



Projet Informatique - Rapport

Taïder MOUSSA
Nioua ILÉNA
Afazouz OMAR
Mai 2023

Professeur titulaire : Hadrien
MÉLOT

Remerciements

Nous tenons à exprimer notre profonde gratitude et notre profond respect à notre professeur titulaire de ce projet informatique M. Hadrien Mélot, son orientation, sa pédagogie, ses conseils et son encadrement ont permis la réalisation et la concrétisation de ce projet.

Nous désirons également faire part de l'expression de notre reconnaissance et de notre considération à M. Bruno Quoitin, professeur titulaire du cours de programmation algorithmique II.

Nous adressons également nos sincères remerciements à nos assistants Romain Absil, Sébastien Bonté, Jeremy Dubrulle, Pierre Hauweele, qui nous ont accompagnés, formés, motivés et qui ont participé au bon déroulement de cette année scolaire ainsi qu'au bon déroulement de ce projet.

Nous rendons hommage également à l'université de Mons qui nous a admis dans ses classes, et qui nous a permis d'avoir accès à cette éducation, au savoir et à la bienséance jusqu'ici.

Table des matières

1	Introduction	3
2	Répartition des tâches	3
3	Thématique du projet	3
4	Points forts du projet	3
5	Algorithmie	4
5.1	La classe PuzzleJeu	4
5.2	La méthode createStage	4
5.3	La méthode start	4
5.4	La classe Controle	4
5.5	La méthode makeForm	4
5.6	La classe Form	4
5.6.1	La méthode Form	4
5.6.2	La méthode addDragEvent	4
5.6.3	La méthode isFull	5
5.6.4	La méthode snapToGrid	5
6	Points faibles du projet	5
7	Apports positifs	5
8	Apports négatifs	5
9	Guide d'utilisation	5
9.1	Prérequis	5
10	Run et compilation	6
10.1	Installation	6
11	Conclusion	6
12	Références	6

1 Introduction

Dans le cadre du cours de programmation algorithmique II, nous avons appris à utiliser le langage de programmation de haut niveau Java, développer par Sun Microsystems, par la suite acquis par Oracle. Subséquemment, il nous a été demandé à travers le cours de projet informatique, d'élaborer un projet logiciel tel qu'un jeu inspiré de « Cats Organized Neatly », un jeu disponible sur Steam qui est de type puzzle, dans lequel l'utilisateur doit disposer des pièces de configuration différentes afin de combler l'aire de jeu. Les objectifs de ce projet logiciel étaient de s'initier au développement de ce type d'activité conséquente, d'appliquer les notions d'algorithmiques abordées aux divers cours de programmation algorithmique I et programmation algorithmique II, et enfin d'apprendre à collaborer en équipe.

Diverses séances obligatoires et facultatives ont été mise en place afin de mener à bien le projet.

2 Répartition des tâches

La répartition des tâches a été tel que Iléna Nioua s'est occupée de l'interface graphique, du graphisme des diverses icônes et boutons du jeu, ainsi que la base de données qui permet la sauvegarde et le générateur automatique de niveaux aléatoires. Moussa Taider s'est occupé également de l'interface graphique concernant l'implantation de fichier json, de la javadoc, de la compilation du projet, et de construire l'application et structurer le projet avec Gradle. Moussa Taider et Iléna Nioua ont collaboré ensemble quant aux tests unitaires. Omar Afazouz s'est occupé de la recherche des différents outils pour mener à bien ce projet, la rédaction d'une partie du rapport, de compresser et d'archiver le projet. Concernant le reste, nous avons tous contribué aux mêmes niveaux et à notre échelle afin de mener à bien ce projet.

3 Thématique du projet

Nous avons décidé de laisser libre arbitre à Iléna concernant le thème à choisir de ce jeu. Cela nous semblait important qu'en tant que seule fille du groupe, elle puisse s'épanouir et ajouter sa touche de fantaisie dans ce projet logiciel, et surtout qu'elle puisse faire preuve d'originalité et de créativité pour le bien de ce projet. Celle-ci a choisi le thème de la mode, et plus précisément le domaine du sac à main. Elle a trouvé important de combiner ses deux passions, l'informatique et la mode, pour plus de fantaisie et de singularité, étant donné que statistiquement, dans notre option, le nombre de garçons soit largement plus élevé que les filles, il était important pour elle qu'elle puisse faire-valoir sa place dans cette option, et ajouter une touche d'originalité, de féminité à ce projet, en espérant avoir marqué les esprits par le biais de celui-ci. Nous avons donc caricaturé le jeu « Cats Organized Neatly » pour le transformer en « Bags Organized Neatly ». Les sacs dessinés sont inspirés des iconiques cabas Kelly et Birkin de la maison Hermès.

4 Points forts du projet

Notre projet logiciel comporte un petit menu qui permet de rentrer dans le jeu ou de quitter l'application et une interface graphique pour chaque niveau. 19 niveaux sont mis en place dans ce projet, partant du niveau 11 et se terminant au niveau 30. Une méthode de sauvegarde ainsi qu'un générateur automatique de niveaux aléatoires valides y sont également implémentés. Tous les skins présents dans notre application font partie d'un thème qui est le thème du sac à main, précisé plus haut. Tous les skins, icônes, boutons et fond d'écran ont été créés et réalisés à la main à l'aide d'outils informatiques et de techniques de graphisme tel que Adobe Photoshop, Picsart, Pixlr, Sketchbook et Canva. Des animations ont également été ajoutées grâce à l'utilisation du framework graphique JavaFX, tel que la rotation des icônes pour changer leurs configurations en effectuant un double click, le retour à la case départ en cas de positionnement invalide, ou le grossissement des pièces lorsqu'elles sont sélectionnées. Plusieurs méthodes ont été implantées pour un maximum d'efficacité.

5 Algorithmie

5.1 La classe PuzzleJeu

La classe PuzzleJeu qui étend la classe application et qui implémente la méthode start(Stage stage). Le programme initialise un niveau à partir d'un fichier JSON et dessine la grille de jeu. Les formes du puzzle sont créées en utilisant la classe Form, qui contient un tableau des rectangles.

Les formes sont affichées sous forme de rectangles. Le joueur peut déplacer les formes en cliquant et en faisant glisser la souris. Le bouton « suivant » permet de passer au niveau suivant.

5.2 La méthode createStage

La méthode createStage (String url, Level level, Stage stage) est appelée pour créer un nouveau niveau. Elle lit le fichier JSON, crée une grille de jeu et ajoute des rectangles bloqués aux positions spécifiées. La grille de jeu est ensuite affichée à l'écran, le programme utilise des effets de lumières et d'ombres pour donner un effet 3D aux rectangles. Il utilise également des transitions pour animer les mouvements de déplacement des formes.

5.3 La méthode start

La méthode start(Stage stage) initialise une nouvelle instance de la classe Level et appelle la méthode createStage() pour créer le premier niveau. La méthode crée également un bouton Suivant qui permet de passer au niveau suivant. Lorsque le bouton est cliqué, la méthode crée un nouveau niveau en appelant la méthode createStage() avec les paramètres mis à jour on se basant sur les niveaux placés dans le dossier ressources.

5.4 La classe Controle

Cette classe initialise les puzzles et leurs positions dans l'espace de jeu. Il définit une méthode appelée makeForm qui crée un puzzle en fonction de son nombre de blocs et de son indice.

5.5 La méthode makeForm

La méthode makeForm utilise la classe Rectangle de JavaFX pour créer les blocs du puzzle. Elle détermine les positions de chaque bloc en fonction du nombre de blocs et de l'indice du puzzle.

Chaque puzzle est nommé en fonction de son nombre de blocs et de sa forme en général, la méthode makeForm initialise des puzzles de différentes formes et tailles. Cette méthode utilise de conditionnelles if-else pour déterminer les positions de chaque bloc de puzzle en fonction du nombre de blocs et de l'indice.

5.6 La classe Form

5.6.1 La méthode Form

La méthode Form (Rectangle[] listRectangles, String name) c'est le constructeur de la classe. Il permet de créer une nouvelle forme de puzzle à partir d'un tableau de rectangles et d'un nom.

La méthode charge également une image selon le nom de la forme.

5.6.2 La méthode addDragEvent

La méthode addDragEvent(Form form, Pane scene, GridPane gridPane) permet de déplacer les rectangles de la forme de puzzle lorsqu'elle est traînée avec la souris. La méthode définit les événements de souris appropriés pour chaque rectangle, tels que "OnMousePressed"

"OnMouseDragged" et "OnMouseReleased". Il ajuste également la position des rectangles pour qu'ils s'alignent sur une grille 50×50.

5.6.3 La méthode isFull

La méthode isFull() vérifie si la grille de puzzle est pleine, c'est-à-dire si tous les emplacements sont remplis par une forme de puzzle.

5.6.4 La méthode snapToGrid

SnapToGrid double x, double y, Form form) cette méthode ajuste la position des rectangles pour qu'ils s'alignent sur une grille 50x50 lorsqu'ils sont relâchés par la souris.

6 Points faibles du projet

Concernant les manquants du projet, nous y trouvons en première position un bug aux niveaux des pièces du puzzle qui n'a pas su être résolu. En effet, lorsque vous cliquez plusieurs fois et de façon rapide sur une pièce du puzzle dans le but de la faire pivoter, et donc l'insérer dans le puzzle cela peut la faire beuguer et détruire l'alignement des icônes. Nous n'avons pas trouver de solution ni d'où provenait le problème algorithmiquement parlant.

Un problème par rapport à la sauvegarde a été également détecté en effet, nous avons constaté qu'il se peut que lorsque nous sauvegardons un niveau résolu, dans le but de le recharger à la demande de l'utilisateur, celui-ci ne se télécharge pas directement dans un format de fichier comme voulu, nous avons élaboré un système de sauvegarde qui permet à l'utilisateur de fermer l'onglet du jeu, sans pour autant que sa partie soit annulée. En effet, malgré qu'il ai quitté l'onglet, il aura la possibilité de recharger sa partie à l'endroit exacte où il s'est arrêté par le biais d'une sauvegarde automatique du niveau. En espérant que notre système de sauvegarde puisse plaire à nos utilisateurs, et à vous qui nous lisez.

Nous avons également remarqué qu'après résolution d'un niveau et après le succès de celle-ci, l'animation «Level Completed» peut prendre du temps à apparaître et peut beuguer. Nous n'avons pas trouvé la cause de cette problématique.

Subséquentement, la vitesse d'exécution n'est pas des plus rapides ni des plus efficaces, néanmoins nous avons tenté notre maximum.

7 Apports positifs

Suite à la réalisation de ce programme, nous avons pu observer une amélioration de nos compétences dans le domaine de la programmation. Certes il a été difficile de l'accomplir, surtout de le rendre à l'image de nos espérances, et de respecter toutes les conditions de recevabilité, mais il nous a permis de nous auto-évaluer et également d'être plus indépendant quant à la conception de ce type de projet. Il a également été satisfaisant et gratifiant de voir l'évolution de notre projet et de faire preuve de fantaisie quant à la présentation de celui-ci, les interfaces graphiques ont été sympathiques à réaliser !

8 Apports négatifs

Évidemment ce projet remplis de rebondissements, riche d'expériences, visant à améliorer notre intégration dans le domaine de l'informatique nous a également fournis beaucoup de stress, par peur de l'échec, peur de ne pas fournir un travail suffisant et à l'image de nos attentes.

9 Guide d'utilisation

9.1 Prérequis

Dans cette application nous utilisons la version 17 du langage de haut niveau Java.

Avec comme outils de construction et de compilation Gradle 7.3, et comme bibliothèque d'interface et framework JavaFX 17.

10 Run et compilation

Afin de run et de compiler l'application, il faut ajouter Add VM Options pour définir la ligne suivante :

```
"-module-path"C : -sdk-17.0.7"-add-modules"  
"javafx.controls,javafx.fxml"
```

10.1 Installation

- Ouvrez le projet dans votre IDE préféré (IntelliJ IDEA, Eclipse sts..)
- Exécutez la commande Gradle build pour construire l'application.
- Exécutez l'application en lançant la classe principale PuzzleJeu. Vous pouvez également exécuter cette commande via l'interface graphique de votre IDE ou via la ligne de commande.

Notre projet logiciel dispose de plusieurs commandes Gradle pré-définies : gradle clean : cette commande supprime les fichiers de build générés précédemment.

- gradle build : cette commande compile les sources du projet et crée un fichier jar exécutable.
- gradle run : cette commande exécute l'application en lançant la classe principale PuzzleJeu.
- gradle test : cette commande exécute les tests unitaires du projet.

11 Conclusion

En conclusion, nous avons été ravis de pouvoir joindre les classes du département informatique et d'avoir pu assister à tout ce cursus, contenant des cours plus enrichissant les uns que les autres. Les cours d'informatique ont été un réel plaisir, et nous avons été contents de pouvoir réaliser ce projet pour donner vie à nos connaissances. Même si il a été extrêmement stressant, difficile, et long, et qu'il a fallut faire preuve d'endurance et de patience, nous sommes fiers d'avoir pu arriver jusqu'ici et d'offrir ce projet. Nous espérons pouvoir nous améliorer d'avantage, et pouvoir continuer à enrichir notre savoir. Malgré ses imperfections, ce projet reste notre premier jet concernant nos réalisations personnelles dans le domaine de la programmation, et nous souhaitons pouvoir partir de cette base pour évoluer vers d'autres horizons, plus lointains.

12 Références

icône sac