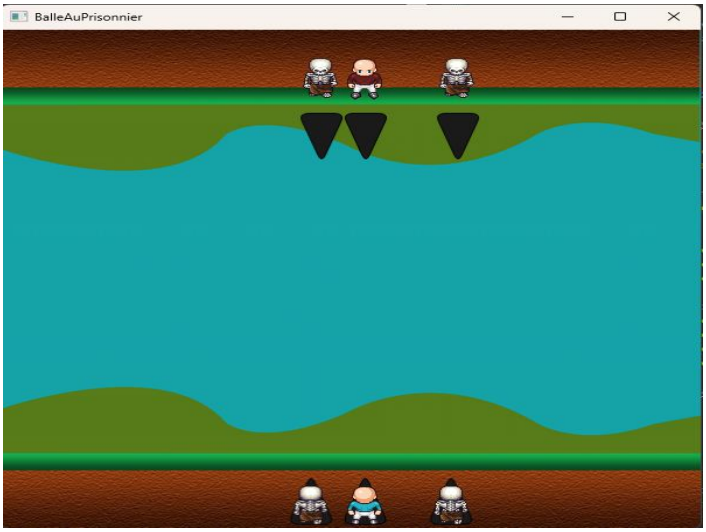
Conception agile de projets informatiques et génie logiciel :

Balle au prisonnier



Réaliser par :

- **Abdelhamid Benkada**
- **Mohammed Azizi**
- **Imad eddine Al hannawi**

Sommaire

Introduction :

Dans ce jeu on aura 2 équipes, chacune est composée d'un joueur contrôlé par le clavier et 2 joueurs contrôlé par l'ordinateur, les joueurs peuvent se déplacer à droite ou à gauche et effectuer des tirs.

Lorsque la balle lancée touche un joueur, une collision aura lieu et le joueur disparaitra.

Le but est d'améliorer ce jeu et de mettre en place des design patterns.

LES TACHES REALISEES :

•Utilisez les touches WASD ou les flèches directionnelles pour déplacer et faire pivoter les joueurs.
•Les touches de contrôle correspondent à un clavier AZERTY plutôt que QWERTY, i.e. ZQSD au lieu de WASD.

• La vitesse (la taille du pas) du Joueur soit variable (que cela devienne une propriété du joueur spécifié à sa création/assigné aléatoirement).

•Chaque joueur utilise une touche différente pour tirer (au lieu d'ESPACE pour les deux joueurs).
•Créer deux listes de joueurs correspondants à deux équipes (au lieu d'une liste).

•Utiliser l'héritage pour faire une distinction entre joueurs contrôlés par un humain et par l'ordinateur.

•Pour l'instant les joueurs contrôlés par l'ordinateur restent statiques.

•Faire en sorte qu'une partie oppose 3 joueurs (1 humain, 2 ordinateurs) de chaque côté.

•Créer une classe projectile, créée lors d'un tir, avec une vitesse et une direction.

•Associer une représentation graphique à cette classe (ajouter une image nommée ball.png dans le dossier assets).

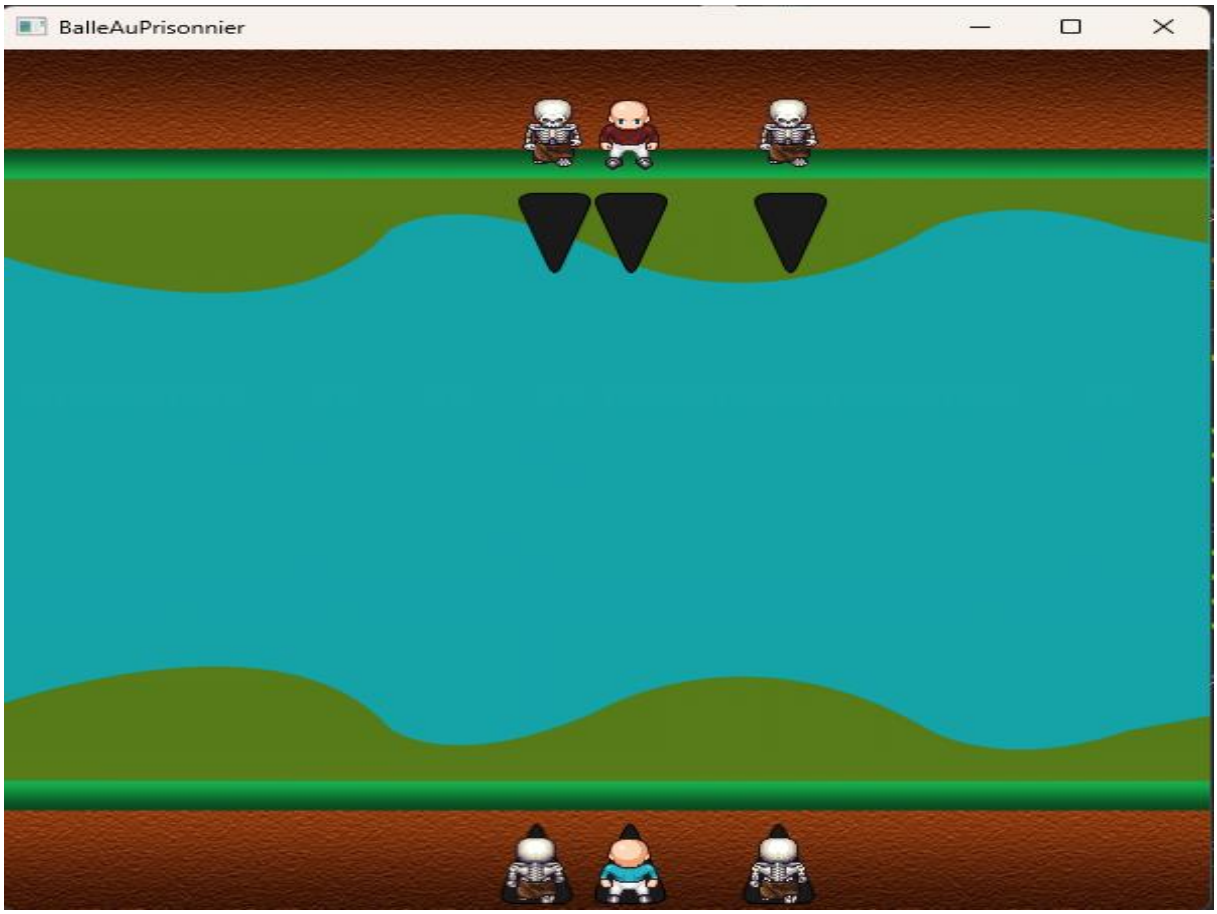
•S'assurer que le projectile se déplace bien lors de la boucle principale du jeu.

•Le projectile s'arrête dans le camp opposé, soit en tombant au sol, soit en touchant un joueur.

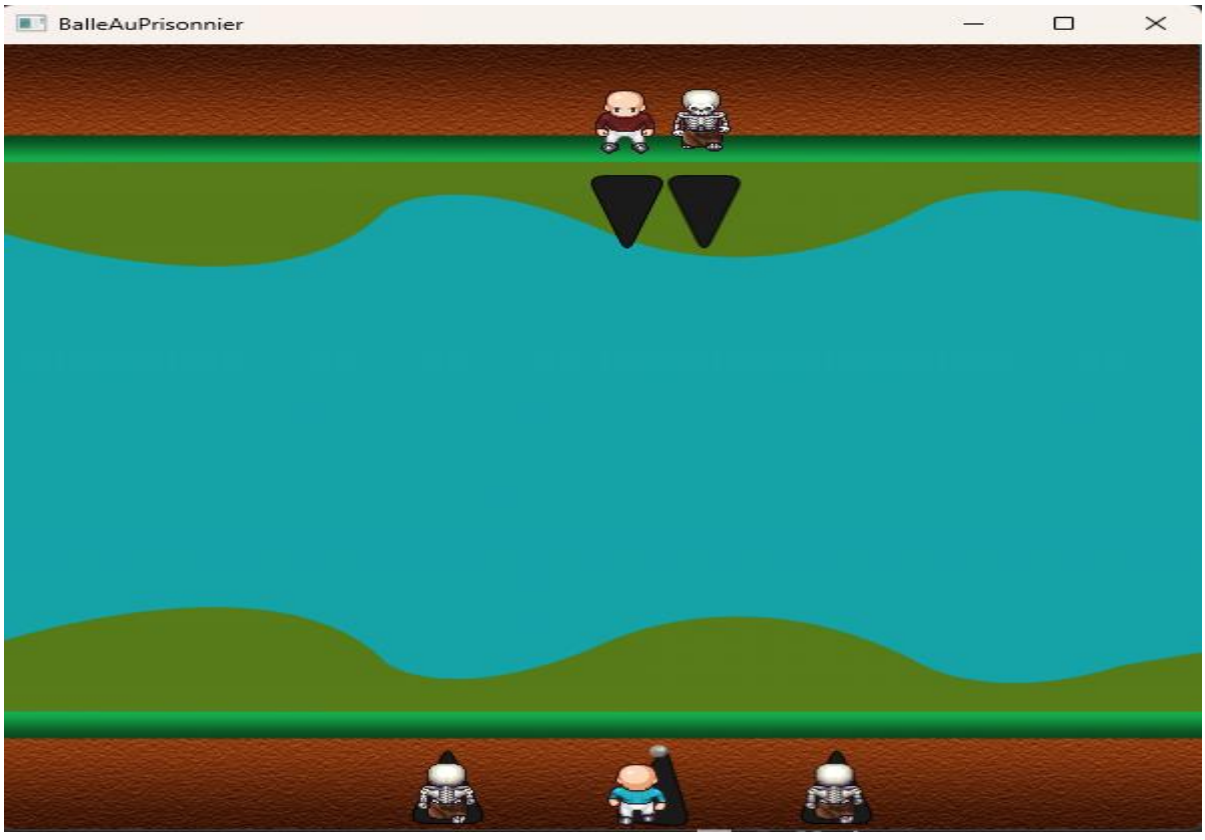
•Le projectile teste s'il rencontre un joueur adverse. Pour cela 1.Créer une méthode testant si deux rectangles se chevauchent : 2.Rajouter une méthode permettant de facilement récupérer les coordonnées de la binding box (du rectangle englobant) des projectiles et des joueurs ; 3.À chaque déplacement du projectile tester s'il rencontre un joueur adverse.

•Le joueur adverse disparaisse du jeu s'il est touché. Pour cela rajouter un attribut à la classe Player.
•Les collisions peuvent être entre : Une balle et un joueur – ce cas doit être géré obligatoirement une balle et les murs une balle et un obstacle

•Chaque joueur avec une IA aura une tactique propre qui consiste à : se déplacer aléatoirement Suivre le joueur contrôlé par un humain



Après collision :



OUTILS DE GESTION DU CODE :

1/OUTILS DE BUILD AVEC MAVEN :

Maven est utilisé pour gérer les processus de génération, de test et de déploiement. Il peut séparer les tests unitaires et les tests d'intégration afin que vous ne les exécutiez que si nécessaire et réduisiez le temps de construction

2/ OUTILS DE GESTION AVEC GITHUB :

GitHub permet aux développeurs de stocker et de partager, publiquement ou non, le code qu'ils créent. La plate-forme accueille ainsi, dit-elle, plus de 80 millions de projets, qu'il s'agisse de logiciels, de sites Web, d'applications pour mobile ou tous autres types de programme informatique

3/LES DESIGN PATTERN :

Les design patterns : On a utilisé 2 designs patterns dans ce projet, Factory et singleton. La méthode 'Factory' dans l'interface permet à une classe de reporter l'instanciation à une ou plusieurs sous-classes concrètes. Étant donné que ces modèles de conception parlent de l'instanciation d'un objet.

On a créé une interface 'Player Interface' et des classes concrètes (Player, Computer) implémentant cette interface. Puis dans la classe Controller Player on 'a pu faire la création des joueurs à l'aide du design pattern 'Factory' selon le type du joueur. En revanche Singleton Pattern indique qu'il suffit de "définir une classe qui n'a qu'une seule instance et qui lui fournit un point d'accès global". En d'autres termes, une classe doit s'assurer qu'une seule instance doit être crée et qu'un seul objet peut être utilisé par toutes les autres classes, dans notre projet on a utilisé ce design pattern pour s'assurer la création d'un seul terrain de jeu, pour cela on a implémenté ce design dans la classe 'Field'.

Conclusion :

En travaillant sur ce projet de conception d'un jeu divertissant qui regroupe 2 équipes de 3 joueurs, nous avons pu exploiter les classes et méthodes appliqués dans le cours et Tps afin de réaliser ce jeu interactif. Toutefois, nous avons rencontré des difficultés pour implémenter le modèle MVC.