Université de Bordeaux I

Architecture logiciel

DM autour d'un framework de jeux 2D

Réalisé par :

LASSOUANI Sofiane

Université Bordeaux 1 (2016–2017) – Master Génie Logiciel (UE AL)

TABLE DES MATIÈRES

Chapitre 1

Introduction

Dans le cadre de l'UE Architecture logiciel, un projet nous ai proposé afin de mettre en pratique ce que nous avons appris tout au long du semestre.

Le projet consiste a réaliser un jeux en s'appuyant sur le framework mis à notre disposition et qui a servis à l'implémentation du jeux PacMan.

Nous devions en s'inspirant de l'implémentation de PacMan fournis et du framework, réalisé un jeux de stratégie en se basant sur le noyau de jeu temps réel mis en œuvre tout au long du semestre lors des séances de TP.

Chapitre 2

Projet

2.1 Introduction

Afin de bien entamer le projet, je devait avant tout bien lire le code fournis, et bien prendre en main le framework dans le but d'avoir une vue d'ensemble sur toute les classes déclarés et connaître leur fonctionnalités ainsi que dépendance vis-à-vis d'autre classes.

2.2 Implémentation

Cette partie du rapport est consacré à l'explication des différents package de l'implémentation du jeux. Afin de construire le jeux un nombre de classe a été implémenter afin de permettre le bon déroulement du jeux avec des soldats des ennemis, et utiliser les méthode existante dans SimpleUnit qui permet de faire des combats entre ces entités

2.2.1 Package ProjectGame

Ce package contient le main ainsi qu'une autre classe qui créer l'environnement du jeux. la classe GameLevelOne permet d'initialiser la map du jeux et d'y inserer les diffirent objets de l'environement du jeux (Tree, Gun, Shield). la création d'un objet de type

2.2.2 Package ProjectGame.entity

Ce package contient les entités du jeux a savoir, une classe Gun qui étend de la classe WeaponAttack qui créer un objet de type gun qui permet d'armer le soldat d'une arme a feu, et une classe Shield qui étend de la class WeaponDefence qui créer un objet de type shield qui permet d'armer le soldat d'un objet de défense qui est un bouclier.

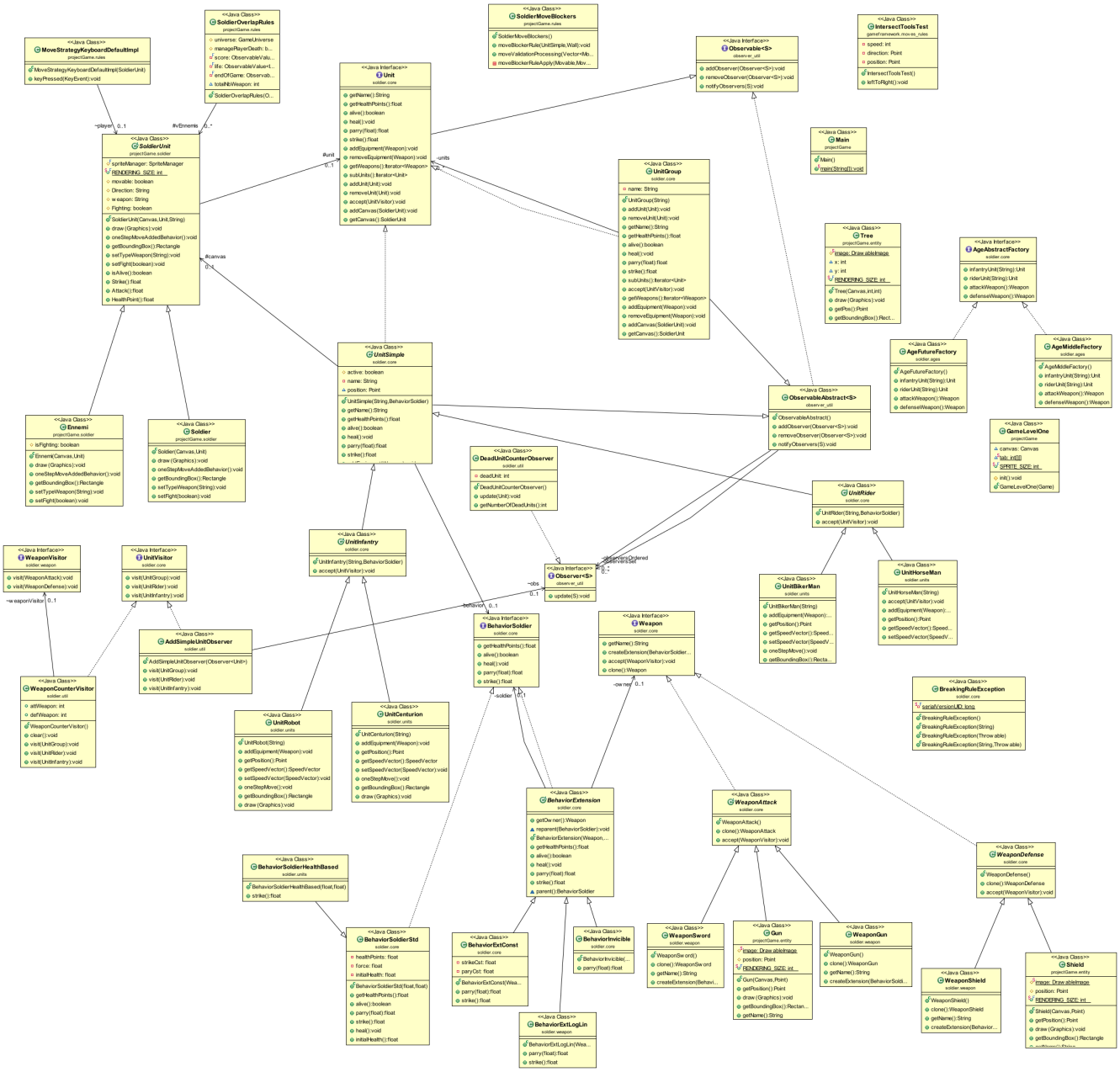
2.2.3 Package ProjectGame.rules

Ce package contient toutes les regles de gestion du jeux. Dans la classe SoldierOverlapRules j'ai une méthode qui fait appel au méthode de SimpleUnit Strick(), qui permet de faire un combat entre un soldat et un ennemi.

2.2.4 Package ProjectGame.soldier

Contient les classe correspondant a la création d'un soldat et d'un ennemi afin de les ajouter à l'environnement du jeux. la class SoldierUnit qui étend de GameMovable qui permet de définir le soldat et les type de mouvement. Les classes Soldier et Ennemi étend toute deux de SoldierUnit et instancie un soldat ou un ennemi selon le sprite passé en paramètre afin de l'ajouter à l'environnement du jeux.

2.3 Diagramme de classe



(a) Diagramme de classe

2.4 Bug

j'ai rencontré un bug que j'ai pas pu corrigé, en lançant le jeux si on essai de dirigé le soldat de gauche a droite vers l'ennemi, une fois prés de l'ennemi le soldat commence a attaquer et se bloque de face, néanmoins ce bug n'est a déclaré que dans ce cas , si on essai de le dirigé de droite a gauche vers l'ennemi une fois prés de lui il s'arrête puis on décide d'attaquer ou pas.

J'ai créer deux objet de type Weapon dans le but d'armer le soldat d'une arme a feu, ou de l'armer avec un objet de défense, en se dirigeant vers ces objets, mais le soldat passe dessus et les objet reste sur l'environnement du jeux, et le soldat n'est pas équiper.

2.5 Comment jouer ?

Les touches suivantes permettent de diriger le soldat dans le jeux :

— Les touches de direction du clavier permettent de faire avancer le soldat.
— La touche "entré" permet de faire arrêter le soldat, de le garder immobile.
— La touche "espace" permet de manipuler l'épée du soldat.

Le jeux se compose d'une aire de jeux, ainsi qu'un soldat crée de type UnitCenturion("soldier") sur qui on à la capacité de manipuler, et un ennemi du même type avec un sprite différent qui reste immobile.

2.6 Amélioration

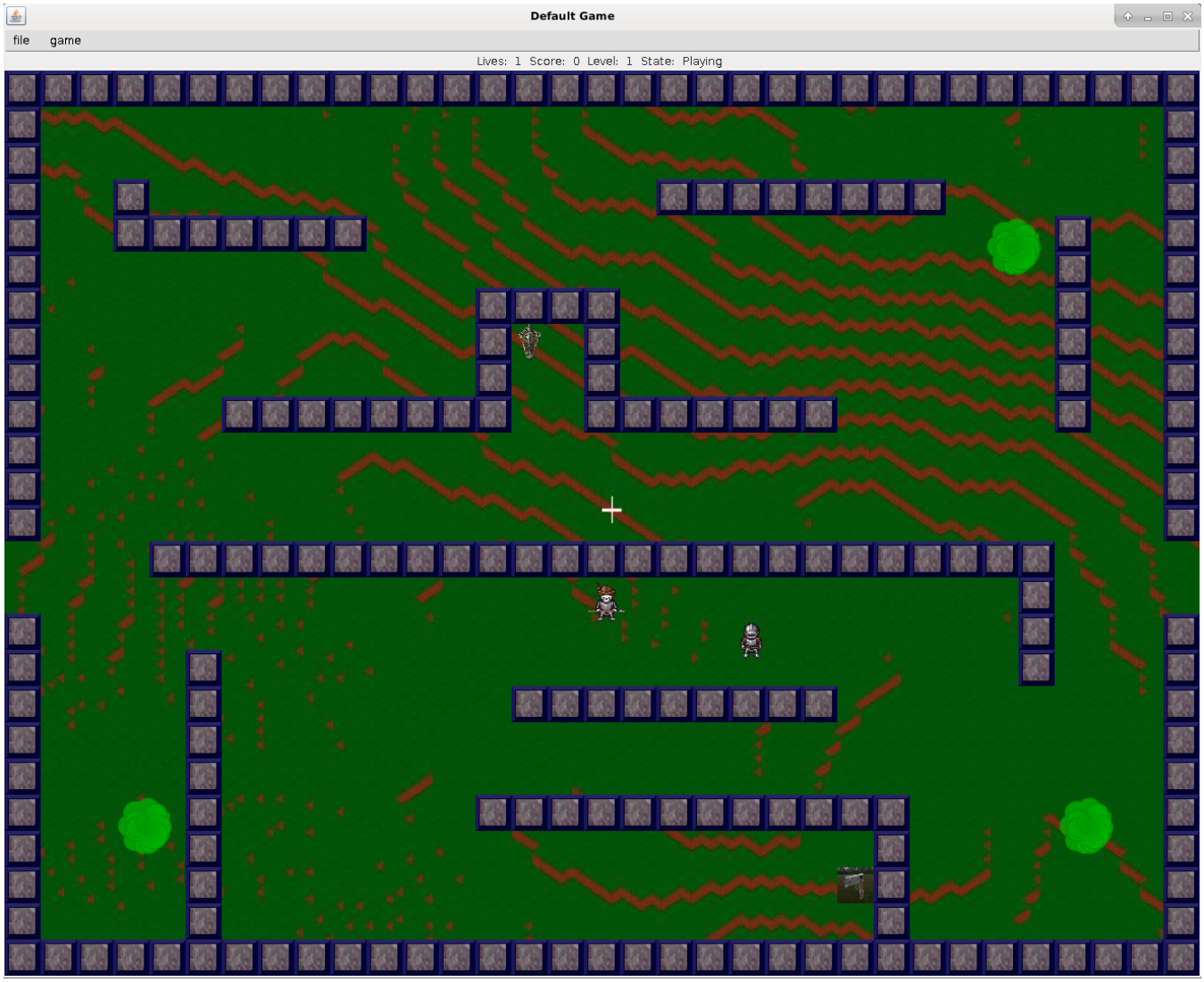
Une amélioration est possible à ce jeux en utilisant la fabrique abstraite qui nous permet de créer des soldat de différent age selon le sprite passé en paramètre ce qui permet d'avoir des monté en puissance du soldat et de sa force d'attaque.

L'ennemi ici reste immobile, on pourrais utilisant la classe MoveStrategyRandom le mener a bouger aléatoirement et se mettant à la poursuite du soldat

2.7 Problèmes rencontrés

Le problème majeur était la compréhension les fonctions des différentes classes du framework, partant de la il est assez simple de commencer a implémenter sa propre stratégie de jeux et mettant en œuvre différentes classe supplémentaire et utilisant celle déjà existante, rajouter a cella les bug que j'ai pas pu corrigé du faite que je savait pas d'où venait le dysfonctionnement.

2.8 Interface du jeux



(a) Interface du jeux

Chapitre 3

Conclusion

Ce projet nous permis de prendre en main un framework de jeux en 2D et d'y implémenter un jeux de stratégie de soldat.

Le jeux développé n'est pas au point du point de vue stratégie, mais ça me permis de comprendre l'utilité d'une architecture dans la programmation et la facilité de réutiliser du code grâce aux patterns.