

Projet Génie Logiciel : LazerChallenge

Rapport d'implémentation

Maazouz Mehdi, Lecocq Alexis

March 29, 2017

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	3
4	Notre jeu	3
5	Notes Annexes	3

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.

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 diagramme de classe se trouve dans le .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 que nous avons pris lors de la phase d'implémentation. Un des choix les plus importants concerne les propriétés disponibles sur les tuiles. Au départ, nous avons respectés 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 appelée restriction. Cette dernière aura comme valeur le nom d'un bloqueur, exemple, si une tuile possède restriction ayant comme valeur source, seul une source pourra être déposée sur cette tuile. Si elle ne possède aucune restriction, tout type de bloqueurs pourront y

être déposés.

Concernant la bibliothèque LibGDX, n'ayant que peu de connaissance au début, 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.

3.2 Difficultés

Une des difficultés principale concernant le projet est l'appropriation de 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 ...

Difficulté car API pas compatible avec Tiled

3.3 Concessions

Suite à un manque de temps, nous avons dû revoir nos objectifs 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 la manière de jouer au jeu et ainsi, d'en profiter pleinement. Vous l'avez donc sans doute remarqué, les extensions ne sont donc pas présentes dans notre jeu final ... ou du moins, elles ne seront 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 pourriez noter la présence de fichier xml spécifiques eux aussi, à cette extension. Bien entendu, cela n'a pas été suffisant pour la qualifier 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. Suite à quelques problèmes, il est venu trouver les assistants et professeurs afin d'obtenir un hébergement, malheureusement, il semblerait que ce n'est pas été le cas.

4 Notre jeu

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

5 Notes Annexes

Nous parlions d'une première version de l'interface graphique dans la section **Nos Choix**, 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 tous les commits pour les retrouver.