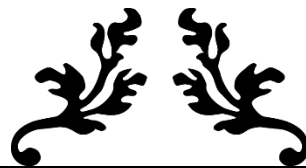




Master 1 Informatique



CONCEPTION AGILE DE PROJETS INFORMATIQUES ET GENIE LOGICIEL



Prénoms & Noms :

Seynabou FAYE

Paule Naomi KALDJOB

Professeur : Valentin Lachand-Pascal

ANNEE SCOLAIRE : 2022/2023

I. Introduction

Tout au long du premier semestre de Master 1 Informatique, nous avons réalisé des projets dont l'un consiste à réaliser le jeu de balle au prisonnier. A l'aide d'une architecture MVC, nous avons construit le jeu de balle au prisonnier en trois parties : vue, modèle et contrôleur. J'ai construit des joueurs, des projectiles, des obstacles, le score et implémentées des stratégies de tirs. Dans ce rapport, nous allons présenter l'architecture du jeu (le modèle MVC) et les design patterns.

II. Modèle MVC

I-Modèle MVC : la vue

1-La classe App

Dans ce projet, j'ai créé une classe App qui affiche les joueurs, les projectiles, les flèches, les balles et les obstacles et aussi le score. Cette classe regroupe toutes les vues du modèle.

2-La classe Field

Cette classe sert à appeler les modèles, écouter les actions de l'utilisateur (clic de clavier, relâchement de clavier), à exécuter des actions spécifiques aux actions de l'utilisateur en appelant la classe Field_controleur et incrémenter ou décrémenter le score. Le score est incrémenté si le projectile d'un joueur de l'équipe touche un joueur de l'équipe adverse (vice-versa) et décrémenté si le projectile touche son partenaire. La classe Field transfère toutes les images et leurs positions au programme principal (la classe App).

3-La classe score

Cette classe permet d'afficher le score de chaque équipe dans la fenêtre. A chaque exécution de la classe Field, le score peut être modifié.

II-Modèle MVC : le modèle

1-La classe projectile

Cette classe fait partie de la vue. Elle permet l'affichage des projectiles et aussi leur déplacement. Le projectile entame son déplacement dans la zone du tireur et change de position continuellement après chaque tir du joueur. Il suit une direction, un angle de tir.

2-La classe Player

Cette classe permet l'affichage des équipes et des flèches lors de l'appel de la classe. On décompte deux équipes. Chaque équipe est constituée de 2 joueurs humains et un joueur d'ordinateur. Cette classe appelle la classe Modèle de Sprite. Elle est intégrée dans le modèle parce qu'elle sert de base pour la création des joueurs humains et des flèches.

3- La classe Sprite

Cette classe sert de modèle pour l’affichage des équipes et leur animation lors d’un déplacement.

4-La classe ordin

Cette classe est une classe qui hérite de Player. Elle sert de base pour la création et l’affichage des joueurs robots. Elle prend en entrée des paramètres définis dans la classe Field.

III-Modèle MVC : le contrôleur

1-La classe Field_contrôleur

Cette classe permet de gérer tout ce qui est direction et déplacement des joueurs, déplacements des balles ,les collisions et les tirs. Pour chaque équipe de joueurs, les joueurs appuient sur des boutons spécifiques pour effectuer des mouvements. Pour l’équipe 1 :

- ✓ Déplacement à gauche : bouton Y pour le joueur1 et bouton A pour le joueur2
- ✓ Déplacement à droite : bouton I pour le joueur1 et le bouton B pour le joueur2
- ✓ Rotation à droite : bouton flèche du bas pour le joueur1 et le bouton O pour le joueur2
- ✓ Rotation à gauche : bouton F pour le joueur1 et le bouton H pour le joueur2
- ✓ Tir : bouton M pour le joueur1 et le bouton T pour le joueur2
- ✓ Le joueur 3 se déplace aléatoirement (à gauche ou à droite)

Pour l’équipe2 :

- ✓ Déplacement à gauche : bouton flèche à gauche pour le joueur1 et bouton Q pour le joueur2
- ✓ Déplacement à droite : bouton flèche à droite pour le joueur1 et le bouton D pour le joueur2
- ✓ Rotation à gauche : bouton flèche du haut pour le joueur1 et le bouton Z pour le joueur2
- ✓ Rotation à droite : bouton flèche du bas pour le joueur1 et le bouton S pour le joueur2
- ✓ Tir : bouton espace pour le joueur1 et le bouton E pour le joueur2
- ✓ Le joueur 3 se déplace comme le joueur 2 de l’équipe adverse

2-La classe Player contrôleur

La classe Player_contrôleur permet de positionner chaque équipe en utilisant la classe Field_contrôleur.

III. Design-pattern : le singleton

Il est un modèle de conception de création qui vous permet de vous assurer qu'une classe n'a qu'une seule instance, tout en fournissant un point d'accès global à cette instance.

On a créé un design pattern dans la classe Field pour faire l'instanciation de cette classe.

IV. Conclusion

En somme, nous avons d'abord construit les joueurs, les flèches et les collisions ensuite construit un code généralisé et identifié le modèle, la vue et le contrôleur dans notre programme. Nous avons créé plusieurs parties du programme suivant le modèle MVC.

Ce projet nous a permis d'améliorer nos compétences en Java surtout Java FX, de construire un modèle MVC et de découvrir les design patterns. Cependant, le code fourni était trop complexe, nous manquions de compétences en Java et le temps d'apprentissage de Java était relativement court. Ainsi, nous n'avons pas pu terminé le projet notamment la construction des boutons, la collision entre la balle et le mur, finalisation du modèle MVC et enfin le changement de tactique en cours de partie.