ECOLE
POLYTECHNIQUE
DE BRUXELLES

ULB

UNIVERSITÉ LIBRE DE BRUXELLES

Rapport projet info

Programmation orientée objet : *H&S*

Auteurs :

DENGUIR Anass
PHAN Trong Quy

6 mai 2016

1 Introduction

Cette année dans le cadre du projet INFO-H-200 à l'Ecole Polytechnique de Bruxelles, il nous a été demandé de développer un jeu vidéo de type Hack and Slash/Dungeon Crawler en Java avec une interface graphique. Ce jeu a eu pour objectif de mettre en pratique les multiples aspects de la programmation orientée objet vus en cours ainsi que de nous familiariser avec le développement de jeux vidéos. Ce rapport aura pour but de détailler les différentes implémentations faites dans notre jeu et les patterns de conception utilisés.

2 Description du jeu

2.1 Personnages

Nous allons tout d'abord détailler les différents acteurs présents dans notre jeu ainsi que le monde dans lequel le joueur est plongé et détailler la manière dont nous les avons implémentés. Les personnages, dont le hero et les monstres, sont hérités d'une classe mère abstraite *Character* implémentant plusieurs interfaces : *Collidable*, *Collision*, *Subject* et *Runnable*. Les personnages possèdent tous de la vie et de la mana qui est utilisée lors de chaque attaque des monstres pour cadencer leurs attaques et le joueur pour lancer des projectiles à distance. La mana est régénérée petit à petit pour les personnages et peut être régénérée plus rapidement pour le joueur grâce à des potions. Un système de collision entre les personnages et les décors a été implémenté en se basant sur le patron de conception visiteur. En fonction de collision entre les entités, différentes actions sont effectuées, par exemple une collision contre un élément de décors repoussera l'entité du décors et une collision entre le joueur et un monstre fera perdre de la vie au joueur.

2.1.1 Hero

Le hero est le personnage principale et est contrôlé par le joueur. Celui-ci interagit avec les monstres soit au corps à corps grâce à une attaque de mêlée soit à distance avec ses flèches mais en consommant alors de la mana. Si le niveau de mana du joueur est insuffisant pour effectuer une attaque à distance, celle-ci ne s'opère pas. Le joueur possède sur lui un inventaire lui permettant de stocker les différents objets qui ramasse sur la carte tels que des potions ou des bonus d'invincibilité. Seul les bonus de vie et de mana peuvent être stockés dans l'inventaire, les bonus d'invincibilité sont eux activés directement lors du contact entre le joueur et le bonus.

2.2 Monstres

Les monstres sont générés à partir de la lecture d'un fichier lors du chargement du level. Ils possèdent comme le joueur une barre de vie et de mana. La mana permet aux monstres d'attaquer le joueur seulement s'ils sont assez proche de celui-ci et seulement si le niveau de mana est suffisant. Cela permet d'éviter que le monstre attaque trop

fréquemment car la régénération de la mana du monstre est lente. De plus lorsqu'un monstre meurt, il a une chance sur 2 de lâcher une potion à sa mort.

2.3 Objet

Les objets sont également issus de la lecture du fichier de construction de la map (P = potion de vie ou de mana, I = potion d'invincibilité) et ont une probabilité de 1 chance sur 4 d'être instancié sur la carte. Parmi les objets possibles, nous avons donc pour l'instant des potions de vie, de mana et d'invincibilité. Seules les potions de vie et de mana peuvent être stockés dans l'inventaire du joueur tandis que les potions d'invincibilité sont utilisées directement sur le joueur.

2.4 Carte

La carte constituée d'un background et des obstacles est aussi générée à partir du fichier texte et les éléments de décors peuvent collisionner avec les personnages. Des portes sont également générées et permettent au joueur de se déplacer de salle en salle lorsque ce dernier rentre en contact avec.

3 Gameplay et interactions

La classe Game de notre jeu possède une boucle principale qui va permettre de checker les collisions entre les éléments du jeu implémentant la collision et va ensuite, si la collision est détectée, mettre en relation les deux entités collisionnées et leurs permettre d'agir en fonction de leur type. Nous avons implémenté pour le cela le patron de conception visiteur, chaque entité A possède leur propre action à appliquer sur elle-même en fonction de l'entité B avec laquelle elle a collisionné et ce sera l'entité B qui dictera à A l'action qu'elle s'auto-subira. Le joueur peut attaquer avec la barre d'espace et switcher d'attaque en pressant la touche A, la touche H et M permettent d'utiliser les potion de soin et de mana.

4 Diagramme de classe

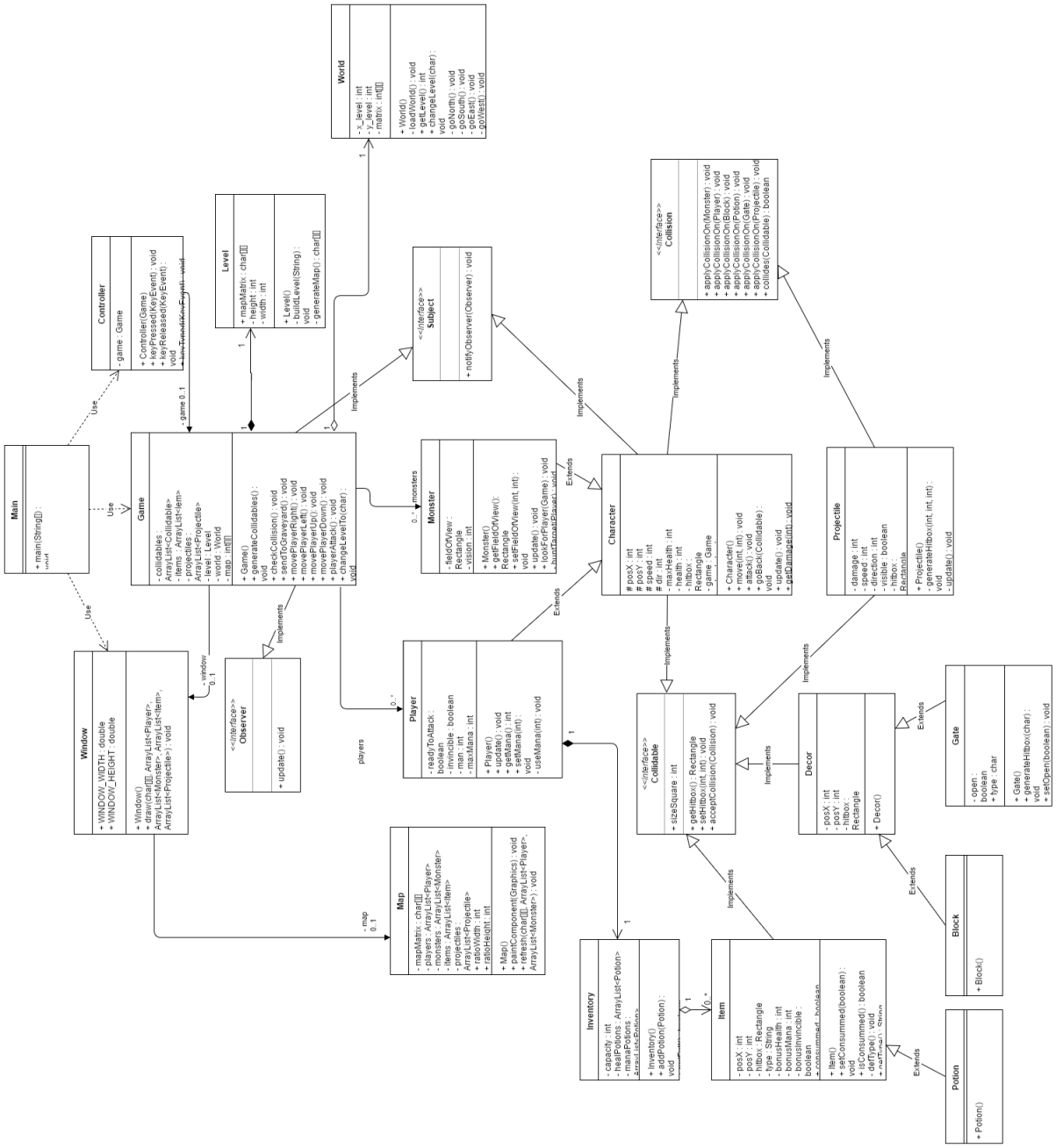

FIGURE 1 – Diagramme de classe

5 Diagramme de séquence

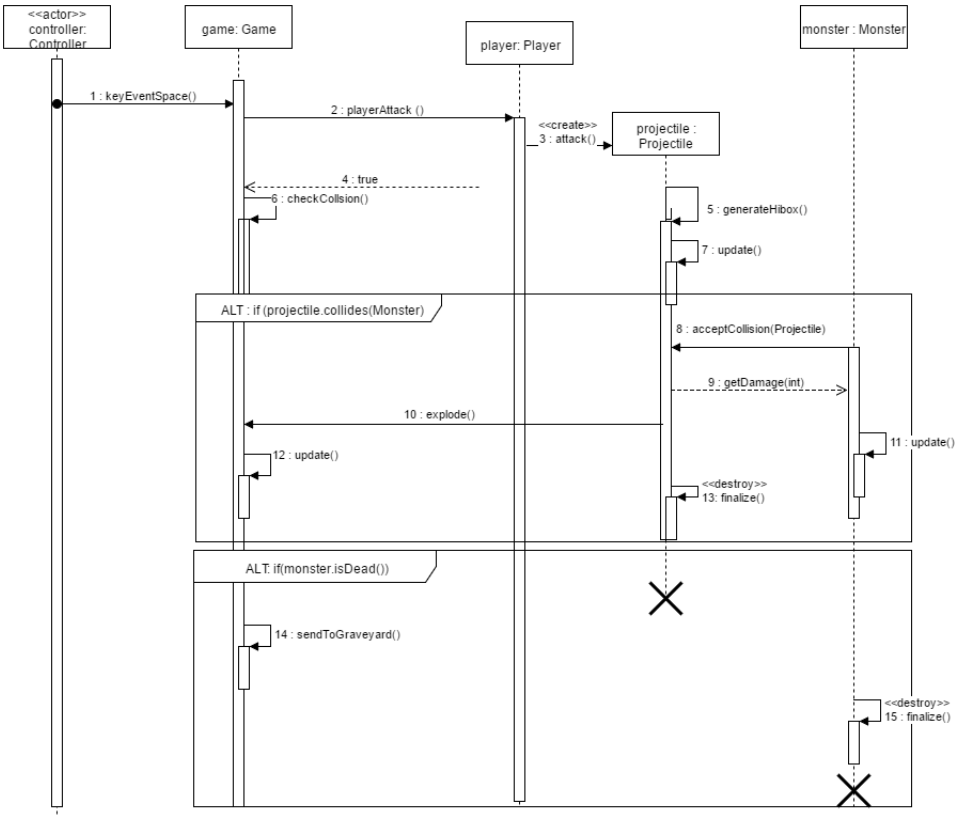

FIGURE 2 – Diagramme de séquence représentant l'attaque à distance du joueur sur un monstre