

## PROJET D'APOO

### ANALYSE DU JEU DE BLOCAGE AVEC POLYOMINOS

---

Antoine CUINET  
Gaspar QUENTIN

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Les fonctionnalités</b>	<b>3</b>
<b>3</b>	<b>Les classe</b>	<b>4</b>
3.1	La classe Main . . . . .	4
3.2	La classe Computer . . . . .	4
3.3	La classe Piece . . . . .	4
3.4	La classe Grid . . . . .	4
3.5	La classe Position . . . . .	4
3.6	La classe Case . . . . .	5
3.7	La classe Matrix . . . . .	5
3.8	La classe Domino . . . . .	5
3.9	La classe Triomino . . . . .	5
3.10	La classe Tetromino . . . . .	5
<b>4</b>	<b>Initialisation de jeu</b>	<b>6</b>
4.1	Dimensions du jeu . . . . .	6
4.2	Distribution des pièces . . . . .	6
<b>5</b>	<b>Description des conditions</b>	<b>7</b>
5.1	Placement des pièces . . . . .	7
5.2	La partie est finie . . . . .	7
<b>6</b>	<b>Jeu de l'ordinateur</b>	<b>8</b>
<b>7</b>	<b>Ajouts/libertés prises</b>	<b>9</b>
<b>8</b>	<b>Conclusion</b>	<b>10</b>

# Introduction

Le projet est un projet de fin d'année de première année de licence informatique à l'université de Franche-Comté.

Ce rapport décrit et explique la conception de ce projet, qui est le jeu de blocage avec polyominos.

Le jeu est codé en Java et le jouer joue contre un ordinateur depuis le terminal.

Le but du jeu est relativement simple, chaque joueur dispose d'un ensemble de pièces, qui sont des polyominos, et qu'il place tour à tour sur un plateau, qui est sous forme de grille rectangulaire.

Il s'agit d'un jeu de blocage : le perdant est le premier joueur qui ne peut plus placer de pièce.

Le jeu se déroule en mode humain contre ordinateur : le joueur humain interagit via des affichages et des saisies sur la console.

Ce jeu à été coder par CUINET Antoine et QUENTIN Gaspard.

# Les fonctionnalités

Les fonctionnalités du jeu sont multiples.

Le joueur peut

// ici à faire

# Les classe

Le jeu dispose de 10 classes.

*// faire les classes, bonne chance à celui qui le fera*

## 3.1 La classe Main

La classe Main est comme son nom l'indique la classe principale de notre projet. Celle ci contient la méthode statique main qui s'occupe de créer les bons objets et d'appeler les bonnes méthodes afin de créer une partie. Cette classe contient également les méthodes permettant à l'utilisateur de jouer.

## 3.2 La classe Computer

La classe Computer s'occupe de faire jouer l'ordinateur contre le joueur. Les pièces placées par l'ordinateur sont toutes choisies de manière aléatoire et placées de manière aléatoire également.

## 3.3 La classe Piece

La classe Pièce est une classe abstraite servant de modèles pour les classes Domino, Triomino et Tetromino qui en hériteront. Cette classe contient plusieurs méthodes virtuelles pures telles que getPosition() (qui renvoie un tableau de Position représentant la pièce dans un espace orthonormé) ou encore la méthode toString() qui s'occupera d'afficher la pièce pour aider l'utilisateur. Ces dernières seront donc entièrement redéfinies dans les classes enfant.

## 3.4 La classe Grid

La classe Grid représente la grille de jeu. Elle se charge également de l'affichage du plateau de jeu. Elle est composée d'un tableau de 2 dimensions d'objets de type Case. Elle comporte également les méthodes permettant au joueur et à l'ordinateur de voir si il est possible de placer une pièce (méthode isPiecePlaceable()) et de la placer (méthode placePiece).

## 3.5 La classe Position

Cette classe est constituée de deux attributs : une position en abscisse sur la grille et une position en ordonnée. Ces deux attributs sont des entiers. Elle possède également plusieurs méthodes utiles comme la méthode add() qui permet d'additionner deux positions entre elles.

### 3.6 La classe Case

Cette classe représente une case sur le plateau. Une case contient une pièce qui peut être nulle. Elle possède également une redéfinition de la méthode toString() permettant d'afficher un rond si une pièce se trouve sur cette case et qu'elle appartient au joueur, un croisillon si une pièce se trouve sur cette case mais que cette pièce est une pièce ennemie et rien sinon.

### 3.7 La classe Matrix

Une pièce étant constituée de plusieurs objets Position, pour trouver l'orientation voulue pour placer la pièce dans la grille, nous avons dû avoir recours à quelques petits calculs matriciels. Cette classe sert donc principalement à calculer des coordonnées rotationnées à l'aide de cette formule mathématique :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

### 3.8 La classe Domino

La classe Domino, qui hérite de la classe Piece, se charge de créer des dominos avec le bon type en fonction du nombre d'instances déjà créés. Cette classe comporte une surcharge de la méthode toString() permettant d'afficher toutes les différentes dispositions des dominos et de la méthode getPositions() renvoyant toutes les positions représentant la pièce créée dans un repère orthonormé.

### 3.9 La classe Triomino

Cette classe fonctionne de façon identique à celle de la classe Domino.

### 3.10 La classe Tetromino

La classe Tetromino comporte la même logique d'implémentation que ces deux classes soeurs, la classe Domino et la classe Triomino.

# Initialisation de jeu

Au début du jeu, un affichage permettant de saisir de pseudo du joueur est mis à l'écran.

Une fois cette saisie faite, une explication du jeu, de l'ordinateur (qui se nome Yumi), ainsi que des pièces disponible est présenté.

## 4.1 Dimensions du jeu

Le plateau de jeu est un rectangle de 12 cases par 10 cases, qui sont numérotés en lettres (de A à L) en abscisse et en chiffre (de 0 à 9) en ordonné.

## 4.2 Distribution des pièces

Au début du jeu, le joueur comme l'ordinateur ont à disposition un même nombre de pièce.

Il est distribué 3 Dominos, 6 Triomino posable en 2 positions avec 3 pièces disponibles par position, et 9 Tétrominos posable en 7 positions avec 1 pièce disponible par position sauf pour les tétrominos de formes I et T qui sont aux nombre de 2.

Ainsi, chaque joueur un nobre de 60 cases sur un plateau faisant en tout 120 cases.

# Description des conditions

Plusieur types de conditions sont utilisé afin de faire fonctionner au mieux le jeu.

## 5.1 Placement des pièces

// ici à faire

## 5.2 La partie est finie

// ici à faire



# Jeu de l'ordinateur

Par manque de temps pour réaliser ce projet, notre ordinateur choisi ses pièce et les places aléatoirement.

# Ajouts/libertés prises

Afin de rajouter de la créativité, l’affichage de la grille est réalisé de façon originale.

Le placement des pièces dans la grilles se fait en couleur, de façon à distinguer facilement et faire un affichage coloré (rouge pour l’ordinateur et bleu pour le joueur).

Le nom du joueur ainsi que celui de l’ordinateur sont eux aussi en couleur (de même, rouge pour l’ordinateur et bleu pour le joueur).

Enfin, l’affichage des pièces en vert, qui est guidé tout au long du processus de choix d’une pièce, permet au joueur de facilement se rendre compte des pièces possible de poser.

# Conclusion

Pour conclure ce rapport, nous avons produit un jeu amusant, autant dans la conception que dans son utilisation.

Certain problèmes ont été rencontrés comme celui de faire les tétraminons qui a été très long (autant dans la création des pièces que dans les affichages). De plus, celui de distinguer les formes possibles de chaque pièce, en plus des différentes pièces et de leurs orientations.

Ces problèmes ont bien évidemment trouvé leurs solutions et nous ont permis de nous surpasser dans la création de ce jeu.

Concevoir ce jeu nous a également permis de mettre en pratique les cours appris tout au long de l'année et spécifiquement la programmation orientée objet.