14 JUIN 2021

RAPPORT DE PROJET

LABYRINTHE

BALENS – GOGUILLON – HAINAUT - HOVART
GROUPE FOXERS
IUT DE LENS

TABLE DES MATIERES

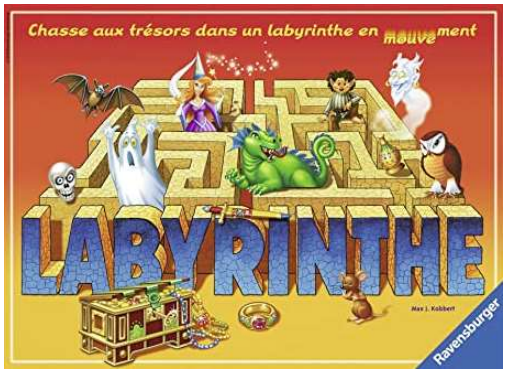
INTRODUCTION

Dans le cadre de notre première année de DUT informatique, notre second projet tutoré consistait en la réalisation d'un plateau fonctionnel du jeu *Labyrinthe*, à l'aide du langage **Java** et à partir d'une **librairie graphique** fournie par les professeurs.

LE PROJET

Les consignes et contraintes du projet

Le projet consistait donc en la réalisation du jeu de société « *Labyrinthe* » sous forme d'un plateau en deux dimensions, à partir de la librairie graphique *swing* fournie avec le sujet. L'ensemble du jeu devait être codé en **Java**.



Pour le développement du projet, nous devions suivre une série d'étapes avec des objectifs précis et des dates butoirs pour garder une avancée régulière et éviter les retards de réalisation. Il fallait également suivre une structure précise dans les fichiers et l'organisation des répertoires. Nous devions à l'étape 6 aboutir à un jeu fonctionnel alliant vrais joueurs et joueurs « ordinateur » de différents niveaux capables de jouer.

L'organisation du travail

Nous avons tout d'abord utilisé plusieurs outils de travail pour la réalisation de ce projet. Tout d'abord, au vu du contexte sanitaire, la communication était primordiale en dehors de l'IUT. Nous avons donc choisi **Discord** comme principal outil de communication, pour sa simplicité d'utilisation et les nombreuses fonctionnalités utiles pour le travail, notamment les possibilités de partage d'écran en appel ou encore l'envoi de code sans affecter sa mise en forme. Pour l'envoi de plus gros fichiers, nous avons utilisé le service **Mega**, facilitant l'envoi rapide des archives et répertoires.

Pour ce qui est du code, nous avons choisi de le réaliser avec **Eclipse IDE**. Il s'agit en effet d'un logiciel complet avec de nombreux outils facilitant le codage, la correction et l'exécution pour les projets en Java.

En ce qui concerne la rédaction des documents annexes, nous avons utilisé la suite **Office de Microsoft**, pour ses nombreuses fonctionnalités et la compatibilité des fichiers, ainsi que **Google G Suite** pour son travail collaboratif.

Grâce à l'ensemble des outils mis à notre disposition, nous avons rapidement pu mettre en place un système de répartition des tâches, et ce malgré quelques difficultés rencontrées au début du projet.

Le mécanisme de travail principal que nous avons mis en place était celui du travail commun. En effet nous nous sommes réparti les tâches, puis nous avons décidé d'avancer ensemble lors d'appels sur **Discord.**

Nous avancions simultanément sur différents morceaux de code en fonction des facilités de compréhension de chaque membre du groupe. Puis lorsque nous nous retrouvions bloqués, nous faisions appel à notre tuteur afin d'obtenir quelques éclaircissements nous permettant de continuer le développement du projet et ainsi pouvoir rendre à temps chaque étape. Cependant, il nous est parfois arrivé de déposer le travail avec un léger retard. Néanmoins celui-ci ne nous a jamais affecté sur les étapes suivantes.

Notre organisation fût rythmée par la présence de travaux externes provenant des autres matières, ce qui a eu pour effet de nous forcer à **travailler notre flexibilité**.

Une fois le mécanisme de répartition des tâches mis en place, **la fluidité** et **l'organisation** de l'écriture du code nous a paru de moins en moins complexe, créant ainsi une **cohésion de groupe** et **un esprit d'entre-aide** permettant le bon déroulement des étapes.

LE DEVELOPPEMENT

Le déroulement des étapes

Premièrement, après la découverte du projet, il nous a été demandé de prendre en main la **librairie graphique** mise à disposition par les professeurs afin de commencer le projet. C'est à ce moment que nous avons décidé de choisir **Eclipse** comme IDE.

Cette première étape nous a permis de choisir nos propres **composants graphiques** (sprites, chemins, objets). Nous avons aussi choisi d'intégrer au jeu une musique composée par nos soins.

Globalement, l'étape une du projet nous a permis de créer **l'identité graphique** de notre **Labyrinthe** mais aussi apprendre à utiliser la librairie graphique et les fonctions lui étant associées.

Ensuite, lors de la deuxième étape, il nous a été demandé de **gérer les pièces** du jeu. Pour cela nous avons dû modifier **quatre classes**, qui définissaient les pièces du tableau, divisées par formes de pièces (un coude, une ligne droite et une intersection). Une classe principale abstraite représente une pièce du jeu, et trois sous-classes concrètes représentent chacune un modèle de pièce.

Cette étape était assez intéressante car c'était la première étape durant laquelle on développait des éléments concrets du plateau de jeu.

La troisième étape était centrée sur les **objets** et le **plateau** de jeu. Nous devions réaliser **deux classes** : une classe permettant de gérer le plateau de jeu, c'est-à-dire un plateau de 49 pièces avec 7 lignes et 7 colonnes avec une pièce en dehors du plateau, et une deuxième classe pour gérer les objets de jeu que le joueur est censé collecter pour gagner la partie.

Cette étape a pour notre groupe été la première grande difficulté : nous avons eu beaucoup de mal avec la classe plateau, notamment avec des méthodes telles que « calculeChemin ».

L'étape 4 concernait l'implémentation d'une classe représentant les **joueurs**. Il était question de développer **une classe principale** avec pour objectif de gérer le joueur, son numéro, ses objets à récupérer, sa progression, ainsi que le nombre de joueurs et leur position.

Cette étape s'est mieux déroulée que la précédente au niveau du code pour notre groupe.

L'étape 5 consistait en la création d'une **partie fonctionnelle de jeu**. Pour cela, **deux classes** étaient à créer. La première classe gérait les éléments de la partie, comme attribuer les objets aux différents joueurs de manière aléatoire ou récupérer les joueurs et les objets de la partie. La deuxième classe gérait quant à elle la création de la partie en elle-même, c'est-à-dire paramétrer ses éléments et la lancer.

Nous n'avons pas eu de soucis avec la classe « ElementsPartie », mais avons eu plus de mal avec la classe « Partie ».

Enfin, l'étape 6 consistait en la création de trois « **intelligences artificielles** » capables de jouer contre un utilisateur humain. Chaque **IA** devait avoir un niveau de difficulté différent.

Par manque de temps et d'idées, nous avons privilégié la dernière semaine pour essayer de combler les quelques éléments manquants au code déjà réalisé, et n'avons donc pas pu optimiser les IA comme le demandait cette étape.

Tout au long de ces étapes, nous devions de plus réaliser des classes de test pour pouvoir essayer nos classes et méthodes au fur et à mesure que nous avancions.

Les méthodes du projet

➢ L'algorithme du tirage aléatoire de valeurs entières

La méthode *genereTabIntAleatoirement* permettait de générer un tableau d'entiers dont la longueur longTab était donnée en paramètre. Le tableau généré devait contenir chaque entier compris entre 0 et longTab-1, et la position de ces entiers dans le tableau devait être aléatoire. Tout d'abord, nous créons un tableau de longueur longTab. Nous avons initialisé un entier i à 0 et un booléen "dansTableau". Dans notre première boucle while, nous vérifions que "i est différent de la longueur du tableau". Si c'est le cas alors nous entrons dans la boucle pour générer un nombre aléatoire et nous mettons "dansTableau" à false, ce qui veut dire qu'aucun nombre ne sera généré. Dans la suite de cette méthode, on retrouve une boucle dans une boucle. La première, une boucle for "j jusqu'à i non inclus", et la deuxième, une boucle if qui vérifie si le nombre actuel est égal aux nombres précédents. Si c'est le cas, nous passons "dansTableau" à true, ce qui fait que nous n'incrémentons pas i. Un nouveau nombre est alors généré. Pour terminer, nous faisons une dernière boucle if pour vérifier le dernier cas, c'est-à-dire si "dansTableau" est égal à false. Lorsque c'est le cas, on incrémente i de 1. A la fin, on retourne le tableau.

➢ L'algorithme de recherche de chemins

Cette méthode est celle qui a causé à notre groupe le plus de problèmes. Nous avons eu beaucoup de difficultés lors des essais que nous avons fait sur son développement, et n'avons toujours pas à l'heure actuelle de méthode fonctionnelle.

Lors de ces différentes tentatives, deux raisonnements nous sont tout de même venus.

Le premier raisonnement consistait en la réalisation de deux étapes. La première étape (sous forme d'une première boucle) consistait, après avoir identifié un point de départ et d'arrivée sur le plateau, à partir simultanément de ces deux points et définir tous les déplacements possibles autour. Ainsi, au premier passage de la boucle, la méthode marquait dans un tableau représentant le plateau toutes les cases voisines des deux points, accessibles depuis ce point, de la valeur 1. Au tour suivant, la méthode marquait de la valeur 2 toutes les cases voisines des cases 1 accessibles depuis celles-ci, et ainsi de suite jusqu'à ce que plus aucune case ne soit accessible.

La deuxième partie consistait alors à parcourir ce tableau ainsi modifié, et déterminer par cette suite de cases voisines s'il était possible de déterminer un chemin entre les deux points. Si oui, on retournait un tableau de dimensions variables (en fonction de la longueur du chemin) contenant les différentes cases de passage et leurs coordonnées.

Nous avons abandonné cette méthode d'une part à cause du manque d'optimisation de l'algorithme et d'autre part par manque de temps pour aller au bout de son développement.

La deuxième méthode envisagée était d'appliquer l'algorithme de **Dijkstra** à notre plateau. Pour rappel, l'algorithme de Dijkstra résout le problème du plus court chemin entre deux points d'un graphe orienté. Il consiste en la création d'un sous graphe dans lequel on classe les différents sommets par ordre croissant de distance au sommet de départ, lorsqu'un passage est possible et qu'il n'y a pas d'obstacle, afin de trouver le chemin le plus court.

En pratique, nous devions traduire notre plateau en matrice d'incidence afin de créer un graphe correspondant. Nous avons néanmoins eu trop peu de temps pour réussir cette étape.

➢ L'algorithme de la boucle principale

Le déroulement d'une partie du jeu est possible grâce à l'implémentation de la classe Partie.java mais aussi ElementsPartie.java, permettant l'initialisation de tous les éléments nécessaires au bon déroulement d'une partie de Labyrinthe.

Après l'initialisation (saisie des paramètres, création du plateau et des joueurs) et le paramétrage de la partie, une partie de jeu est prête à être lancée.

Le lancement de la partie se fait grâce à la librairie graphique IG, la classe prend en compte un ensemble de paramètres ayant été définis par les classes précédemment implémentées comme Plateau.java, pour créer un nouveau plateau, prêt pour le jeu mais aussi Joueur.java afin de créer autant de joueur que demandé dans les paramètres.

Enfin, la méthode main contient la boucle principale du jeu, grâce à une boucle infinie, on lance une nouvelle partie du jeu.

L'utilisation des modules de POO et COO

Tout d'abord, le module de **POO** nous a été indispensable lors de la réalisation de ce projet. Nous avons acquis beaucoup de connaissances de base lors des cours magistraux, qui nous ont permis de comprendre le fonctionnement de la programmation en Java. Les travaux dirigés et les travaux pratiques nous ont quant à eux appris à utiliser les différents outils et structures nécessaires au développement du projet.

Le module de **COO** nous a permis d'avoir une vision plus globale du projet et de son fonctionnement, de comment les différentes classes, méthodes et autres éléments devaient communiquer entre eux et être reliés pour que toute la structure fonctionne. Par ailleurs, les connaissances pratiques acquises pendant les travaux dirigés et pratiques nous ont facilité la tâche lors de la réalisation des différents diagrammes demandés aux étapes 4 et 5 du projet.

LE PROJET FINALISE

Capture d'écran du plateau de jeu finalisé :

NOTRE RETOUR D'EXPERIENCE

Nos retours personnels

➢ Bilan personnel de Anne-Sophie

Pour ma part, ce projet m'a permis de me rendre compte à quel point la programmation orientée objet était importante. J'ai également rencontré des difficultés lorsqu'il s'agissait de coder des méthodes plus complexes que d'autres, mais le fait de travailler ensemble m'a permis de comprendre certaines choses, et a également permis de diminuer mes craintes et le stress. J'ai beaucoup aimé le fait de devoir coder un jeu. C'était un projet assez concret et prenant, et celui-ci m'a aidé à me faire une idée sur ce qui pourrait me plaire à l'avenir au niveau professionnel.

➢ Bilan personnel de Nicolas

Personnellement, ce projet m'a apporté beaucoup au niveau de l'autonomie. J'ai rapidement rencontré des difficultés avec le code et les étapes à suivre, et afin de ne pas perdre le rythme par rapport au groupe, il m'a fallu compter sur leur aide, et m'accorder avec eux pour nous organiser, pour que chacun, dont moi, puisse travailler sur les parties qu'il comprenait le mieux et saurait le mieux avancer. Cela m'a permis de travailler à la fois sur l'entraide en groupe comme sur ma capacité à travailler seul. Ce fut un projet conséquent mais enrichissant.

➢ Bilan personnel de Mathéo

Le projet du second semestre fût à mon sens, l'étape la plus importante de cette première année de DUT Informatique. La programmation du Labyrinthe, bien que demandant de nombreux efforts individuels mais aussi coopératifs, m'a réellement permis de prendre une décision sur mon avenir dans l'informatique. Après un départ compliqué de mon côté, le projet a su me faire comprendre l'importance de la rigueur et de la cohésion de groupe. Grâce à notre méthode de travail, j'ai pu faire de nombreux progrès en programmation orientée objet mais aussi combler quelques lacunes en algorithmique. En conclusion, le projet m'a été bénéfique en tout point et le fait de travailler méthodologiquement dans un groupe uni m'a permis de comprendre encore un peu mieux les attentes du domaine de l'informatique.

➢ Bilan personnel de Baptiste

Pour ma part, la programmation du jeu du Labyrinthe a été une expérience vraiment enrichissante car j'ai pu mettre en œuvre mes compétences acquises en POO et en programmation lors de cette première année de DUT informatique ainsi que mes capacités acquises en tant qu'autodidacte.

Ensuite, j'ai trouvé que ce projet était assez long à réaliser. En effet, concevoir certaines méthodes qui fonctionnent correctement avec nos compétences actuelles peut d'avérer difficile et coûteux en temps même si ce fût très formateur pour les années d'études et de travail à venir.

Pour finir, je suis satisfait du travail que nous avons effectué car nous avons bien su gérer notre projet malgré les difficultés techniques et temporelles, tout cela grâce à une bonne cohésion dans le groupe et une bonne répartition des tâches.

Notre retour global

Ce projet de semestre a pour nous été très formateur et nous a appris beaucoup de choses dans différents domaines. Nous avons tout de suite été motivés par le sujet, grâce à son **aspect concret** : devoir **coder un jeu de société** fonctionnel, réputé dans le domaine des jeux de société, est plus motivant qu'un exercice plus court et moins concret à notre échelle d'étudiants.

Ce projet nous a également permis de découvrir et de nous former à l'utilisation du langage **Java** : il s'agissait d'une nouveauté que nous avions découvert quelques semaines auparavant avec le **module de POO**, mais devoir s'engager dans un projet plus conséquent et de manière plus **autonome** nous a permis de mieux comprendre l'utilité et les usages possibles de ce langage, et nous a poussé à mieux nous y intéresser.

Nous avons été libres de découvrir et choisir nos outils de développement, et particulièrement **Eclipse**, un nouvel **IDE** que nous avons choisi d'utiliser tout au long de ce projet. En autonomie toujours nous avons pu découvrir les différentes fonctions et possibilités qu'offraient cet outil et qui nous ont facilité la tâche au fur et à mesure du code.

Le contexte actuel a eu une influence sur notre expérience : avec la crise sanitaire et l'adaptation des cours au respect des mesures et gestes barrières, notre temps à l'IUT et donc en présence de notre groupe était très réduit, et nous avons dû adapter nos séances de travail et nos **outils de communication** ; bien que contraignante, cette situation a renforcé notre autonomie et notre capacité à travailler à distance.

Nous avons bien sûr rencontré des **difficultés** durant le projet, que ça soit au niveau des problèmes de développement, du travail à distance, de manque de temps ou de la pression accumulée avec le reste des cours à gérer en parallèle, mais globalement, nous pouvons dire que ce projet nous a beaucoup intéressé et nous a apporté de nombreuses **connaissances** et **compétences**, en informatique comme en gestion de travail, en autonomie comme en équipe.

LEXIQUE

Classe : fichier exécutable dans le langage Java.

COO : Conception Orientée Objet.

Algorithme de Dijsktra : algorithme en théorie des graphes permettant d'identifier le plus court chemin entre deux points donnés. Il a été inventé par Edsger Dijkstra.

Discord : logiciel de communication basé sur la création de serveurs, salons et catégories de salon, développé par Discord Inc.

Eclipse : IDE spécialisé dans le développement de codes et programmes en Java, mais adaptable à d'autres langages, développé par la fondation Eclipse.

IA : Intelligence Artificielle

IDE : Integrated Development Environment (Environnement de développement Intégré) : logiciel de codage et développement en informatique.

Identité graphique : ici, l'identité graphique du projet est l'ensemble d'éléments visuels produits par l'interface graphique permettant d'identifier, comprendre et jouer au jeu du labyrinthe.

Java : langage de programmation orientée objet développé par James Gosling et Patrick Naughton

Librairie graphique : bibliothèque logicielle contenant les fonctions graphiques d'un programme afin de lui créer une interface.

Mega : service de stockage en ligne développé par Mega Limited.

POO : Programmation Orientée Objet.

Sprites : Eléments graphiques pouvant se déplacer sur l'écran, et qui peut être animé.