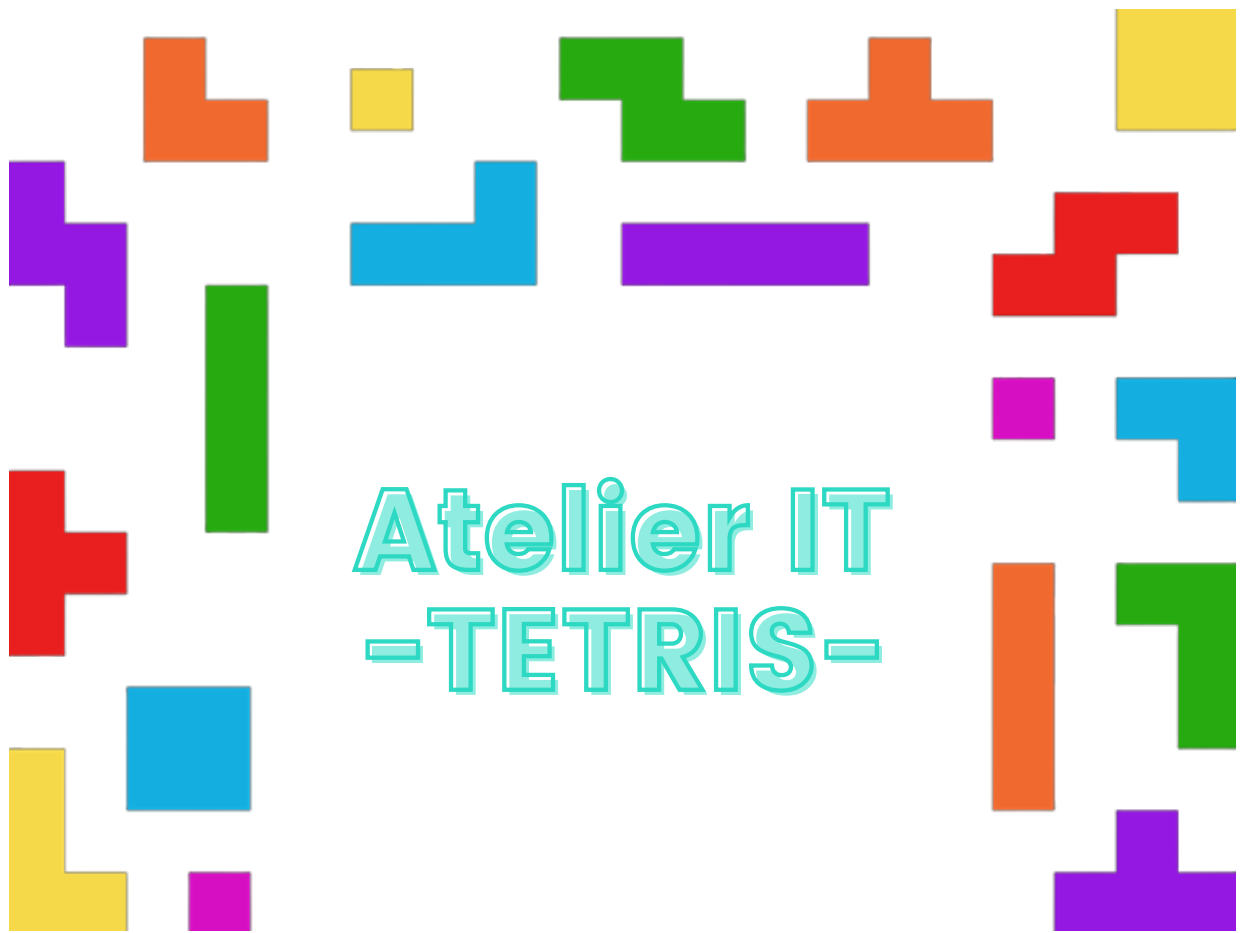


**CLORINDA NGUOKO  
RÉMI MATHIVANAN  
SARAH PLICHTA**



---

**2023**

# Sommaire

---

- 01.** Introduction
- 02.** Objectifs à atteindre
- 03.** Choix du langage
- 04.** Collaboration
- 05.** Difficultés Rencontrées
- 06.** Visualisation du projet (UML)
- 07.** Présentation des fonctionnalités
- 08.** Manuel d'utilisation

# Introduction

---

Le jeu Tetris est un classique intemporel qui a captivé des millions de joueurs à travers le monde depuis sa création. Le défi de manipuler des blocs de différentes formes pour créer des lignes complètes a intrigué et divertit les amateurs de jeux vidéo pendant des décennies.

Dans le cadre de notre mini-projet, notre groupe de trois personnes (Clorinda, Rémi et Sarah) a entrepris de développer une version du jeu propre à nous.

L'objectif principal de notre projet était de répondre aux contraintes imposées tout en proposant des fonctionnalités originales et innovantes en reproduisant au préalable les mécanismes de base du jeu original. Afin de produire de bons résultats, nous avons commencé par choisir un langage de programmation adéquat pour la conception du jeu et ayant pour tous les membres.

# Objectifs à atteindre

---

“

## Objectif Spécifique

---

**"Créer un jeu de Tetris"**

Contraintes à Respecter:

---

Fonctionnalités de l'application

- L'Application devra contenir une interface graphique pour représenter les blocs et l'interface utilisateur.
- Possibilité de déplacer les blocs de gauche à droite.
- Possibilité de faire des rotation des pièces du jeu en utilisant une touche spécifique.
- Une interface dédiée pour le login et des sign up.

Contraintes techniques

- Réaliser en python, java ou javascript
- Utiliser un outil de gestion de version

# Choix du langage

---

Afin de réaliser ce projet, nous avons décidé d'utiliser comme langage de programmation : JavaScript, CSS et HTML.

Pourquoi ce choix?

La réalisation de ce projet nous paraissait plus simple en utilisant du JavaScript, en particulier pour avoir une belle interface.



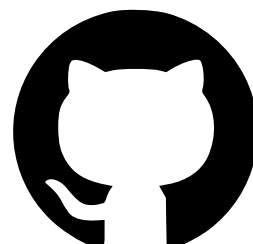
## Collaboration

---

Pour permettre une bonne organisation et communication dans l'équipe, nous avons utilisé Discord et GitHub.

Discord afin de discuter par rapport au projet.

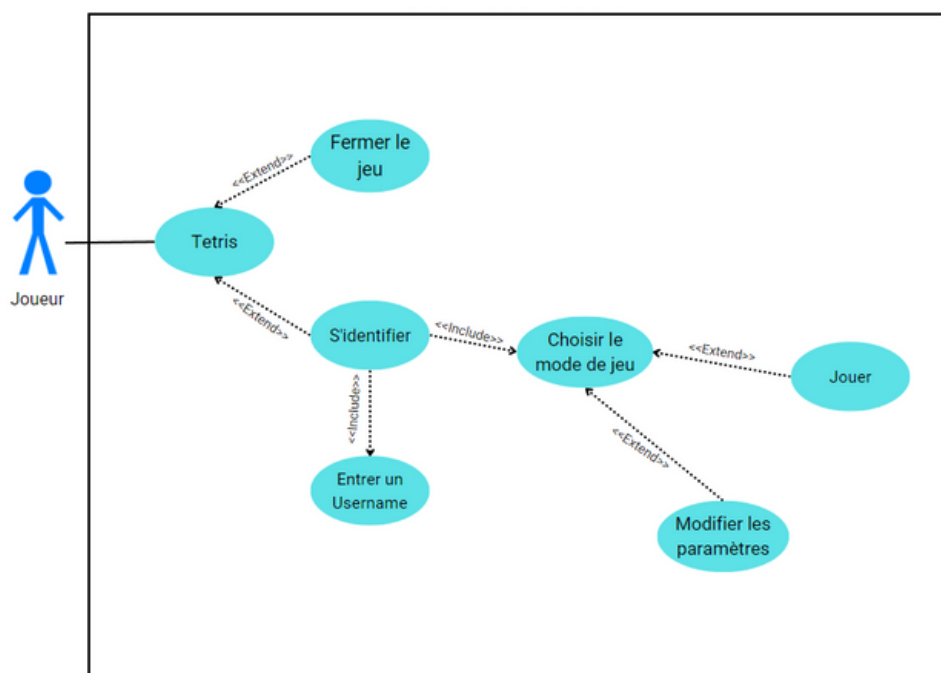
GitHub afin de faciliter les modifications que chacun faisait et permettre aux autres d'avoir le code modifié facilement.



# Diagrammes UML

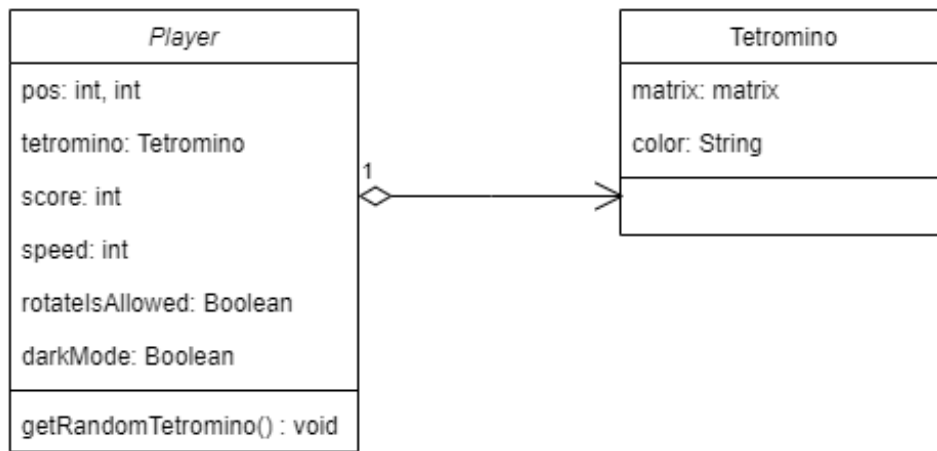
Afin d'avoir une bonne visualisation du projet et de ses fonctionnalités, nous avons tracé des diagrammes UML.

## Use Case



Ce diagramme est utile pour voir ce qu'on peut faire avec le projet : à partir du moment où l'on ouvre le jeu, nous pouvons soit le fermer, soit s'identifier afin de jouer. Nous sommes obligés d'entrer un pseudo pour jouer. Le joueur pourra après choisir un mode de jeu et modifier des paramètres si il le veut.

# Diagramme de Classe



La classe Player détermine/contrôle la position du bloc actuellement en chute (pos), la pièce qui est entrain de descendre dans la grille (tetromino), le nombre de points marqués par le jouer (score), la possibilité de faire tourner les blocs (rotate), le choix entre le mode sombre et éclairé (darkMode) et la vitesse de chute des blocs (speed).

La classe tetromino contient les différentes formes de blocs utilisées dans le jeu(matrix) et leurs couleurs respectives. Pour dessiner les blocs, nous avons utilisées des matrices.

# Difficultés Rencontrées

---

Pendant le processus de développement, nous avons rencontré plusieurs défis auxquels nous avons dû faire face de manière collaborative, en particulier les problèmes liés à la sauvegarde de données ou de paramètres.

Nous avons essayé de relever le défi de créer ce jeu sans utiliser de base de données. Ceci n'est pas été facile notamment pour l'enregistrement du nom des joueurs et l'affichage des statistiques.

En plus de l'aspect technique, nous avons accordé une attention particulière à l'interface utilisateur et à l'expérience utilisateur. Nous avons cherché à créer un design attrayant et convivial, tout en veillant à ce que les contrôles du jeu soient intuitifs et réactifs.

En conclusion, notre mini projet de développement du jeu Tetris nous a permis d'explorer divers aspects de la programmation et de la conception de jeux. Nous sommes fiers du résultat final que nous avons atteint en tant qu'équipe et nous espérons que notre jeu offrira une expérience divertissante à tous ceux qui y joueront.



# Présentation des fonctionnalités réalisées

---

## 3 modes de jeu :

- le mode Classic : ce mode correspond au Tetris que tout le monde connaît.
- le mode Metabloc : ce mode propose des formes de pièces différentes du mode Classic.
- le mode Rush : ce mode contraint le joueur à une minute de jeu avant de se stopper.

## Affichage :

Pour chaque mode de jeu, le joueur peut voir la pièce suivante, le niveau auquel il est, son temps de jeu et son score

## Paramètre modifiable :

- la vitesse : le joueur a le choix de régler la vitesse des pièces entre 11 niveaux.
- la rotation : le joueur peut choisir de bloquer la rotation des pièces pendant sa partie.
- Dark / Light : le joueur peut choisir de jouer en mode Dark (en noir) ou en mode Light (en blanc)

# Manuel d'utilisation

---

- 1 Lancer login.html
- 2 Entrer un pseudo et cliquer sur play
- 3 Sélectionner le mode de jeu
- 4 Utiliser la flèche vers le haut pour effectuer une rotation sur la pièce en cours de jeu
- 5 Utiliser les touches directionnelles pour se déplacer
- 6 Vous pouvez appuyer sur les boutons dédiés pour activer/désactiver les options "Rotate on/off" et "dark/light mode"
- 7 Vous pouvez utiliser les boutons en forme de flèche pour ajuster la vitesse de jeu
- 8 Cliquer sur le bouton "RESTART" pour recommencer la partie
- 9 Pour quitter le jeu, fermer la page Web