

Projet Génie Logiciel : Lazer Challenge

Rapport d'implémentation

Maazouz Mehdi, Lecocq Alexis

March 30, 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	Visuel Du Jeu	4
6	Notes Annexes	5
7	Conclusion	5
8	Remerciements	5

1 Introduction

Nous débutons notre rapport d'implémentation par un léger rappel du projet qui nous a été demandé initialement. 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 dû être nécessaires. La première consistait en la planification de notre projet, cette étape 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, empêchement qui pourrait survenir et nous ralentir dans la conception de notre jeu. Ensuite est venue l'étape de la modélisation, où nous avons dû créer une maquette permettant aux professeurs et assistants d'avoir une vision de la forme que prendra notre jeu, c'est à dire la partie graphique. Mais également la conception de plusieurs diagrammes UML, notamment un diagramme de classe qui a permis de montrer le fond de notre jeu, autrement dit les classes qui formeront notre projet. TODO Nous voici désormais à l'étape de l'implémentation où nous allons maintenant vous expliquer notre travail dans son ensemble, les difficultés et problèmes survenus, nos choix d'implémentations, quelques concessions faites également ... TODO suite.

2 Préparatifs

2.1 Changement de groupe

Dès le début de la phase d'implémentation, suite à plusieurs abandons dans les autres groupes, un membre de notre groupe a été annexé dans un autre groupe. 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 possèdera, ou non, une propriété appelée restriction. Cette dernière aura comme valeur le nom d'un bloc, 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 de 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.

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 et que le mode "continuous laser beam" en mode "practice" (mode "entraînement" dans notre interface graphique). Nous en parlons déjà dans notre rapport de modélisation.

De plus, un mode Pause nous était demandé. Nous n'avons pas souhaité l'implémenter car étant 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 entraînerait 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 LibGDX officielle ou bien sur des sites tierces

Une autre difficulté vient du fait qu'une méthode `render` venant de `Tiled` et qui permet 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 qu'il n'est pas permis par Java.

3.3 Concessions

Suite à un manque de temps , nous avons dû revoir nos objectifs à la baisse et avons privilégié ce qui nous semble être le plus important. Ces objectifs ont été de pouvoir rendre un projet jouable, bien documenté, composés de quelques tests unitaires sur le côté , mais également un rapport fourni et une vidéo expliquant les fonctionnalités de notre jeu. Les extensions ne sont donc pas présente dans notre jeu final ... ou du moins , elles ne sont pas finalisées. En effet , pour l’extension “Diagonal directions” de Maazouz Mehdi, les maps avaient été créées, certains sprites spécifiques à ce mode de jeu également et vous pourrez noter la présence de fichiers Xml spécifiques eux aussi, à cette extension. Bien entendu, cela n’a pas été suffisant qualifier cette dernière de finie. Alexis Lecocq devait initialement, s’occuper de l’extension “Saving + Multiple users + Social network”. Après réflexion, son choix s’est tourné 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 , il semblerait que ça n’est pas été le cas .

Même si cela n’était pas obligatoire, nous pouvons également noter l’absence de fichiers XSD pour vérifier la structure de nos fichiers XML, cependant, nous aimerions préciser que nous suivions/suivons le cours de Base de Données 2 en Bac3, nous avons donc créé quelques DTD qui nous avaient permis vérifier nos XML. Mais en utilisant la classe XmlReader de LibGDX , des soucis sont apparus dû à la présence de ces DTD (souci de compatibilité ?) qui nous ont obligé à les retirer.

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, cliquer 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.

5 Visuel Du Jeu

Vous pouvez visualiser un aperçu commenté de notre jeu en cliquant sur cette URL

6 Notes Annexes

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 .

7 Conclusion

En conclusion, ce projet fût long , laborieux mais il nous a permis de nous apprivoiser avec certains outils, certaines notions pas forcément déjà vus tels que le format XML, la création et la gestion des maps avec l'outil Tiled, la bibliothèque LibGDX, Gradle, Git qui, une fois “maîtrisé” a pu grandement nous aider à gérer ce projet, et dans une moindre mesure , l'outil Gimp pour les sprites ainsi que TODOOBS pour l'enregistrement de notre vidéo explicative. Nous avons également pu nous refamiliariser avec la JavaDoc, JUnit vus, quand à eux, en Bac1. Notons aussi le langage LaTeX, utilisé pour la rédaction de ce rapport.

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

8 Remerciements