Option IS

Rapport PLT

Final fantastique

FALLOURD Yohann - VAN ESPEN Jérémie

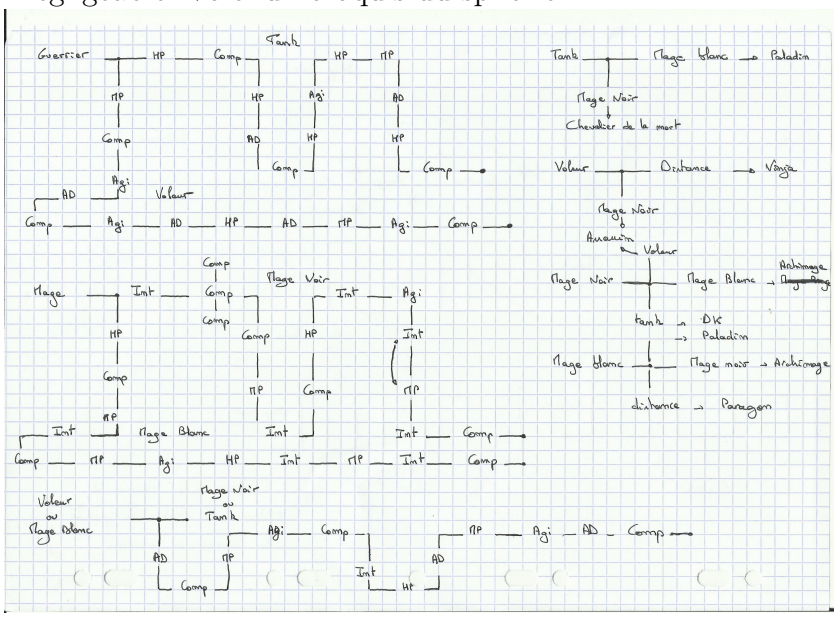
Table des matières

Chapitre 1

Objectif

1.1 Présentation générale

Le jeu est basé sur l'archétype de Final Fantasy X, se distinguant par un système de combat au tour par tour ainsi que son emblématique "sphérier" correspondant à un arbre de talent apportant une liberté de personnalisation non négligeable. Voici un croquis du sphérier :



A la différence du véritable jeu, il n'y aura pas de déplacement libre en temps réel ; le gameplay se résume à un déplacement de points d'intérêts en points d'intérêts, poussé par une histoire typique de cet archétype, où se déroulent différents évènements/combats.

1.2 Règles du jeu

Le coeur du jeu sont les deux personnages auquel le joueur (si il joue seul) a accès. Chacun commence avec une spécialisation particulière de mage et de guerrier choisissant directement de se diriger vers différentes spécialisations. Via une carte du monde, le joueur avance de point en point et déclenche différentes rencontres. Les combats se déroulent en tour par tour et si l'un des joueurs meurt, il revient à la vie avec 1 point de vie. La mort des deux joueurs entraine un retour au dernier point de sauvegarde automatique. Gagner un combat donne de l'expérience, qui donne des niveaux, qui permettent de progresser dans le sphérier.

Il existe également différents objets permettant de rendre de la vie ou d'avoir d'autres effets. De plus, un système d'équipement existe pour rendre le joueur plus fort au cours de l'aventure. En effet, un personnage possède des caractéristiques (Force, Agilité, Intelligence, Points de vie (HP) et Points de magie (MP)) qu'il peut améliorer.

Le jeu se finit lorsque l'aventure est terminée !

1.3 Conception Logiciel

Voici les packages de notre projet :

Package state. Package central qui gère l'état du jeu.

Package engine. Package qui modifie l'état de jeu en fonction de différents inputs.

Package ai. Package qui gère le contrôle par l'ordinateur des monstres et, potentiellement, un personnage principal.

Package server. Package contenant la gestion de l'API du jeu, que ce soit par réseau ou localement.

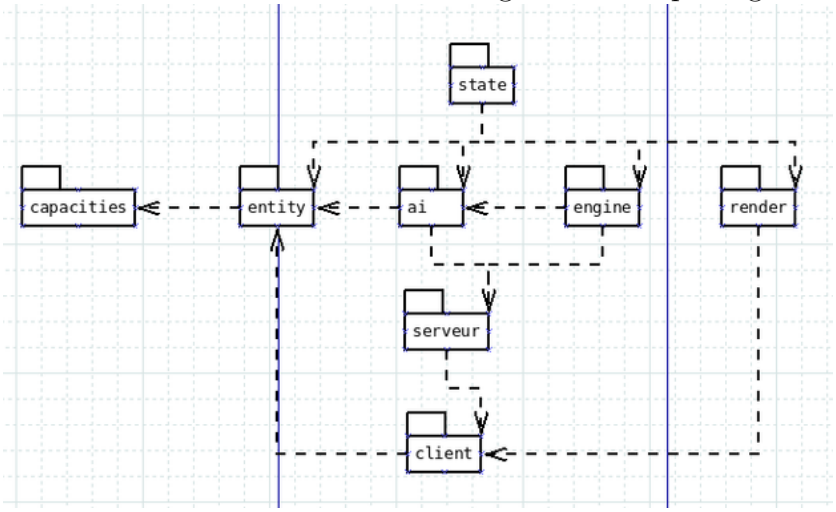
Package render. Associe un affichage graphique du jeu à l'état de jeu.

Package client. Package contenant les différents traitements et commandes faites localement, sur la machine du joueur, afin de produire le comportement souhaité du jeu.

Package entity. Package contenant toutes les entités du jeu. Cela rassemble les Personnages Non Joueurs, les personnages principaux et les différents ennemis.

Package capacities. Package contenant les différents sorts et capacités utilisées par les personnages principaux et les ennemis.

FIGURE 1.1 – Diagramme des packages



Chapitre 2

Description et conception des états

2.1 Description des états

Un état du jeu est formé par un ensemble d'éléments vivants et non vivants ainsi que la situation dont se trouve le joueur.

2.1.1 Etat éléments vivants

Les éléments vivants sont des entités ayant tous des caractéristiques propres : santé max, santé actuelle, mana max, mana actuel, force, agilité et intelligence. On distingue deux types d'éléments vivants :

Character. Ce sont les héros du jeu. Ils pourront s'équiper d'une arme et d'une protection ajoutant des bonus d'attaque et de défence. Ils évolueront grâce à un système de gain d'expérience et de personalisation du joueur.

Monster. Comme le nom l'indique, ce sont les monstres du jeu. Ils ne peuvent pas gagner d'expérience et leurs caractéristiques sont générées avec le niveau actuel des héros. Leurs compétences seront fixées suivant le type du monstre (élémentaire, boss ect...). Ils ne portent pas d'équipement. Seul les boss auront une capacité spéciale (comme les héros) appelé "Overdrive".

2.1.2 Etat éléments non vivants

Les éléments non vivants sont au nombre de trois et ne portent aucune caractéristiques communes.

Item. Ce sont les objets utilisables par les héros. Ils peuvent changer leurs attributs (augmentation d'une caractéristique ect...).

Equipment. Ce sont les armes et protections que peuvent possèder les héros. Leurs effets seront pris en compte dans la méthode Takedomage() présent dans la class Element.

Node. Ce sont les points clefs du jeu. Les héros pourront se déplacer sur une carte de noeud en noeud. Chaque noeud comporte des évènements aléatoires et non aléatoires. Les éléments aléatoires sont des combats contre des monstres aléatoires tandis que les non aléatoires sont des élèments de l'histoire. Cela peut être un simple dialogue ou un combat contre un boss. Il est possible uniquement de ce déplacer au noeud suivant ou au noeud précédent. Si ce noeud a déjà été visité, l'évènement non aléatoire n'aura plus lieu.

2.1.3 Situation du joueur

Les situations possibles dans lequelles se trouve le joueur sont au nombre de quatre. Elles représentent la ligne directive du jeu.

Déplacement sur un noeud. Le joueur se déplace dans un nouvel endroit qui va générer des évènements.

Evènement aléatoire. Cet évènement se traduit la plus part du temps par un combat contre des monstres aléatoires.

Aubergiste. En arrivant dans un nouveau lieu, le joueur a accès à un menu lui permettant d'acheter des objets utilisables en combat et de se préparer à l'évènement non aléatoire.

Evénement non aléatoire. Il dépend de l'histoire du jeu.

2.2 Conception logiciel

Dans cette section, nous expliciterons le diagrammes des classes pour les états présenté en fin de chapitre.

Classes Element. Cette classe et ses classes filles contiennent tout les éléments vivants du jeu. Elle est intrasèquement liée aux classes Capacities, Equipment et Spherier qui viennent complèter la classe Character.

Fabrique d'éléments. Cette classe a pour but de rendre plus facile la construction d'éléments vivants, notamment des monstres provenant d'un évènement aléatoire (considéré comme non boss).

Liste d'éléments. Elle va contenir toute les informations sur l'ensemble des éléments présent dans le jeu.

Classes Node. Cette classe permet de faire le lien entre l'apparition d'évènements et les éléments. C'est une liste chainée dont le déplacement est bidirectionnel.

Fabrique de noeuds. Nous reprenons le même schéma que la classe Element et les classes liées à celle-ci. Ainsi la fabrique de noeuds permettra une mise en place plus facile des points clefs du jeu.

Liste de noeuds. Elle va contenir toute les informations sur l'ensemble des noeuds présent dans le jeu.

Classe State. Cette contiendra toutes les informations liées aux noeuds et aux éléments vivants / non vivants.

FIGURE 2.1 – Diagramme des classes du package State