, appliquons  $n_{i,M}(a|b)$ . Ceci donne un nouvel hypergraphe  $H_{i+1}$  qui définit les locii accessibles au joueur. Une telle étape est atomique et « instantanée ».

Cet algorithme sera explicité dans une forme plus générale au paragraphe suivant.

Pour illustrer ce modèle, prenons par exemple le jeu Silent Hill 2 [Konami02] du genre « horreur-aventure ». Le scénario du jeu se déroule dans une ville presque déserte. Le joueur contrôle le personnage de James Sunderland, un jeune homme qui revient dans un lieu dénommé Silent Hill après avoir reçu une lettre mystérieuse de son épouse décédée qui lui demande de la rencontrer là-bas. Dans cette ville se passe d'étranges événements et les habitants qui la peuplent sont soit des déséquilibrés mentaux, soit des monstres.

La carte de Silent Hill 2 correspondant à la première partie du jeu est montrée dans la figure 4.12. Les bâtiments sont indiqués par les zones en gris et les rues par les zones en blanc. Les zones en noir signalent des impasse. Une description des zones est associée aux sommets de la carte pour mieux comprendre l'exemple.



Zones (locii) à explorer :

- 7. Parking
- 8. Fontaine
- 9. Eglise
- 10. Jardin
- 11. Tunnel
- 12. Monument
- 13. Bar
- 14. Caravane
- 15. Porte d'appartement
- 16. Appartement

Figure 4.12 : La carte de Silent Hill 2

Le modèle pour cette partie du scénario du jeu est donné par le réseau de transaction de la figure 4.13, les hypergraphes des figures 4.14, 4.15 et 4.16 ou les remplacements et connexions sont décrits dans la suite.

La liste des transactions à exécuter pour finir le premier niveau est présenté ci-dessous. Pour faciliter la lecture du graphe de R, au lieu d'indiquer le nom de chaque élément des transactions, nous les avons identifiés par une lettre de l'alphabet (en minuscule):

- (a) Récupérer la carte dans la voiture de James. Ensuite partir vers Silent Hill.
- Une fois dans la ville, (b) gagner le combat contre le monstre pour pouvoir continuer.
- Après avoir vaincu le monstre, James peut réaliser les quatre actions suivantes, dans n'importe quel ordre :
- (c) Récupérer l'inscription du monument étrange.
- (d) Regarder la carte dans la caravane.
- (e) Trouver une deuxième carte dans le bar Neely.
- (f) Récupérer la clé de l'appartement.

(c), (d) et (e) sont des transactions indépendantes du point de vue de l'ordre et facultatives (leur ordre d'exécution n'est pas imposé et n'a pas de conséquence dans la réalisation de la quête).

(f) est obligatoire et détermine la fin du premier niveau du jeu. Après ceci, James peut (g) rentrer dans l'appartement et le joueur commence le niveau suivant.

Dans les hypergraphes (figures 4.14, 4.15, 4.16 et 4.17), les cercles modélisent les sommets (locii) et les flèches les arcs. Le joueur est libre de visiter les locii connectés par les arcs. Nous utilisons des lettres en majuscule pour identifier les arcs facilement.

Le jeu commence dans les toilettes d'un parking aux alentours de la ville (locus 1). Le joueur peut partir à l'aventure uniquement après avoir examiné la voiture et récupéré le plan de la ville. Tant que cette action n'est pas réalisée, il est obligé de rester dans le parking. Cette situation est modélisée par le fait que le sommet 1 de l'hypergraphe dans la figure 4.14 est isolé.



De la même manière, après avoir vaincu le monstre dans le tunnel (la transaction (b) a été exécutée), la topologie de la carte est modifiée permettant au joueur de retourner au jardin ou d'aller au monument, à la grille de l'appartement, au bar ou à la caravane.

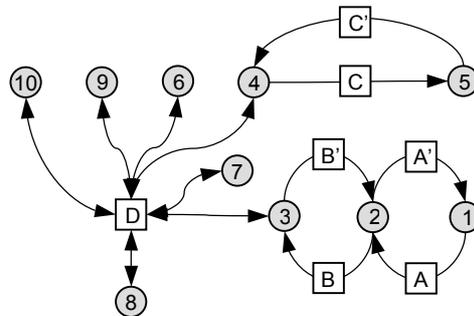
Nous modélisons ces possibilités par le remplacement de l'hyperarc D:

$$r_2 = \{ (\{\}, \{\}) | (\{5\}, \{4\}), (\{3,4\}, \{3,4\}) | (\{3,4,6,7,8,9\}, \{3,4,6,7,8,9\}) \}$$

L'hypergraphe de la figure 4.16 montre la nouvelle topologie après l'application des connexions correspondantes.

Enfin, après avoir récupéré la clé, le joueur a accès à l'appartement (dernier remplacement en fonction de la transaction (f)). La figure 4.17 donne la nouvelle topologie obtenue par application des connexions. Le remplacement effectué est:

$$r_3 = \{ (\{3,4,6,7,8,9\}, \{3,4,6,7,8,9\}) | (\{3,4,6,7,8,9,10\}, \{3,4,6,8,9,10\}) \}$$



**Figure 4.17 : Nouvelle hypergraphe après avoir récupéré la clé de l'appartement.**

Le modèle que nous avons présenté permet de modéliser l'évolution de la topologie du jeu en fonction des transactions exécutées par le joueur. En revanche, nous ne pouvons pas représenter le fait que l'exécution d'une transaction dépende de l'accès à certains objets qui sont localisés dans l'espace et donc pas toujours accessibles. Nous allons étendre notre modèle pour prendre en compte cet aspect en utilisant les réseaux de Petri Colorés [Jensen81].

#### **4.2.2. Un modèle permettant de faire évoluer le réseau de Pétri en fonction de la topologie**

##### **Les réseaux de Pétri colorés**

Dans ce paragraphe nous présentons une définition informelle et limitée des réseaux colorés. Elle sera suffisante pour l'usage que nous en faisons. Le lecteur pourra se reporter à [Diaz01, Jensen81] pour une présentation plus détaillée.

Nous appelons ensemble de couleur un ensemble fini. Dans notre exemple les couleurs sont associés aux lieux possibles d'exécution d'une transaction, par exemple  $V = \{ \text{parking, fontaine, église, bar, caravane, tunnel, jardin, monument, appartement, porte d'appartement, noir} \}$ . Dans un réseau coloré, les jetons (marques) sont colorés et le marquage n'est plus une application de  $P$  dans  $N^{|P|}$  mais une application  $P$  dans l'ensemble des applications de  $V$  vers  $N$ . Un marquage est constitué du nombre de jetons de chaque couleur qui sont contenus dans chaque place. Nous distinguons une couleur privilégiée, le noir par exemple, qui est utilisée pour les jetons « ordinaires ».

Dans notre application, la distinction des jetons nous permet de tracer les déplacements du joueur dans la topologie. Les places qui contiennent un jeton qui n'est pas noir sont les places internes des transactions (i.e. Place A de la transaction A) et la place du joueur (Player). La couleur de A, quand la transaction est en cours, est le locus d'exécution. La couleur courante de la marque de la place du joueur est le locus de la dernière transaction exécutée.

Dans les réseaux colorés, à chaque transition  $t_i$  est associée un sous ensemble de  $w(i)$  qui détermine les modalités de franchissement de  $t_i$ .

Dans notre application nous commençons par fixer un des lieux  $l$  où une transaction A peut être exécutée (e.g. tuer un monstre), par exemple  $l$  peut être le tunnel ou le parking.  $l$  étant fixé, pour pouvoir tirer la transition  $\text{Begin}_A$ , le joueur doit être localisé dans un locus qui permet d'accéder au lieu de la transaction  $l$ . Notons  $\text{antécédent}(l)$  l'ensemble de ces locii à une étape donné du jeu. Par exemple, supposons qu'à une étape du jeu, l'hypergraphe topologique (voir figure 4.18) donne  $\text{antécédent}(\text{parking}) = \{ \text{parking, appartement} \}$  et  $\text{antécédent}(\text{tunnel}) = \{ \text{tunnel} \}$ . Les modalités de  $\text{Begin}_A$  sont constituées d'un couple  $(l, l')$  où  $l$  est le locus d'exécution de A et  $l'$  un élément de l'ensemble des locii antécédent de  $l$ . Par exemple si  $l = \text{tunnel}$ ,  $l' = \text{tunnel}$ , si  $l = \text{parking}$ ,  $l' = (\text{tunnel, appartement})$ . Franchir la transition selon cette modalité signifie que le joueur est allé de  $l'$  à  $l$  pour exécuter la transaction A en  $l$ .

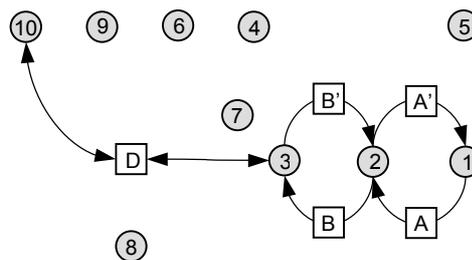


Figure 4.18 : Hypergraphe pour tuer le monstre.

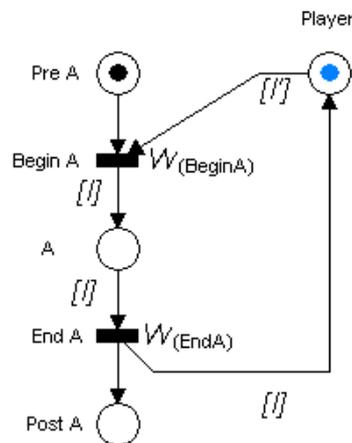
A chaque modalité  $m$  de  $w(i)$ , à chaque couleur et à chaque arc  $(p_j, t_i)$  du réseau de Petri est associé un nombre  $C^-(m, c, j, i)$ . Ce nombre représente le nombre de jetons de couleur  $c$  que doit contenir  $p_j$  pour que  $t_i$  soit franchissable dans la modalité  $m$ . Il représente également le nombre de jetons de couleurs  $c$  qui sont ôtés de la place  $p_j$  quand  $t_i$  est franchie dans la modalité  $m$  (pré-condition). De même, à chaque modalité  $m$  de  $w(i)$ , à chaque couleur et à chaque arc  $(t_i, p_j)$  du réseau de Pétri est associé un nombre  $C^+(m, c, j, i)$ . Ce nombre représente le nombre de jetons de couleur  $c$  qui est ajouté de la place  $p_j$  quand  $t_i$  est franchie

dans la modalité  $m$  (post condition). Quand les modalités et les couleurs d'une des fonctions d'incidence ne sont pas précisées, se sont les jetons noirs qui sont ajoutés ou retirés.

Dans notre application, supposons que la transaction est exécutée en  $l$  et que le joueur est initialement en  $l'$  (couleur du jeton dans la place joueur). La pré condition de Begin est donc que la place du joueur contienne un jeton d'une couleur  $l'$ , qui par définition de  $w(\text{BeginA})$  appartient à  $\text{antécédant}(l)$ . Le joueur doit donc être localisé dans un locus qui permet de rejoindre  $l$  et d'exécuter la transaction.

La post condition de BeginA consiste à rajouter dans la place A un jeton de couleur  $l$ , lieu de la transaction.

Les modalités de EndA sont les locii possibles A. A doit donc contenir un jeton de cette couleur, qui lors du franchissement de EndA, est ôté et ajouté à la place du joueur. Ceci détermine sa nouvelle localisation.



**Figure 4.11 : Modèle de base d'une transaction en Réseau de Pétri Coloré.**

Pour montrer la construction d'un modèle de transaction en RdP coloré, nous avons adapté l'exemple de Silent Hill 2 utilisé précédemment. Nous supposons une topologie très simple avec un espace constitué de trois zones : un appartement, un parking et un tunnel. Dans notre scénario, une même action peut se passer dans deux endroits différents : le joueur doit tuer un monstre dans le tunnel ou dans le parking. Pour modéliser ces situations, nous devons prendre en compte les différentes possibilités de la topologie. Supposons que, dans l'état courant, le tunnel soit isolé et que l'on puisse accéder au parking à partir uniquement de l'appartement. Soit la figure suivante :

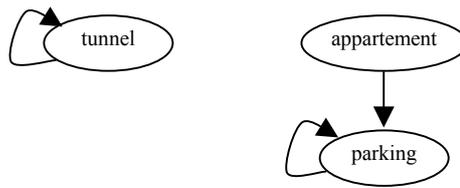


Figure 4.20 : Scénario réduit de Silent Hill 2.

Donc, si le joueur est initialement dans le tunnel, le lieu du drame et les antécédents sont le même : le tunnel. Dans le second cas, le lieu du drame est le parking mais les antécédents sont l'appartement (si le joueur vient de là) et le parking (s'il s'y trouve déjà).

Le modèle de la transaction est constitué de l'ensemble Couleur = { appartement, parking, tunnel, noir}, de l'ensemble lieux  $l \in \{\text{tunnel, parking}\}$  et de l'ensemble des antécédents de lieux : antécédent (tunnel) = {tunnel}, antécédent (parking) = {appartement, parking},  $W_{(\text{BeginA})} = \{(\text{tunnel, tunnel}), (\text{parking, parking}), (\text{parking, appartement})\}$ ,  $W_{(\text{EndA})} = \{\text{tunnel, parking}\}$ .

Supposons qu'au début de cette petite aventure, le joueur est dans le tunnel. Il n'est peut aller nul part tant qu'il n'a pas vaincu le monstre. A cette étape du jeu, Player=tunnel, Pre<sub>i</sub>=noir, et modalité de BeginA =(tunnel, tunnel), qui fait partie de  $W_{(\text{BeginA})}$  alors franchissable. Quand BeginA est franchie, le jeton noir de PreA et le jeton tunnel de Player sont retirés et un jeton tunnel est ajouté à A. Ceci signifie que le joueur est en train de tuer le monstre dans le tunnel. Quand EndA est franchie, le jeton tunnel de A est retiré, un jeton noir est ajouté à PostA et un jeton tunnel est ajouté à Player. Ceci veut dire que la transaction A, tuer le monstre, a été exécuté dans le tunnel.

Les RdP colorés ajoutent des nouvelles informations aux nœuds et aux arcs permettant de définir une sémantique de fonctionnement qui tient compte de celles-ci (dans notre cas, la topologie du jeu). Cette abréviation peut être dépliée en un réseau de Petri ordinaire (voir figure 4.21). Cependant, pour distinguer plusieurs instances d'un même état, les places et les transitions se dupliquent. Chaque place du réseau coloré se déplie en autant de places que le nombre de couleurs de jetons que peut contenir cette place. Chaque transition du réseau coloré se démultiplie selon le cardinal de ses modalités [Diaz01].

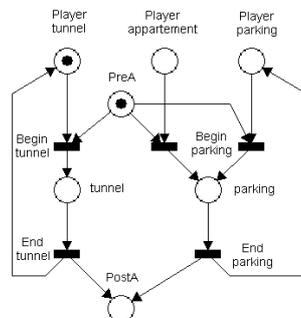


Figure 4.21 : Modèle de la transaction A (tuer le monstre dans le tunnel ou dans le parking) déplié en RdP ordinaire.

### 4.2.3. Un modèle d'un jeu avec localisation et dépendance réciproque des transactions et de la topologie

Dans ce paragraphe nous formalisons le modèle présenté au paragraphe précédent.

Nous définissons donc un modèle de jeu par :

- Un hypergraphe initial  $H_0$  dont les sommets sont notés  $V$ ,
- Un réseau de transaction  $R$  coloré non autonome (voir figure 4.19) défini de la façon suivante :
  - o L'ensemble des places et des transitions sont définis comme au chapitre 3
  - o  $\{V, \text{noir}\}$  est l'ensemble des couleurs
  - o Lieu est une application de l'ensemble des transactions de  $R$  dans les parties de  $V$ , noté  $P(V)$ .  $C$ 'est le lieu d'exécution de la transaction  $i$  et l'ensemble des modalités de franchissement de  $\text{End}_i$ .
  - o  $\text{WBEGIN}$  est une application de  $\text{BEGIN}_i \times H(V) \times V$  dans  $V \times P(V)$  ou  $\text{BEGIN}$  est l'ensemble des transitions de la forme  $\text{Begin}_i$ .  $\text{WBEGIN}$  est défini de la façon suivante : à une transition  $\text{Begin}_i$ ,  $H(V,E)$  un élément de  $H(V)$  et  $x$  un sommet de  $\text{Lieu}(i)$ ,  $\text{WBEGIN}$  fait correspondre le couple  $(x, y)$ , où  $y \in \text{antécédent}(x)$  l'ensemble des antécédents de  $x$  dans l'hypergraphe  $H(V,E)$ .  $\text{WBEGIN}$  est l'ensemble des modalités de franchissement de  $\text{Begin}_i$ .
  - o  $\forall l \in \text{Lieu}(i), \forall (l,l') \in \text{WBEGIN}(\text{Begin}_i, H(V,E), l), C^-(l,l', l', \text{Player}, \text{Begin}_i) = 1$
  - o  $\forall l \in \text{Lieu}(i), \forall (l,l') \in \text{WBEGIN}(\text{Begin}_i, H(V,E), l), C^-(l,l', \text{noir}, \text{Pre}_i, \text{Begin}_i) = 1$
  - o  $\forall l \in \text{Lieu}(i), \forall (l,l') \in \text{WBEGIN}(\text{Begin}_i, H(V,E), l), C^+(l,l', l, P, \text{Begin}_i) = 1$
  - o  $\forall l \in \text{Lieu}(i), C^-(l, l, P, \text{End}_i) = 1$
  - o  $\forall l \in \text{Lieu}(i), C^+(l, l, \text{Player}, \text{End}_i) = 1$
  - o  $\forall l \in \text{Lieu}(i), C^+(l, \text{noir}, \text{Post}_i, \text{End}_i) = 1$
  - o Le marquage initial de la place  $\text{Player}$  est un jeton dont la couleur est un élément de  $V$
  - o Le marquage initial des place  $\text{Pre}_i$  est un jeton noir
  - o Le marquage initial des autres place est nul.
- Un ensemble de remplacements  $\text{RMP}$  défini sur  $H(V)$
- Un ensemble de connexions  $n_{M}(a|b)$  comme défini au paragraphe précédent.

Ce modèle définit l'exécution du jeu selon l'algorithme suivant, qui évalue tous les états possibles du jeu, comme une extension d'un graphe de marquage. Un état est constitué du marquage proprement dit, de l'hypergraphe courant et la dernière transaction exécutée.

Début

Etat\_non\_évalué :=  $\{(M_0, H_0, \emptyset)\}$

Tant Etat\_non\_évalué n'est pas vide

Choisir un élément Etat\_courant =  $(M, H, t)$  dans Etat\_non\_évalué

```

    Soit R le réseau et M son marquage
    Tant que dans R il reste une transaction i tel que Begini soit franchissable
      Pour toutes les transitions tel que Begini soit tirable
        Tirer Begini, Tirer Endi
        Soit M' le marquage atteint
        calculer_replacement_hypergraphe(H,i,H') ;
        Etat_non_évalué := Etat_non_évalué ∪ {M',H',i} ;
      Fpour
    Ftant_que
    Etat_non_évalué := Etat_non_évalué - {M,H,t} ;
  Fin

```

Procédure calculer\_replacement\_hypergraphe(in H,in i,out H')

```

  H_courant:=H
  S'il existe une connexion  $n,M$  substituant b à a et si a est un hyperarc de H et b
  n'en est pas un, H' := rM(H)
  Sinon H' := H
  Fsi
  Ftant que

```

#### 4.2.4. Mise en œuvre du modèle précédent

L'idée principale que nous présentons dans les prochains paragraphes consiste à montrer que le modèle précédent s'insère naturellement dans la méthode utilisée dans l'industrie pour concevoir et programmer les niveaux (notion de labyrinthe introduit dans le chapitre 2). Elle permet d'obtenir une représentation telle que le concepteur soit capable d'identifier des cheminements possibles d'un point de vue topologique et de positionner correctement les objets qui contribuent à une énigme dans une zone de la carte.

##### *Etape 1 : Game Design*

Nous supposons donc que le Game Design est matérialisé par :

- La conception des classes d'objets existant dans l'univers. Appelons O l'ensemble des classes d'objets de l'univers. Un élément de O est lui même un ensemble puisqu'il s'agit d'une classe d'objet (un animal, un habitant). Pour créer un objet, il faut éventuellement spécialiser la classe (un mouton), instancier les paramètres de la classe (blanc) et créer un nouvel objet (nouveau\_mouton(blanc)). Ceci sera fait lors de la conception des niveaux.

Dans notre optique, les méthodes d'un objet qui peuvent être appelées à la suite d'une action du joueur définissent des transactions : l'objet épée peut être pris, agité, cassé. Parmi les objets, certains ont un statut particulier. Ils permettent de modifier la topologie (nous les appelons objets de modification topologique). Ce peut être par exemple une porte, un mur, etc... Selon les cas, les méthodes d'un objet de modification topologique, donc les transactions associées, peuvent être : ouvrir, fermer ou les deux. Ces classes

d'objet dépendent de paramètres qui, quand ils sont instanciés, déterminent les zones géographiques qui sont mises en relation.

- La conception des classes de quêtes qui vont servir de mécanisme de base au *game play*. Une classe de quête peut être par exemple :

Vaincre au moins  $n$  petits monstres pour gagner de l'argent, acheter un nouveau type d'arme et vaincre un gros monstre avec la nouvelle arme.

Une telle classe de quête comporte d'une part un ordonnancement d'actions, représentable par un réseau de quête et un ensemble d'objets qui sont instanciés lors de la création de la quête proprement dite. Par exemple, la classe de quête permet de créer des instances qui dépendent du type de petits monstres et de gros monstre, de la valeur de  $n$ .

Formellement une classe de quête CQ est définie par :

- Q Un réseau de quête
- O Un ensemble de classes d'objets, où  $o_q \subset O$
- $f$  Une application de l'ensemble des transactions de Q dans l'ensemble des parties de  $o_q$ .  $f$  dépend d'un paramètre formel lequel sera associé au lieu d'exécution de la transaction lors de la création des objets.

Cette application fait correspondre à chaque transaction tous les types d'objets qui sont impliqués dans son exécution. En pratique, ces opérations se font séquentiellement (création d'une quête, positionnement des objets, création d'une autre quête...). Mais, à une étape donnée de ce processus, le résultat est identique.

Le concepteur utilise ensuite certains types d'objets pour établir les relations de voisinage entre les zones, créer des allégories d'espace (au sens d'Aarseth, voir chapitre 2) et définir ainsi les différents parcours possibles dans la carte.

### *Etape 2 : Level Design*

Nous supposons que la carte du niveau est donnée. Elle détermine une description topologique représentable par un hypergraphe. Nous expliquons plus loin comment à partir d'une description géométrique que peut fournir un modéleur et la connaissance des objets topologiques, il est possible de construire cet hypergraphe.

Le concepteur crée alors toutes les quêtes qu'il va utiliser dans son niveau en instanciant certaines classes de quêtes définies dans le Game Design. Ceci a pour conséquence de créer les transactions  $T_n$  et les objets correspondants. A la fin de cette opération, on dispose de tous les objets du niveau  $O_n$ . La dernière phase consiste à placer les objets sur la carte.

Nous supposons que tous les objets nécessaires pour réaliser une transaction sont situés en un même niveau. Si la même action atomique peut être exécutée indifféremment dans plusieurs lieux, on crée alors une transaction ayant autant des modalités Begin que de lieux d'exécution. A chaque modalité est associé une instance des objets liés à la transaction. Enfin, si une action du jeu nécessite des objets situés dans différents lieux, elle est

décomposée en autant de transactions. Ceci veut dire que si  $t$  est une transaction et  $l$  un lieu d'exécution tous les objets  $f(t,l)$  sont situés dans le lieu  $l$ .

Par contre, un même objet peut être associé à plusieurs transactions. Par exemple, une porte peut être ouverte ou fermée et donc nous avons une transaction d'ouverture et une transaction de fermeture. Donc, créer les transactions détermine  $T_n$  comme l'ensemble des transactions du niveau  $n$ . Ceci définit  $O_n = \cup f(t,l)$ , l'ensemble des objets du niveau. Le placement des objets fait correspondre à toute transaction un lieu  $l$  qui est tel que les objets de  $f(t,l)$  sont placés au lieu  $l$ .

Considérons les objets de modification de topologie (objets topologiques). Ce sont par exemple des portes, des ponts, des escaliers, etc... Ils sont associés à un comportement dynamique qui altère la topologie : crée une possibilité d'accessibilité (ouvrir) ou la supprime (fermer). Ils ont un état initial (ouvert, fermé) et sont associés à des classes de transactions qui peuvent être : ouvrir(objet) ou fermer(objet). La création d'un objet modificateur de la topologie dépend d'un paramètre constitué de deux listes de zones. Une transaction d'ouverture autorise des déplacements entre les zones initiales et les zones finales, une transaction de fermeture supprime cette possibilité. Ceci permet de définir les connexions qui sont la représentation formelle des objets de modification topologique et les remplacements qui associent ces objets avec une transaction d'ouverture ou de fermeture.

Enfin, le concepteur programme l'interaction du jeu en définissant les relations entre les transactions et les objets dans le monde selon certaines règles plus ou moins déclaratives ou impératives [Combs04]. Selon l'outil employé, l'interactivité est spécifiée sous forme d'un script où les comportements sont très attachés à un scénario de jeu (programmation impérative), ou d'un programme normalement orienté objet, basé sur un ensemble de règles adaptables à plusieurs scénarios selon certains paramètres (programmation déclarative).

#### **4.2.5. Implantation dans les outils de conception d'un jeu**

Dans un outil de montage de jeu, comme Renderware Studio [Criterion04] ou Virtools [Virtools04], cette méthode devient plus évidente : l'interface de l'éditeur de jeu se présente comme une carte et des décors (voir figure 4.22. Interface Virtools). Le concepteur choisit les objets de la quête, les positionne dans la carte et conçoit l'interactivité mettant en relation leurs comportements. Les objets et les comportements sont sélectionnés à partir d'une bibliothèque d'objets et d'une bibliothèque de comportements disponibles dans l'interface d'édition de l'outil [Natkin04]. Enfin, la composition est exécutée par le moteur de jeu en fonction des relations spatiales et logiques des objets et du mode de programmation (e.g. script pour Virtools, déclarative pour Renderware).

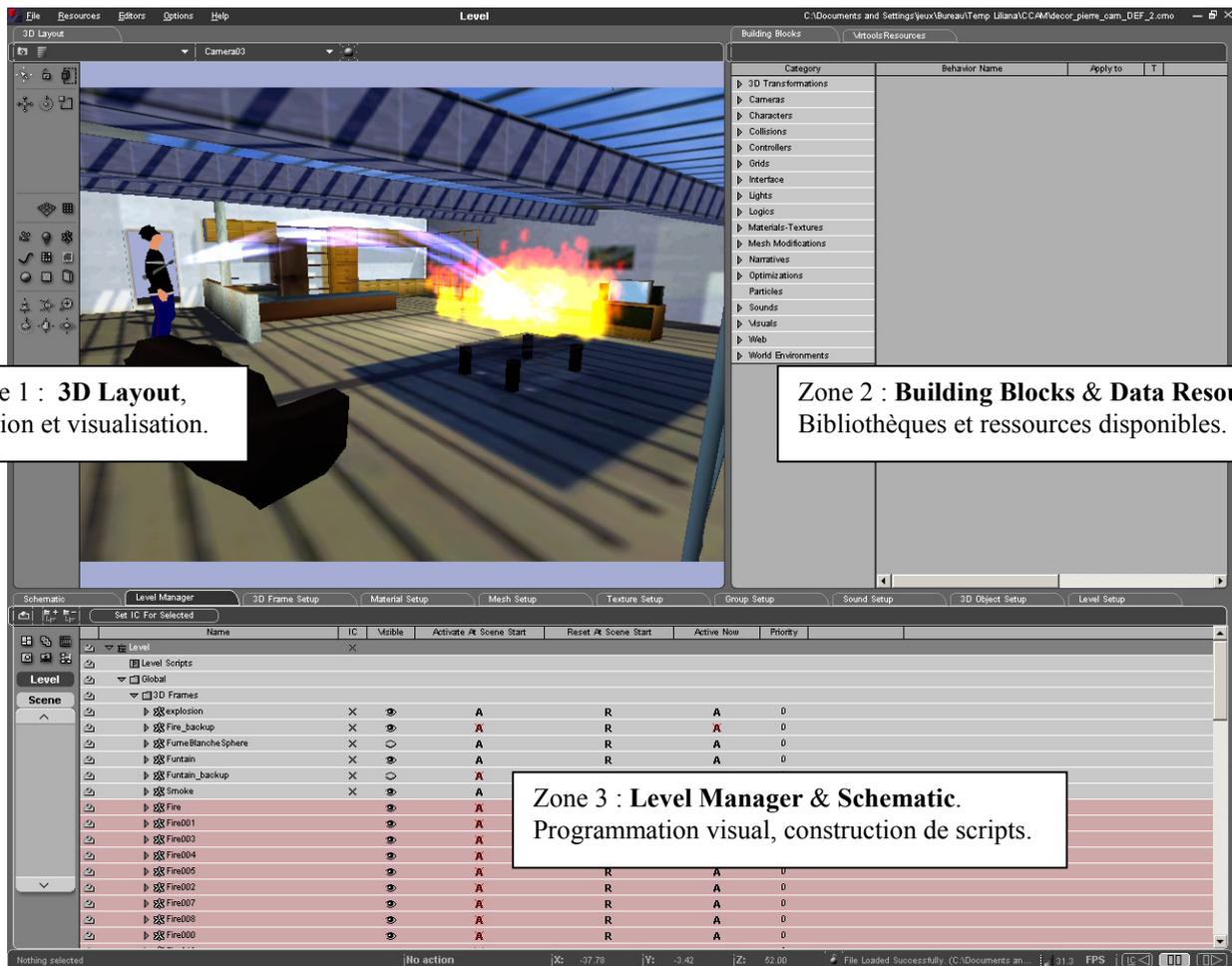


Figure 4.22 : Interface Virtools.

L'implémentation du modèle nécessite donc de fournir une interface au concepteur pour assurer la cohérence entre la topologie et le réseau de quête. D'une part, l'interface doit vérifier l'inexistence de blocages (les objets des transactions franchissables d'un marquage donné sont accessibles dans la carte à un instant du jeu), indiquer les éventuels blocages et assurer que l'ordre des transactions est cohérent avec les cheminements possibles du joueur. D'autre part, il faut que l'interface calcule le remplacement, applique les connexions pour actualiser la hypergraphe et affiche la nouvelle topologie.

Une fois que la carte du jeu a été créée dans un modèleur 3D comme Maya ou 3ds Max (description géographique), nous créons et plaçons dans la carte tous les objets topologiques définissant les ouvertures entre des zones.

Pour produire le premier hypergraphe (H0), nous proposons de récupérer la description géométrique du niveau (e.g. fichier 3ds Max) et, à partir de l'utilisation d'un algorithme de pathfinding (e.g. A\* [Sanchez04]), de parcourir la carte et de reconstituer la description topologique.

Ceci est fait de la façon suivante : dans une première étape, l'état de chaque ouverture (les objets de modification topologique) est déclaré « fermé ». Ensuite, l'algorithme de pathfinding est lancé et tous les composants fortement connexes (ou locus) détectés sont

répertoriés pour constituer les sommets de l'hypergraphe. On détermine une fonction qui a une position  $(x,y,z)$  associe le locus correspondant.

Dans une seconde étape, pour tous les objets de modification topologique, les zones mises en correspondance sont calculées. La position géométrique d'un tel objet est située dans une zone et dans le voisinage d'autres zones.

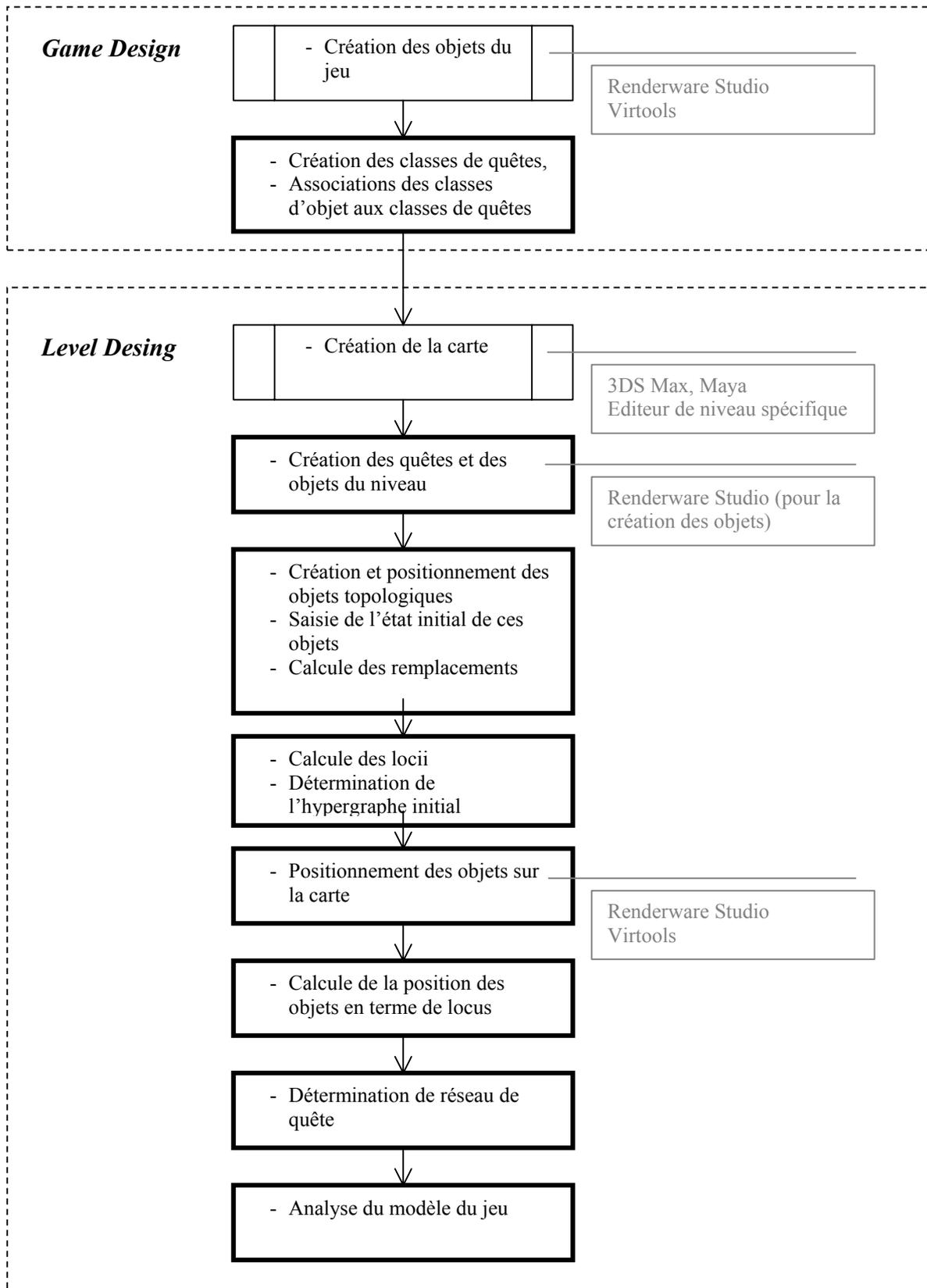
Le concepteur déclare alors l'état initial de toutes les ouvertures du niveau. En accord avec cette déclaration et conformément à la liste de connexions, les remplacements sont appliqués pour produire ainsi tous les hyperarcs de l'hypergraphe initial (H0).

Le concepteur doit alors définir les quêtes qu'il va utiliser dans le niveau en instanciant des classes de quêtes définies dans le Game Design. Ceci suppose de définir, pour le Game Design, un outil de description de quête basé sur le langage, qui décrit le réseau de quête et qui associe à chaque quête des paramètres déterminant les objets utilisés et le lieu. Lors du Game Design, l'instanciation des quêtes crée les objets correspondants.

Le concepteur de niveau va positionner les objets du jeu dans l'espace. Ceci va permettre de :

- Déterminer le locus de chaque objet
- Déterminer les modalités lieu / d'exécution des transactions associées et, par voie de conséquence, les locii qui sont la modalité de la transition Begin et la modalité de la transition End.

Tous les éléments formels de notre modèle sont alors implicitement ou explicitement créés. Il est possible de générer le réseau de transactions et d'appliquer, soit une simulation comportementale, soit une analyse complète par model checking. L'algorithme d'analyse a été décrit au paragraphe 2.3. La figure suivante schématise le processus et outils associés en mettant en évidence les briques à développer (en noir sur la figure 4.23).



**Figure 4.23 : Processus d'analyse proposé.**

## 4.3. Conclusion

Nous avons présenté une nouvelle méthodologie pour combiner les rapports spatio-temporels dans les jeux. Pour modéliser ces rapports spatiaux, nous avons employé des hypergraphes. Parce qu'il existe des nombreux liens entre graphes, hypergraphes et réseaux de Pétri, nous aurions pu aussi utiliser ces derniers pour caractériser les rapports spatiaux du jeu. Dans ce cas, le modèle est composé de deux réseaux de Petri, chacun des ensembles de contraintes (topologiques ou d'ordre) devient une précondition nécessaire pour le tir des transitions concernées de l'autre réseau. Toutefois, nous avons privilégié les hypergraphes (pour la modélisation topologique) pour leurs qualités de représentation. Ils caractérisent d'une manière plus intuitive les relations spatiales, permettant une abstraction et une construction sémantique plus descriptive concernant la structure topologique.

La méthode proposée n'est pas centrée sur le problème d'une localisation correcte des objets de la quête dans la carte du jeu. Cet aspect est traité mais n'est pas un problème complexe de conception des niveaux. L'importance de la méthode réside dans les autres avantages dérivés de l'analyse du processus de conception du jeu :

- La possibilité de produire tous les cheminements possibles du joueur, pour analyser et choisir les meilleurs par rapport à certains mécanismes dramatiques.
- L'amélioration de la conception du niveau, altérant la localisation des objets pour obtenir des quêtes plus intéressantes.
- La détection d'incohérences, lors de modifications dans la logique du *gameplay* ou de la carte du niveau.
- L'intégration naturelle de ce modèle dans les pratiques et outils utilisés dans l'univers du jeu vidéo.

Le modèle ne prend pas en considération les objets mobiles (e.g. un PNJ qui se promène de façon indépendante entre les localités ou qui peut suivre le personnage principal). Si l'objet mobile reste dans le périmètre d'un même locus, aucune modification du modèle n'est nécessaire. Dans le cas contraire, une extension de la méthode proposée doit être envisagée.

# CHAPITRE 5

## *Analyse du jeu*

---

### **Contenu du Chapitre**

Dans ce chapitre, nous appliquons notre méthode d'analyse à un exemple concret. L'étude a été effectuée sur le jeu d'action DOOM<sup>3</sup>. Nous avons choisi ce jeu particulier pour tester l'utilisabilité et les limites de notre méthode par rapport à ce type de scénario.

### **Organisation du chapitre :**

Section 5.1. Analyse du jeu DOOM<sup>3</sup>. – p. 104 : Introduction à l'analyse du jeu DOOM<sup>3</sup> et aux caractéristiques du genre.

Section 5.2. *Game design* et *level design*. – p. 105 : Application de notre méthode pour l'analyse de ces deux niveaux de conception.

Section 5.3. Conclusions – p. 118 : Présentation des aspects techniques et de l'analyse du jeu.

## 5.1. Analyse du jeu DOOM3

### 5.1.1. Introduction

*« Nous sommes en Mars 2045, dans l'Union Aerospace Corporation, un complexe scientifique hautement protégé. Les scientifiques s'activent autour de leurs ordinateurs : ils semblent travailler sur un projet ultra secret. Jusqu'au moment où l'un d'entre eux, l'air inquiet, se dirige dans une petite salle... et là, une explosion se produit dans le complexe : des crânes fantomatiques s'échappent d'une machine endommagée. Ils foncent sur les gardes et les scientifiques, qui deviennent de véritables zombies. Bientôt, tout les occupants de la base subissent le même sort. [...] un seul individu échappe à ce terrible maléfice. Seul, il devra survivre à bord de l'Union Aerospace Corporation, infestée de créatures démoniaques. Ce personnage, c'est bien entendu VOUS. »<sup>13</sup>*

Ce type de scénario correspond à celui d'un jeu d'action FPS (*First Person Shooter*). Le but principal dans ces jeux consiste à tuer des ennemis. Chaque niveau du jeu semble être un véritable labyrinthe où le joueur doit progresser en évitant de se faire tuer par ses adversaires.

Nous avons indiqué dans les chapitres précédents notre intérêt pour les jeux solo dont les caractéristiques correspondent aux genres aventure, RPG et stratégie. Nous avons cependant choisi DOOM<sup>3</sup> pour effectuer cette analyse pour plusieurs raisons :

- Tester l'utilisabilité de notre méthode d'analyse pour ce genre de jeu.
- Comparer les résultats obtenus avec ceux des jeux d'aventure, RPG et de stratégie.
- Analyser la complexité de la topologie des jeux FPS.

Nous prendrons comme support à notre analyse deux niveaux du jeu. Le premier niveau, par sa simplicité, nous semblait intéressant pour une première analyse. Une seconde analyse, plus complexe, a pu être menée sur le neuvième niveau.



DOOM<sup>3</sup>. Id Software 2003.

<sup>13</sup> Description extrait de Chroniques et points de vue. Disponible à partir du site AMAZON jeux vidéo : DOOM3 à l'adresse électronique : <http://www.amazon.fr>

## 5.2. Game design et level design

### 5.2.1. Game design

#### Synopsis<sup>14</sup>

Le joueur incarne un soldat d'une unité spécial de sécurité envoyé dans un laboratoire de recherche de la Corporation Union Aerospace sur Mars. Des expériences de téléportation sont réalisées dans cette base spatiale. Peu après son arrivée sur la base, le joueur doit exécuter une mission de routine.

En revenant à la base, le joueur se rend compte que tous les soldats sont morts ou transformés en zombis (mort-vivants). Un seul survivant, le commandant Serge, reste en contact avec le joueur et lui donnera des instructions par radio. Le joueur a pour mission de tuer les démons pour les empêcher d'envahir Mars et la Terre.

Il y existe différents endroits à visiter pendant le déroulement du jeu : la base spatiale UAC, certains lieux à l'extérieur de Mars et l'enfer.

#### Scénario du niveau 1

Le joueur arrive à la base spatiale de Mars. Dans le hangar, un garde lui propose de se présenter à la réception. Le soldat a accès à la base après avoir été soumis à une procédure de sécurité (scanner). Une fois à la réception, on lui donne son équipement personnel (un PDA – *Personal Digital Assistant*) et on lui révèle son premier objectif : se rendre au Quartier Général.

Au Quartier Général, le commandant de la base lui donne des instructions pour atteindre son second objectif : aller à la recherche d'un scientifique. Un robot lui est attribué pour l'aider dans cette tâche.

#### Scénario du niveau 9

Le joueur arrive au niveau « communication » de la base. A l'entrée, il doit activer un bouton pour ouvrir la porte que lui donne accès au garage. Une fois à l'intérieur, il peut récupérer une armure et, grâce à une cinématique, prendre connaissance d'un premier sous-objectif : localiser la salle de communication principale. Le joueur devra traverser un tiers du niveau pour pouvoir y accéder. Le parcours s'avère difficile avec la confrontation de plusieurs types de monstres et créatures de l'enfer qui bloquent le chemin. Après avoir atteint la salle de communication, le second sous-objectif est révélé : traverser la zone d'ingénierie et localiser le centre de contrôle satellite.

---

14 Information extrait de PC GAMES E3 2004: DOOM 3 Q&A et traduis de l'anglais. Disponible à l'adresse électronique : <http://pc.ign.com/articles/511/511995p1.html>

Pour arriver au centre de contrôle satellite, le joueur devra traverser le niveau entier en courant les mêmes risques qu'auparavant. Tout au long du parcours, le joueur récupère des objets précieux (munitions, PDA's, ...) et des informations sur des bornes électroniques qui l'aideront à atteindre son objectif.

Dans le centre de contrôle satellite, le joueur sera informé par son supérieur qu'un signal de détresse doit être envoyé à la flotte orbitale. Le joueur a le choix d'envoyer ou non le signal. Un nouvel objectif lui est donné : retourner au salon et aller à la station du monorail pour atteindre les Laboratoires Delta.

### 5.2.2. Level design

Dans le prochain paragraphe nous présentons le modèle pour les scénarios des niveaux 1 et 9 analysés selon la méthode proposée (voir §2 du chapitre 4). Cette représentation est composée de deux étapes : la première consistant en une description des quêtes à exécuter pour finir les niveaux et leur description dans le langage d'ordonnancement (présenté dans le chapitre 3). La seconde est construite à partir des différentes phases dans l'évolution de la topologie (chapitre 4).

#### Description des quêtes du niveau 1

Les actions que nous avons identifiées pendant cette première étape du jeu correspondent à une période d'entraînement pour le joueur. Il s'agit de parcourir le niveau en repérant les différents types d'interactions : rencontrer des personnages non joueurs, récupérer certains objets (PDA's, disques vidéo), activer des robots, consulter des bornes électroniques pour récupérer des informations téléchargeables sur les PDAs, etc.

*La liste des quêtes exécutables dans le premier niveau est la suivante :*

#### **Quête 1 : Aller au Quartier Général.**

- A. Activer\_borne\_information
- B. Rencontrer\_personnage
- C. Passer\_scanner
- D. Recuperer\_equipement
- E. Recuperer\_Vidéodisque\_1
- F. Recuperer\_PDA2
- G. Jouer\_jeu\_video
- H. Fermer\_volets

Le réseau de transactions de cette première quête est défini par  $TR=\{A..H\}$ , où le langage d'ordonnancement des actions est :

Precond( Activer\_borne\_information ) =1

Precond( Rencontrer\_personnage ) =1

Precond( Passer\_scanner ) =1

Precond( Recuperer\_equipement ) = Avant( Passer\_scanner )  
Precond( Recuperer\_Vidéodisque\_1 ) = Avant( Passer\_scanner )  
Precond( Recuperer\_PDA2 ) = Avant( Recuperer\_equipement)  
Precond( Jouer\_jeu\_video ) = Avant( Recuperer\_equipement)  
Precond( Fermer\_volets ) = Avant( Recuperer\_equipement)

## Quête 2 : Aller à la recherche d'un scientifique hors de la base.

- I. Suivre\_robot
- J. Trouver\_sortie
- K. Appeler\_ascenseur

Le réseau de transactions décrivant la seconde quête est définie par  $TR=\{I..K\}$ , où le langage d'ordonnancement des actions est :

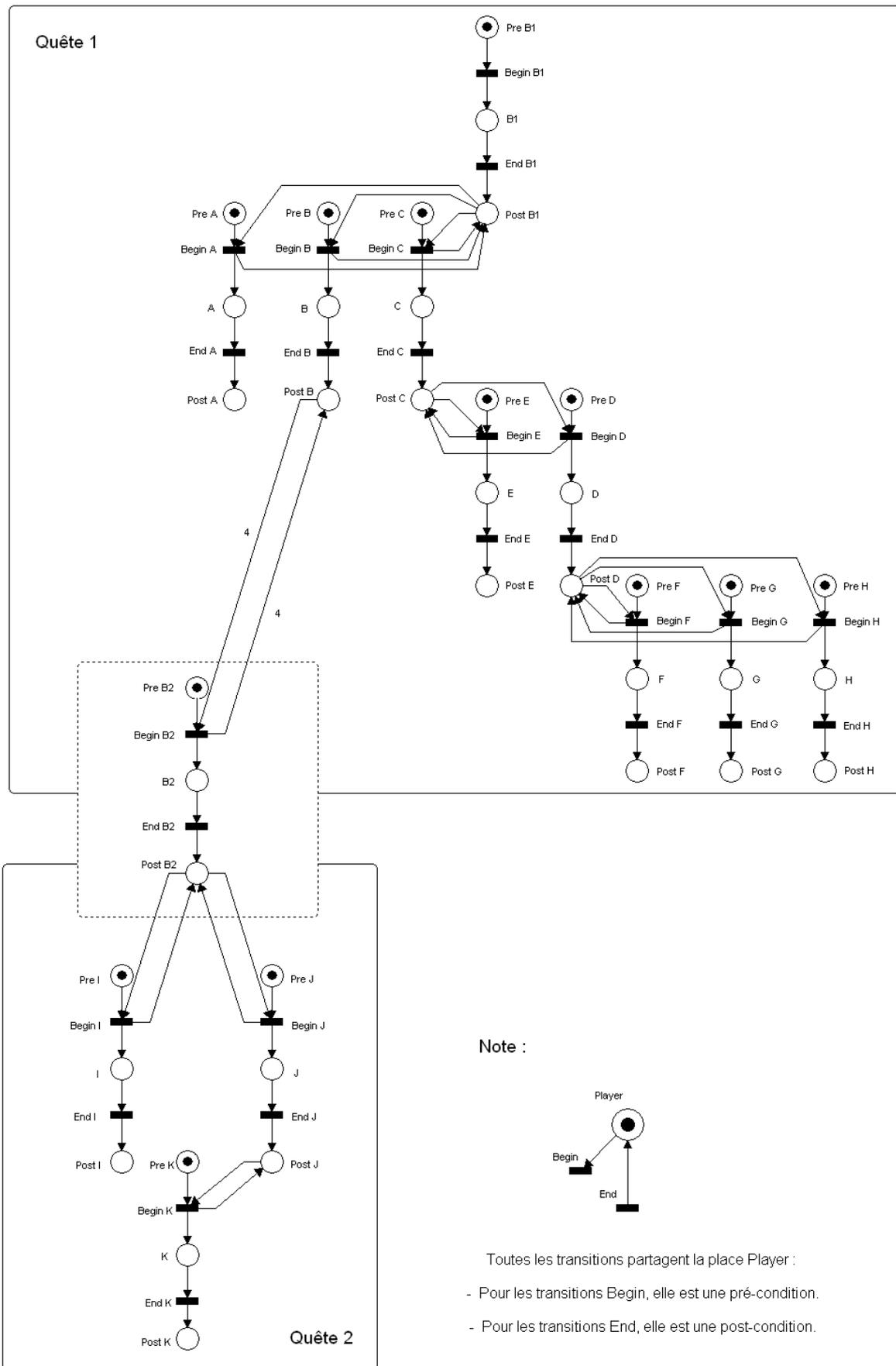
Precond( Suivre\_robot ) = Avant( Rencontrer\_personnage)  
Precond(Trouver\_sortie) = Avant( Rencontrer\_personnage)  
Precond( Appeler\_ascenseur ) = Avant(Trouver\_sortie)

Le modèle pour ce scénario est donné par le réseau de quêtes de la figure 5.1 et les hypergraphes des figures 5.3, 5.4 et 5.5.

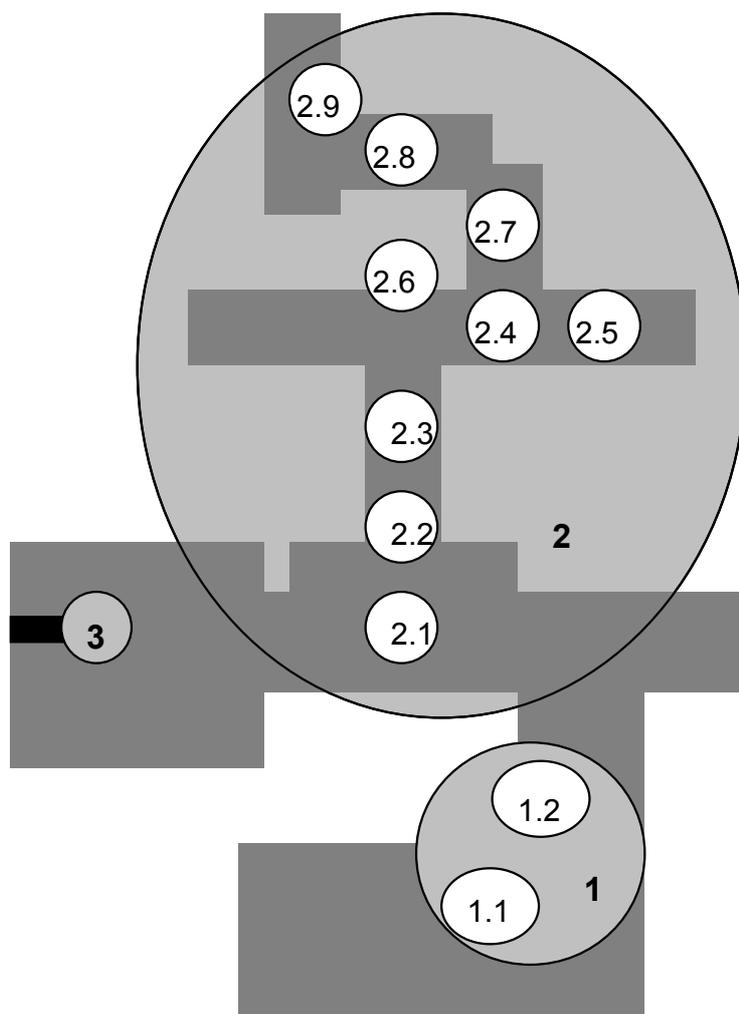
## Modélisation de l'espace du jeu du niveau 1

La carte de DOOM<sup>3</sup> correspondant au premier niveau du jeu (base spatiale UAC) est dessinée dans la figure 5.2. La topologie de la base est en gris foncé. Trois locii ont été identifiés et leur composition est détaillée en bas de la figure. La zone en noire signale un ascenseur.

Le jeu commence dans le hangar de la base (locus 1). Le joueur peut entrer dans la base (locus 2) uniquement après avoir été scanné au point de sécurité (locus 1). Tant que le scanner n'est pas terminé, l'accès à la base est interdit. La hypergraphe dans la figure 5.3 montre de quelle manière le locus 1 est isolé du reste des installations de la base. Comme nous l'avons indiqué dans le chapitre précédent, les cercles, dans les hypergraphes, modélisent les locii et les flèches, les arcs. Ces derniers sont identifiés par des lettres en minuscules correspondant à celles de la liste de transactions (lettres en majuscules) à exécuter dans la quête.



**Figure 5.1 : Réseau de transactions du premier niveau de DOOM<sup>3</sup>.**



Zones (locii) à explorer :

**1. Hangar**

- 1.1. Hangar
- 1.2. Point de sécurité

**2. Accès central**

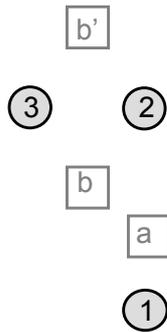
- 2.1. Salon de départ
- 2.2. Reception
- 2.3. Accès central

- 2.4. Accès commandement

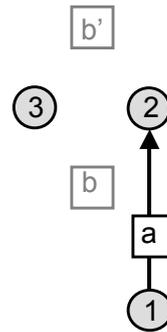
- 2.5. Toilettes
- 2.6. Accès QG Marines
- 2.7. Zone commune
- 2.8. QG Marines
- 2.9. Contrôle QG Commandement

**3. Maintenance**

**Figure 5.2 : Carte de la base spatiale UAC.**



**Figure 5.3 : Hypergraphe initial (H0) du premier niveau de DOOM<sup>3</sup>.**



**Figure 5.4 : Nouvelle hypergraphe (H1) après avoir passé le scanner au point de sécurité.**

Après avoir passé le scanner, le joueur a le droit d'accéder à la base. Maintenant, il a la possibilité de se déplacer dans les installations de la base comprises entre le salon du départ et le contrôle du QG commandement (zones comprises dans le locus 2). Ceci est modélisé par le remplacement suivant :

$$r_1 = \{ (\{\}, \{\}) | (\{1\}, \{2\}) \}$$

Le nouvel hypergraphe est montrée dans la figure 5.4.

Après être arrivé à la réception (zone scénarisé), une cinématique est activée. Pendant son déroulement, le soldat à l'accueil reçoit le joueur, lui fournit son premier objectif et lui donne son PDA. Grâce au PDA, le joueur pourra se déplacer dans les zones pour lesquelles il possède les autorisations.

Le premier objectif consiste à aller au Quartier Général, localisé tout au fond de la base. Le joueur est, de cette façon, contraint de parcourir toute la zone (voir figure 5.2). Les déplacements sont simples, les couloirs n'ont pas d'embranchement compliqué. La distribution des zones est presque séquentielle et le joueur peut trouver le Quartier Général facilement.

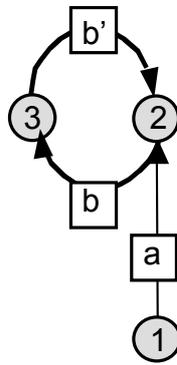
Pendant ce parcours, le joueur traverse l'accès central (zone scénarisée) où une deuxième cinématique est activée. Les informations révélées n'ont aucune influence sur le jeu, celles-ci contribuant uniquement à l'immersion du joueur.

Une fois que le joueur est arrivé au Quartier Général, le premier objectif est atteint. Alors, au travers d'une troisième cinématique, le deuxième objectif lui est confié : rechercher un scientifique hors de la base. Pour sortir de la base, le joueur doit prendre l'ascenseur localisé dans la zone de maintenance (locus 3), initialement inaccessible. C'est grâce à la prise de connaissance de l'objectif 2 que la topologie de la carte est modifiée, permettant ainsi au joueur d'accéder à cette zone.

Nous modélisons cette possibilité par le remplacement :

$$r_2 = \{ (\{\}, \{\}) | (\{2\}, \{3\}), (\{\}, \{\}) | (\{3\}, \{2\}) \}$$

L'hypergraphe de la figure 5.5 montre la nouvelle topologie de la carte.



**Figure 5.5 : Nouvelle hypergraphe (H2) après avoir appris le deuxième objectif.**

Après avoir appelé l'ascenseur, le joueur peut partir à la recherche du scientifique. C'est la fin du premier niveau.

### **Description des quêtes du niveau 9**

Les actions repérées dans ce niveau sont différentes de celles du premier, étant plus orientées vers le *gameplay* de ce type de jeu. Le but pour le joueur consiste à affronter les ennemis rencontrés sur son parcours. Même si l'évitement de certains combats reste techniquement possible, ce choix entraîne une accumulation d'adversaires et de danger à proximité du joueur. Pour l'analyse suivante, nous supposons que le joueur choisit une stratégie d'affrontement.

Comme dans le niveau précédent, le joueur réalise des actions élémentaires pour accéder aux informations et progresser dans le niveau.

Notons que pour cette partie de l'analyse nous utilisons le modèle de transactions représenté en réseaux de Pétri Colorés. De cette manière, nous définissons une liste généralisée des transactions à exécuter pour finir le niveau :

1. Ouvrir\_porte
2. Prendre\_armure
3. Prendre\_arme
4. Appeler\_ascenseur
5. Activer\_borne\_information
6. Recuperer\_PDA
7. Activer\_robot
8. Tuer\_monstre

## La structure du jeu pour ce niveau est la suivante :

Objectif 1 : Envoyer un signal de détresse à la flotte orbitale.

Sous-objectif 1.1 : Localiser la salle de communication principale.

Sous-objectif 1.2 : Traverser l'ingénierie et localiser le centre de contrôle satellite.

Objectif 2 : Atteindre les Laboratoires Delta.

Sous-objectif 2.1 : Retourner au salon et aller à la station du monorail.

Nous nous sommes basés sur l'ordre dans lequel le joueur exécute ces objectifs pour les traduire dans le modèle de *réseau de niveau* suivant :

Quête 1 = Sous-objectif 1.1

Quête 2 = Sous-objectif 1.2

Quête 3 = Objectif 1

Quête 4 = Objectif 2 et Sous-objectif 2.1

Quête 1 est avant Quête 2, Quête 2 est avant Quête 3 et Quête 3 est avant Quête 4.

Les tableaux ci-dessous décrivent les transactions à exécuter pour chacune des quêtes du niveau. Les lieux d'exécution ainsi que leurs antécédents de lieux sont indiqués (voir la carte du niveau dans la figure 5.7).

### Quête 1 : Localiser la salle de communication principale.

<i>transaction</i>	<i>lieux</i>	<i>antécédents de lieux</i>
A. Ouvrir_porte	{entrée_garage} {hall_comm}	{{entrée_garage}, {pont, hall_comm}}
B. Prendre_armure	{garage_comm}	{entrée_garage, garage_comm}
C. Prendre_arme	{garage_comm}	{entrée_garage, garage_comm}
D. Tuer_monstre*	{garagé_comm, jonc_accès1, ...}	{{entrée_garage, garage_comm}, {jonc_accès1, garage_comm}, ...}
E. Appeler_ascenseur	{garage_comm}	{entrée_comm, garage_comm}
F. Activer_borne_information	{hall_comm., salle_comm}	{{pont, hall_comm}, {jonc_accès3, salle_comm}}

Le réseau de transactions correspondant à la quête 1 est défini par  $TR=\{A..F\}$ , où le langage d'ordonnancement des actions est :

Precond(Ouvrir\_porte)= 1

Precond(Prendre\_armure)= Avant(Ouvrir\_porte)

Precond(Prendre\_arme)= Avant(Ouvrir\_porte)

Precond(Tuer\_monstre)= Avant(Prendre\_armure)  $\wedge$  Avant(Prendre\_arme)

---

\* Pour des raisons de clarté, nous avons indiqué seulement les deux premières instances de la classe employée dans le niveau. Voir les annexes pour la description intégrale des instances des classes.

Precond(Appeler\_ascenseur)= Avant(Ouvrir\_porte)  
 Precond(Activer\_borne\_information)= Avant(Appeler\_ascenseur)

*Note : Le langage d'ordonnancement décrivant le reste des quêtes du niveau est donné en annexe.*

### Quête 2 : Traverser l'ingénierie et localiser le centre de contrôle satellite.

	<i>lieux</i>	<i>antécédents de lieux</i>
G. Recuperer_PDA*	{salle_comm, sist_comm, ...}	{{jonc_accès3, salle_comm}, {unité_ing_comm1_b, sist_comm}, ...}
D. Tuer_monstre*	{jonc_accès3, unité_ing_comm1_haut, ...}	{{jonc_accès3, salle_comm}, {unité_ing_comm1_haut, jonc_accès3}, ...}
A. Ouvrir_porte	{jonc_accès3, unité_ing_comm2}	{{salle_comm, jonc_accès3}, {bureau_securité, unité_ing_comm2_entrée}}
E. Appeler_ascenseur*	{unité_ing_comm1_haut, unité_ing_comm1_b, ...}	{{unité_ing_comm1_haut, jonc_accès3}, {unité_ing_comm1_a, unité_ing_comm1_b}, ...}
H. Activer_robot	{bureau_securité}	{unité_ing_comm2_entrée, bureau_securité}
F. Activer_borne_information	{bureau_securité, ctrl_sat}	{{unité_ing_comm2_entrée, bureau_securité}, {ctrl_termique, ctrl_sat}}

### Quête 3 : Envoyer un signal de détresse à la flotte orbitale.

	<i>lieux</i>	<i>antécédents de lieux</i>
F. Activer_borne_information	{ctrl_sat}	{ctrl_termique, ctrl_sat}

### Quête 4 : Retourner au salon et aller à la station du monorail pour atteindre les Laboratoires Delta.

	<i>lieux</i>	<i>antécédents de lieux</i>
G. Récuperer_PDA	{ctrl_sat}	{ctrl_termique, ctrl_sat}
E. Appeler_ascenseur*	{ctrl_sat, entretien_comm, ...}	{{ctrl_sat}, {ctrl_sat, entretien_comm}, ...}
D. Tuer_monstre	{entretien_comm}	{ctrl_sat, entretien_comm}
A. Ouvrir_porte	{hall_comm}	{pont, hall_comm}

Le réseau de Pétri coloré de la figure 5.6 et les hypergraphes des figures 5.8, 5.9, 5.10 et 5.11 montrent le modèle pour le scénario de la première quête du neuvième niveau<sup>15</sup>.

<sup>15</sup> Le RdPC du niveau complet comprenant les quêtes 1, 2, 3 et 4 est présentée en annexes, ainsi que la totalité des hypergraphes générés par les modifications du niveau.

## Modélisation de l'espace du jeu du niveau 9

Lors de l'analyse du niveau 1, nous avons constaté la possibilité de regrouper plusieurs zones du niveau dans un même locus. Ce regroupement était possible lorsque le joueur pouvait accéder aux différentes zones sans contrainte particulière. En revanche, l'analyse du niveau 9 présente des contraintes supplémentaires. Le combat contre les différents types de monstres représente un obstacle important à la progression du joueur. L'accès aux différentes zones ne dépend plus uniquement de la présence des objets topologiques, mais aussi et surtout du succès du joueur lors de ses combats. Pour cette raison nous n'avons pu fusionner certaines zones dans un même locus. La liste des zones décrivant les locii du niveau reste donc presque la même que la liste originale des zones décrites dans le jeu.

En ce qui concerne la description de la topologie à partir des hypergraphes, il n'existe pas d'hyperarcs au sens strict du terme. L'utilisation d'arcs simples a suffi pour décrire la dynamique de la topologie pendant le déroulement du jeu. Ceci résulte de la simplicité de la topologie du niveau. On peut accéder à une zone à partir de deux zones connexes au maximum. La plupart des zones sont distribuées séquentiellement dans l'espace du niveau. Il n'existe pas de carrefour ou de rond-point.

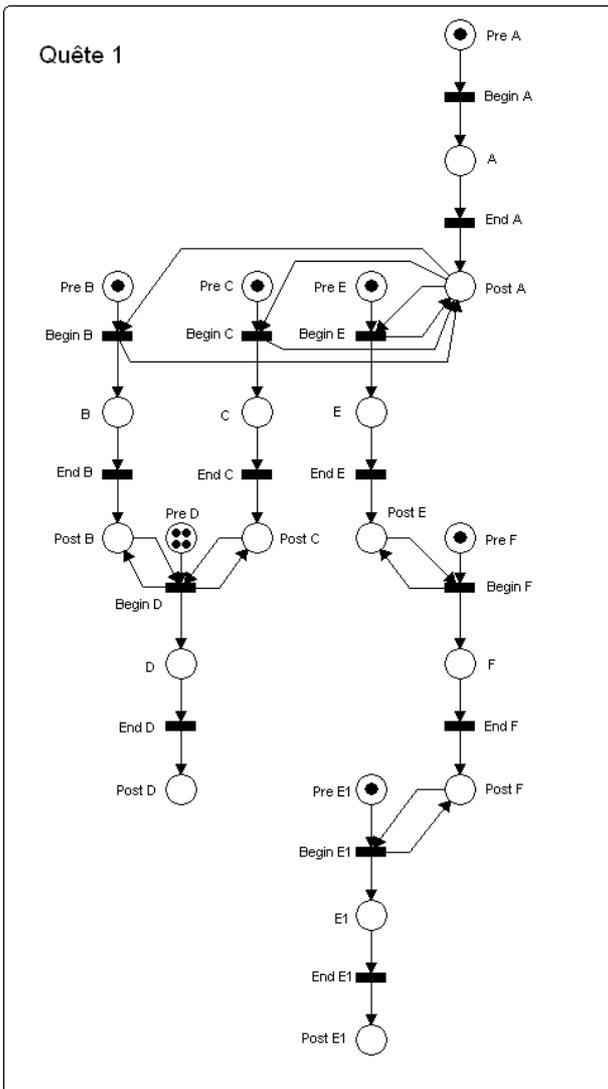
La carte correspondant au neuvième niveau de DOOM<sup>3</sup> est montrée dans la figure 5.7. La topologie que nous proposons ne correspond pas forcément à une distribution séquentielle entre les différents étages. Certains d'entre eux ont des mezzanines ou des niveaux intermédiaires qui compliquent la description stricte de la topologie du niveau. Les différents secteurs du bâtiment sont indiqués en gris. Les zones en noir signalent des ascenseurs ou des monte-charges.

Le niveau commence à l'entrée du garage (locus 1). Le joueur doit appuyer sur un bouton pour ouvrir la porte d'entrée et rentrer dans le garage (locus 2). La porte se referme juste après, empêchant le joueur de revenir à l'entrée. Les hypergraphes montrés dans les figures 5.8 et 5.9 modélisent ce scénario (le locus 1 reste isolé du locus 2 tant que le joueur n'a pas ouvert la porte).

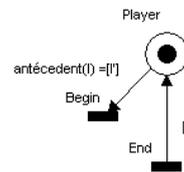
Le remplacement suivant modélise l'ouverture de la porte d'entrée (figure 5.9) :

$$r_1 = \{ (\{\}, \{\}) | (\{1\}, \{2\}) \}$$

Le joueur continue son parcours et, après un deuxième combat, peut traverser le garage pour accéder à l'ascenseur. En appelant l'ascenseur, la topologie est modifiée à nouveau. La figure 5.10 montre le nouvel hypergraphe.

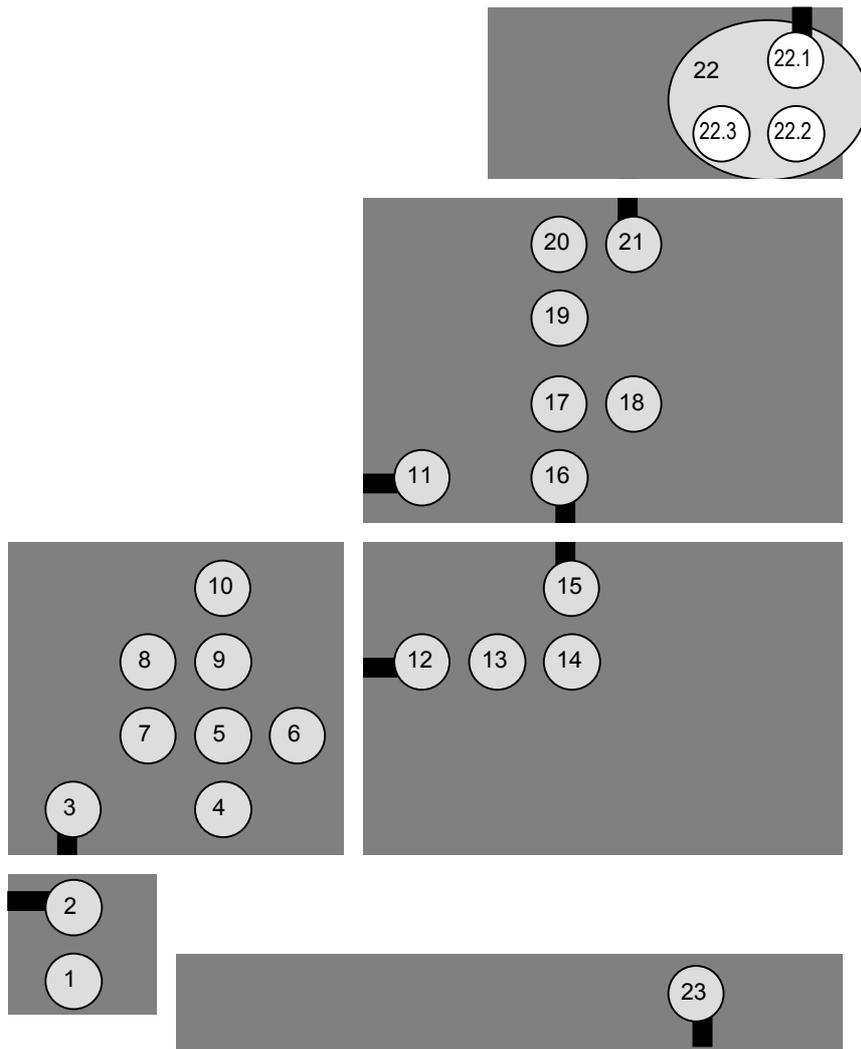


**Note :**



- Toutes les transitions partagent la place Player :
- Pour les transitions Begin, elle est une pré-condition.
  - Pour les transitions End, elle est une post-condition.

**Figure 5.6 : Réseau de transactions de la quête 1 du neuvième niveau de DOOM<sup>3</sup>.**



Zones (locii) à explorer :

- |   |  |
|---|--|
| <ol style="list-style-type: none"> <li>1. Entrée</li> <li>2. Garage communication</li> <li>3. Jonction accès 1</li> <li>4. Pont</li> <li>5. Hall communication</li> <li>6. Accès Monorail</li> <li>7. Jonction accès 2</li> <li>8. Couloir communication</li> <li>9. Jonction accès 3</li> <li>10. Salle communication</li> <li>11. Unité ingénierie communication 1 (haut)</li> <li>12. Unité ingénierie communication 1 (bas)</li> <li>13. Unité ingénierie communication 1(a)</li> </ol> | <ol style="list-style-type: none"> <li>14. Stockage ingénierie communication 1</li> <li>15. Unité ingénierie communication 1 (b)</li> <li>16. Systèmes communications</li> <li>17. Unité ingénierie communication 2 (entrée)</li> <li>18. Bureau sécurité</li> <li>19. Unité ingénierie communication 2</li> <li>20. Annexe commun</li> <li>21. Contrôle thermique</li> <li>22. Contrôle satellite             <ul style="list-style-type: none"> <li>22.1 Accès au contrôle</li> <li>22.2 Pont</li> <li>22.3 Centre de contrôle</li> </ul> </li> <li>23. Entretien communication</li> </ol> |
|---|--|

Figure 5.7 : Carte du secteur communication de la base spatiale UAC.

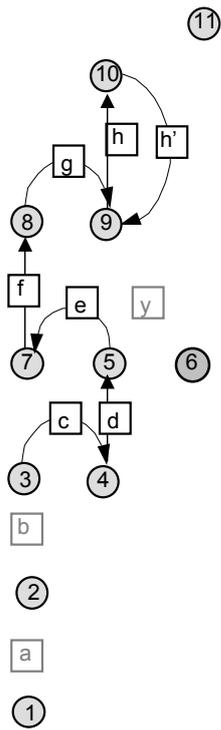


Figure 5.8 : Hypergraphe initial (H0) du neuvième niveau de DOOM<sup>3</sup>.

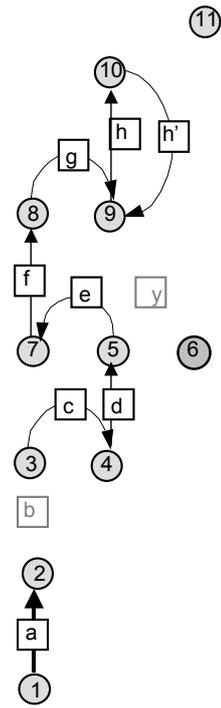


Figure 5.9 : Nouvelle hypergraphe (H1) après avoir ouvert la porte d'entrée.

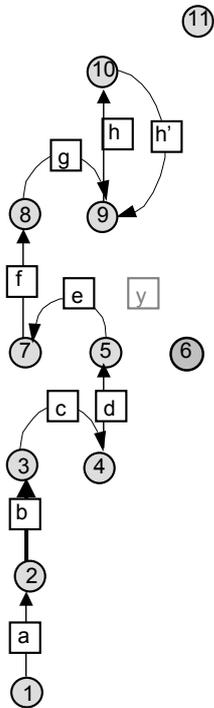


Figure 5.10 : Hypergraphe (H2) après avoir appelé l'ascenseur.

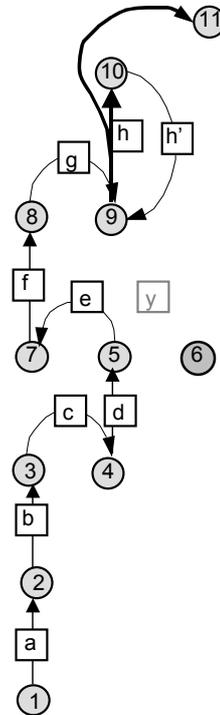


Figure 5.11. Hypergraphe (H3) après avoir récupéré la PDA.

Le remplacement suivant génère l'hypergraphe de la figure 5.10 :

$$r_2 = \{ (\{\}, \{\}) | (\{2\}, \{3\}) \}$$

Pendant l'exécution de cette première quête, une troisième modification dans la topologie est provoquée par la récupération du PDA de M. Blake localisé dans la salle de communication (locus 10). Ceci donne accès à la porte permettant d'aller de la jonction d'accès (locus 9) à l'unité d'ingénierie communication (locus 11).

Le remplacement indiquant cette modification est (figure 5.11) :

$$r_3 = \{ (\{9\}, \{10\}) | (\{9\}, \{10, 11\}) \}$$

Notons que ceci est le seul remplacement du modèle utilisant des hyperarcs.

## 5.3. Conclusions

### 5.3.1. Aspects techniques

Pendant le déroulement du premier niveau, l'évolution de la topologie du jeu est modifiée en fonction des transactions exécutées par le joueur. Aucune relation de dépendance entre les transactions et l'accès à certains objets n'a été détectée. Par conséquent, nous pouvons réaliser l'analyse avec l'utilisation de modèles de réseaux de Pétri places transitions et d'hypergraphes. Nous avons montré cette représentation dans la section 4.2.1.

A l'opposé, dans l'analyse du niveau 9, l'exécution de transactions particulières dépend de l'accès et de l'activation de certains objets localisés dans des zones *a priori* inaccessibles. Dans ce cas, l'application de réseaux de Pétri Colorés et d'hypergraphes pour la représentation de ce niveau a été nécessaire. Nous avons présenté ce modèle dans la section 4.2.2.

L'utilisation de notre méthode nous permet d'extraire des modèles peu complexes. Les RdP modélisés sont constitués des quêtes très simples, presque linéaires, à l'exception de deux quêtes (2 et 4 du niveau 9) qui partagent une transaction commune. Quelques transactions obligatoires sont suffisantes pour terminer les niveaux. Le langage généré pour expliquer la sémantique des réseaux est aussi très simple.

La simplicité des relations spatio-temporelles du système du jeu sous-emploie les capacités de description des RdP et des hypergraphes. Ceci a pour conséquences :

- Le remplacement simple des hypergraphes.
- L'utilisation d'hyperarcs inutiles, à l'exception du dernier remplacement ( $r_3$ ) où un hyperarc a été nécessaire pour modéliser le changement dans la topologie.

Nous décrivons ensuite la seconde partie de l'analyse.

### 5.3.2. Analyse du jeu

La manière dans laquelle l'information est présentée au joueur dans le premier niveau suggère un certain apprentissage de l'utilisation des objets et du type d'interactions nécessaires pour leur activation (*gameplay*).

Avant d'effectuer l'analyse, le niveau 9 semblait intéressant, en particulier au niveau de sa topologie. Les textures, les sons et l'ambiance générale donnent l'impression d'un niveau compliqué. Toutefois, après l'analyse, on se rend compte que la difficulté est basée principalement sur les combats (*gameplay* de ce type de jeux) et que la topologie est en réalité très simple. Les déplacements sont très limités et linéaires. Il n'existe pas de possibilités d'explorer librement le niveau. Les parcours possibles sont imposés par la distribution séquentielle des zones.

Les autres niveaux pratiqués (20 au total pour ce jeu) donnent la même expérience de *gameplay*. La topologie maintient une distribution séquentielle. La progression du joueur est toujours linéaire, sauf quelques retours en arrière parfois possibles mais assez limités (dans la même zone).

*Half Life, Soldier of fortune, Call of Duty, Unreal tournament, etc...* sont des jeux FPS comparables à ceux de la série DOOM. Les caractéristiques qui nous intéressent pour effectuer notre analyse restent les mêmes. Les jeux d'action-aventure comme *Tomb Raider* méritent d'être analysés pour valider les aspects qui font des meilleurs scénarii.



# CHAPITRE 6

## *Conclusion Générale*

---

Notre travail de recherche part d'un point de vue analytique. Il permet de comprendre et de décrire de manière formelle le scénario d'un jeu. Cette caractérisation est fondée sur la description de l'ordre des actions dans un jeu vidéo et de leur relations logiques. Notre étude constitue une contribution à l'analyse formelle et à la spécification des scénarios de jeux vidéo.

L'état actuel des travaux de recherche sur les méthodes d'analyse et de spécification des jeux vidéo met en évidence la convergence des domaines de la narration audiovisuelle, du génie logiciel et de l'analyse cognitive et sociale des comportements du joueur. Notre étude se situe plus précisément au carrefour de la narration audiovisuelle et du génie logiciel. Nous nous sommes intéressés aux relations temporelles, logiques et spatiales définissant les séquences des actions du joueur.

A travers l'analyse des séquences des actions que nous identifions comme « quêtes » [Guardiola00], nous décrivons la structure narrative non linéaire du jeu. De telles structures correspondent aux jeux qui, entre autres, racontent une histoire au travers des actions exécutées par le joueur [Sanchez]. Ces actions sont issues d'un processus analytique ou des choix offerts au joueur pendant le déroulement du jeu. En considérant un jeu vidéo comme un système hiérarchique de buts définis par une structure narrative, notre travail s'adresse principalement aux jeux d'aventure, les RPGs et aux jeux de stratégie. Nous avons basé notre étude sur 3 axes principaux :

- La description formelle de l'ordonnement des séquences des actions du joueur.
- La représentation formelle de la notion de localité, décrivant la relation entre les séquences des actions et l'univers du jeu.
- La proposition d'une méthode d'analyse, qui tient compte de la dépendance réciproque entre les transactions et la topologie.

En ce qui concerne l'ordonnement des séquences des actions du joueur, un modèle de quête a été proposé comme élément central pour la définition de la structure narrative. Un jeu est décomposé en niveaux. Les niveaux décrivent la structure linéale (trame) d'un jeu. Chaque niveau est constitué d'une ou plusieurs quêtes. Les quêtes sont composées des transactions et définies par leurs relations d'ordre. Les transactions sont la représentation atomique des actions du joueur.

En partant d'une description scénaristique nous avons exprimé les relations d'ordre entre transactions sous forme d'un langage. La sémantique de ce langage correspond à une logique temporelle définie par la synchronisation propre aux réseaux de Pétri : la causalité, le

parallélisme, l'attente, le non-déterminisme et les conflits [Diaz01]. Ces modèles permettent à la fois de montrer l'aspect causal propre à la narration et le non-déterminisme qui permet au joueur de s'exprimer.

Notre approche permet de formaliser des concepts utilisés par le « *game designer* » comme les notions de quêtes facultative et obligatoire. Elle constitue une base de compréhension et de comparaison de la dynamique des jeux.

Après avoir présenté les principes et le processus d'écriture utilisés dans l'industrie pour définir les éléments de base nécessaires à la conception d'un jeu, un modèle d'écriture spatiale est proposé. Le modèle combine les rapports spatio-temporels et, en définissant l'espace physique de l'univers du jeu, les objets et leurs emplacements dans cet espace. L'espace physique est décrit à partir des hypergraphes représentant la topologie du niveau du jeu. Les objets sont classifiés en deux catégories : ceux modifiant la topologie et ceux servant comme outils pour la réalisation de la quête. Nous avons montré l'importance d'un modèle apte à valider la cohérence des niveaux créés par rapport aux relations existantes entre l'ordre des transactions, l'emplacement de ces objets, la localisation du joueur et l'accessibilité aux lieux. Nous avons montré comment ce modèle peut s'intégrer dans le processus et les outils de *game design* qui existent.

## Travaux futurs

Plusieurs aspects de notre étude révèlent des perspectives intéressantes pour continuer la recherche autour de l'analyse formelle des jeux vidéo. Le plus immédiates, qui découlent de notre proposition, sont :

– *L'analyse de jeux multijoueurs coopératifs*

Le modèle pourrait être étendu pour traiter les jeux multijoueurs coopératifs du fait que ceux-ci sont une extension des jeux solo. Les jeux multijoueurs coopératifs relèvent des mêmes types de constructions narratives que leurs versions solo et conservent en plus les mêmes principes de *gameplay*. En supposant que les actions et le scénario restent les mêmes pour tous les participants, un modèle d'analyse basé sur des réseaux de Pétri colorées pourrait être utilisé. Un nouvel ensemble de couleurs de jetons servant à distinguer les différents joueurs serait défini. Dans notre application, la couleur courante des marques de la place du joueur indiquerait le locus de la dernière transaction exécutée par chaque participant. Le modèle étendu permettrait de tracer les déplacements des différents joueurs dans la même topologie. Les modalités de franchissement des transactions resteraient donc les mêmes. Ce nouveau modèle devrait considérer l'augmentation de la complexité liée à la dépendance réciproque des transactions et de la topologie. A un instant donné, un des joueurs pourrait modifier la topologie et empêcher d'autres participants de continuer le jeu.

– *La définition d'un langage de description scénaristique plus concis*

Le langage que nous avons proposé est directement lié aux opérateurs élémentaires qui permettent de construire un réseau de Pétri. Nous avons proposé quelques abréviations simples qui n'ont pas été développées dans le langage. Il est certainement possible de définir d'autres abréviations permettant, par exemple, d'exprimer les priorités. Une

analyse de besoins et l'étude de nombreux exemples permettrait de spécifier ces abréviations et le langage associé.

– *Le développement d'un prototype*

Les modèles de construction des réseaux de Pétri sur lesquels nous avons basé notre modèle d'analyse induisent une méthode constructive qui pourrait constituer une contribution à la validation de la conception d'un jeu.

Le développement d'une interface d'édition des quêtes permettrait la mise à jour simultanée des transactions, des objets et des lieux impliqués. L'outil illustrerait de manière pratique l'efficacité et la pertinence de la méthode proposée. La conception d'un tel prototype pourrait prendre la forme d'une extension aux outils existants de création de jeux (e.g. Renderware Studio ou Virtools).

L'implantation dans les outils de conception du modèle d'analyse proposé a été présenté. Le diagramme du processus d'analyse pourrait servir de base pour la spécification d'une architecture.

Certains aspects importants de l'ordonnancement et de la topologie des jeux n'ont pas été pris en compte : le *game over*, les points de reprise, les actions dont la durée est contrainte et les objets ayant une grande mobilité. Il serait nécessaire d'étudier comment ces aspects pourraient éventuellement s'intégrer dans notre modèle ou nécessiteraient de développer une extension.

Malgré les limites précédentes nous avons montré les possibilités d'application et l'intérêt de notre modèle sur plusieurs exemples significatifs.

– *La prise en compte de l'immersion et l'apprentissage du joueur dans l'analyse*

Le modèle d'analyse proposé se limite à l'étude de la structure narrative non déterministe d'un jeu. Une amélioration de notre modèle serait de prendre en compte des aspects perceptifs du jeu. D'un point de vue émotif ou intellectuel perceptif, il existe deux types de plaisirs esthétiques : l'immersion et l'engagement [Douglas01]. Fondamentalement, le plaisir ludique, qui est l'objectif essentiel d'un jeu, ne peut relever du type d'analyse que nous proposons. Un modèle formel d'une oeuvre littéraire, cinématographique ou vidéo ludique ne peut traduire son caractère émotionnel. L'étude de tels aspects relèvent plutôt de méthodes psychologiques ou sociales. Cependant, nous considérons que certains facteurs comme le niveau d'apprentissage et le niveau de difficulté, importants pour évoquer chez le joueur les sentiments d'immersion et d'engagement, peuvent être partiellement pris en compte.

En tant que système de réalité virtuelle, un jeu est un système d'apprentissage implicite. La compréhension des lois de l'univers du jeu fait partie du plaisir de jouer. Nous avons abordé ce sujet du point de vue du *gameplay* en prenant compte une construction générique des quêtes orientées objet. Le modèle proposé permet au concepteur la spécialisation des transactions. Par exemple, deux quêtes peuvent être des instances de la même classe et avoir des niveaux de difficulté différents (e.g. plus ou moins de monstres à tuer).

Le *game* et le *level design* définissent un deuxième niveau d'apprentissage. Les premiers niveaux du jeu servent à introduire le joueur dans le jeu. Ils sont également des espaces d'entraînement (salles d'isolement, îles d'entraînement,...) où le joueur peut aller à tout moment pour s'exercer. L'exécution du jeu peut être modifiée selon les résultats obtenus lors de ces sessions d'entraînement. Le but étant de permettre au gameplay d'estimer les capacités du joueur et au joueur de prendre conscience de ses progrès.

Notre approche permet de prendre en compte l'implication de cet apprentissage dans l'ordonnancement des actions (simplification ou complexification). Il serait sans doute intéressant d'introduire des variables d'état indiquant le degré supposé d'apprentissage d'un joueur qui a atteint un niveau et d'analyser la complexité des quêtes réalisées. Une telle démarche pourrait s'inspirer des modèles relevant de l'intelligence artificielle.

Par exemple, dans l'analyse de *Myst* que nous avons effectuée, l'exécution ou non de certaines quêtes facultatives suggère un certain niveau de connaissance du joueur. La validation de ces observations permettrait d'étendre le modèle de manière à ce que ces quêtes contribuent à la définition d'un mécanisme de tension basé sur le profil du joueur. Il s'agirait d'adapter les cheminements possibles dans le jeu à l'évolution du niveau de connaissance du joueur.

Il s'agit là d'une extension d'une toute autre ampleur que celles proposées précédemment. Elle ne peut, à notre avis, être envisagée qu'au terme d'une phase importante d'expérimentations.

## *Publications*

- [1] Gal, V. et C. Le Prado, S. Natkin, L. Vega. 2002. *Writing for video games* - Proceedings VRIC, Laval France, 2002.
- [2] Gal, V. et C. Le Prado, J.B. Merlan, S. Natkin, L. Vega. 2002. *Processes and tools used in the sound design for computer games*. ICMC 2002. Goetborg, Suede, Septembre 2002.
- [3] Gal, V. et C. Le Prado, S. Natkin, L. Vega. *Quelques aspects de l'économie du jeu vidéo* - Proceedings Journées d'études "jeu et socialisation", Eclou des télécommuniactions, Paris, Décembre 2002.
- [4] Natkin, S. et L. Vega. 2003. *A Petri Net Model for the Analysis of The Ordering of Actions in Computer Games*, - Proceedings GAME ON, London, Octobre 2003.
- [5] Natkin, S. et S.Grünvogel, L. Vega 2004. *A New Methodology for Spatiotemporal Game Design*. – Proceedings CGAIDE 2004, UK Novembre 2004.



## Références

- [Aarseth00] Aarseth, Espen. 2000. *Allegories of Space: The Question of Spatiality in Computer games*. Dans Markku Eskelin et Raine Koskimaa eds. Cybertext. Université de Jyväskylä. Disponible sur <http://www.hf.uib.no/hi/espen/papers/space/>
- [Aarseth03] Aarseth, Espen. 2003. *Playing Research: Methodological approaches to game analysis*. Melbourne, DAC 2003.
- [Andrieu03] Andrieu, J. et Jeff Rawlings. *Player-Protagonist Motivation in First-Person Interactive Drama: A Framework for Aristotelian Interactive Drama*. 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment, TIDSE 2003, Darmstadt, Allemagne, mars 2003.
- [Bates00] Bates, B. 2000. *Game Design : the Art & Business of Creating Games*, Prima Tech Ed. ISBN 0-7615-3165-3
- [Bethke03] Bethke, Erik. 2003. *Structuring Key Design elements*. Gamasutra. <http://www.gamasutra.com>
- [Booch99] Booch Grady, Rumbaugh James et Ivar Jacobson. 1999. *The Unified Modeling Language User Guide*. Addison Wesley. 1999. ISBN 0-201-57168-4
- [Castellani01] Castellani, X. et H. Jiang. 2001. *Analysis and Design of Multimedia Scenarios with Navigation Diagrams and Navigation Constraints*. Rapport technique CEDRIC No 199, 2001.
- [Cerny02] Cerny, Marc. 2002. *The Method – A Model Game Design*. GDC Europe 2002. Londres 27-29 aout.
- [Chion00] Chion, Michel. 2000. *L'audio-vision son et image du cinéma*. Ed. Nathan. 2ème édition. 2000.
- [Demanchy02] Demanchy, Thomas. 2002. « Programming a Game Design – Compliant engine using UML » dans *Game Programming Gems3*, Charles River Media. Inc., 2002. pp 73-82.
- [Diaz01] Diaz, M. et al. 2001. *Les réseaux de Pétri – Modèles fondamentaux*. Hermes Science Publications Ed. Paris. ISBN 2-7462-0250-6

- [Dico01] Dictionnaire en ligne disponible à l'adresse <http://encyclopedia.thefreedictionary.com>
- [Dico02] Dictionnaire en ligne disponible à l'adresse <http://www.free-definition.com/Game.html>
- [Dico03] Le Grand Dictionnaire Terminologique disponible sur <http://www.granddictionnaire.com>
- [Douglas01] Douglas, J. Yellowlees et Andrew Hargadon. *The pleasures of immersion and engagement : schemas, scripts ant the fifth business*, dans Digital Creativity. Vol 12, No. 3, pp 153-166.
- [Durard97] Durard, Alain, Julien Huart et Sylvie Leleu-Merviel. 1997. *Vers un modèle de programme pour la conception de document* dans Hypertextes et hypermédias. Ed. Hermes, Volume 1, No 1/1997, pp 79-101.
- [Flury03] Flury, T. et al. 2003. "A model and Software architecture for location-management in Smart Devices/Ambient communication environments", dans *Communicating with smart objects*, Kintzing, C. G. Poulain, G. Privat, et P.N. Favennec. 2003. Ed. Kogan Page Science, ISBN 1-9039-9636-8, Chapitre 7, pp 71-90.
- [Fritz00] Fritz, J. 2000. *Schemata und Computerspiel*. Computerspiele auf dem Prüfstand. No 12, 94-98 Ed. Bundeszentrale für politische Bildung. 2002.
- [Gal02] Gal, V. et C. Le Prado, S. Natkin, L. Vega. 2002. Writing for video games - Proceedings VRIC, Laval France, 2002.
- [Genvo03] Genvo, S. 2003. *Introduction aux enjeux artistiques et culturels des jeux video*, L'Harmattan Ed. Paris.
- [Guardiola00] Guardiola, E. 2000. *Ecrire pour le jeu*, Dixit Ed. Paris. ISBN 2-84481-025-X
- [Hunicke04] Hunicke Robin, Marc LeBlanc et Robert Zubek. 2004. *MDA : A formal approach to Game Design and Game Reseach*. Game Design and Tuning Workshop. GDC, San Jose, CA.
- [Jenkins02] Jenkins, H. 2002. *Game Design as Narrative Architecture*, dans First Person. Pat Harrington et Noah Frup-Waldrop (Eds.), Cambridge: MIT Press, 2002.
- [Jensen81] Jensen, Kurt. 1981. *Coloured Petri Nets and the invariant-method*. Theoretical Computer Science 14 (1981) North-Holland Publishing Company. pp 317-336.
- [Juul99] Juul, Jesper. 1999. *A Clash between Game and Narrative*. A thesis on computer games and interactive fiction. Thèse en maîtrise. Institut Nordique de Langues et Littérature, University of Copenhagen. Février

1999. –Traduction à l’anglais Avril 2001.

- [Kerlow00] Kerlow, Isaac V. 2000, *The art of 3-D computer animation and imaging*. Ed. Wiley. ISBN 0-471-36004-X. pp 77-82.
- [Kreimeier03a] Kreimeier, B. 2003. *Game desing methods*. IGDA Roundtable. GDC, San José, CA.
- [Kreimeier03b] Kreimeier, B. 2003. *Game design methods: A 2003 survey*. Gamasutra, March. <http://www.gamasutra.com/features/20030303/kreimeier.shtml>
- [Laurel91] Laurel, Brenda. 1991. *Computers as Theatre*. Ed. Addison-Wesley.
- [Levi-Strauss58] Levi-Strauss, C. 1958. *Anthropologie Structurale*. Ed. Plon, Paris.
- [Lindley02] Lindley, Craig. 2002. *The Gameplay Gestalt, Narrative, and Interactive Storytelling*. Computer Games and Digital Cultures Conference, juin 6-8, Tampere, Finland, 2002.
- [Mateas03] Mateas, M. et A. Stern. 2003. *Integrating Plot, Character And Natural Language Processing in the Interactive Drama Façade*. 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment, TIDSE 2003, Darmstadt, Allemagne, mars 2003.
- [Natkin02] Natkin, S. 2002. *L’utilisation de la théorie des jeux pour les jeux vidéo*. Master class supports DESS JVMI at CNAM, Paris.
- [Natkin03] Natkin, S. 2003. *Une architecture pour jouer à un million de joueurs*, Les Cahiers du Numérique, Paris.
- [Natkin04] Natkin, S. 2004. *Jeux Vidéo et Médias du XXIe siècle*. Ed. Vuirbert. Paris 2004.
- [Pearce02] Pearce, Celia. 2002. *Towards a Game Theory of Game*. MIT Press.
- [Printz02] Printz J., *Le génie Logiciel*, Ed QSJ, Paris, 2002
- [Propp70] Propp, Vladimir. 1970. *Morphologie du conte*. Ed. Seuil, Paris.
- [Rollings00] Rollings, A. and D. Morris. 2000. *Game Architecture and Design*, Coriolis Ed. Scottsdale. ISBN 1-57610-425-7
- [Rollings03] Rollings, A. et Ernest Adams. 2003. *Andrew Rollings and Ernest Adams on Game Design*. Ed. New Riders. ISBN 1-5927-3001-9
- [Rucker02] Rucker, Rudy. 2002. *Software engineering and Computer Games*. Ed.

Addison Wesley. San Jose CA.

- [Sanchez04] Sanchez-Crespo, Daniel. 2004. *Core Techniques and Algorithms in Game Programming*. Ed. New Riders.
- [Siang04] Siang, A.C. et G.S.V. Radha Krishan Rao. 2004. *Designing Interactivity in computer games: a UML Approach*. International Journal of Intelligent Games and Simulation. Vol. 3 Numéro 2. Août 2004. pp 62-69.  
<http://www.scit.wlv.ac.uk/~cm1822/ijigs32.htm>
- [Silva85] Silva, M. 1985. *Las redes de Pétri: en la automatica y la informatica*. Ed. AC, Madrid. Espagne.
- [Southey04] Southey, Finnegan et Holte Robert. 2004. *Semi-Automated Gameplay Analysis*. Dep. d'informatique, Université d'Alberta. AAAI 2004 Workshop.
- [Szilas03] Szilas, N. 2003. *IDtension: a narrative engine for Interactive Drama*. 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE 2003), Darmstadt, Allemagne. March 24–26 2003.
- [Thorhauge03] Thorhauge, Anne Mette. 2003. *Player, reader and social actor*. Université de Copenhagen.
- [Todorov73] Todorov, Tzvetan. 1973. *Qu'est ce que le structuralisme? Tome 2, Poétique*. Ed. Seuil, Paris.
- [Tosca03] Tosca, S. 2003. *The Quest Problem in Computer Games*. IT University of Copenhagen. TIDSE, Damstandt, 2003.
- [Vanoye01] Vanoye, F. and A. Goliot-Lété. 2001. *Précis d'analyse filmique*, Nathan Université Ed. France.
- [Vidal92] Vidal-Naquet, G. and A. Choquet-Geniet. 1992. *Réseaux de Pétri et systèmes parallèles*. Armand Colin Ed. Paris.
- [Villette04] Villette Numérique. 2004. Table ronde : *La narration interactive ou le roman d'un joueur*. Cité des Sciences et de l'industrie, Parc de la Villette. Octobre 2 du 2004.
- [Watt03] Watt, A. et F. Policarpo. 2003. *3D Games, Animation and Advanced Real-Time Rendering*, Ed Addison-Wesley. Vol II, ISBN 0-201-78706-7.

Jeux Vidéo et  
Logiciels:

- [B&W03] *Black & White*. Electronics Arts, Black & White Studios and Lionhead Studios 2000-2003. Jeu de stratégie (PC).

- [Blizzard01]      *Diablo*. 2001. Blizzard Entertainment.
- [Criterion04]      *Renderware Studio*. 2004. Criterion Software. Logiciel Moteur de jeu.
- [Konami02]      *Silent Hill 2*. 2002. Konami, Horreur-adventure game (PC).
- [Myst]      *Myst*. 1994. Ubisoft, Cyan. Adventure game (PC).
- [Polyphony98]      *Grand Turismo*. 1998. Polyphony Digital Inc. Sony Computer Entertainment Europe.
- [Virtools04]      *Virtools Dev. 3.0*. 2004. Virtools.



## Annexes

Dans le chapitre 5 nous avons présenté l'application de notre méthode d'analyse à l'exemple concret de DOOM<sup>3</sup>. A cette occasion et pour donner une explication plus compréhensible, nous avons indiqué seulement quelques-uns des éléments. Dans le paragraphe qui suit nous présentons la totalité des éléments générés par le modèle d'analyse.

Les tableaux ci-dessous décrivent les transactions à exécuter par chacune des quêtes du niveau 9 du jeu DOOM<sup>3</sup>. Nous rappelons que, pour les transactions multiples, nous indiquons de manière exhaustive les lieux d'exécution possibles dans le scénario. Les lieux d'exécution ainsi que leurs antécédents de lieux sont indiqués selon la carte du niveau dans la figure 5.7 (page 111).

### Quête 1 : Localiser la salle de communication principale.

	<i>lieux</i>	<i>antécédents de lieux</i>
A. Ouvrir_porte	{entrée_garage}, {hall_comm}	{{entrée_garage}, {pont, hall_comm}}
B. Prendre_armure	{garage_comm}	{entrée_garage, garage_comm}
C. Prendre_arme	{garage_comm}	{entrée_garage, garage_comm}
D. Tuer_monstre	{garagé_comm, jonc_accès1, jonc_accès2, couloir_comm}	{ {entrée_garage, garage_comm}, {jonc_accès1, garage_comm}, {jonc_accès2, hall_comm}, {jonc_accès2, couloir_comm}}
E. Appeler_ascenseur	{garage_comm}	{entrée_comm, garage_comm}
F. Activer_borne_information	{hall_comm., salle_comm }	{{pont, hall_comm}, {jonc_accès3, salle_comm}}

Le réseau de transactions correspondant à la quête 1 est défini par  $TR=\{A..F\}$ , où le langage d'ordonnancement des actions est :

$Precond(Ouvrir\_porte)= 1 \vee Avant(Activer\_borne\_information)$

$Precond(Prendre\_armure)= Avant(Ouvrir\_porte)$

$Precond(Prendre\_arme)= Avant(Ouvrir\_porte)$

$Precond(Tuer\_monstre)= Avant(Prendre\_armure) \wedge Avant(Prendre\_arme)$

$Precond(Appeler\_ascenseur)= Avant(Ouvrir\_porte)$

$Precond(Activer\_borne\_information)= Avant(Ouvrir\_porte) \vee Avant(Appeler\_ascenseur)$

## Quête 2 : Traverser l'ingénierie et localiser le centre de contrôle satellite.

	<i>lieux</i>	<i>antécédents de lieux</i>
G. Recuperer_PDA	{salle_comm, sist_comm, ctrl_sat}	{{jonc_accès3, salle_comm}, {unité_ing_comm1_b, sist_comm}, {ctrl_termique, ctrl_sat}}
D. Tuer_monstre	{jonc_accès3, unité_ing_comm1_haut,  unité_ing_comm1_bas,  unité_ing_comm1_a,  stock_ing_comm1,  unité_ing_comm,  sist_comm, unité_ing_comm2,  unité_ing_comm2,  annexe_commun,  ctrl_termique}	{jonc_accès3, salle_comm}, {unité_ing_comm1_haut, jonc_accès3}, {unité_ing_comm1_haut, unité_ing_comm1_bas}, {unité_ing_comm1_bas, unité_ing_comm1_b, stock_ing_comm1, unité_ing_comm1_a}, {stock_ing_comm1, unité_ing_comm1_a}, {unité_ing_comm1_a, unité_ing_comm}, {unité_ing_comm1_b, sist_comm}, {sist_comm, bureau_securité, unité_ing_comm2}, {unité_ing_comm2_entrée, unité_ing_comm2}, {unité_ing_comm2, annexe_commun}, {annexe_commun, ctrl_termique}}
A. Ouvrir_porte	{jonc_accès3, unité_ing_comm2}	{{salle_comm, jonc_accès3}, {bureau_securité, unité_ing_comm2_entrée}}
E. Appeler_ascenseur	{unité_ing_comm1_haut,  unité_ing_comm1_b,  ctrl_termique }	{{unité_ing_comm1_haut, jonc_accès3}, {unité_ing_comm1_a, unité_ing_comm1_b}, {annexe_commun, ctrl_termique}}
H. Activer_robot	{bureau_securité}	{unité_ing_comm2_entrée, bureau_securité}
F. Activer_borne_information	{bureau_securité,  ctrl_sat}	{{unité_ing_comm2_entrée, bureau_securité}, {ctrl_termique, ctrl_sat}}

Le réseau de transaction pour cette deuxième quête est défini par  $TR = \{A, D, E, F, G, H\}$ . Le langage d'ordonnancement est le suivant :

Precond(Recuperer\_PDA)= 1

Precond(Ouvrir\_porte)= Avant(Recuperer\_PDA)  $\vee$  Avant(Activer\_robot)

Precond(Tuer\_monstre)= Avant(Prendre\_armure)  $\wedge$  Avant(Prendre\_arme)

Precond(Appeler\_ascenseur)= 1

Precond(Activer\_borne\_information)= 1

Precond(Activer\_robot)= 1

## Quête 3 : Envoyer un signal de détresse à la flotte orbitale.

	<i>lieux</i>	<i>antécédents de lieux</i>
F. Activer_borne_information	{ctrl_sat}	{ctrl_termique, ctrl_sat}

Le réseau de transaction,  $TR=\{F\}$  définit cette petite quête. Un seul énoncé en langage d'ordonnement est nécessaire pour décrire cette quête :

$Precond(Activer\_borne\_information)=1$

**Quête 4 : Retourner au salon et aller à la station du monorail pour atteindre les Laboratoires Delta.**

	<i>lieux</i>	<i>antécédents de lieux</i>
G. Récuperer_PDA	{ ctrl_sat }	{ ctrl_termique, ctrl_sat }
E. Appeler_ascenseur	{ ctrl_sat, entretien_comm, accès_monorail }	{{ctrl_sat}, {ctrl_sat, entretien_comm}, {hall_comm, accès_monorail}}
D. Tuer_monstre	{entretien_comm}	{ctrl_sat, entretien_comm }
A. Ouvrir_porte	{hall_comm}	{pont, hall_comm}

Le réseau de transaction pour cette quête est défini par  $TR=\{A, D, E, G\}$ . Nous donnons ci-dessous le langage d'ordonnement décrivant cette dernière quête :

$Precond(Récuperer\_PDA)=1$

$Precond(Appeler\_ascenseur)=1 \vee Avant(Récuperer\_PDA)$

$Precond(Tuer\_monstre)=Avant(Prendre\_armure) \wedge Avant(Prendre\_arme)$

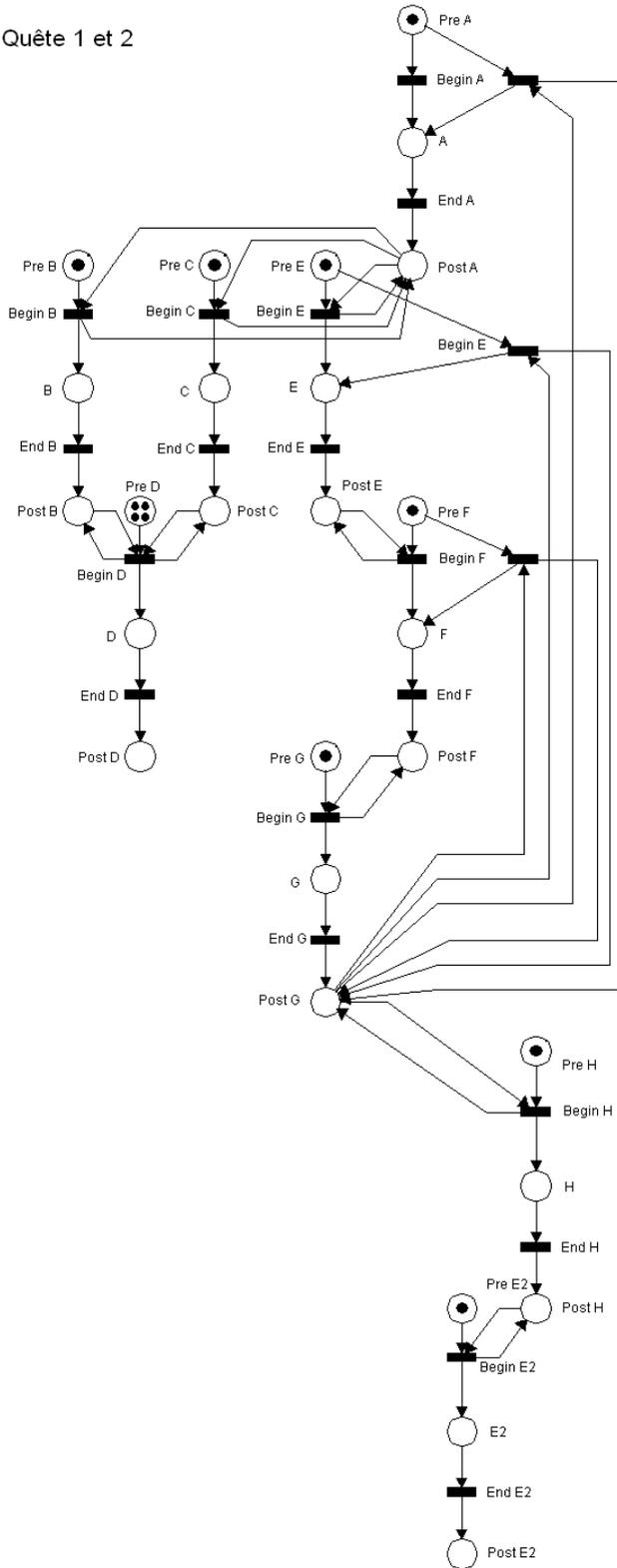
$Precond(Ouvrir\_porte)=Avant(Activer\_borne\_information)$

Le réseau de Pétri coloré de la figure A.1 et les hypergraphes des figures A.2 à A.14 montrent intégralement le modèle d'analyse pour le scénario du neuvième niveau. Les figures A.2, A.3, A.4 et A.5 illustrent les changements de la topologie pendant l'exécution de la première quête. Nous avons donné une description détaillée du scénario pour cette quête dans le chapitre 5. Cependant, dans ce paragraphe, nous limitons cette description textuelle à la description des actions conséquentes dans la topologie en termes de remplacements. Ceux-ci sont exprimés graphiquement par des hypergraphes.

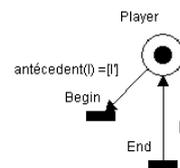
Comme nous avons indiqué précédemment (voir chapitre 5), dans les hypergraphes les cercles modélisent les locii et les flèches, les arcs. Ces derniers sont identifiés par des lettres en minuscule. Les flèches en gras indiquent les nouveaux arcs remplacés. Les lettres en gris montrent les arcs vides à remplacer éventuellement.

Les quatre quêtes du niveau partagent plusieurs transactions. Ceci complexifie la modélisation graphique du réseau de Pétri quant à la distinction entre les quêtes. Cependant, l'utilisation des réseaux de Pétri colorés permet la construction d'un modèle réduit tenant compte de la topologie. Ceci permet la distinction entre chacune des exécutions nécessaires à une quête en particulier. Dans la figure A.1, nous modélisons les quêtes 1 et 2.

Quête 1 et 2



Note :



Toutes les transitions partagent la place Player :

- Pour les transitions Begin, elle est une pré-condition
- Pour les transitions End, elle est une post-condition

Figure A.1 : Réseau de transactions (quêtes 1 et 2) du neuvième niveau de DOOM<sup>3</sup>.



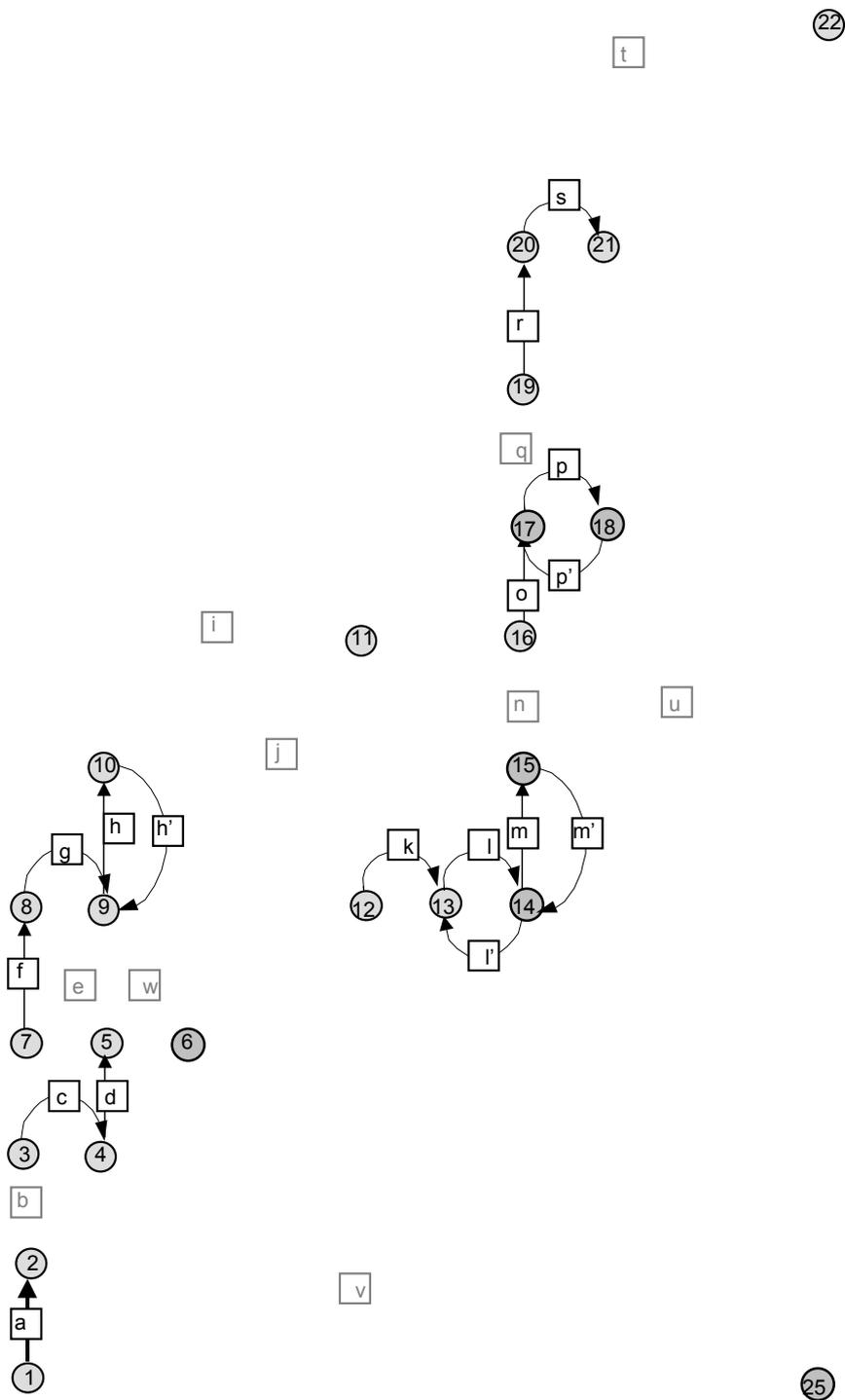


Figure A.3 : Nouvelle hypergraphe (H1) après avoir ouvert la porte d'entrée.

Le remplacement modelisant l'ouverture de la porte d'entrée du garage est le suivant :

$$r_1 = \{ (\{\}, \{\}) | (\{1\}, \{2\}) \}$$

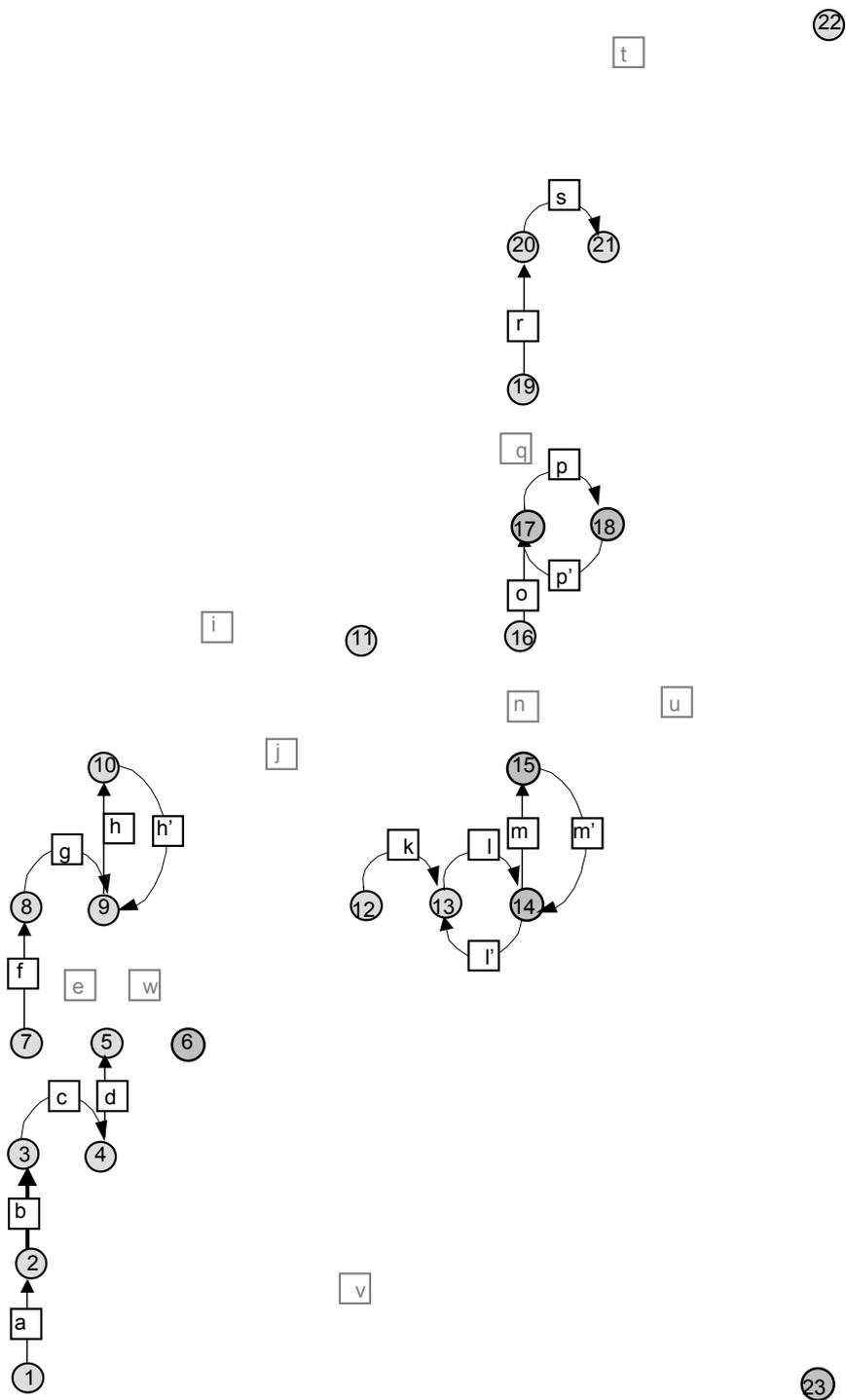


Figure A.4 : Hypergraphe (H2) après avoir appelé l'ascenseur.

Le remplacement suivant indique l'utilisation de l'ascenseur du garage :

$$r_2 = \{ (\{\}, \{\}) | (\{2\}, \{3\}) \}$$

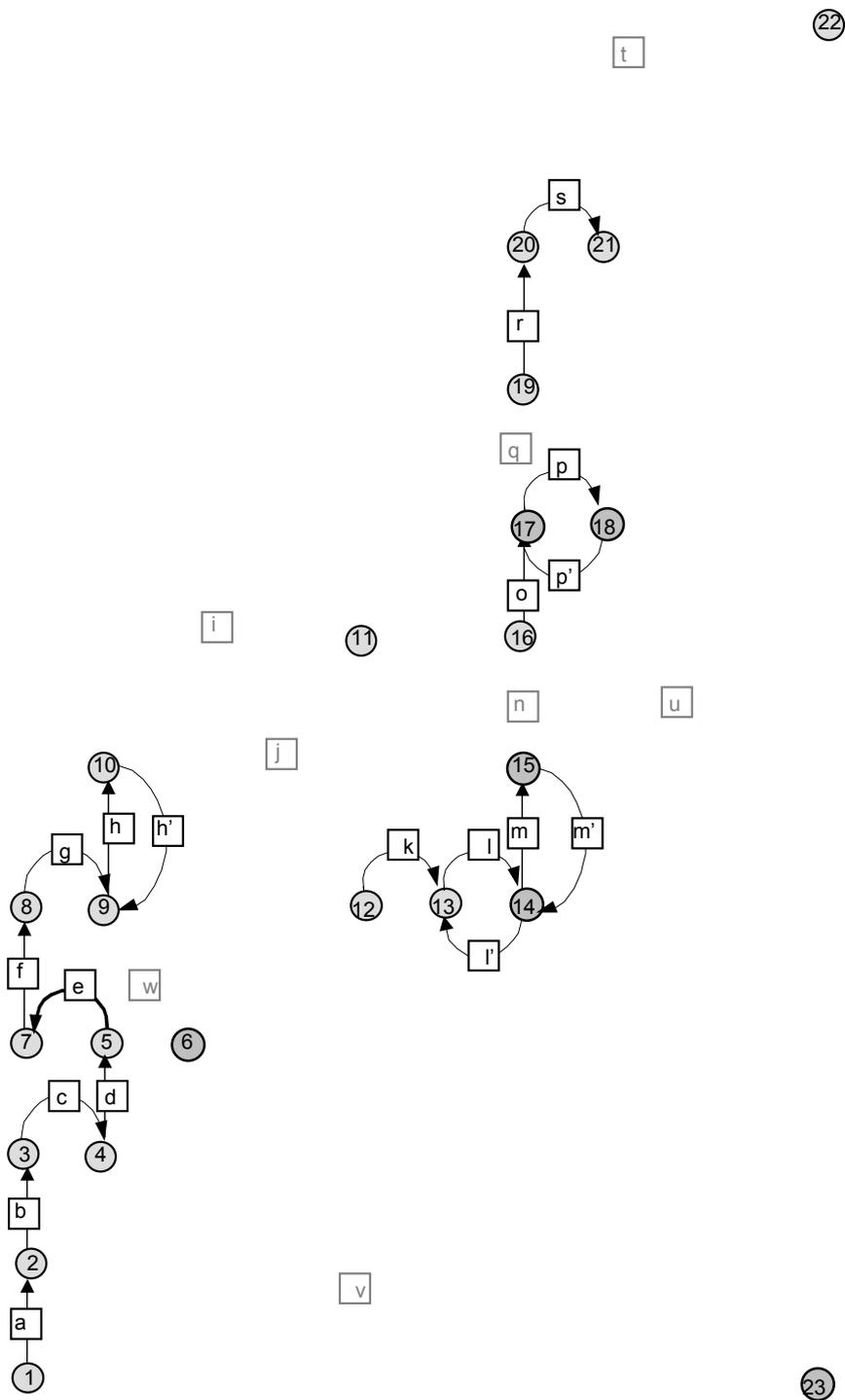


Figure A.5 : Hypergraphe (H3) après avoir activé la borne d'information.

L'activation de la borne d'information disponible au hall permet le passage à jonction d'accès 2.

Cette nouvelle possibilité d'accès est modélisée par le remplacement suivant :

$$r_3 = \{ (\{\}, \{\}) | (\{5\}, \{7\}) \}$$

t

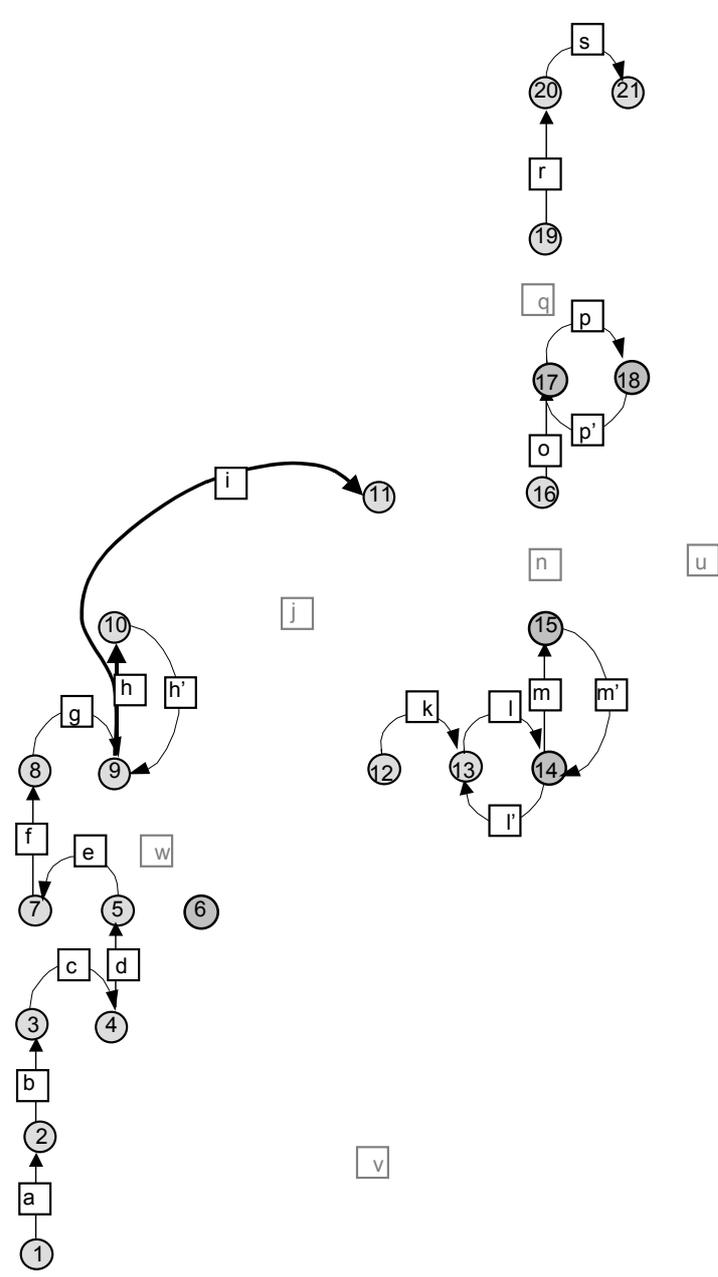


Figure A.6 : Hypergraphe (H4) après avoir récupéré le PDA.

Dans la jonction d'accès 3, une porte devient accessible grâce au PDA récupéré. Le remplacement suivant permet de communiquer les locus 9 et 11 :

$$r_4 = \{ (\{9\}, \{10\}) \mid (\{9\}, \{10, 11\}) \}$$

t

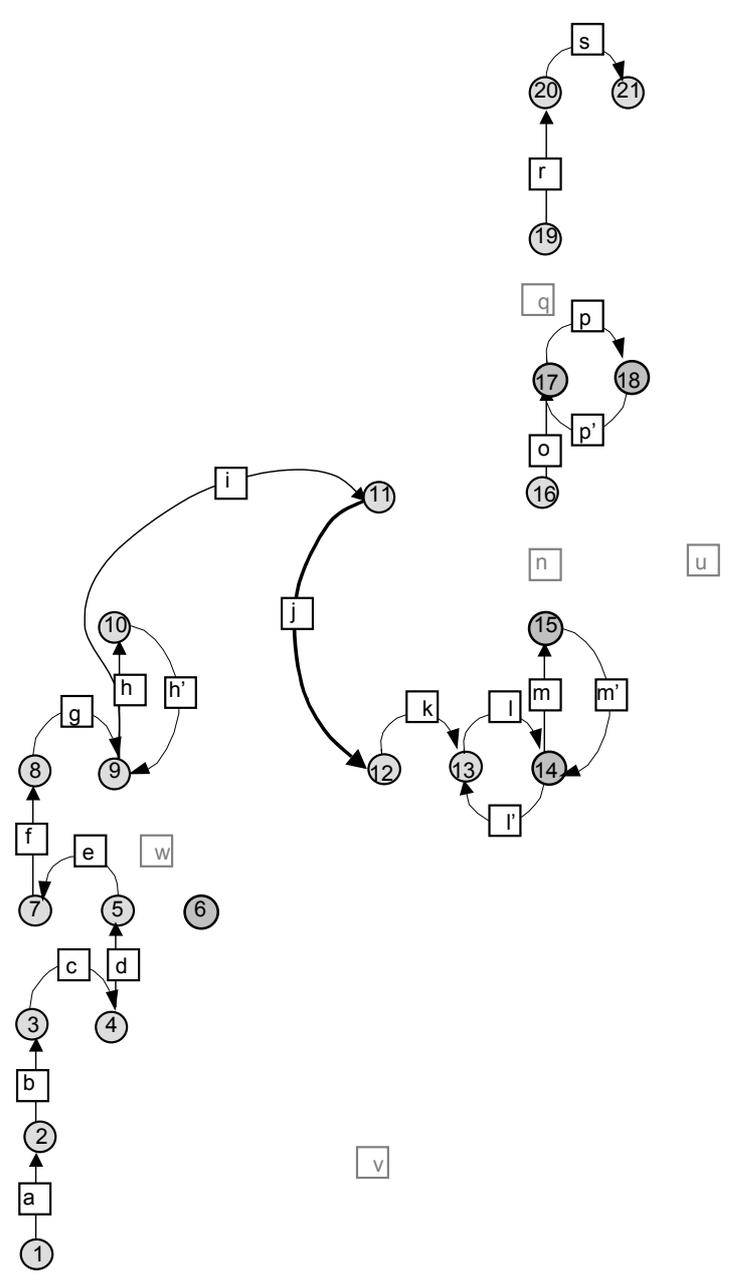


Figure A.7 : Hypergraphe (H5) après avoir appelé l'ascenseur.

Le remplacement suivant modélise la possibilité de descendre, par l'ascenseur, de la mezzanine dans l'unité d'ingénierie communication :

$$r_5 = \{ (\{\}, \{\}) | (\{11\}, \{12\}) \}$$

t

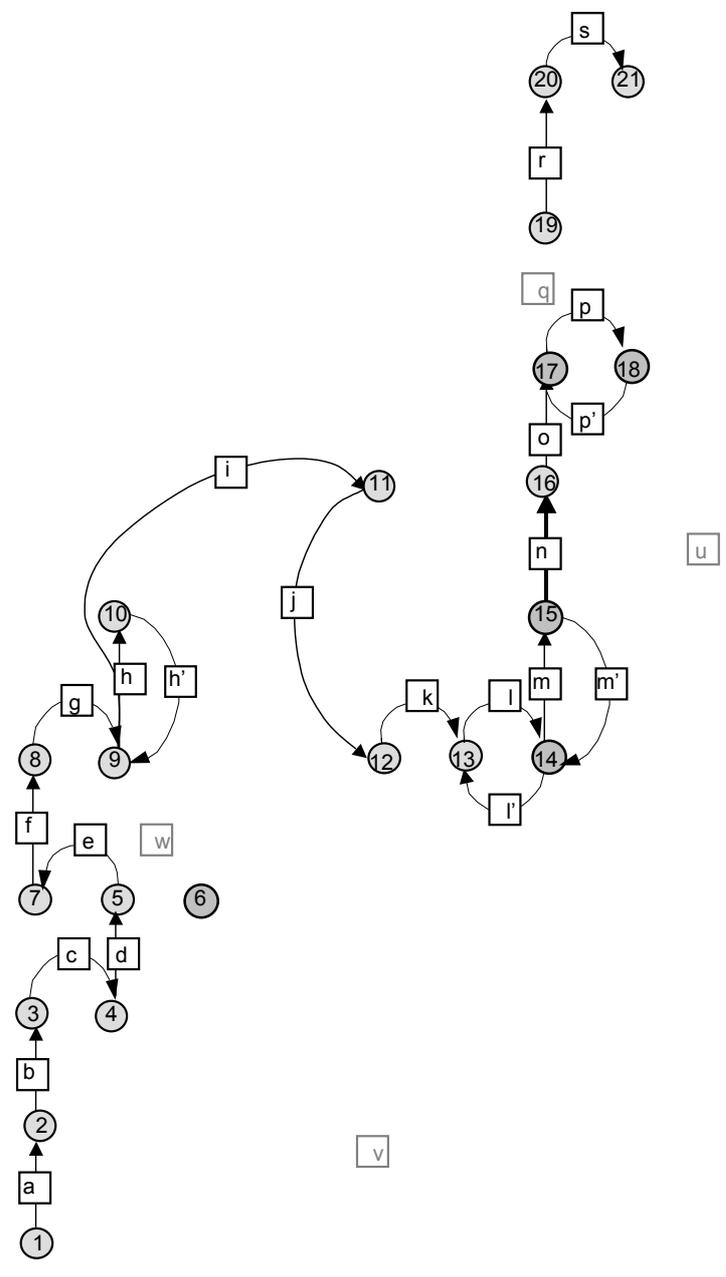


Figure A.8 : Hypergraphe (H6) après avoir appelé l'ascenseur.

Les locus 15 et 16 communiquent par un ascenseur. Le remplacement suivant modélise cette relation :

$$r_6 = \{ (\{\}, \{\}) | (\{15\}, \{16\}) \}$$

t

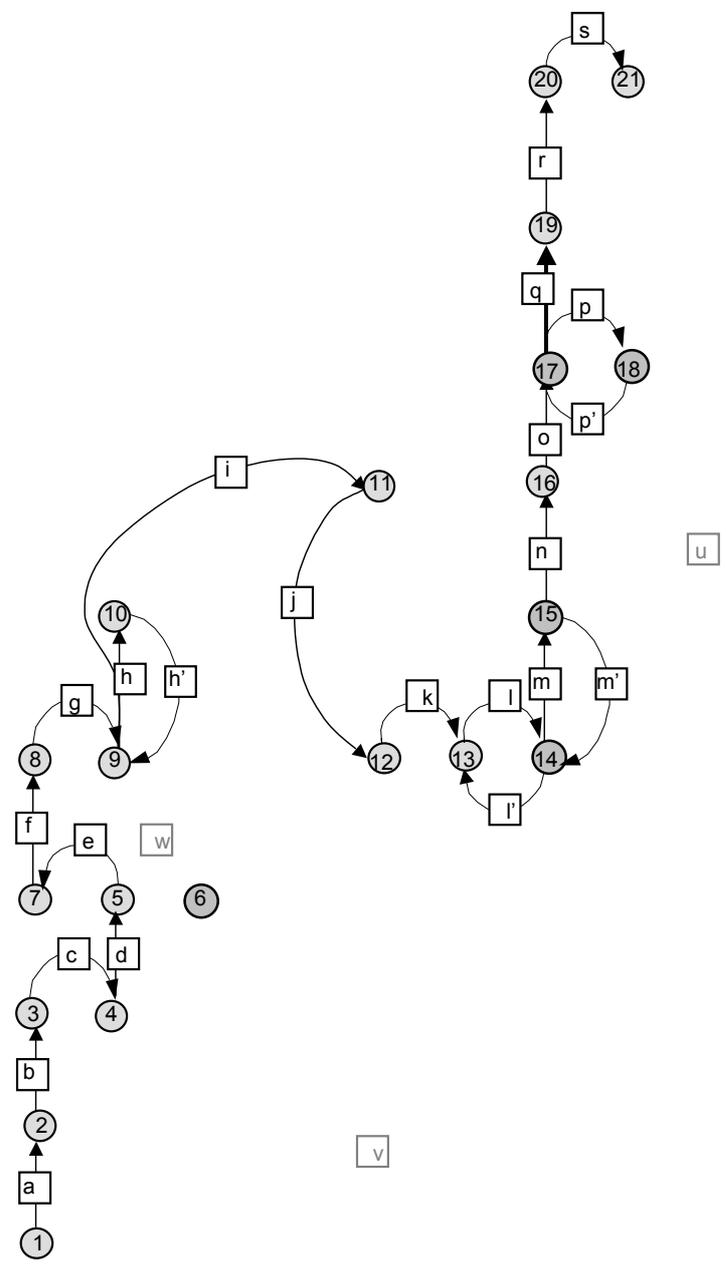


Figure A.9 : Hypergraphe (H7) après avoir activé le robot.

La porte d'entrée à l'unité d'ingénierie communication 2 ne peut être ouverte que par des robots. Le remplacement suivant modélise cette possibilité.

$$r_7 = \{ (\{\}, \{\}) | (\{17\}, \{19\}) \}$$

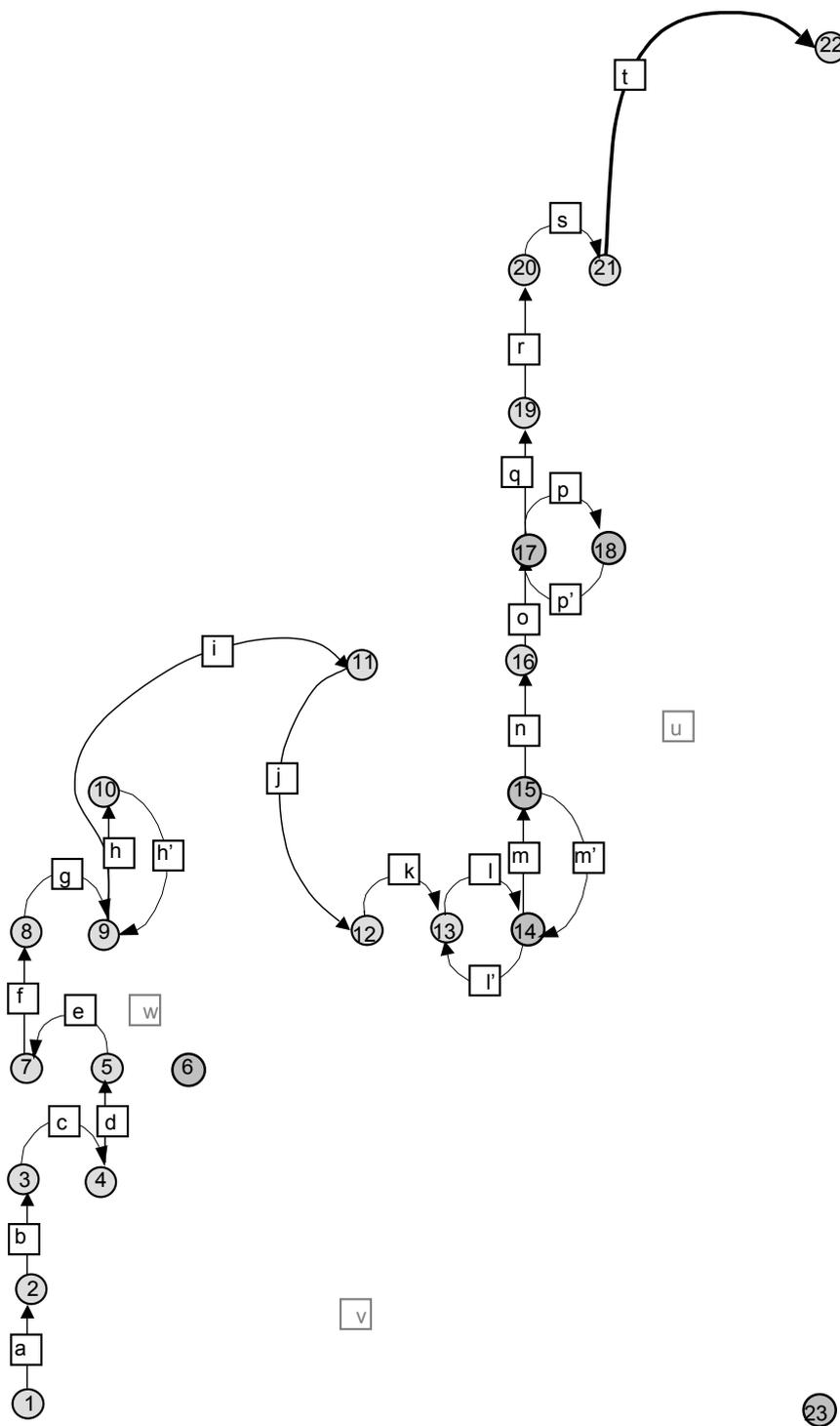


Figure A.10 : Hypergraphe (H8) après avoir appelé l'ascenseur.

Les zones de contrôle thermique et de contrôle satellite communiquent par un ascenseur. Le remplacement suivant modélise l'utilisation de cet ascenseur :

$$r_8 = \{ (\{\}, \{\}) | (\{21\}, \{22\}) \}$$

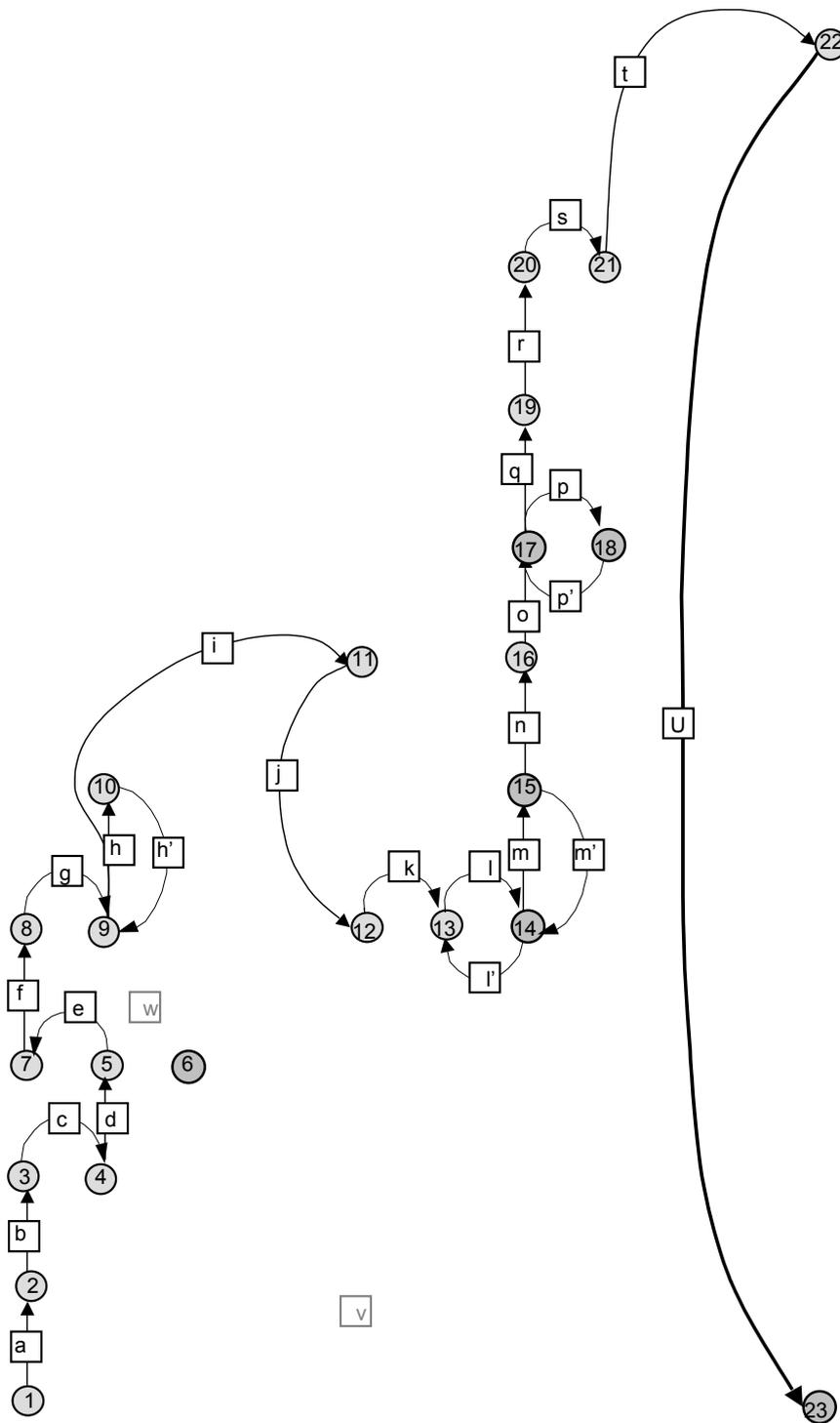


Figure A.11 : Hypergraphe (H9) après avoir récupéré la PDA.

Grâce au nouveau PDA récupéré, l'ascenseur permet l'accès à la zone d'entretien. Cette possibilité est modélisée par le remplacement suivant :

$$r_9 = \{ (\{\}, \{\}) | (\{22\}, \{23\}) \}$$

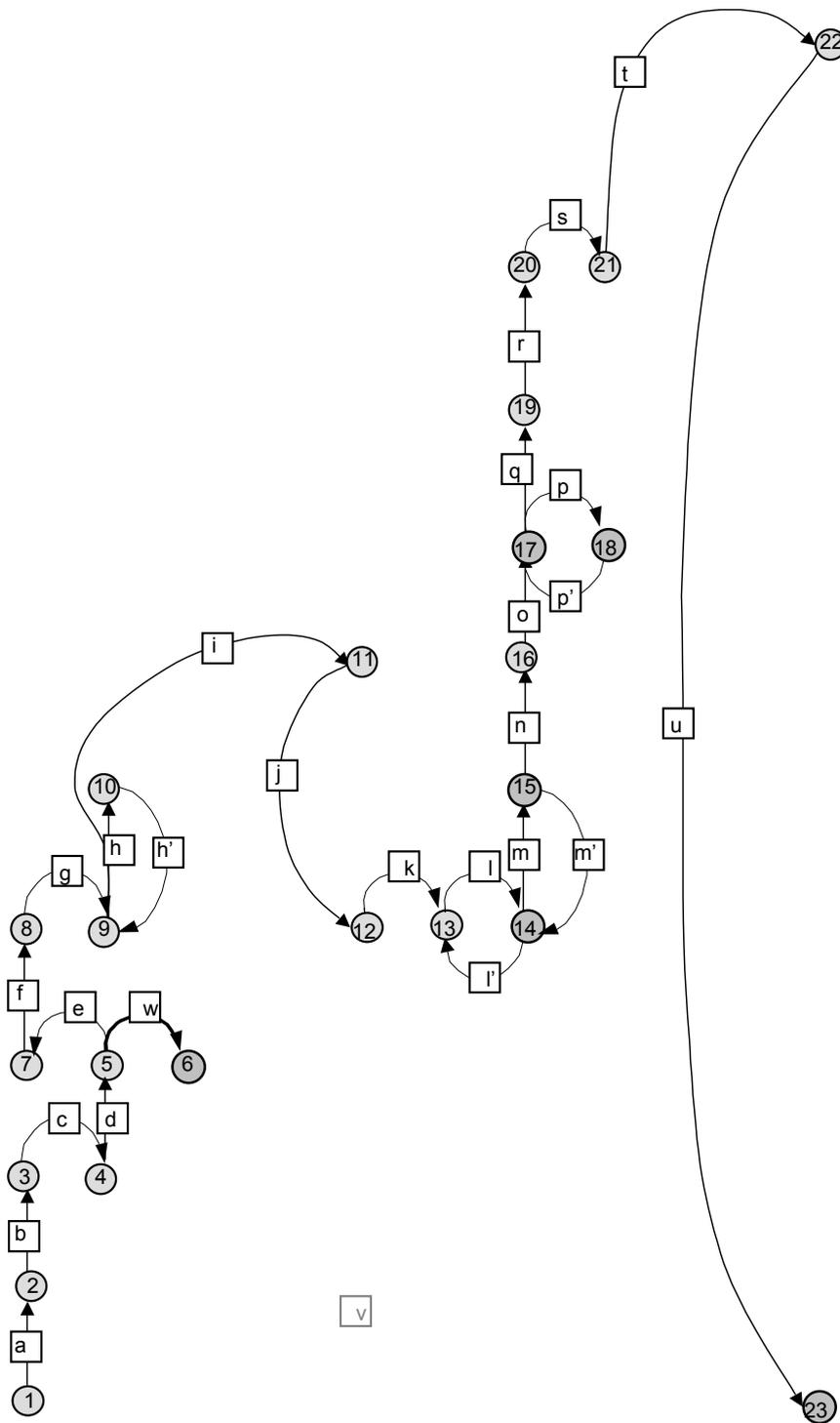


Figure A.12 : Hypergraphe (H10) après avoir envoyé ou non le signal de détresse.

Une fois que le joueur est arrivé au centre de contrôle satellite et après avoir envoyé le signal de détresse (ou non), l'objectif 2 lui est confié. En même temps, la porte du hall communication donnant au monorail est débloquée. Cette possibilité est modélisée par le remplacement suivant :

$$r_{10} = \{ (\{\}, \{\}) | (\{5\}, \{6\}) \}$$

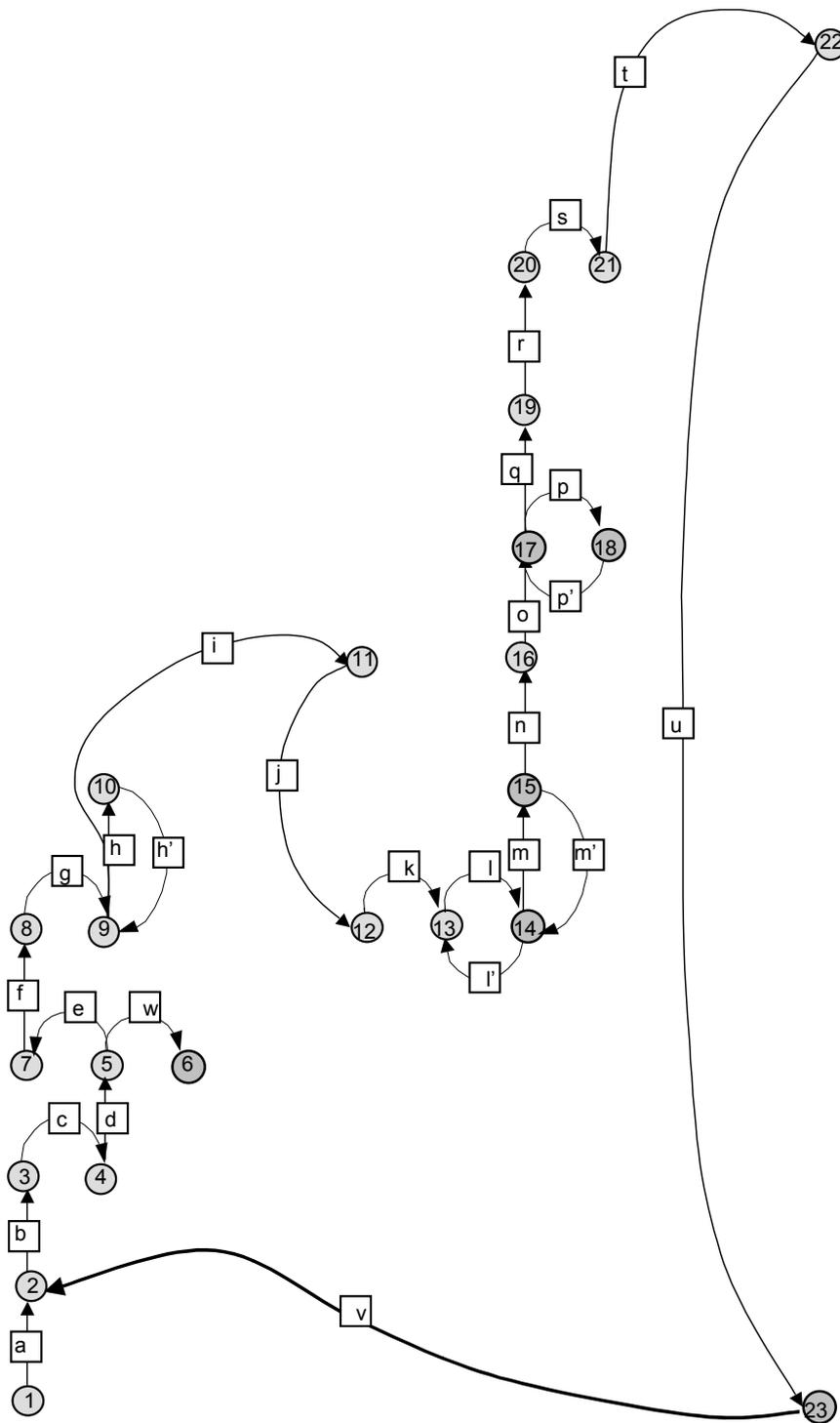


Figure A.13 : Hypergraphe (H11) après avoir appelé l'ascenseur.

Un deuxième ascenseur dans le niveau de maintenance permet de communiquer de la zone d'entretien au garage. Le remplacement suivant modélise l'utilisation de cet ascenseur :

$$r_{11} = \{ (\{\}, \{\}) | (\{23\}, \{2\}) \}$$

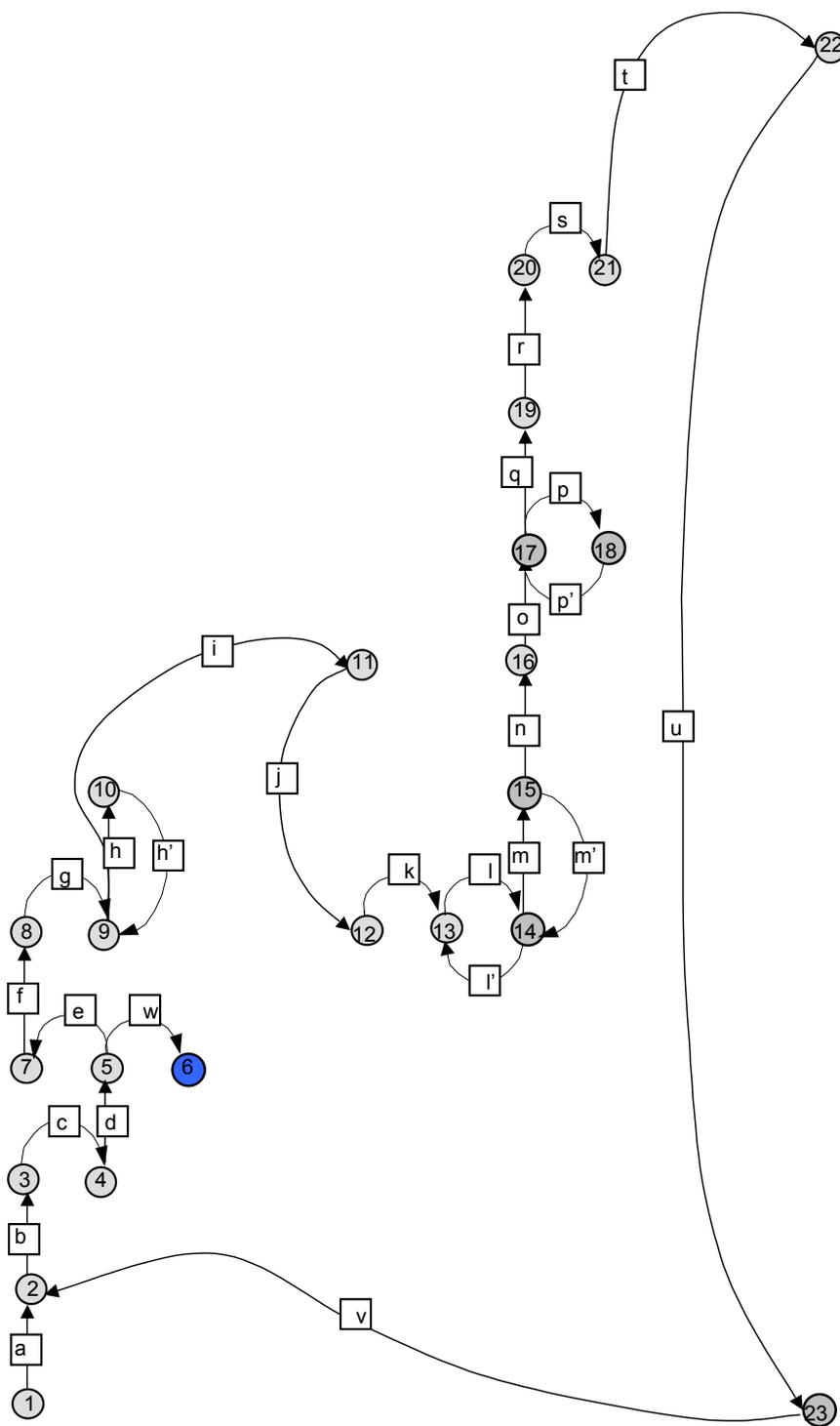


Figure A.14 : Hypergraphe (H12) après avoir appelé l'ascenseur.

Après avoir traversé la porte du hall communication donnant au monorail (voir figure A.12), le joueur appelle un ascenseur pour arriver au quai. Cette transaction lui permet de finir le niveau.

## ***Résumé***

Les jeux vidéo sont étudiés à partir d'un point de vue analytique. Notre travail de recherche est fortement axé sur la méthodologie actuelle d'écriture utilisée dans l'industrie. Cette méthode d'écriture spatiale n'est pas formalisée et la manque d'outils spécialisés complique son processus de création et d'analyse.

Nous proposons l'application de techniques formelles pour la définition des rapports spatio-temporels existants dans les scénarii de jeu. L'analyse est effectuée en deux phases réciproquement dépendantes prenant compte les relations entre l'ordre des actions du joueur et l'univers du jeu. Les réseaux de Pétri sont utilisés pour décrire l'ordonnancement des actions. La topologie des niveaux du jeu est représentée par des hypergraphes.

Nous avons montré comment notre modèle permet l'analyse et la validation de la cohérence des rapports spatio-temporels dans les niveaux de jeu. Enfin, nous avons présenté une possibilité d'intégration de ce modèle dans le processus de conception.

### **Mots-clés :**

Réseaux de Pétri, hypergraphes, modélisation jeux vidéo, analyse jeux vidéo, scénario jeux vidéo.

## ***Abstract***

Video games are studied from an analytical point of view. Our research work is strongly centred on the writing methodology currently used in industry. This original space-writing method is not formalized and the absence of specialized tools complicates its design process and analysis.

We propose the application of formal techniques for definition of existing space-time relationships in the game scenario. The analysis is carried out in two interdependent phases taking into account the relationships between player's actions order and the game univers. Petri Nets are used to describe action sequencing. For game level topologic representation, we used hypergraphs.

We have showed how our model allows analysis and validation of coherence for space-time relationships in game levels. As for design tools, we presented how this model can be integrated in the existing process of *Game Design*.

### **Key Words:**

Petri Nets, Hypergraphs, Video Games Modelling, Video Games Analysis, Video Games Scenario.