



INSA RENNES

DÉPARTEMENT INFORMATIQUE, 4^{ÈME} ANNÉE

Rapport de Conception

Jeu du Space Battle

Auteurs

Nicolas MALOEUVRE
Bastien VANNIER

Encadrants

27 octobre 2011

Table des matières

Introduction	3
1 Analyse conceptuelle	4
1.1 Diagrammes de cas d'utilisation	4
1.2 Diagramme d'activité	4
1.3 Diagramme d'états-transitions	5
1.4 Diagramme de séquence	5
1.5 Diagramme de classes	5
1.6 Diagramme de composants	5
2 Analyse des choix	6
2.1 Choix des classes	6
2.2 Choix des designs patterns	6
Conclusion	7

Introduction

Space Battle est un jeu de société qui dispose d'un plateau de jeu représentant une partie du système solaire Alpha du Centaure composé de deux astéroïdes, quatre stations orbitales de quatre couleurs différentes situées aux quatre coins du plateau, quatre cases « artefact », quatre cases « station d'énergie ». La boîte de jeu contient également deux dés à six faces, douze pions (trois vaisseaux par joueur : un vaisseau-mère, un battlecruiser et une capsule de survie) et quatorze artefacts.

Chaque joueur est le commandant d'une flotte. Il choisit la couleur de sa station orbitale et de sa flotte. A quatre joueurs, chacun prend une flotte. A trois joueurs, une flotte est laissée de côté. A deux joueurs, chacun prend deux flottes. Une flotte est un groupe de trois vaisseaux : vaisseau-mère, battlecruiser et capsule de survie.

Chaque joueur part pour la course aux artefacts. Sur le plateau de jeu, il y a quatre cases oranges. Ce sont les cases « artefact ». Le but du jeu réside dans le fait d'atteindre l'une de ces cases, prendre un artefact, l'emporter avec son vaisseau et le ramener à sa station orbitale. Pour gagner, il faut être le premier à réussir cette prise trois fois. Nous pouvons également noter la présence de quatre cases grises. Ce sont les cases « station d'énergie ». A partir de ces cases, un joueur recharge son vaisseau en énergie ce qui lui permet dans les tours suivants de viser et tirer sur l'un de ses adversaires depuis n'importe où sur la carte.

Le joueur commence à jouer avec son vaisseau-mère. Si ce dernier se fait détruire à la suite du tir d'un adversaire, le vaisseau-mère est remplacé par le battlecruiser. Enfin si le battlecruiser est détruit, le joueur doit rentrer à sa station orbitale avec sa capsule de survie. Une fois rentré à la station, il lui sera alors possible de repartir avec son vaisseau-mère.

Notre projet est de modéliser, en C++, le jeu de société se basant sur les règles principales énoncées précédemment. Pour y arriver, la première étape consiste en la modélisation de notre future application. Ainsi, dans ce rapport de conception, nous allons vous présenter cette modélisation grâce à divers diagrammes UML.

Chapitre 1

Analyse conceptuelle

L'étude UML¹ nous a permis de modéliser l'application par abstraction des détails inutiles pour la simplifier. Elle nous a également aidé à clarifier les besoins du jeu et les spécificités de l'application afin de la développer. Nous nous sommes concentrés sur différents diagrammes UML qui nous semblaient les plus adaptés à notre besoin. Nous avons donc choisi de modéliser notre projet à travers des diagrammes de cas d'utilisation, un diagramme d'activité, des diagrammes d'états-transitions, un diagramme de séquence, un diagramme de composants et un diagramme de classe.

1.1 Diagrammes de cas d'utilisation

Le diagramme de cas d'utilisation permet la représentation de l'implication de l'utilisateur dans l'application et des interactions entre l'utilisateur et le système. On peut ainsi décrire ce que le futur système devra faire, sans spécifier de quelle manière il le fera. Pour notre projet, nous avons deux diagrammes de cas d'utilisation.

Le premier diagramme représente le lancement du jeu et donc l'implication de l'utilisateur dans l'application pour initialiser une partie.

Le second diagramme représente le joueur en pleine action, c'est-à-dire durant une partie. Les différentes actions qu'il pourra effectuer sont directement liées aux principes du jeu et les relations d'inclusion et d'exclusion permettent d'identifier les enchainements et les interactions avec le système.

1.2 Diagramme d'activité

Le diagramme d'activité permet la représentation du processus d'un tour. On peut ainsi voir toutes les possibilités et tous les cas possibles offerts aux joueurs lors d'un seul tour.

1. Unified Modeling Language

1.3 Diagramme d'états-transitions

Afin de clarifier le déroulement d'un tour effectué par un jour, nous allons décomposer les différentes étapes et possibilités à travers différents diagrammes d'états-transitions.

1.4 Diagramme de séquence

Le premier rectangle symbolise la Vue, représentant l'IHM (Interface Homme Machine). Cette interface permet de faire communiquer les actions choisies par l'utilisateur (lancé de dés, choix de cases) au contrôleur (la partie). Le contrôleur du jeu récupère les données envoyées par la vue afin d'exécuter les traitements, et s'assurer que les règles du jeu soient respectées. Une fois que le contrôleur a effectué les traitements nécessaires, il envoie le résultat aux modèles. Le second rectangle représente le contrôleur (Partie), c'est lui qui va gérer l'ensemble du jeu, comme par exemple le lancement d'une partie. Les autres rectangles symbolisent les modèles. Les lignes verticales pointillées représentent les lignes de vie des objets. Les actions à faire sont énumérées et symbolisées par des guillemets.

1.5 Diagramme de classes

1.6 Diagramme de composants

Le diagramme de composants décrit l'organisation du système du point de vue des éléments logiciels comme les modules (paquetages, fichiers sources, bibliothèques, exécutables), des données (fichiers, bases de données) ou encore d'éléments de configuration (paramètres, scripts, fichiers de commandes). Ce diagramme permet de mettre en évidence les dépendances entre les composants (qui utilise quoi).

Chapitre 2

Analyse des choix

2.1 Choix des classes

2.2 Choix des designs patterns

Conclusion

A travers les informations recueillies durant cette première phase d'analyse, nous avons acquis une première vision du cadre d'évolution de notre projet : « Numérisation de documents en mobilité sur iPhone/iPad pour Mac OS X ». De grands objectifs s'en sont dégagés, nous permettant d'identifier clairement les besoins réels liés à ce dernier. Il s'agit ici de concevoir un système d'applications simple, ergonomique, répondant à un fort contexte de mobilité et pouvant s'adapter à de nombreuses situations, le tout en offrant un confort d'utilisation accru, notamment grâce à une synchronisation transparente et efficace entre divers périphériques de type Apple.

De plus, la réalisation de ce projet nous permettra de prendre en main et d'allier différentes technologies. A travers lui, nous allons découvrir et utiliser des ressources spécifiques (Objective-C, Cocoa, etc) ainsi que des outils informatiques. Notamment, nous verrons plus en détails le développement mobile ainsi que les modes de connexion entre différents supports matériels.

Nous allons maintenant poursuivre notre projet en étudiant et définissant les spécifications générales et fonctionnelles. Pour cela, nous nous intéresserons au fonctionnement interne et à l'architecture des deux applications nécessaires à la réalisation du cahier des charges.

Plus tard, nous verrons en détails l'impact en terme d'échéances de notre participation à l'IT Challenge. Nous aurons à formaliser dans notre dossier d'inscription différents aspects n'étant pas évoqués dans le projet (modèle économique, bénéfices pour les marques partenaires, etc) ce qui nous apportera un point de vue différent sur les attentes que peut avoir un client vis à vis de son produit.