

Rapport de projet

Conception agile de projets informatiques et génie logiciel

Balle au prisonnier

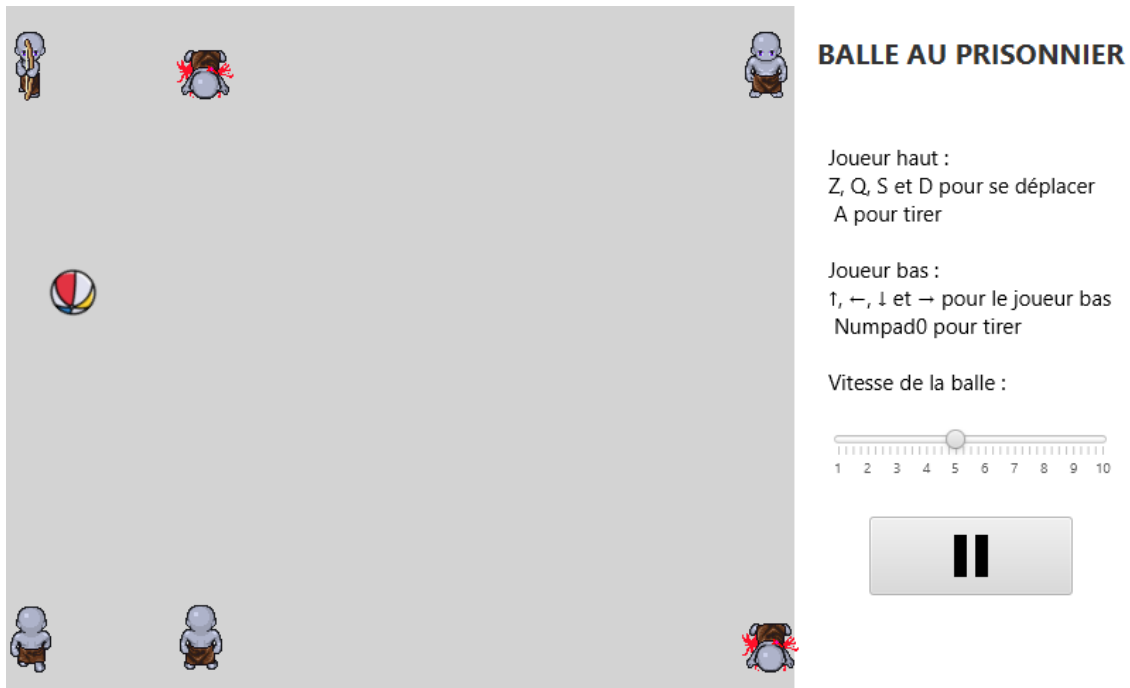


Table des matières

Présentation du projet	3
Choix d'architecture	4
Classes abstraites	4
ObjectController, GameObject et ViewObject	4
PlayerController et Player	4
Singleton	4
Classe finale	5
MVC	6
Implémentation des IAs	7
Déplacement	7
Lancement de la balle	8

Présentation du projet

Ce projet a pour but de réaliser une balle au prisonnier en Java et en utilisant le software de création d'application JavaFX. Dans ce jeu, deux camps s'affrontent (avec des IAs ou non). Le but est de toucher l'adversaire avec la balle pour l'éliminer. Le premier des deux camps à n'avoir plus aucun joueur vivant a perdu.

Choix d'architecture

Classes abstraites

ObjectController, GameObject et ViewObject

Nous avons réfléchi à ce que nos classes ont en commun. C'est-à-dire, une position x et y, une largeur, une hauteur, une vitesse de déplacement, une vitesse de rotation et un angle. Donc, nous avons généralisé ces propriétés par deux classes abstraites, nous permettant de centraliser ces types de données. C'est aussi le cas pour ViewObject qui concerne l'affichage des objets. En effet, la fonction *rotate(...)* est la même dans tous les cas et chaque objet possède une image à afficher.

PlayerController et Player

Un joueur pouvant être un humain ou une intelligence artificielle, il nous a fallu créer une classe pour chaque cas. Or peu importe le type de joueur, certaines actions étaient communes, par exemple, le fait de se déplacer à droite ou à gauche. Nous avons donc ces classes qui permettent à la fois de généraliser l'instanciation des joueurs en ayant des listes de Player, mais aussi de gérer les spécificités de chaque type de joueur.

Singleton

Nous avons utilisé le design pattern Singleton pour le ProjectileController. C'est-à-dire qu'il doit n'y avoir qu'une seule instance de ProjectileController car il n'y a qu'une seule balle sur le terrain.

Classe finale

Nous avons une classe finale Const qui contient un grand nombre de constantes utilisée dans le code.

Nom de la constante	Valeur	Utilisation
SCREEN_DIM	800, 600	Taille de l'écran
FIELD_DIM	800 * 0.7, 600	Taille du terrain
NB_EQ1, NB_EQ2	3	Nombre de joueurs de l'équipe X
NB_IA_EQ1, NB_IA_EQ2	2	Nombre de joueurs IA de chaque équipe
MOVESPEED	2	Vitesse de déplacement des joueurs
SIDE_TOP, SIDE_BOT	1, 2	Valeur représentant le camps en haut ou en bas
HITBOX_PLAYER	64	Taille de la boîte de collision des joueurs
HEIGHT_EQ1	600 - 20 - 32	Position en Y du côté de l'équipe 1
HEIGHT_EQ2	40	Position en Y du côté de l'équipe 2
STATE_MOVE, STATE_PREPARE_SHOOT, STATE_SHOOT, STATE_DEAD	0, 1, 2, 3	États de la machine à état de l'IA.
CD_SHOOT_MAX, CD_SHOOT_MAX_OFFSET	150 et 20	Temps d'attente avant de lancer la balle. Entre [130; 170] ticks.

MVC

L'architecture du jeu est en MVC. À savoir, qu'il y a un contrôleur qui met à jour la vue en utilisant les données du modèle.

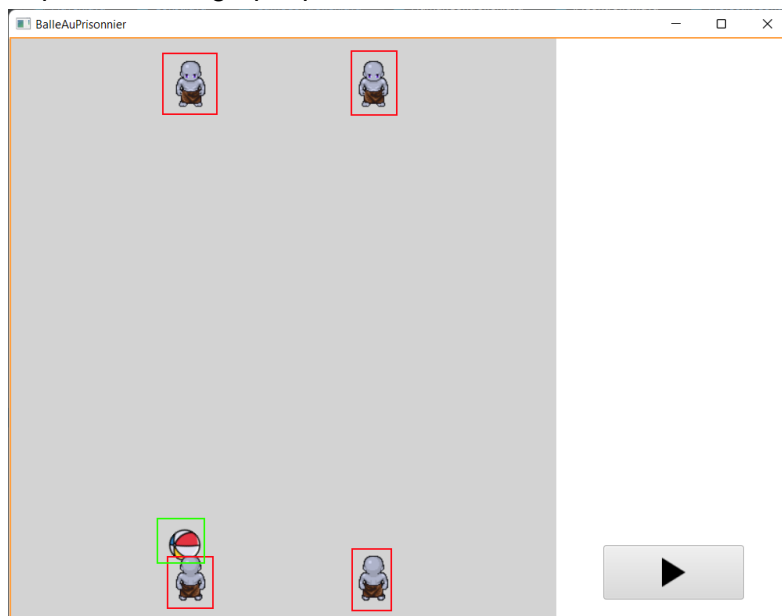
Nous avons 4 contrôleurs :

- GameController : Générer le jeu et la boucle infini de jeu et de mettre à jour la vue à chaque ticks
- ProjectileController : Singleton représentant la balle de jeu
- HumanController et IAController : Représente un joueur ordinateur ou humain

Lors de la mise à jour de la vue de GameController, les autres vues sont aussi mise à jour au travers de la fonction *handleInput()* de la vue de Game.

Ce GameController est créé lors du lancement de l'application pour initialiser le jeu.

Représentation graphique des contrôleurs :



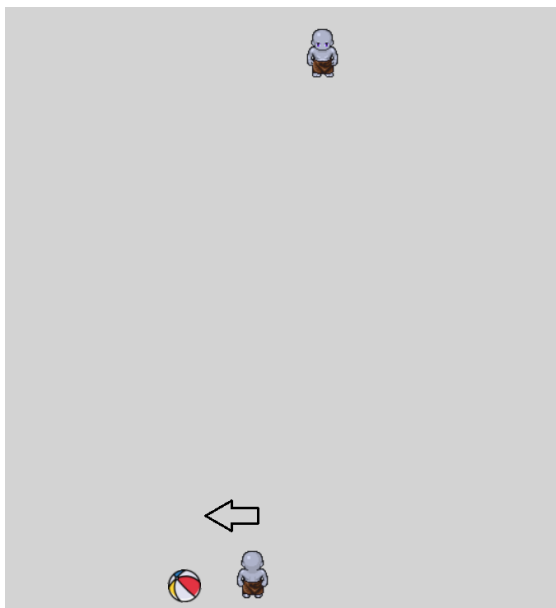
Sur cette capture d'écran, il y a 4 contrôleurs de joueurs représenté par des cadre rouge. Ensuite, entouré par un cadre vert, le contrôleur ProjectileController. Et pour finir, le contrôleur englobant le jeu (GameController) est représenté par le cadre orange.

Implémentation des IAs

Déplacement

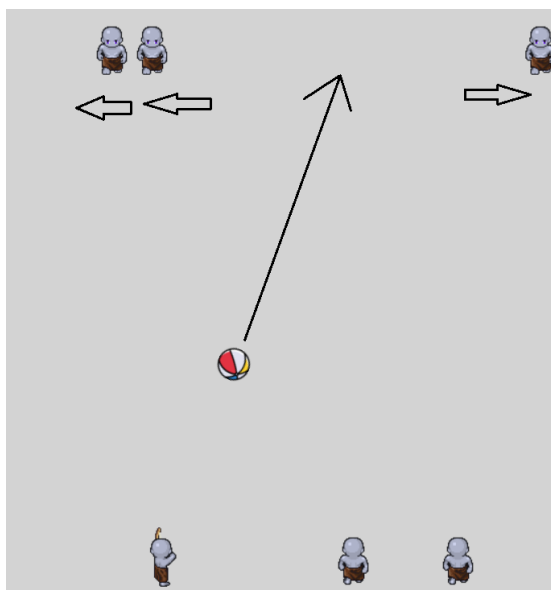
Pour les déplacements de l'IA, il y a 3 cas de figures différents.

Tout d'abord, lorsqu'une balle est arrêtée sur son terrain, toutes les IAs de ce côté se déplaceront pour attraper la balle.



Dans ce cas, le joueur du bas se dirige vers la balle à sa gauche.

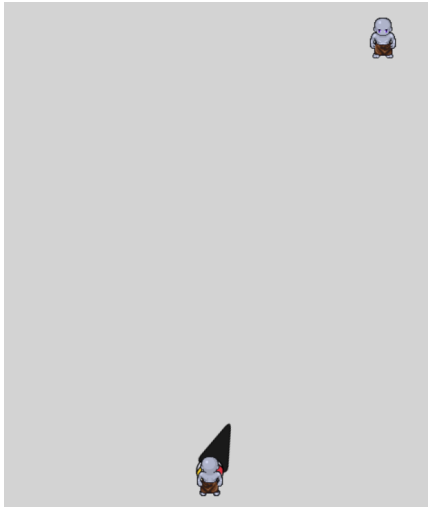
Ensuite, si une balle est lancée vers son côté, tous les joueurs essayeront de l'esquiver en partant au plus loin du point d'impact de la balle.



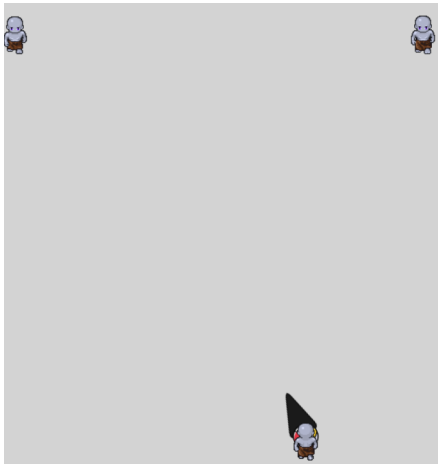
Dans ce cas, les deux joueurs en haut à gauche se déplacent vers la gauche pour esquiver la balle tandis que le joueur en haut à droite se déplace à droite pour esquiver la balle.

Lancement de la balle

Pour finir, le dernier cas de figure est le lancement de la balle. Il y a plusieurs situations. Dans le cas où il reste qu'un ennemi en vie. Le joueur le visera :



Quand il y a autant de joueurs à droite et à gauche du tireur, il choisira aléatoirement un des ennemis :



Lorsqu'il y a plus de joueur d'un côté que de l'autre, il choisira la position moyenne entre tous ceux qui sont de ce côté :

