# Projet Jeu de la Balle

## Introduction

Le but de ce projet est de créer un jeu de balle où deux joueurs séparés par des murs formés de 6 cases, eux-mêmes séparés par un espace, s’affrontent en se lançant à tour de rôle une balle. Les joueurs ont chacun 3 vies et ont chacun un score qui augmente au cours de la partie, jusqu’à un maximum de 3. Le joueur qui a atteint le score de 3 gagne.

## Besoins, contraintes et description du jeu

Au début de la partie, le nombre de vies de chaque joueur (3) et le score de chaque joueur (0) s’affichent en haut à gauche de la console, cinq points clignotants s’affichent sur le joueur qui commence, lorsque l’on appuie sur la touche espace, sans la lâcher, le dernier point qui clignotait s’arrête et il détermine donc l’axe de tire de la balle. Au même moment une barre de progression s’affiche en haut de la console et commence à progresser, lorsque le joueur décide de lâcher la touche espace, la progression affichée détermine la puissance du tir.

La balle est ensuite tirée depuis sa position de départ et parcours une distance sous forme d’arc de cercle en suivant l’axe de tir, sa chute est en fonction de la gravité, et atterrit sur le mur ou de l’autre côté du mur de l’autre joueur. Si la balle touche le mur du joueur qui l’a lancée, il perd un point. Si la balle touche le mur adverse, la case du mur touchée disparait, le joueur qui a tiré marque un point et c’est à l’autre joueur de commencer son tour de jeu. Si la balle touche l’autre joueur ce dernier perd une vie et le joueur qui a tiré et touché marque un point également. Le jeu continue comme cela jusqu’à ce qu’un des deux gagne, soit en ayant atteint le score maximal (7), soit lorsque l’un des deux joueurs a perdu tous ses points de vie.

Du son est implémenté dans le jeu lorsque le jeu commence, lorsque le premier joueur choisit l’angle de tir, lorsque la barre de progression a terminé de progresser, lorsque la balle part, lorsque la balle touche soit le mur soit le joueur et lorsque la partie se termine.

## Structure du jeu

### Fonctionnalités principales

* **Initialisation du jeu**
  + Afficher le titre du jeu, les scores de chaque joueur, le nombre de vies de chaque joueur, les murs formés de 6 cases, les avatars des joueurs, les cinq points qui clignotent au-dessus du premier joueur.
* **Gestion du début de tour de jeu**
  + Choisir le joueur qui commence.
  + Calculer l’angle de tir en appuyant sur espace via un mécanisme de clignotement de points.
  + Faire progresser la barre de progression en lâchant la barre espace.
  + Faire partir la balle.
* **Simulation de la trajectoire et de la puissance**
  + Calcul de la trajectoire en fonction de l’angle, de la puissance et de la gravité.
  + Calculer la puissance via la barre de projection.
* **Mise à jour des scores et des vies**
  + Vérifier les collisions avec les murs ou l’adversaire.
  + Gérer les règles de points et de pénalités.
* **Sonorisation**
  + Jouer des sons à chaque événement cité.
* **Fin de partie**
  + Déterminer le gagnant et afficher un message de fin et de recommencement de partie.

### Les classes

* **Game** : gère la logique globale du jeu (début, fin, alternance des tours)
* **Player** : contient les informations sur le joueur (vies, score, position).
* **Ball** : gère la trajectoire et les interactions avec l’environnement.
* **Wall** : représente les murs des joueurs (positions, gestion des impacts)
* **SoundManager** : gère les sons à jouer lors des évènements spécifiques.
* **UserInterface** : affiche les informations dans la console (scores, vies, barre de progression).

### Les méthodes

* Pour la classe Game
  + StartGame() ; // Initialise le jeu (score, vies, murs, joueurs).
  + PlayTurn() ; // Gère un tour de jeu (alternance entre les joueurs)
  + DisplayStatuts() ; // Affiche les scores, vies et état des murs.
  + IsGameOver() ; // Vérifie si la condition de fin est atteinte.
  + EndGame() ; // Affiche le gagnant et termine le jeu.
* Pour la classe Ball
  + SetInitialPosition(); // Définit la position de départ de la balle.
  + CalculateTrajectory(); // Calcule la trajectoire selon l'angle, la puissance, et la gravité.
  + CheckCollisionWall(); // Vérifie si la balle touche un mur.
  + CheckCollisionPlayer(); // Vérifie si la balle touche un joueur.
  + Move(); // Fait avancer la balle sur la trajectoire calculée. }
* Pour la classe Wall
  + RemoveBlock(); // Supprime un bloc du mur en cas de collision.
  + IsWallIntact(); // Vérifie si le mur contient encore des blocs.
* Pour la classe SoundManager
  + public void PlayStartSound(); // Joue le son au début de la partie.
  + PlayAngleSound(); // Joue le son lors du choix de l'angle.
  + PlayPowerSound(); // Joue le son à la fin de la barre de progression.
  + PlayShootSound(); // Joue le son lorsque la balle est tirée.
  + PlayCollisionSound(); // Joue le son lors d'une collision (mur/joueur).
  + PlayGameOverSound(); // Joue le son lorsque la partie se termine.
* Pour la classe UserInterface
  + public void DisplayScores(); // Affiche les scores et les vies des joueurs.
  + DisplayProgressBar(); // Affiche et anime la barre de progression.
  + DisplayBlinkingPoints(); // Gère les points clignotants pour le choix de l'angle.
  + ClearScreen(); // Efface la console pour actualiser l'affichage.
  + DisplayWinner(); // Affiche le gagnant de la partie.

## Semaine 1

Création du rapport de projet, des différentes classes Player, Wall, Ball, Game et SoundManager et des différentes méthodes dans chaque classe.

## Semaine 2

Début de la programmation, création de l’interface graphique composée des informations des deux joueurs (points de vie, scores), des murs et des personnages représentant les joueurs.

## Semaine 3

// A remplir

## Semaine 4

Programmation : implémentation de la sélection de l’angle de tir, de la puissance, de la collision avec le joueur adverse.

## Semaine 5

## Semaine 6

## Semaine 7

## Semaine 8