Projet : Jeu de la Balle



# 1. Introduction

Ce projet consiste à développer un jeu de balle interactif en C#, où deux joueurs s'affrontent à tour de rôle en tentant de détruire les murs adverses ou d'atteindre directement leur adversaire à l'aide d'une balle lancée selon un angle et une puissance définis. Le jeu se déroule dans la console, avec une interface ASCII, un système de gestion des scores et des vies, et une intégration complète de sons pour rendre l'expérience plus immersive.  
  
Le projet a été enrichi et corrigé grâce à l'aide de l'IA pour résoudre plusieurs problèmes techniques, notamment la gestion des collisions et l'ajout de sons contextuels.

# 2. Description générale du jeu

## Objectifs du jeu :

- Deux joueurs sont séparés par des murs constitués chacun de 6 cases.  
- À chaque tour, un joueur sélectionne un angle de tir via un système de points clignotants, puis une puissance via une barre de progression.  
- La balle suit une trajectoire réaliste, influencée par la gravité.  
- Objectifs pour marquer :  
 \* Toucher le mur adverse : supprimer une case et gagner un point.  
 \* Toucher l'adversaire : enlever une vie à l'adversaire et gagner un point.  
- Attention : si la balle touche son propre mur, le joueur perd une vie et l'adversaire gagne un point.  
- Le premier joueur à atteindre 7 points ou faire perdre toutes les vies de l'adversaire gagne.

# 3. Contraintes et fonctionnalités spécifiques :

- Interface console avec affichage des scores, vies, murs, et joueurs ASCII.  
- Sélection d'angle et de puissance via interactions clavier.  
- Trajectoire réaliste influencée par gravité.  
- Gestion précise des collisions avec mur, propre mur, et adversaire.  
- Sons distincts pour tir, impact mur, impact joueur.  
- Fin de jeu avec annonce du gagnant.

# 4. Améliorations, corrections et nouveautés (IA)

a) Correction des collisions murales précises :  
- Recalcul précis des coordonnées du mur, alignement exact des colonnes et lignes.  
- Correction des formules pour collisions.  
- Sécurité pour ne pas dépasser les limites du tableau.  
  
b) Gestion des collisions avec son propre mur :  
- Implémentation d'une méthode spécifique.  
- Pénalité : perte de vie et point pour l'adversaire.  
  
c) Correction des positions de départ :  
- Décalage des joueurs et murs pour éviter tir immédiat sur soi-même.  
- Recalcul dynamique de la sélection d'angle.  
  
d) Ajout de sons différenciés :  
- Tir : laser-shot-ingame-230500.wav.  
- Impact mur : hit-by-a-wood-230542.wav.  
- Impact joueur : metal-hit-15-193280.wav.

# 5. Structure finale des classes et responsabilités

- Game : gestion du déroulement global du jeu.  
- Player : informations sur les joueurs.  
- Ball : gestion des trajectoires et collisions.  
- Wall : gestion des murs.  
- Mur : cellule individuelle.  
- SoundManager : gestion des sons.

# 6. Récapitulatif des méthodes corrigées ou ajoutées grâce à l'IA

- Ball : CalculateTrajectory (corrigée), CheckWallCollision (corrigée), CheckOwnWallCollision (ajoutée).  
- Game : PlayTurn (corrigée), ChooseAngle (corrigée).  
- SoundManager : PlayShot, PlayWallHit, PlayPlayerHit (ajoutées pour gestion sons spécifiques).

# 7. Conclusion et apport de l'IA

L'IA a permis de corriger les collisions, éviter les problèmes de décalage, ajouter des sons contextuels, clarifier la structure du code, améliorer l’interface du jeu, et proposer des fonctions manquantes.  
  
Le jeu est désormais complet, fluide et équilibré, offrant une bonne expérience en console C#.

# 8. Remerciements

Merci à l'enseignant pour ce projet motivant et à l'assistant IA pour l'accompagnement dans les corrections et l'amélioration du gameplay.

Matias Denis - FID1 - I320