

Gronier Thibaut

Gaspard Corentin

Caudron Jonathan

Dubois Maxence

Table des matières

[I) Introduction 3](#_Toc74415349)

[II) Développement 3](#_Toc74415350)

[**2.1** **Etape 1 : Prise en main de la librairie graphique** 3](#_Toc74415351)

[2.1.1 Contexte et objectifs : 3](#_Toc74415352)

[2.1.2 Thème : 4](#_Toc74415353)

[2.1.3 Programme : 4](#_Toc74415354)

[**2.2** **Etape 2 : Implémentation des pièces du jeu** 4](#_Toc74415355)

[2.2.1 Contexte et objectifs : 4](#_Toc74415356)

[2.2.2 Programme : 5](#_Toc74415357)

[**2.3** **Etape 3 : Implémentation des objets et du plateau de jeu** 5](#_Toc74415358)

[2.3.1 Contexte et objectifs : 5](#_Toc74415359)

[2.3.2 Programme : 6](#_Toc74415360)

[**2.4** **Etape 4 : Implémentation de la classe Joueur** 6](#_Toc74415361)

[2.4.1 Contexte et objectifs : 6](#_Toc74415362)

[2.4.2 Programme : 6](#_Toc74415363)

[2.4.3 Diagramme de classe : 8](#_Toc74415364)

[**2.5** **Etape 5 : implémentation d’une première version du jeu** 8](#_Toc74415365)

[2.5.1 Contexte et objectif : 8](#_Toc74415366)

[2.5.2 Programme : 8](#_Toc74415367)

[2.5.3 Diagramme de classe : 10](#_Toc74415368)

[**2.6** **Etape 6 : Développement de joueurs « ordinateur »** 11](#_Toc74415369)

[2.6.1 Contexte et objectif : 11](#_Toc74415370)

[**2.7** **Comment les modules de POO et COO ont été utilisés** 11](#_Toc74415371)

[**2.8** **Répartition du travail** 11](#_Toc74415372)

[III) Conclusion : 11](#_Toc74415373)

[**3.1 Bilan global** 11](#_Toc74415374)

[**3.2 Bilan personnelle** 11](#_Toc74415375)

[3.2.1 Bilan de Corentin Gaspard 11](#_Toc74415376)

[3.2.2 Bilan de Maxence Dubois 11](#_Toc74415377)

[3.2.3 Bilan de Jonathan Caudron 11](#_Toc74415378)

[3.2.4 Bilan de Thibaut Gronier 11](#_Toc74415379)

[**3.3 Ce qui pourrait être amélioré** 11](#_Toc74415380)

# Introduction

Ce projet consiste à créer un jeu en java. Nous allons donc concevoir le jeu Labyrinthe avec la librairie graphique swing pour l’interface graphique.

# Développement

## **Etape 1 : Prise en main de la librairie graphique**

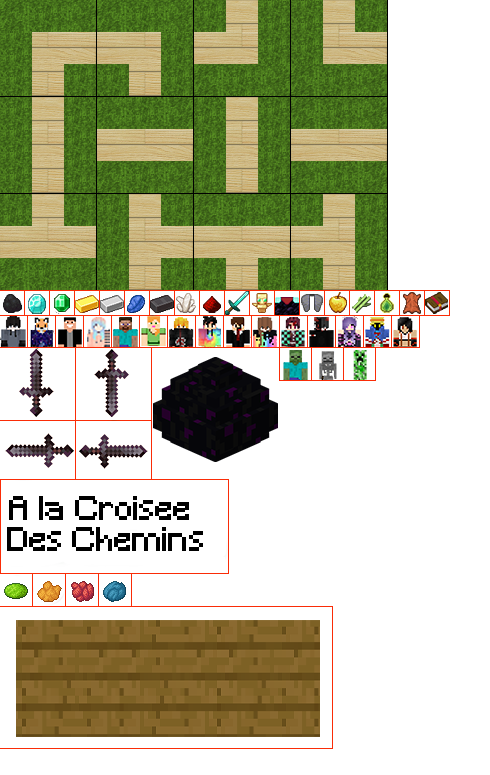
### Contexte et objectifs :

Nous allons donc utilisez la librairie swing qui permet d’ouvrir la fenêtre de paramètre du jeu et la fenêtre du jeu avec le plateau du jeu. Nous avons inclus la librairie dans nos programmes avec la classe IG.

Les objectifs de cette étape :

* Créer un projet java dans un IDE de programmation Java
* Utilisé la librairie graphique
* Utilisé un dépôt GitHub pour le partage du travail

### 2.1.2 Thème :

Nous avons créé notre propre thème qui a été intégré dans la librairie.

### Programme :

Nous avons commencé à utiliser cette librairie. Dans cette partie, nous avons regardé comment placer des pièces sur le plateau, faire une rotation des pièces, placer des objets et des personnages sur le plateau.

## **Etape 2 : Implémentation des pièces du jeu**

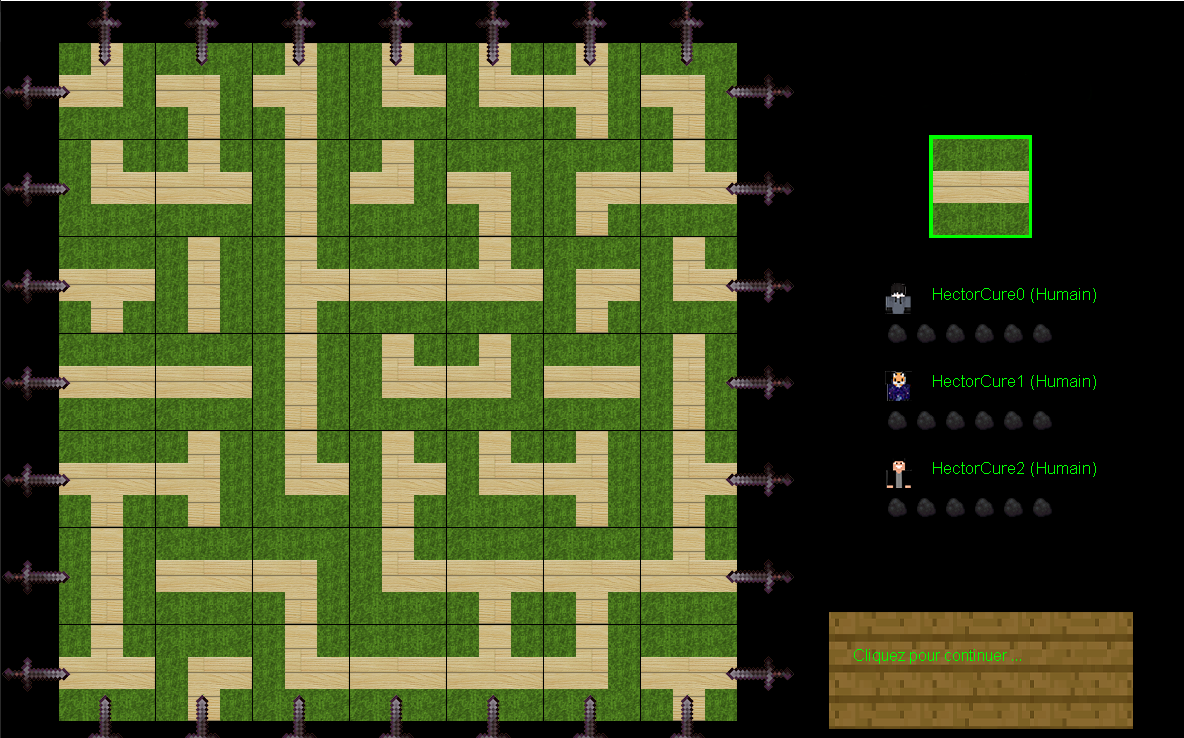
### Contexte et objectifs :

Dans cette étape nous allons implémenter les classes qui permettent la représentation des composants du jeu. Nous allons implémenter les classes qui correspondent aux pièces du jeu, au plateau du jeu, aux joueurs et une partie.

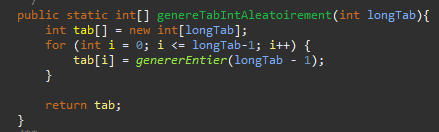
Notre objectif principal de cette étape est de faire la classe qui représente les pièces du jeu.

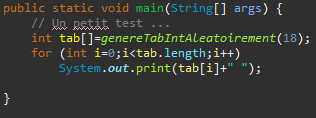
### Programme :

*Testpieces.java* :

Tout d’abord nous avons placé des pièces aléatoirement sur le plateau puis nous avons fait une rotation des pièces.

*Utils* : Explication de l’algorithmes du tirage aléatoire de valeur entières

 Des entiers entre *0* et *longTab - 1* placé aléatoirement dans un tableau.

Dans cette partie les entiers générés aléatoirement sont affichés.

*Piece.java* :

Cette classe permet de faire une rotation à une pièce. La classe génère un tableau avec les 50 pièces qui seront placées sur le plateau.

## **Etape 3 : Implémentation des objets et du plateau de jeu**

### Contexte et objectifs :

Pour cette étape nous allons continuer le développement des composants du jeu. Notre objectif est la création des classes pour les objets et le plateau.

### Programme :

*Objet.java :*

Cette classe permet de construire un tableau avec les 18 objets du jeu et de les placer sur le plateau. Cette classe peut positionner un objet à un autre lieu de départ ou même l’enlever du plateau. Nous pouvons savoir si à une ligne et colonne spécifique se trouve un objet.

*TestObjet.java* :

Nous avons placé des objets aléatoires sur des pièces aléatoires du plateau.



## **Etape 4 : Implémentation de la classe Joueur**

### Contexte et objectifs :

Dans l’étape précédente, nous avons ajouté les composants physiques du jeu. Maintenant nous allons poursuivre le développement avec les joueurs et les éléments d’une partie.

Notre objectif principal est le développement de la classe Joueur.

### Programme :

Joueur.java :

Dans cette partie nous avons juste complété la classe Joueur. Cette classe affecte à chaque nouveau joueur une position sur le plateau, des objets qu’il devra récupérer durant la partie. La classe permet de savoir le prochain objet à récupérer par le joueur et si le joueur a trouvé tous ses objets. Pour pouvoir récupérer les objets la classe permet le déplacement en modifiant la position du joueur. Le joueur pourra choisir l’orientation de la pièce hors plateau pour la placer sur le bord du plateau.

TestJoueur.java :

Dans ce test nous avons généré un plateau aléatoire et permettre à chaque joueur de se déplacer une fois.

### Diagramme de classe :

## **Etape 5 : implémentation d’une première version du jeu**

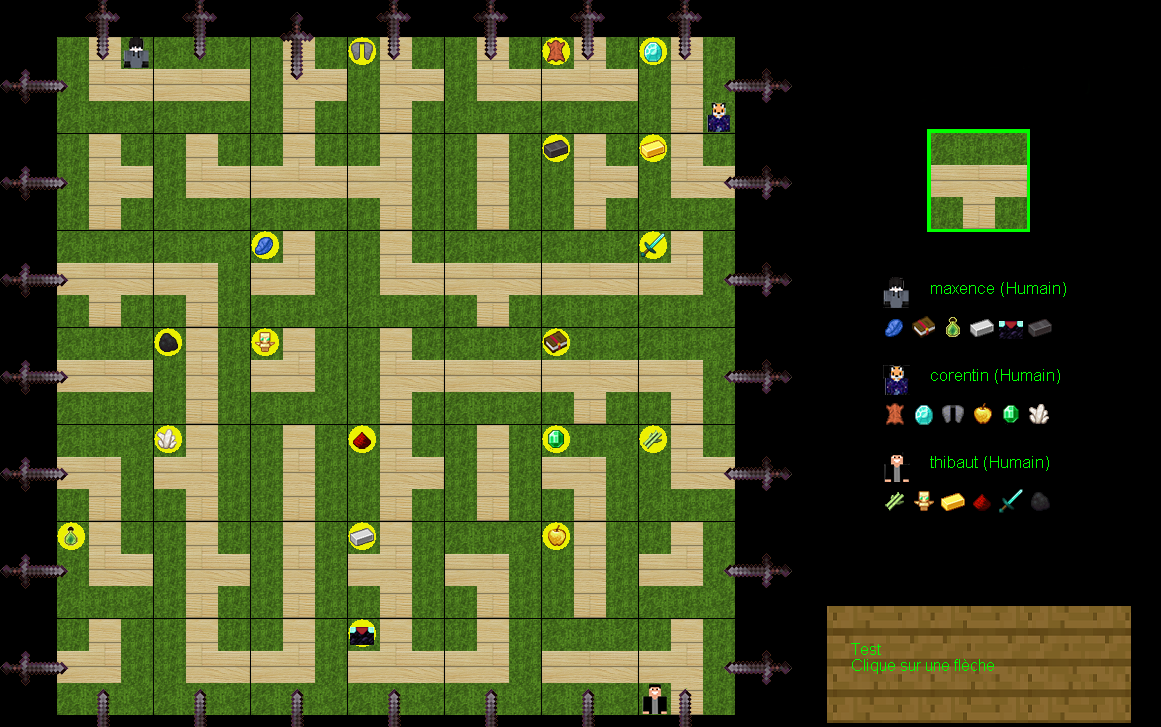
### Contexte et objectif :

Dans cette étape le développement va être poursuivi avec les éléments qui constituent une partie du jeu, pour obtenir la première version jouable du Labyrinthe.

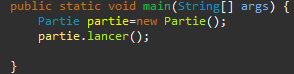
### Programme :

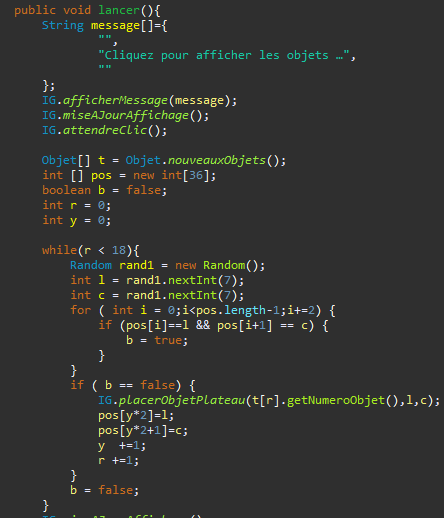
*ElementsPartie.java* :

Cette classe permet d’attribuer des objets à chaque joueur de manière aléatoire. Elle permet la chose la plus importante de ce jeu, le fait de prendre une pièce est de la placer sur le bord du plateau pour que toutes les autres pièces se décalent.

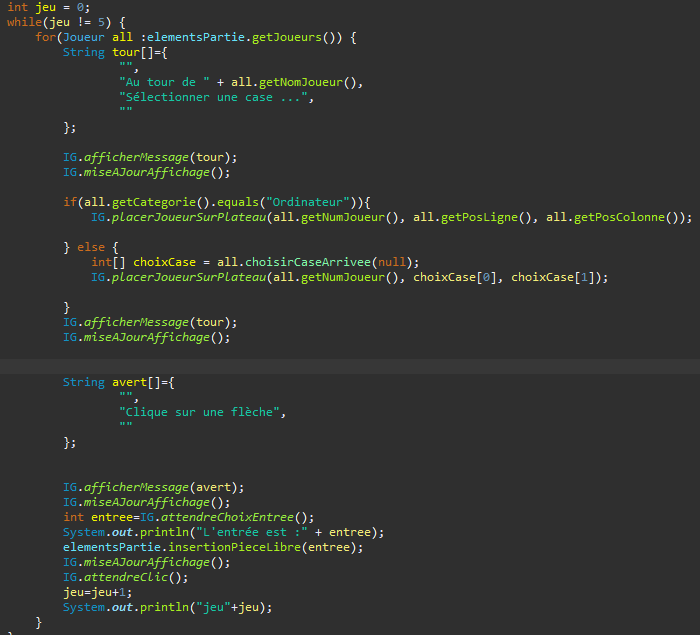
Le résultat de *TestElementPartie* :

*Partie.java :* L’algorithme de la boucle principal

Cette classe permet l’initialisation de la partie avec les éléments des joueurs, du plateau et les objets. Avec cette classe nous pourrons faire une partie.

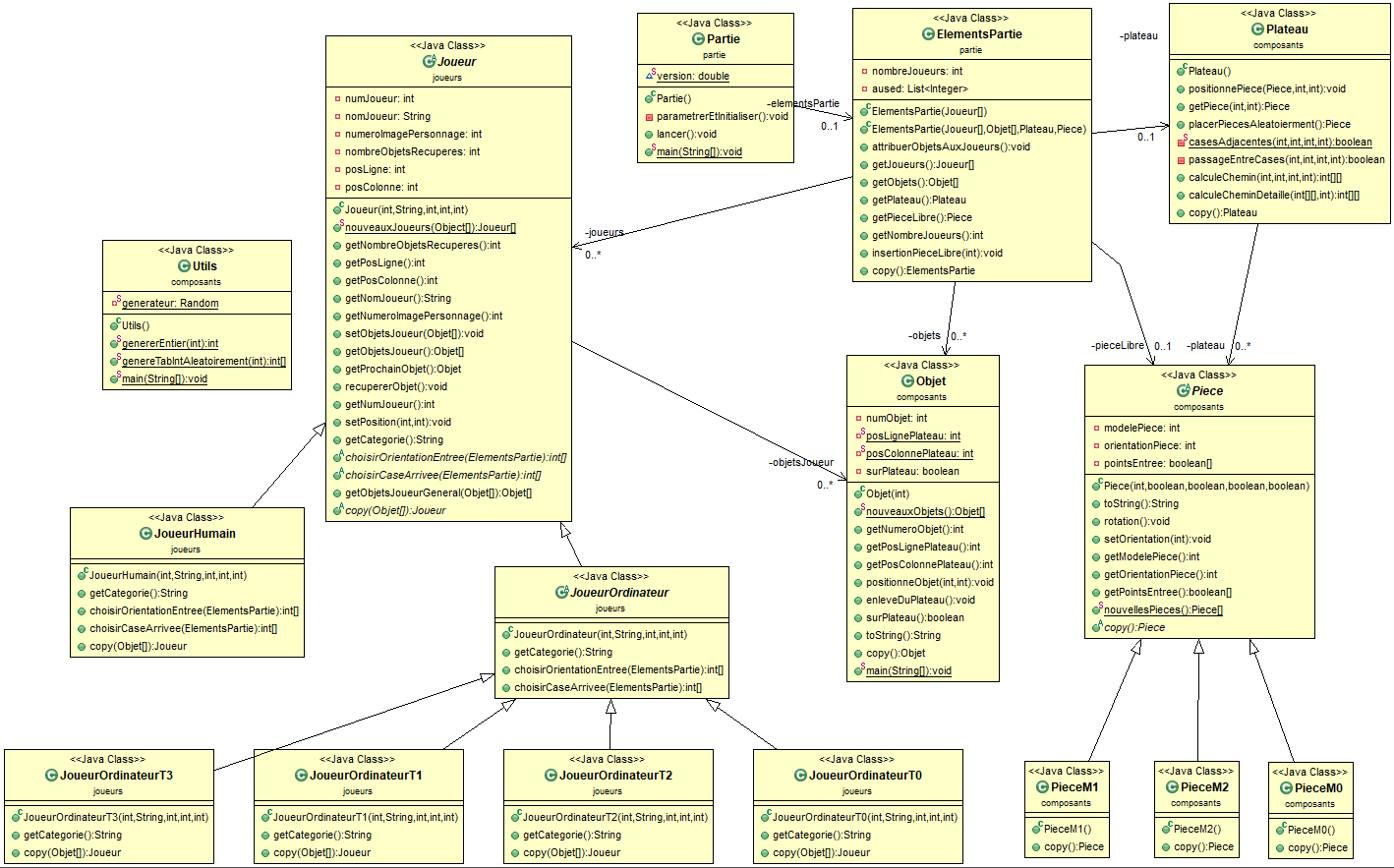
Avec cette partie de programme, nous allons d’abord lancer la partie, on va appeler lancer ().

Quand lancer () sera appelé, nous allons créer neuf objets aléatoires par personne puis les placer sur le plateau.



Ici dès que chaque joueur aura joué 5 fois la partie sera terminée. Pendant la partie, d’abord le joueur 1 se déplace et fait rentrer la pièce hors plateau sur le plateau, puis c’est au tour du joueur 2 et joueur 3. Voilà un tour de fait, plus que cinq.

### Diagramme de classe :



## **Etape 6 : Développement de joueurs « ordinateur »**

### Contexte et objectif :

Dans cette étape nous allons terminer le jeu avec les joueurs de type ordinateur. Nous allons donc implémenter dans le jeu des joueurs ordinateurs améliorer par rapport à celui de base.

## **Comment les modules de POO et COO ont été utilisés**

A remplir

## **Répartition du travail**

A remplir

# Conclusion :

## **3.1 Bilan global**

## **3.2 Bilan personnelle**

### 3.2.1 Bilan de Corentin Gaspard

### 3.2.2 Bilan de Maxence Dubois

### 3.2.3 Bilan de Jonathan Caudron

### 3.2.4 Bilan de Thibaut Gronier

## **3.3 Ce qui pourrait être amélioré**