

Gronier Thibaut

Gaspard Corentin

Caudron Jonathan

Dubois Maxence

Année 2020/2021

PROJET TUTORE SEMESTRE 2

Table des matières

[I) Introduction 3](#_Toc74509373)

[II) Développement 3](#_Toc74509374)

[**2.1** **Etape 1 : Prise en main de la librairie graphique** 3](#_Toc74509375)

[2.1.1 Contexte et objectifs : 3](#_Toc74509376)

[2.1.2 Thème : 3](#_Toc74509377)

[2.1.3 Programme : 4](#_Toc74509378)

[**2.2** **Etape 2 : Implémentation des pièces du jeu** 4](#_Toc74509379)

[2.2.1 Contexte et objectifs : 4](#_Toc74509380)

[2.2.2 Programme : 4](#_Toc74509381)

[**2.3** **Etape 3 : Implémentation des objets et du plateau de jeu** 5](#_Toc74509382)

[2.3.1 Contexte et objectifs : 5](#_Toc74509383)

[2.3.2 Programme : 5](#_Toc74509384)

[**2.4** **Etape 4 : Implémentation de la classe Joueur** 6](#_Toc74509385)

[2.4.1 Contexte et objectifs : 6](#_Toc74509386)

[2.4.2 Programme : 6](#_Toc74509387)

[2.4.3 Diagramme de classe : 7](#_Toc74509388)

[**2.5** **Etape 5 : implémentation d’une première version du jeu** 7](#_Toc74509389)

[2.5.1 Contexte et objectif : 7](#_Toc74509390)

[2.5.2 Programme : 7](#_Toc74509391)

[2.5.3 Diagramme de classe : 9](#_Toc74509392)

[**2.6** **Etape 6 : Développement de joueurs « ordinateur »** 10](#_Toc74509393)

[2.6.1 Contexte et objectif : 10](#_Toc74509394)

[**2.7** **Comment les modules de POO et COO ont été utilisés** 10](#_Toc74509395)

[**2.8** **Répartition du travail** 10](#_Toc74509396)

[III) Conclusion : 11](#_Toc74509397)

[**3.1 Bilan global** 11](#_Toc74509398)

[**3.2 Bilan personnelle** 11](#_Toc74509399)

[3.2.1 Bilan de Corentin Gaspard 11](#_Toc74509400)

[3.2.2 Bilan de Maxence Dubois 11](#_Toc74509401)

[3.2.3 Bilan de Jonathan Caudron 11](#_Toc74509402)

[3.2.4 Bilan de Thibaut Gronier 12](#_Toc74509403)

# 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**

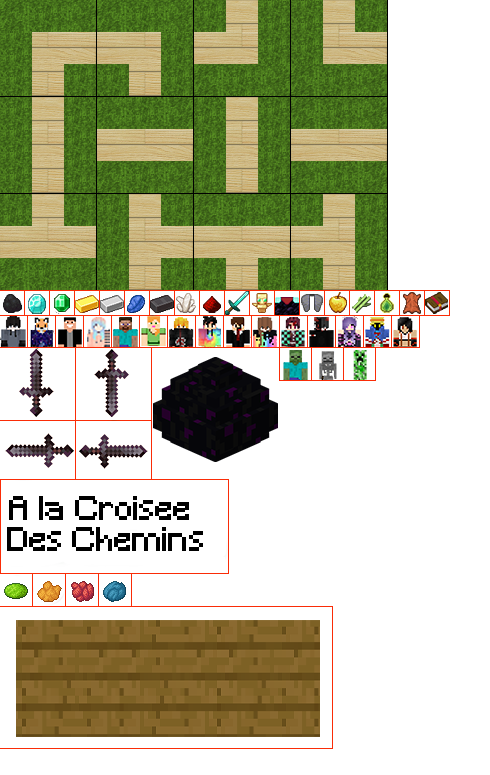
### Contexte et objectifs :

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**

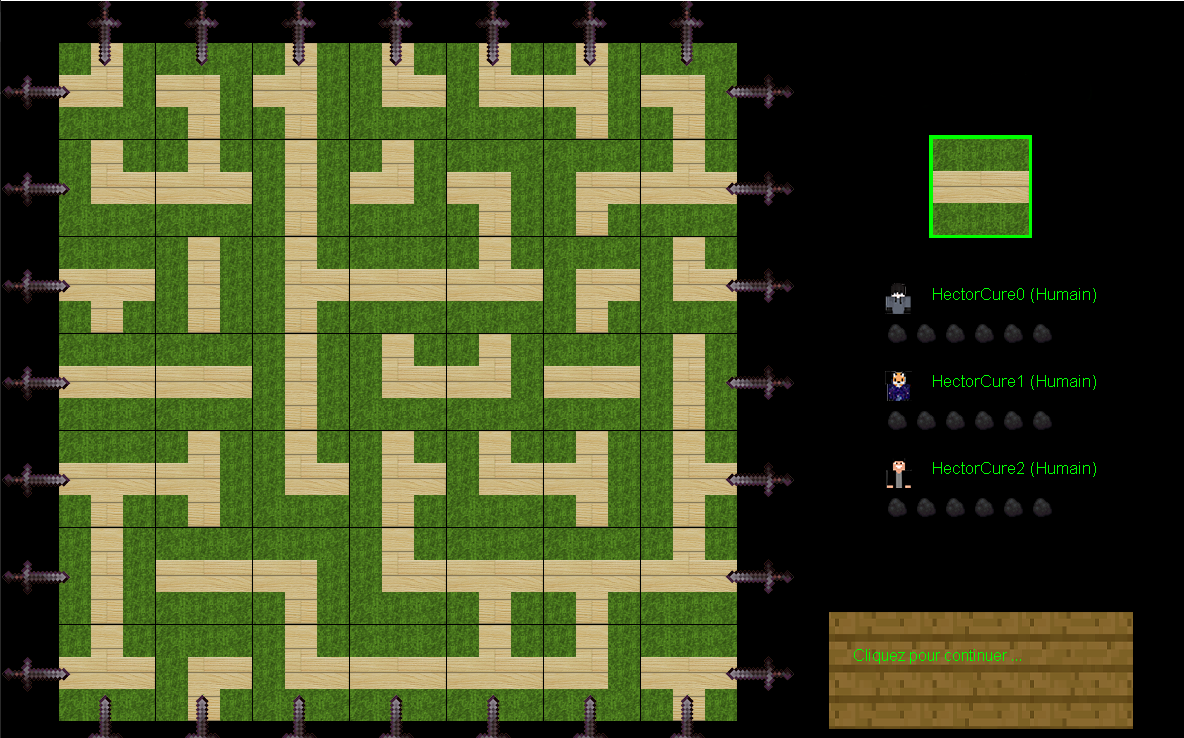
### Contexte et objectifs :

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.

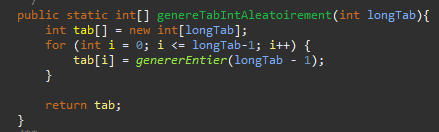
Notre objectif principal de cette étape est de faire la classe qui représente les pièces du jeu.

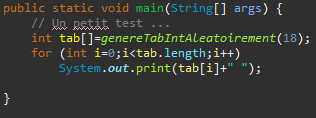
### Programme :

*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’algorithme 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**

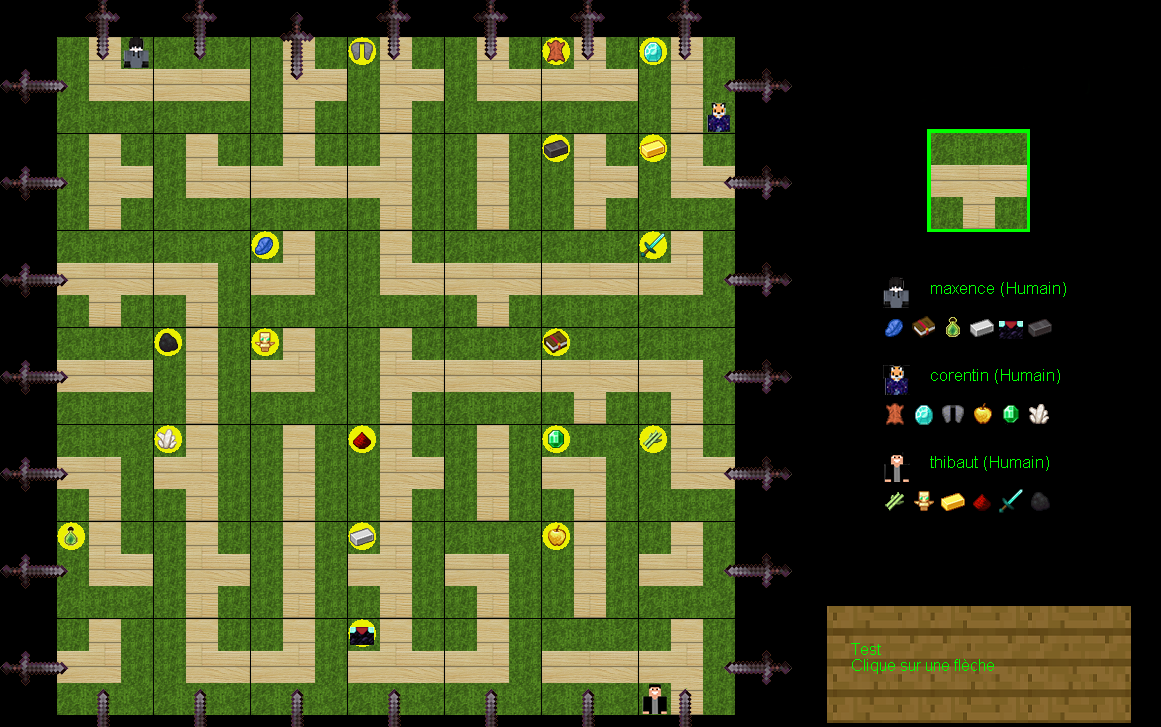
### Contexte et objectif :

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.

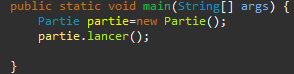
### Programme :

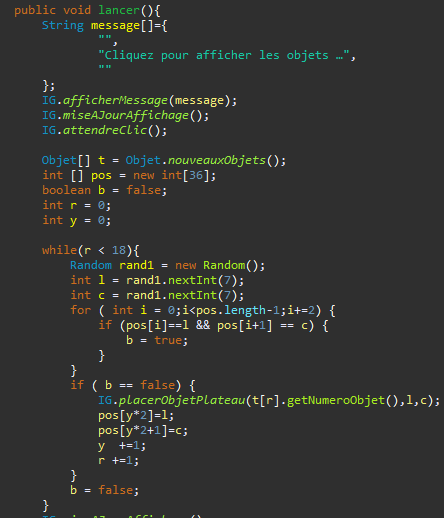
*ElementsPartie.java* :

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.

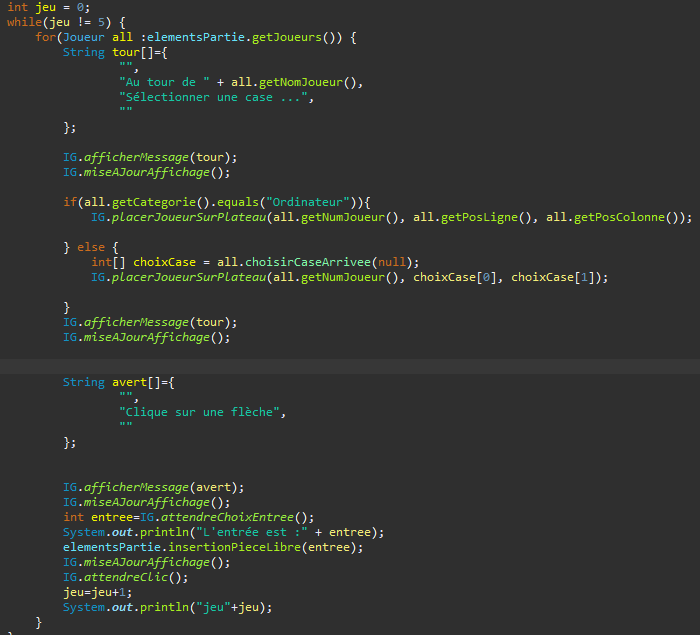
Le résultat de *TestElementPartie* :

*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.

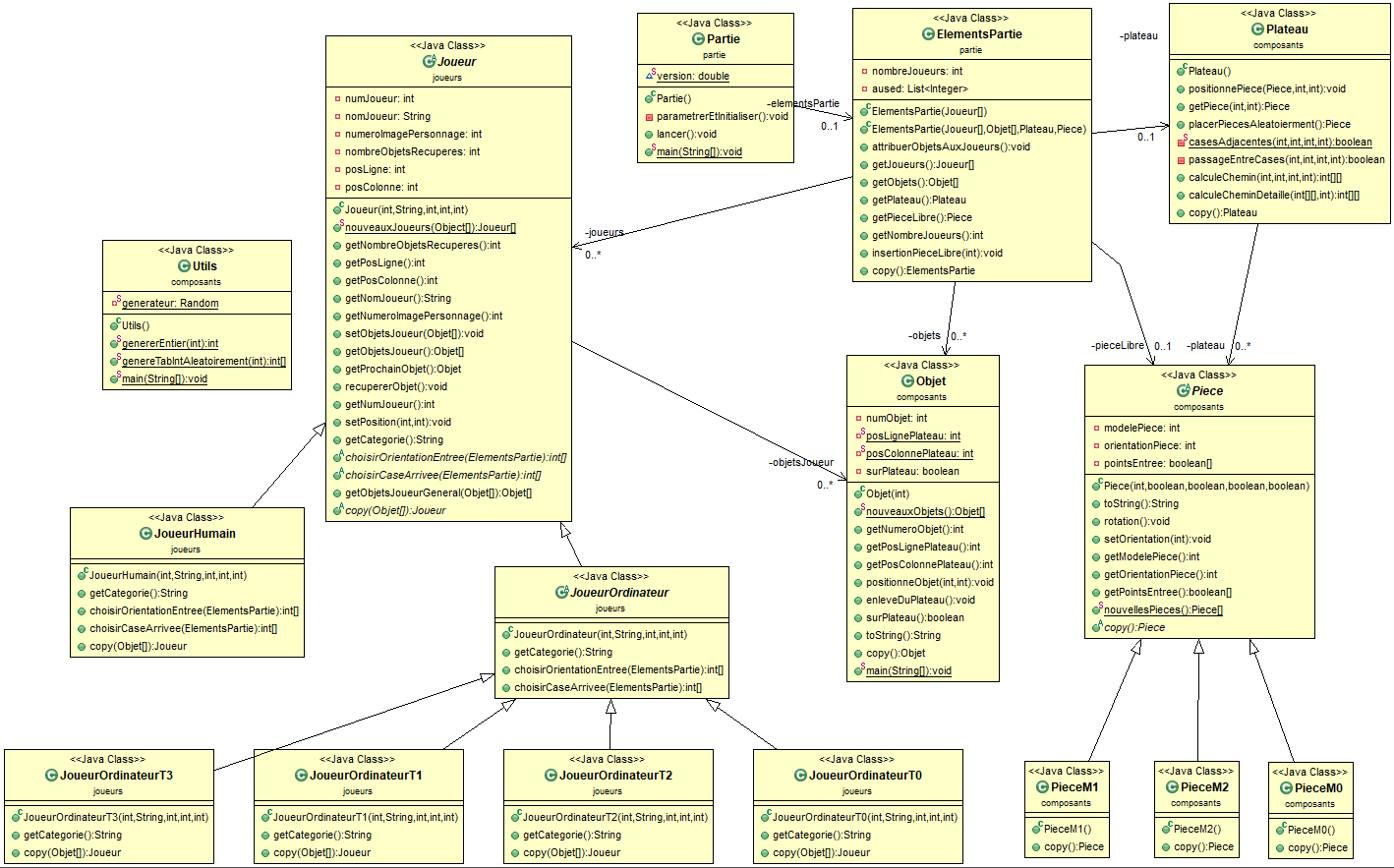
Avec cette partie de programme, nous allons d’abord lancer la partie, on va appeler lancer ().

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**

Lors du déroulement de notre projet les modules de POO et de COO ont été utilisés. Pour la POO, nous avons utilisé les cours, tp, td pour terminer ce projet. Ce module nous a permis de mieux connaître la syntaxe java. Nous avons créé des diagrammes de classes dans l’étape 4 et 5 grâce à nos de connaissance dans le module de COO.

## **Répartition du travail**

Notre groupe :

* Gronier Thibaut *chef de projet*
* Gaspard Corentin
* Caudron Jonathan
* Dubois Maxence

Pour le projet, nous avons créé un GitHub pour se répartir le travail. Nous avons créé un serveur discord pour pouvoir travailler ensemble et s’aider pour trouver les erreurs, faire les programmes.

# Conclusion :

## **3.1 Bilan global**

Sujet :

Ce sujet était intéressant, nous nous somme améliorer en java durant tout le projet.

Outils :

Pour ce projet, les outils utilisés sont :

* GitHub : pour se partager le travail et voir ce que chaque personne fait (avec les commit)
* Discord : pour s’écrire sur un salon textuel et discuter sur un salon vocal
* Zoom : pour les réunions avec notre tuteur (une réunion par semaine)

Difficultés :

Au début du projet, avant l’étape 3. Nous avons eu pas beaucoup de difficulté pour la réalisation de l’étape 1 et 2. Puis nous avons eu beaucoup de difficulté avec la fonction calculChemin() , notre problème sur cette fonction été que, nous avons compris la raisonnement mais le refaire en java été plus compliqué .

## **3.2 Bilan personnelle**

### 3.2.1 Bilan de Corentin Gaspard

Pour ce projet, j’ai appliqué ce que j’avais appris en java lors de ce second semestre. J’ai pu améliorer mes connaissances en java et j’ai pu apprendre à utiliser la librairie IG. Le problème que ce projet ma pause est la fonction calcule chemin qui nous a bloqués lors des étapes suivantes ce qui ne nous a pas permis de rendre le jeu fonctionnel.

### 3.2.2 Bilan de Maxence Dubois

Pour l'avis personnel : avec ce projet, j'ai pu acquérir de nouvelles connaissances en programmation, en l'occurrence Java. J'avais un petit peu de mal à comprendre comment Java fonctionnais vu que ce langage est bien différent de Python. J'ai donc réussi, grâce à ce projet, à comprendre un petit peu mieux ce langage qu'au début ou j'avais beaucoup de mal et aussi à l'aide de mes camarades qui pouvait m'expliquer aussi le fonctionnement même si cela a été assez éprouvant.

### 3.2.3 Bilan de Jonathan Caudron

Pour mon avis personnel, ce projet m'a permis de me relancer sur le Java et a renforcé mes connaissances sur ce langage sur certains points comme les tableaux, auparavant j'utilisais seulement les ArrayList inclus dans la librairie java.utils intégré à Java, l'utilisation de git et d'autres. Mon expérience s'est créée avec de l'utilisation de librairie et ça me rappelle bien mes débuts en Java. Pour moi ce projet est plutôt un mélange d'autres matières qu'on a eu au cours du semestre, plus précisément POO et COO, on reste très limité à de l'utilisation de librairie et c'est dommage qu’on n’ait pas pu créer nous-même la librairie graphique pour nous laisser plus de libertés au niveau des méthodes. Ayant discuté avec un étudiant de seconde année, c'est bien réduit par rapport à l'année dernière, avec l'utilisation de JavaFX par exemple et d'autres bases de Java qu’on n’a pas apprises au cours de ce semestre.

### 3.2.4 Bilan de Thibaut Gronier

Avant le projet, j’avais des lacunes en java mais grâce au projet je me suis amélioré en programmation java. J’ai appris à utiliser une librairie graphique. Grâce à mes camarades on a pu s’entraider en travaillant ensemble, c’était une bonne chose. Ce projet m’a permis de mieux utiliser Eclipse. J’ai eu des difficultés avec calculChemin() qui m’a bloqué sur cette étape. Maintenant je comprends beaucoup plus rapidement, comment faire une fonction.