

# Conception agile de projet Informatiques et génie logiciel.

**Sujet : Jeu de Balle au Prisonnier**

Encadré Par : M. Valentin Lachand-Pascal

Réalisé Par : Wendyam Maximilien Ivan YAMEOGO

Rami Ayman

## 1- Introduction :

Dans le but de se remémorer les grands principes de la programmation orientée objet , un projet nous a été confié : jeu de Balle Au prisonnier. Le but est d'améliorer le programme en faisant :

- Refactoring du code
- Application de design pattern
- Ajout des classes (Projectile , Machine ) et gestion d'héritage.

## 2- Modélisation :

### 2-1 Acteurs :

- *Principaux* : Player, Machine

### 2-2 Besoins par acteurs:

- Player:

Déplacement du joueur de la gauche vers la droite

Rotation de visé

Tirer

Mettre en Pause

Quitter

Voir les paramètres (score de jeu)

- Machine :

Machine hérite de la classe Player et donc peut faire :

Déplacement du joueur de la gauche vers la droite

Rotation de visé

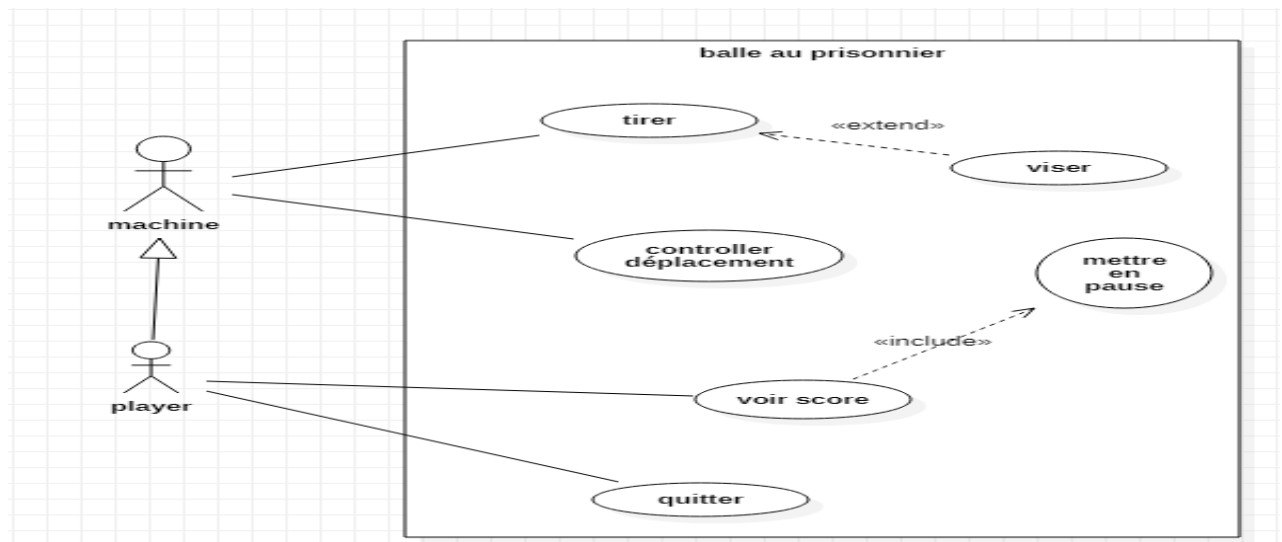
Tirer

## 3-Diagramme de Use Case :

### 3-1 Définitions :

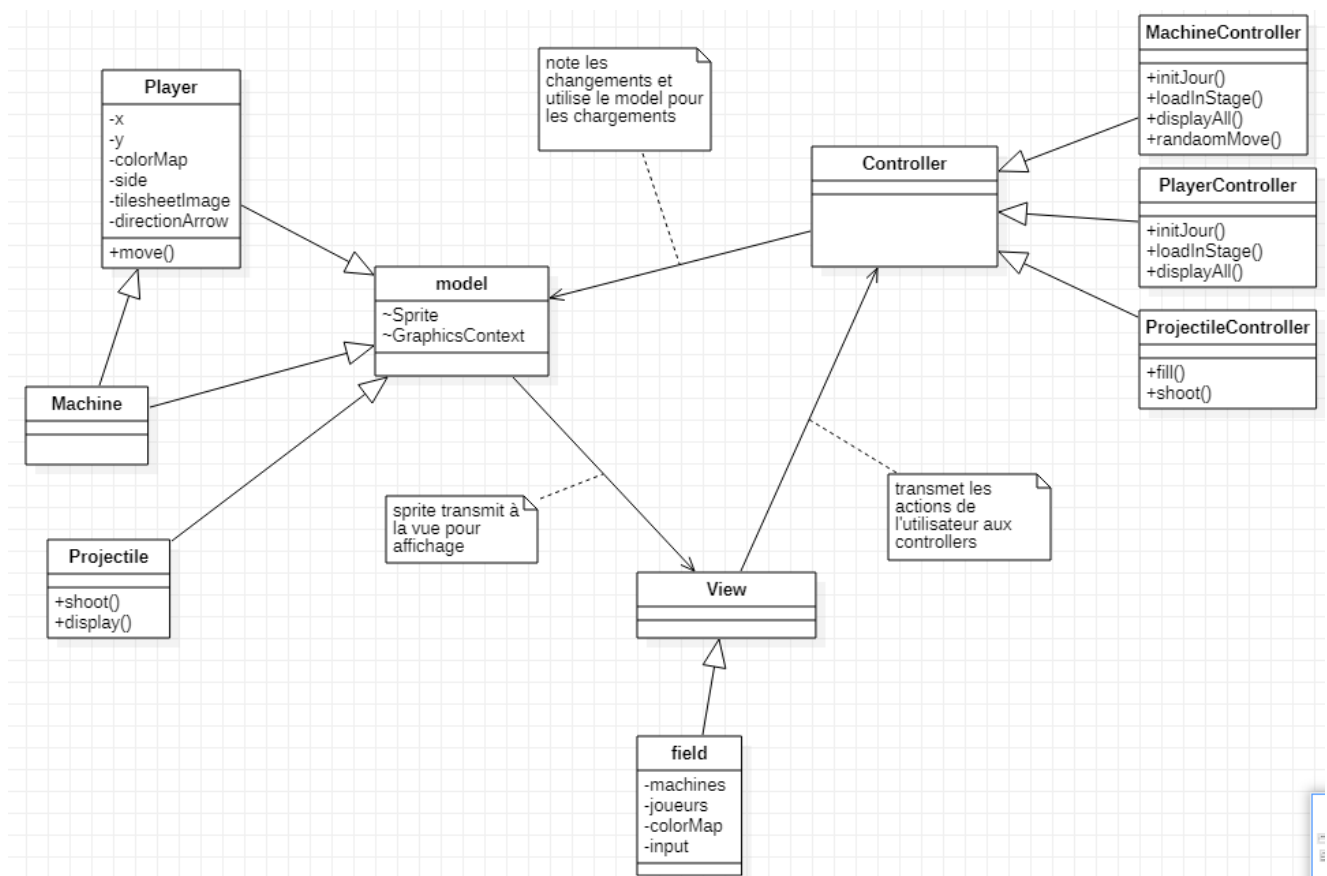
Le Diagramme ci-dessous nous indique que le Player hérite de Machine et qu'ils ont les mêmes méthodes sauf que le Player a aussi le droit de voir le score et quitter le jeu .

### 3-2 Diagramme



### 4- Diagramme de Classe :

#### 4-1 Diagramme :



## **5- Design Patterns :**

### **5-1 le choix du modèle MVC :**

Le Modèle MVC (model view controller ) permet d'avoir plus de lisibilité sur le code ; la modification est plus rapide et plus facile car on connaît les différentes interactions entre les fonctions des différents packages

Utilisation : Model (Player , Machine , Projectile , Sprite ) View (Field ) Controller (MachineController pour contrôler les actions de la machine , PlayerController celles du joueur principale et ProjectileController pour gérer les projectiles ) .

### **5-2 Decorateur :**

Lombok est une bibliothèque qui met à disposition des décorateurs permettant de gérer automatiquement nos classes. Nous avons eu à utiliser @Getter @Setter.

- @Getter : crée automatiquement les accesseurs de nos attributs
- @Setter : crée automatiquement les mutateurs de nos attributs

### **6- Conclusion :**

En somme ce travail a été très enrichissant pour nous du fait qu'il nous a permis de revoir la POO avec le polymorphisme, les héritages, le casting, l'utilisation de design pattern pour plus de lisibilité et une meilleure organisation.