

Projet Génie Logiciel : Lazer Challenge

Rapport d'implémentation

Maazouz Mehdi, Lecocq Alexis

March 31, 2017

Annee Academique 2016-2017
Groupe 6

Contents

1	Introduction	2
2	Préparatifs	2
2.1	Changement de groupe	2
3	Conception	2
3.1	Choix d'Implémentation	2
3.2	Difficultés	3
3.3	Concessions	4
4	Eventuelles Améliorations	4
5	Outils Utilisés	4
6	Visuel Du Jeu	5
7	Remarques	5
8	Conclusion	5
9	Remerciements	6

1 Introduction

Nous débutons notre rapport d'implémentation par un léger rappel du projet qui nous a, initialement, été demandé. L'énoncé mis à notre disposition en début d'année parlait de la réalisation du jeu nommé "Lazer Challenge" en Java en nous aidant de la bibliothèque LibGDX. De plus, une extension était demandée individuellement à chaque étudiant.

Pour pouvoir réaliser ce projet, plusieurs étapes ont dûes être nécessaires. La première consistait en la planification de ce dernier, elle comprenait la création d'un graphe Gantt, PERT, ainsi qu'une gestion des risques, une répartition du travail, et de pouvoir déterminer les ressources qui allaient nous être nécessaires dans l'optique de nous préparer au mieux à d'éventuels obstacles, empêchements qui pourrait survenir. Ce qui nous aurait ralenti dans la conception de notre jeu.

La deuxième, quand à elle, nous a amenée à créer une maquette permettant aux professeurs et assistants d'avoir un aperçu de la forme que prendrait notre jeu, c'est à dire de la partie graphique. Mais nous avons dû également travailler sur plusieurs diagrammes UML, notamment un diagramme de classe qui a permis de montrer le fond de notre jeu, autrement dit les classes qui formeraient notre projet.

Nous voici désormais à l'étape de l'implémentation, où il nous était demandé d'implémenter le jeu LazerChallenge ainsi que nos deux extensions "Diagonal Directions" et "Saving + Multiple Users + Social Network". Mais également un rapport d'implémentation, des tests unitaires, une vidéo explicative et une javadoc complète.

Nous pouvons désormais vous décrire notre travail dans son ensemble, les difficultés et problèmes survenus, nos choix d'implémentations, quelques concessions faites également mais aussi les améliorations que nous pourrions apporter dans le futur.

2 Préparatifs

2.1 Changement de groupe

Dès le début de la phase d'implémentation, suite à plusieurs abandons dans les autres groupes, le notre a dû se voir séparé de l'un de ses membres. En conséquence, nous avons dû immédiatement revoir notre diagramme de classe et la maquette graphique afin de pouvoir nous baser directement sur quelque chose de concret. Le nouveau diagramme de classe se trouve dans l'archive .jar et est donc à votre disposition (core/assets/images/ClassDiagram.png).

3 Conception

3.1 Choix d'Implémentation

Nous allons ici, essayer d'argumenter de la meilleure manière nos choix d'implémentations que nous avons pris lors de cette phase. Un des choix les plus importants concerne les propriétés disponibles sur les tuiles. Au départ, nous avons respecté les consignes qui nous avaient été données, mais après quelques

soucis d'implémentation (pas adapté à notre Design, quelques bugs également difficiles à régler), nous avons opté pour un mécanisme de restriction , autrement dit, chaque tuile possédera, ou non, une propriété appelé restriction. Cette dernière aura comme valeur le nom d'un bloc. Par exemple, si une tuile possède une restriction ayant comme valeur source, seul une source pourra être déposée sur cette tuile. Si elle ne possède aucune restriction, tout type de blocs pourront y être déposés.

Concernant la bibliothèque LibGDX, n'ayant que peu de connaissance au début du projet, nous étions partis sur une interface graphique composée de sprites qui faisaient office de boutons. C'était une solution facile à mettre en oeuvre , qui nous avait permis de pouvoir passer à d'autres étapes dans notre projet.

Cependant, le fait de créer des images de boutons devenaient redondants et après réflexion, nous nous sommes dit que si nous voulions, dans un futur hypothétique, traduire notre jeu. Il nous aurait fallu refaire tous les sprites. La décision de revenir en arrière et de recréer entièrement nos menus du jeu a été prise en exploitant au mieux les propriétés de LibGDX.

Par rapport à notre maquette graphique remis durant la phase de modélisation, nous avons essayé de la respecter au mieux, seul quelques changement esthétiques ont pu faire leur apparition.

D'un autre côté, nous avons fait le choix de définir nos blocs dans des fichiers XML, nous pensons que cela rend notre projet plus modulable. En effet, nous n'avons pas besoin de recompiler notre projet si nous voulons ajouter un nouveau type de bloc. Nous avons procédé de la même manière pour les orientations en les définissant également dans des fichiers XML.

De plus, nous avons pris la peine de fournir des fichier .dtd afin de vérifier si nos fichiers XML étaient corrects.

Par ailleurs, l'énoncé parlait d'un mode "arcade/practice" ainsi que d'un mode "continuous laser beam/one-time-only laser beam". Concernant les deux premiers modes , nous n'avons à noter aucun changement par rapport à l'énoncé initial. Cependant, nous n'avons autorisé que le mode "one-time-only laser beam" en mode arcade. Le mode "continuous laser beam", est quand à lui, exclusif au mode "practice"(mode "entraînement" dans notre interface graphique). Nous en parlions déjà dans notre rapport de modélisation.

Un mode Pause nous était aussi demandé. Cependant, nous n'avons pas souhaité l'implémenter. Etant donné qu'en mode "entraînement", aucune limite de temps n'est fixée , nous ne voyons donc pas l'intérêt d'un tel mode. En mode "arcade", une limite de temps est fixée, de ce fait, un mode pause entrainerait également la pause du timer ... Ce que nous considérons comme étant de la triche.

3.2 Difficultés

Une des difficultés principale concernant le projet est la capacité à nous adapter à la bibliothèque LibGDX, ce qui est normal étant donné , qu'aucun des membres n'avaient utilisés cette bibliothèque auparavant , cependant nous regrettons un manque de documentation accessible sur internet, que ce soit sur le site contenant la documentation officielle de LibGDX ou bien sur des sites tierces

Une autre difficulté vient du fait que notre objet Orthogonal TileMap Renderer qui permettait d'afficher correctement les maps n'est pas compatible avec l'API sceneUI.2D utilisée pour l'interface graphique. Une solution a été de créer une

table contenant des boutons qui ont permis d’afficher plusieurs textures. Concernant l’extension “Diagonal Directions”, une difficulté nous est apparue en fin de projet. En effet, nous avons utilisé des énumérations pour les orientations. De ce fait, afin de pouvoir implémenter les orientations diagonales pour notre extension, il nous aurait fallu hériter notre énumération, ce qui n’est pas permis par Java.

3.3 Concessions

Nous parlions, dans la section **Difficultés**, de l’extension “Diagonal Directions” et des difficultés que nous avons rencontré avec notre choix des énumérations concernant les orientations. Malheureusement, cela a pris un certain temps à régler, de ce fait, l’intensité n’a pas pu être implémentée correctement.

4 Eventuelles Améliorations

Malgré le fait que l’on avait un énoncé contenant les consignes pour la réalisation de ce projet, nous avons quand même décidé de penser à d’éventuelles améliorations que l’on pourrait apporter à notre jeu après sa remise.

Tout d’abord, on sait que LibGDX possède quelques classes liés à l’implémentation de sons, de musiques qui apporteraient un plus non négligeable à notre jeu, bien qu’esthétiques.

On pourrait également remanié notre interface graphique, ainsi, au lieu de devoir cliquer sur un bloc, de cliquer ensuite sur un bouton déplacement et choisir la case où l’on souhaiterait déposer notre bloc; on implémenterait un système de “drag and drop” où il suffirait de prendre le bloc, et de le glisser sur la case voulue sur la map. On considère cette façon de faire plus intuitive et elle pourrait faire gagner du temps au joueur en mode arcade.

Pour le score, notre jeu le calcule seulement en fonction du temps qui s’écoule, on pourrait aussi le calculer en fonction du nombre de blocs allant de l’inventaire à la zone de jeu. Ainsi, plus nous mettrions de bloc en jeu, plus le score aura diminué.

Nous avons aussi pensé à inclure un “Redo”, vu que nous avons pris la peine d’implémenter un “Undo” qui permet de garder un historique des coups en mémoire.

De plus, nous avons intégré une sauvegarde pour les niveaux en mode entraînement, cependant seul la position des blocs sur la carte a été sauvegardé. Autrement dit, à chaque recharge du niveau, les orientations des blocs, que le joueur avait placé au préalable sur la carte, reprendront leur orientations par défaut.

Nous pourrions également ajouter un avertissement qui prévient le joueur lorsqu’il réussi un niveau en mode entraînement.

5 Outils Utilisés

Pour la réalisation de ce projet, nous avons utilisé plusieurs outils mis à notre disposition, certains étaient demandés par les professeurs et assistants, tel que Git qui nous a grandement aidé dans la gestion de notre projet, Gradle qui nous a permis de compiler le programme facilement et d’effectuer les tests unitaires

avec JUnit.

La création des maps s’est faites avec l’outil Tiled, comme également, demandé par les professeurs et assistants.

Par ailleurs, nous avons utilisé certains outils qui n’étaient pas demandés explicitement, tel que Gimp pour la création des sprites de tous les blocs et lasers, la capture de la vidéo explicative s’est faite avec le logiciel de capture Kazam et le son avec le logiciel Audacity. Le montage, quand à lui, a été réalisé grâce au programme Pitivi.

Qui plus est, ce rapport a été rédigé avec le langage LaTeX (avec comme complément, l’IDE Kile).

6 Visuel Du Jeu

Vous pouvez visualiser un aperçu commenté de notre jeu en cliquant sur l’URL ci-dessous.

<https://youtu.be/0XHiiyTEw3s>

7 Remarques

Nous parlions d’une première version de l’interface graphique dans la section **Choix d’implémentation**, Maazouz Mehdi s’était occupé majoritairement de cette version en ayant fait une vingtaine de commits, cependant, aucun de ses commits n’apparaît sur GitHub (dans des sections de statistiques, de graphiques, ...) . Il faut donc intégralement remonter la liste contenant tous les commits pour les retrouver .

Alexis Lecocq devait initialement, s’occuper de l’extension “Saving + Multiple users + Social network”. En discutant avec l’assistant M.Devillez, ce dernier lui a conseillé de se tourner vers le réseau social Twitter (à noter que ce choix est différent de celui qui avait été préconisé dans le diagramme de classe)

. Suite à quelques problèmes, il est venu trouver les assistants et professeurs afin d’obtenir un hébergement, malheureusement , ces derniers n’ont pas répondu a sa demande. .

Nous avons donc pris la décision de nous occuper, ensemble de l’extension “Diagonal Directions”

8 Conclusion

En conclusion, ce projet fût long , laborieux, mais nous pensons avoir implémenté tout ce qui nous était demandé.

De plus, il nous a permis de nous apprivoiser avec les outils cités plus haut. Ils n’ont pas forcément été facile à maîtriser mais une fois une certaine maîtrise acquise, ces derniers se sont révélés très utile pour la conception de notre projet et sans doute, d’autres projets futurs.

Mais ce projet a surtout été enrichissant au niveau des discussions, des échanges des idées entre les membres du groupe.

9 Remerciements

Nous tenions à remercier le professeur Tom Mens ainsi que les assistants , Gauvain Devillez et Jeremy Dubrulle qui ont pu nous aider à concevoir les idées qui ont menées à la réalisation de ce projet.