

Projet informatique

SAMUEL GIVOIS

ENSAE

samuel.givois@ensae-paristech.fr

16 mai 2018

I. INTRODUCTION

L'objectif fixé pour ce projet informatique est la création d'un jeu, ou plus précisément la mise en place d'une structure minimale de jeu, pouvant être par la suite enrichie facilement. Le jeu choisi est « Bomberman », créé par l'éditeur Hudson Soft en 1983. Il s'agit d'un jeu en 2 dimensions, avec vue isométrique (l'univers est vu d'en haut). Le joueur est dans un labyrinthe, et peut poser des bombes. Celles-ci peuvent détruire des murs de brique ou des ennemis. En détruisant un mur, des bonus (ou malus) apparaissent. Le succès du jeu est venu de son mode multijoueur, le but étant d'être le seul survivant. On a cherché ici à développer en priorité un jeu pour un seul joueur, plus facile à réaliser. Cependant, le résultat obtenu devra permettre de passer facilement à un mode multijoueur.

II. CHOIX DES TECHNOLOGIES

La plupart des langages existant et l'écosystème informatique permettent facilement le développement de jeux sur ordinateur. Cela est dû à l'ancienneté de ces systèmes, avec pour conséquence une communauté informatique mature et des sources documentaires en profusion. Le développement se faisant essentiellement sur ordinateur, les tests sont par ailleurs facilités pour les applications de bureau. Cependant, il est difficile d'ignorer la prépondérance du mobile aujourd'hui, en particulier pour les jeux simples comme « Bomberman ». On a donc cherché lors de ce projet à développer un jeu qui soit compatible pour ces deux types de terminaux, les ordinateurs non tactiles

a priori et dotés d'un clavier, et les mobiles et tablettes.

N'ayant jamais développé de jeux vidéo et afin d'avoir le temps d'assimiler les contraintes et technologies associées à ce type de projet, on a choisi de se baser sur un langage qu'on maîtrisait déjà, le langage Java. Pour permettre la réalisation d'un jeu multiplate-forme, on a étudié plusieurs solutions possibles reposant sur l'écosystème Java. Une première piste à paraître intéressante fut l'API JavaFX. Cependant, il est rapidement apparu que la portabilité sur mobile était un peu plus compliquée que prévu. On s'est alors orienté sur le « framework » libGDX, créé par Mario Zechner en 2014. Celui-ci, une fois installé et correctement configuré, permet une réalisation facilitée de livrables dédiés pour chacun des environnements cibles (bureau, Android, iOS, ...).

Par habitude, on a utilisé l'environnement de développement Eclipse pour l'écriture du code. Cependant, Google a interrompu le support des outils de développement pour Android intégrés à Eclipse en novembre 2016, et a créé son propre environnement de développement Android Studio. Les outils d'Eclipse ne fonctionnant qu'imparfaitement, on a également utilisé l'IDE de Google pour les besoins spécifiques aux applications mobiles (test et construction des livrables en particulier).

Le « framework » libGDX repose en outre sur le moteur de production Gradle, permettant notamment la gestion des dépendances et la construction de livrables (à l'instar de Maven). Eclipse et Android Studio sont capables de gérer cet outil. On a enfin choisi git pour la gestion de version, avec un dépôt central sur le site GitHub.

III. DE LA CARTE À L'ÉCRAN

Un avantage supplémentaire de libGDX, est qu'il permet de manipuler des fichiers de carte au format TMX. Ceux-ci définissent des cartes carroyées (« tiled map »), souvent utilisées pour les jeux en vue isométrique. Le principal intérêt est de pouvoir découpler la partie développement de la partie création des cartes. Pour réaliser ces dernières, on a utilisé le logiciel Tiled, créé par Thorbjørn Lindeijer.

Permettre le passage de la carte au jeu à proprement parler est ce qui a constitué la majeure partie du travail de ce projet (disons 80 %). En effet, cela a nécessité un temps conséquent pour comprendre le « framework » et ses possibilités. En contrepartie, peu de développement seront nécessaires pour l'extension du jeu (création de nouveaux niveaux ou de nouveaux personnages).

LibGDX fournit un modèle complexe pour représenter l'univers du jeu (à l'instar de l'univers qu'on connaît, avec la gravité par exemple). Après de nombreux essais infructueux, on a choisi d'utiliser une représentation simplifiée, qu'on a implémenté spécifiquement. On a également adapté un autre outil, la caméra, afin de permettre la création de niveaux dépassant l'écran (dans l'optique de livrables pour mobile en particulier). On a ainsi réussi à obtenir une vision centrée sur le joueur si la carte dépasse le cadre de l'écran, dans la limite du bord de la carte.

IV. DÉVELOPPEMENT DU JEU

La principale difficulté est dans la gestion du temps.

V. NAVIGATION

Facile à réaliser une fois que le framework est compris.

VI. PAR LA SUITE