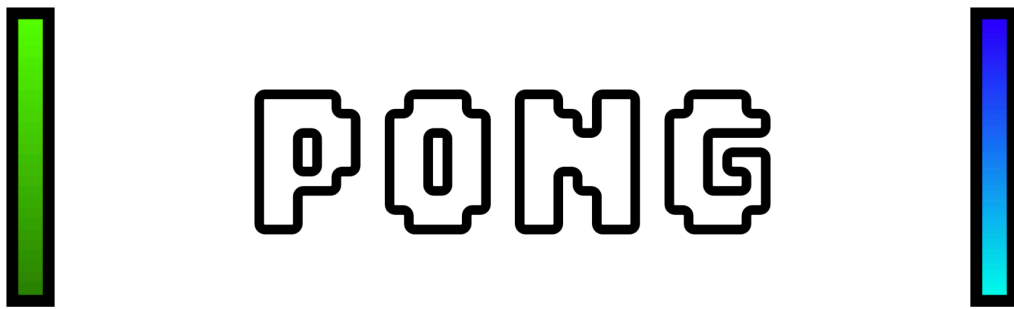


Documentation technique

– Projet Pong



BUT3 DevALT (R5-8)

Équipe de travail :

- Théo Tornatore - Pierrick Marquillies - Théo Le Bastard - Ilyas Sahabi

SOMMAIRE

I. Introduction	3
A. Présentation du jeu	3
A. Objectifs de la documentation technique	3
II. Architecture du jeu	3
A. Moteur de jeu	3
B. Système de rendu	3
C. Gestion des ressources	3
III. Conception du gameplay	4
A. Mécaniques de jeu	4
B. Systèmes de contrôle	4
C. Interactions joueur-environnement	4
IV. Graphismes et Animation	4
A. Modélisation	4
B. Animation	4
V. Optimisation et Performance	5
A. Gestion des ressources système	5
B. Optimisation du code	5
C. Tests de performance	5
VI. Documentation du Code	5
A. Structure du code source	5
B. Commentaires et documentation interne	5
C. Bonnes pratiques	5
VII. Déploiement et Maintenance	6
A. Configuration requise	6
B. Procédure d'installation	6
C. Gestion des mises à jour	6
VIII. Annexes	6
A. Schémas	6
B. Captures d'écran	7

I. Introduction

A. Présentation du jeu

Pong est un jeu vidéo en 2D qui simule un match de tennis de table entre deux joueurs. Chaque joueur contrôle une raquette verticale pour renvoyer une balle qui rebondit sur les murs et les raquettes.

A. Objectifs de la documentation technique

- Décrire l'architecture logicielle du projet.
- Expliquer les choix techniques.
- Fournir des informations pour la maintenance et l'évolution du projet.

II. Architecture du jeu

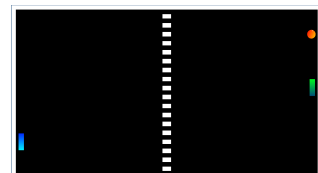
A. Moteur de jeu



Utilisation de Godot comme moteur de jeu et IDE (C#).

B. Système de rendu

Rendu en 2D : terrain rectangulaire, deux raquettes et une balle.
Gestion du rafraîchissement via la boucle de jeu principale.



C. Gestion des ressources

Sprites 2D simples pour la balle, les raquettes et le fond. Pas d'utilisation avancée de sons ou animations complexes.

III. Conception du gameplay

A. Mécaniques de jeu

Balle qui se déplace selon un angle et rebondit sur les murs/raquettes.
Incrémentation du score lorsqu'un joueur laisse passer la balle.

B. Systèmes de contrôle

Joueur 1 : touches Z (haut) et S (bas).
Joueur 2 : flèches ↑ (haut) et ↓ (bas).

C. Interactions joueur-environnement

La balle interagit avec les murs et les raquettes.
Le score s'actualise automatiquement après un point.

IV. Graphismes et Animation

A. Modélisation

Représentation en 2D uniquement : rectangles pour raquettes et terrain, cercle pour la balle.



B. Animation

Déplacement fluide de la balle et des raquettes via calculs de position.

V. Optimisation et Performance

A. Gestion des ressources système

Peu de ressources nécessaires, le jeu reste léger.

B. Optimisation du code

Utilisation de constantes pour éviter les 'magic numbers'.

Boucle de jeu optimisée pour limiter la charge CPU.

C. Tests de performance

Vérification de la fluidité des déplacements.

Contrôle du bon fonctionnement de tous les boutons.

Vérification du calcul correct du score (score final, attribution correcte des points).

Vérification des interactions correctes de la balle avec les parois.

Vérification de la remise en jeu correcte après un but.

VI. Documentation du Code

A. Structure du code source

- Game : boucle principale et gestion du rendu.
- Ball : logique des déplacements et rebonds.
- Paddle : gestion des contrôles et collisions.
- Score : gestion et affichage des points.

B. Commentaires et documentation interne

Chaque classe et méthode est documentée.

C. Bonnes pratiques

Respect des conventions de nommage.
Modularité et réutilisabilité du code.

VII. Déploiement et Maintenance

A. Configuration requise

PC standard avec support du langage choisi (C#).

B. Procédure d'installation

Téléchargement du dépôt Git.
Compilation/lancement selon le langage (exécutable ou script).

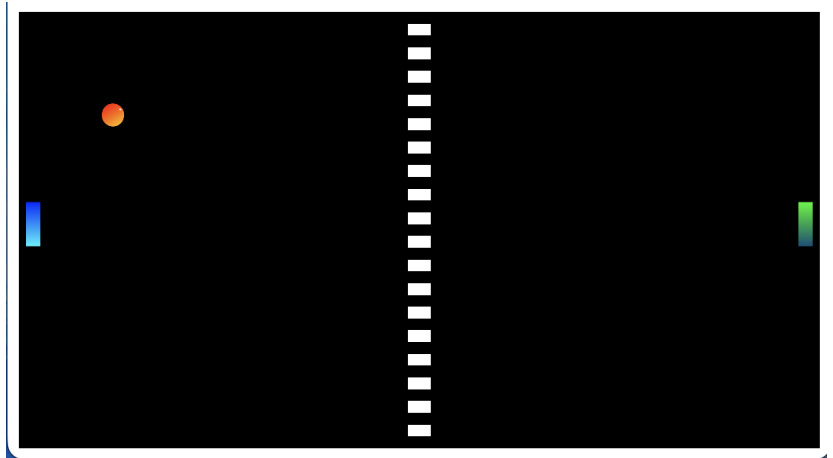
C. Gestion des mises à jour

Versionnement avec Git.
Possibilité d'ajouter de nouvelles fonctionnalités par modules.

VIII. Annexes

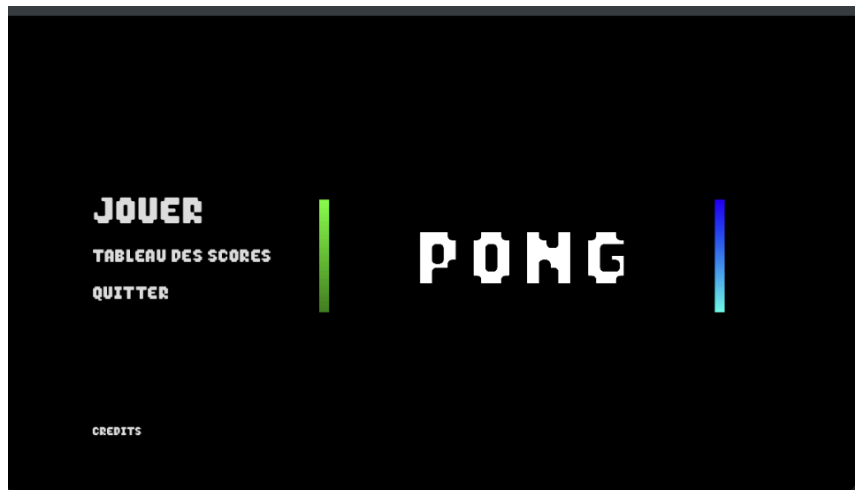
A. Schémas

Schéma du terrain de jeu.



B. Captures d'écran

Menu du Jeu PONG



Credit du jeu Pong

