

# Rapport projet conception logicielle

Romuald GAFPE      Yanis COSNEFROY  
Vy Vu NGUYEN PHUONG      Alexandre BOURGOIN

27 mars 2020

# Table des matières

<b>I</b>	<b>Objectifs du projet</b>	<b>2</b>
0.1	Présentation du concept . . . . .	3
0.2	Cahier des charges . . . . .	3
0.3	Exemples du jeux publiés . . . . .	3
<b>II</b>	<b>Fonctionnalités implémentés</b>	<b>5</b>
0.4	Mode solo . . . . .	6
0.5	Mode multijoueur . . . . .	6
0.5.1	Joueur vs Intelligence artificielle . . . . .	6
0.5.2	Joueur vs Joueur en local . . . . .	6
0.5.3	Joueur vs Joueur en ligne . . . . .	6
<b>III</b>	<b>Éléments techniques</b>	<b>7</b>
0.6	La grille . . . . .	8
0.7	Les pièces . . . . .	8
0.8	L'aléatoire . . . . .	8
0.9	L'intelligence artificielle . . . . .	8
0.10	Le réseau . . . . .	8
<b>IV</b>	<b>Architecture du projet</b>	<b>9</b>
0.11	Diagrammes des modules et des classes . . . . .	10
0.12	Cas d'utilisation . . . . .	10
<b>V</b>	<b>Conclusion</b>	<b>11</b>

Première partie

**Objectifs du projet**

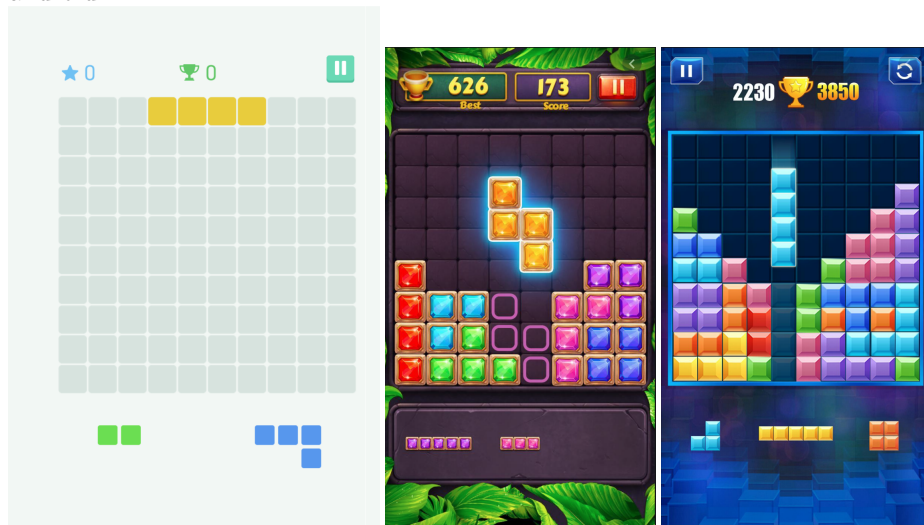
## 0.1 Présentation du concept

Tout d'abord, nous avons choisi de créer un puzzle block puisque nous avions plus d'affinité et d'idées en rapport avec ce sujet. Nous voulions que ce jeu soit simple d'utilisation afin que tout le monde puisse y jouer. Le jeu que nous avons développé est composé de 4 modes de jeux, le premier étant le mode de jeu solo, il se présente sous forme d'une grille carré de 10 cases de côté. Le joueur obtient 3 pièces à placer sur la grille tirées aléatoirement parmi un total de 30 pièces (toutes orientations inclus). Les pièces qui apparaissent doivent toutes être placées dans la grille avant que les suivantes soit tirés. Si le tirage courant du joueur ne peut pas être placé alors un écran de fin de jeu s'affiche, montrant le score de la personne. Sur ce menu, le joueur aura la possibilité de retourner au menu principal ou de recommencer une partie. Ensuite le jeu se compose d'un mode multijoueur. Le premier étant contre une intelligence artificielle ayant le même mode de fonctionnement que le mode solo. Seul le score actuel de l'intelligence artificielle figure sur l'écran du joueur afin d'éviter d'en copier la stratégie. Le second mode multijoueur est, lui, un mode joueur contre joueur en local sur une même machine. Les grilles des joueurs s'affichent l'une après l'autre afin de limiter, de même, l'imitation de la stratégie adverse. Le dernier mode de jeu est un mode joueur contre joueur également mais en ligne où le but est de faire le plus de points.

## 0.2 Cahier des charges

## 0.3 Exemples du jeux publiés

Voici plusieurs block-puzzle provenant de jeux.fr, et des applications pour android :



On peut y retrouver les mêmes caractéristiques que notre projet. Notre projet a été conçu sur une base classique de block-puzzle qui réunit ces trois images montrées précédemment. Les différences entre ces block-puzzles et le notre sont la décoration du jeu ainsi que la sauvegarde du meilleur score établis sur le jeu par le joueur. On peut constater à l'essai de notre block-puzzle uniquement l'apparition des scores qui vont être actifs en fonction du jeu du joueur. Cependant nos règles sont un peu différentes par rapport au jeu déjà publié. On a décidé de ne pas rajouter de bonus ou d'aide à chaque fois qu'une ligne ou une colonne est remplie. C'est un choix volontaire pour provoquer une augmentation de concurrence de jeu avec d'autres joueurs mais aussi pour une adaptation plus développée de la difficulté de jeu.

Deuxième partie

**Fonctionnalités  
implémentés**

## **0.4 Mode solo**

## **0.5 Mode multijoueur**

### **0.5.1 Joueur vs Intelligence artificielle**

### **0.5.2 Joueur vs Joueur en local**

### **0.5.3 Joueur vs Joueur en ligne**

Troisième partie

Éléments techniques



## **0.6 La grille**

La grille est mise sous la forme d'une classe,

## **0.7 Les pièces**

## **0.8 L'aléatoire**

## **0.9 L'intelligence artificielle**

## **0.10 Le réseau**

Quatrième partie

**Architecture du projet**

**0.11 Diagrammes des modules et des classes**

**0.12 Cas d'utilisation**

**Cinquième partie**

**Conclusion**