CHAPITRE 8

Autres domaines d'application

Ce chapitre suggère des applications possibles du modèle acteur-observateurs aux conceptions d'interface utilisateur et de jeux vidéo.

8.1 Application à la conception de jeu

De tous les produits multimédia, le plus avant-gardiste et novateur est assurément le jeu vidéo. A tel point que la direction de Nintendo refuse de voir catégoriser son entreprise sous le terme galvaudé de multimédia. Les consommateurs de jeu vidéo sont en quelque sorte les plus grands utilisateurs domestiques d'informatique et ont été nourris à une forme de logiciel très distincte du logiciel bureautique :

- **Un logiciel actif**. La concurrence et l'utilisation de télécommunications pour la gestion d'environnements partagés font presque systématiquement partie des technologies informatiques employées dans les jeux vidéo.
- Un logiciel interactif. Le rôle de l'utilisateur et les modes de contrôle du logiciel constituent souvent le fondement et l'intérêt du logiciel.
- Le recours à la 3D et au son. De façon presque systématique, le logiciel de jeu a recours désormais à des interfaces

multimodales tridimensionnelles permettant de simuler le monde réel.

- Le zéro-défaut. Les exigences de qualité et de performance des logiciels de jeu sont renforcées par leur utilisation dans un contexte véritablement grand-public et par la pratique du jeu qui ne doit souffrir ni de lenteurs ni d'erreurs de programmation fatales[1].

Ces trois premières caractéristiques pourraient rendre le modèle acteur-observateur adapté à la conception de jeu vidéo

8.1.1 L'ingénierie traditionnelle du jeu vidéo

Les méthodes de conception de jeu vidéo se rapprochent plus de l'artisanat que de la production industrielle de biens culturels ou matériels.

L'expérience de NAMCO

La société Namco[2], qui est l'un des premiers producteurs mondiaux de consoles de jeu pour salles et de logiciels pour consoles familiales, a récemment produit une console de jeu de simulation de course automobile d'un réalisme jamais atteint : ***Ridge Racer***. Une équipe d'environ dix personnes a conçu et mis en oeuvre en un an :

- Un circuit graphique à haute performance fondé sur une architecture multiprocesseur comportant une centaine de processeurs de traitement du signal.
- Un système d'exploitation temps-réel adapté à cette architecture (codé en langage machine).

1. On peut se demander dans quelle mesure la distribution de logiciels de jeu sous forme de cartouches a conduit à assimiler les jeux vidéos à un bien matériel donc soumis aux mêmes impératifs de qualité que les produits courants, ce qui n'est pas le cas encore du logiciel de bureautique.

2. Namco est le créateur de Galaxian

- Un logiciel de rendu et d'animation en temps réel (codé en langage machine).
- Un système de détection de choc et de sonorisation (codé en langage machine).
- De nombreux modèles géométriques, textures et bruitages.
- Un meuble avec volant, siège de course et levier de changement de vitesse

Le succès d'un tel projet est en partie dû à l'expérience unique acquise par Namco dans chacun de ces domaines et à l'esprit d'équipe qui règne au sein de chaque projet.

Un besoin de formalisation

Cependant, malgré ses atouts, Namco considère que la conception de jeu vidéo est arrivée à un tournant majeur : les méthodes artisanales de conception de jeux vidéo devront évoluer en raison de la complexité sans cesse croissante des modèles d'interaction et des bases de données graphiques. Dans un contexte où l'arrivée de systèmes grand public à haute performance comme la console *Playstation* de Sony ou *Saturn* de Sega, stimule la concurrence entre concepteurs de jeux vidéo haut de gamme, l'intégration de méthodes avancées de génie logiciel devient un atout capable de faire émerger de nouvelles entreprises.

Rocket Science : une tentative de formalisation

Partant de ce constat, la société Rocket Science a construit un système de formalisation (secret) lui permettant de produire des jeux vidéo à interaction limitée en deux fois moins de temps que le leader mondial, Sega. Ce système est en partie fondé sur les méthodes de production cinématographiques hollywoodiennes. Rocket Science espère tirer de son avance conceptuelle une place de premier plan sur un marché du jeu déjà supérieur en chiffre d'affaires à celui du cinéma.

8.1.2 Utilisation d'acteurs et d'observateurs

Le modèle acteur-observateur pourrait s'appliquer à la conception de jeu vidéo dans la mesure où il améliore les environnements à base de librairies de classes en permettant la spécification d'interactions génériques entre objets. En effet, on peut distinguer dans un jeu deux aspects :

- **le comportement** : il correspond aux personnages autonomes du jeu.
- **la structure interactive** : elle correspond aux règles mettant en relation les personnages du jeu.

Acteurs : les personnages du jeu

Le comportement autonome dans un jeu vidéo est modélisé au sein d'acteurs que l'on peut conserver dans une base de données accessibles à plusieurs équipes de création de jeu. L'utilisation d'objets actifs, distribués ou persistants rend aisée la réalisation de jeu faisant appel à la télévirtualité et plus généralement aux réseaux de communication grand-public.

Observateurs : le scénario du jeu

La structure interactive d'un jeu vidéo est modélisée par une collection d'observateurs (la forme interactive) associée à un graphe d'observations (le scénario d'interaction). Un même scénario d'interaction peut donner lieu, par remplacement des acteurs voire des observateurs, à plusieurs versions d'un même jeu, adaptées aux contraintes d'âge ou de culture du marché visé.

Inspecteurs : création graphique d'un jeu

La conception du jeu repose alors presque exclusivement sur l'édition des acteurs ou des observateurs du jeu grâce au mécanisme d'inspecteur et sur la mise en place de graphes d'interaction grâce aux inspecteurs d'observateurs.

8.2 Application à la conception d'interface utilisateur

La généralisation des interfaces graphiques à base de souris a permis au logiciel de toucher un marché large, celui de la bureautique. Les nouvelles interfaces multimodales fondées sur la simulation semblent être aujourd'hui une voie d'avenir pour une informatique qui, sous le nom de multimédia, se veut d'usage universel ou plus immédiat[1].

8.2.1 Les nouvelles interfaces graphiques

Les interfaces utilisateur intègrent aujourd'hui de plus en plus de caractéristiques provenant de l'informatique graphique, musicale et de la théorie des langages de programmation.

Interfaces multimodales

La notion d'interface graphique a été supplantée par celle d'interface multimodale. Le système d'exploitation MacOS intègre aujourd'hui des mécanismes de synthèse et de reconnaissance de la parole permettant de remplacer l'apparition fréquente de zones de dialogues et d'accéder à des commandes plus rapidement qu'au moyen de menus déroulants.

Nouveaux périphériques d'entrée

Au même moment, on voit apparaître de nouveaux périphériques d'entrée comme la caméra vidéo qui, couplée à un logiciel de reconnaissance simple, permet de saisir les mouvements de mains ou les expressions d'un visage humain pour accélérer l'entrée de paramètres multiples dans une machine [TakeNaga93].

Nouvelles techniques d'interface : l'intrusion de la 3D

La troisième dimension est aussi promise à un grand avenir dans la mesure où elle permet de produire des interfaces

1. Dans le domaine de la gestion de réseau, de nouveaux produits fondés sur la simulation et des interfaces en trois dimensions font leur apparition [Robi95].

reposant sur une simulation du monde réel, plus proche à la fois des jeux vidéo et de ce dont les non informaticiens ont l'habitude dans la vie courante.

Nouveaux langages de programmation

La conception d'interface utilisateur a aussi été l'une des premières disciplines à s'emparer des notions modernes de programmation et à les exploiter largement :

- **Programmation à objets**. Le langage Smalltalk a été conçu en même temps que la première mise en oeuvre d'une interface graphique au travers du modèle MVC [Gold84].
- **Programmation par contraintes** : L'environnement de programmation de la station NeXT utilise des contraintes mécaniques représentées par des ressorts pour spécifier les positions respectives des boutons, champs etc. Le système Allien [BeauCour91] fonde la conhérence d'une interface utilisateur sur une résolution de contraintes.
- **Programmation par prototypes**. La plupart des systèmes d'écriture multimédia [Appl88, AmarBureSoqu93, Macr90] utilisent pour la construction d'interfaces un modèle de programmation proche de celui des prototypes [Lieb86, UngaChamChan91].

Un besoin de structuration

Une partie importante des travaux de recherche menés en interface utilisateur consiste à décomposer un système de manière à rendre aussi indépendants que possible ses divers constituants, et à modéliser leurs relations [HelmHollGang90].

8.2.2 Utilisation d'acteurs et d'observateurs

Toutes ces remarques nous conduisent à envisager la conception d'interfaces utilisateur comme un terrain d'application du modèle acteur-observateur. En effet, ce modèle permet à la fois de décomposer un système en composants

indépendants et de formaliser les relations entre composants du système sous forme de composants réutilisables. Enfin, la notion d'inspecteur se prête bien à la mise en oeuvre d'interfaces, ou de constructeurs d'interfaces, à manipulation directe.

Acteurs : éclatement du programme

Une mise en oeuvre du modèle acteur-observateur consiste à éclater une application en composants actifs élémentaires :

- **Des composants de modélisation**. Ces composants fournissent un moyen de représenter les données formelles d'un problème et les services permettant d'exploiter ces données.
- **Des composants d'interface**. Ces composants proposent une représentation multimodale, générique et active, directement manipulable par un utilisateur.
- **Des composants utilisateur**. Ces composants permettent de représenter au sein du système, un utilisateur réel ou virtuel (voir paragraphe 6.4.1 page 122).

Observateurs : définition de l'interaction

Une fois définis les acteurs élémentaires, on cherche à les structurer en niveaux, parties et plus généralement en interactions. La notion d'observateur permet alors de mettre en oeuvre le formalisme développé. Citons à titre d'exemple deux formalismes courants :

- **Chaîne de commande**. Dans la plupart des applications, l'appel d'une commande d'un menu conduit à chercher à quelle cible doit être appliquée la commande. En général, on maintient d'une part, une structure arborescente et d'autre part, un pointeur d'état sur l'une des feuilles de l'arbre. La recherche de la cible est effectuée en partant de l'état courant et en « remontant » jusqu'à la racine de l'arbre.
- **Navigation**. Ce modèle d'interaction est une application aux interfaces utilisateur des structures d'état. On met en

interaction d'une part, un ensemble statique de composants de modélisation et, d'autre part, en ensemble dynamique de composants d'interface. La sélection par l'utilisateur d'un composant d'interface induit une modification des composants d'interface.