

# Analyse du projet IN204

## *Création d'un jeu Tétris multijoueur*

Réalisé par :

Cyriac SALIGNAT  
cyriac.salignat@taep.fr

Théotime GALMICHE  
theotime.galmiche@taep.fr

Encadrants :

Bruno MONSUEZ  
Valentin PERELLE

**ENST2**



**IP PARIS**

Année universitaire 2025–2026

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Analyse et spécifications</b>	<b>2</b>
2.1	Objectifs du projet . . . . .	2
2.2	Fonctionnalités attendues . . . . .	2
2.3	Utilisation des notions vues en cours . . . . .	3
2.4	Choix effectués . . . . .	3
<b>3</b>	<b>Structure du projet</b>	<b>3</b>

# 1 Introduction

Cette analyse intervient dans le cadre du cours IN204, et porte sur la création d'un jeu Tétris en C++ en binôme. Le but est notamment d'utiliser les éléments vus lors des séances du cours : classes, constructeurs, conteneurs, etc.

Ce présent document détaille la direction prise et les choix effectués dans le lancement du développement du projet, et vise à structurer les objectifs ainsi que la manière d'y parvenir.

## 2 Analyse et spécifications

### 2.1 Objectifs du projet

Le but du projet est d'obtenir une version fonctionnelle de Tétris, utilisable en suivant les instructions du README en clonant dépôt Git. Il faut ainsi reproduire toutes les fonctionnalités présentes initialement dans le jeu et détaillés dans la section suivante. Nous avons choisi d'implémenter l'affrontement multijoueur plutôt qu'une possible version 3D, puisque la notion d'hébergement du jeu nous intéressait particulièrement.

Une attention particulière sera donnée à l'expérience utilisateur, en essayant de garantir une interface de jeu claire et intuitive, et des visuels agréables. Dans le même sens, les menus de sélections du mode de jeu ou de la difficulté se voudront ergonomiques.

### 2.2 Fonctionnalités attendues

Afin de fournir un jeu le plus similaire au Tétris original, nous comptons implémenter les Fonctionnalités suivantes :

- Création d'une grille de 22 lignes pour 10 colonnes.
- Apparitions aléatoires et périodiques des blocs parmi la liste des 7 motifs possibles ;
- Rotation des pièces sous quatre angles différents ( $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  et  $270^\circ$ ) ;
- Disparition des blocs d'une ligne lorsqu'elle est entièrement remplie ;
- Chute des blocs supérieurs lorsqu'une ligne disparaît ;
- Arrêt du jeu lorsqu'un bloc dépasse la limite haute de la grille.
  
- Existence d'un système de points, prenant en compte le nombre de lignes supprimées simultanément et la difficulté de la partie (choisie avant le lancement de celle-ci, et allant de 1 à 15) ;
- Définition du système de difficulté : joue sur la vitesse de chute des blocs, et, par conséquent, sur le délai de réaction que possède le joueur.
  
- Possibilité de jouer en multijoueur ;
- Existence d'un mode "Affrontement solo", dans lequel le joueur pourra s'exercer en duel contre un Bot.

## 2.3 Utilisation des notions vues en cours

- **Classes et héritage de classes** : créer une classe avec toutes les fonctionnalités communes aux pièce, puis des classes héritières avec les spécificités de chacune des pièces.
- **Conteneurs** : créer les matrices pour la grille et pour les pièces.

## 2.4 Choix effectués

Un premier choix a été d'utiliser SFML pour l'affichage graphique, ainsi que la gestion du temps et des sons.

Ensuite, nous pensons opter pour SFML Network pour la partie multijoueur, afin de maximiser la compatibilité avec SFML

## 3 Structure du projet

```
IN204/  
| src/  
|   |----main.cpp  
|   |---- grille.cpp  
|   |---- pieces.cpp  
|   |---- score.cpp  
|   |---- interface.cpp  
|   |---- res/  
|       |---- font.ttf  
|---- build/  
|---- README.md
```