PG4 : Akropolis

Nidhal Moussa & Mathéo Piget & Benzerdjeb Reyene & Gbaguidi Nerval & Chetouani Bilal

5 mai 2024

Table des matières

1 Introduction

1.1 Présentation du Projet

Le projet consiste en la réalisation d'un jeu de société nommé Akrapolis, qui est un jeu tour par tour dont le but est de construire une cité antique. Le jeu se déroule sur un plateau de jeu composé de cases hexagonales. Chaque joueur possède un certain nombre de ressources qu'il peut utiliser pour construire des bâtiments sur les cases du plateau. Chaque bâtiment rapporte des points de victoire à son propriétaire. Le joueur qui a le plus de points de victoire à la fin de la partie est déclaré vainqueur.

Réalisé en java et en utilisant la bibliothèque graphique Swing, le jeu est jouable en local sur un ordinateur.

1.2 Objectifs

L'objectif principal du projet est de réaliser un jeu de société complet et fonctionnel, avec une interface graphique permettant de jouer en local sur un ordinateur. Les objectifs secondaires ont été de réaliser un jeu agréable à jouer, avec une interface graphique intuitive et des graphismes de qualité.

2 Gestion du projet

2.1 Méthode

Pour la gestion du projet, nous avons opté pour une méthode de développement agile, basée sur des itérations courtes et des réunions régulières pour faire le point sur l'avancement du projet et discuter des problèmes rencontrés. Nous avons également utilisé un dépôt Git pour gérer les différentes versions, ce qui nous a permis de travailler en parallèle sur les différentes parties du jeu et de fusionner nos modifications facilement.

2.2 Répartition des taches

Pour la réalisation du projet, nous avons réparti les tâches en fonction des compétences de chacun, la polyvalence de chacun nous a permis de travailler sur plusieurs aspects du jeu. Nous avons également organisé des réunions régulières (plus ou moins obligatoire grace a ce cours) pour discuter de l'avancement du projet et des problèmes rencontrés (voir section 4).

2.3 Architecture du Projet

Pour ce projet, nous avons opté pour une architecture orientée objet, avec une séparation claire entre les différentes parties du jeu, basé sur un modèle MVC (Modèle-Vue-Contrôleur) car beaucoup plus modulable et facile à maintenir, pour faire des modifications ou ajouter des fonctionnalités. Nous avons également utilisé des classes utilitaires pour gérer le jeu. La partie graphique du jeu est gérée par la bibliothèque Swing, qui permet de créer des interfaces graphiques en Java. Avec une grande partie assez abstraite, le jeu est facilement modifiable et extensible, principalement pour les jeux de société.

3 Développement et Fonctionnalités

Le développement du jeu a été réalisé en plusieurs mois, en plusieurs étapes, en commençant par la réalisation des mécaniques de jeu de base, puis en ajoutant des fonctionnalités supplémentaires et en améliorant l'interface graphique. Dans un premier temps, étant donné que le modèle de conception du jeu est basé sur le modèle MVC, nous avons commencé par la réalisation du modèle, la vue et le contrôleur plus ou moins en même temps. Parallèlement à cela, nous avons travaillé sur les tâches annexes comme la gestion des images, des sons, mais également sur des scripts bash et powershell. Mais également sur des tests unitaires pour vérifier le bon fonctionnement de notre code.

3.1 Organisation du Code

Le code du jeu est organisé en plusieurs packages, chacun regroupant des classes ayant un rôle adéquat. Le modèle du jeu est regroupé dans le package `model`, la vue dans le package `view` et le contrôleur dans le package `controller`, eux même contenant certain d'autre package pour une meilleure organisation. Les classes utilitaires sont regroupées dans le package `test`. Les ressources du jeu (images, sons, etc.) sont stockées dans le package `res`. Pour rentrer dans les détails, le modèle est composé de plusieurs classes, chacune représentant un élément du jeu (joueur, plateau, bâtiment, etc.). La vue est composée de plusieurs classes, chacune représentant un élément graphique du jeu (fenêtre principale, plateau de jeu, etc.). Pour plus de lisibilité et de clairet, nous avons décidé de faire un schéma UML de notre projet, que vous pouvez retrouver en annexe, voir section **??**.

3.2 Fonctionnalités Principales

Etant donné que le jeu est un jeu de société, les fonctionnalités principales sont les suivantes :

— **Plateau de Jeu :** Le plateau de jeu est composé de cases hexagonales, sur lesquelles les joueurs peuvent placer de tuiles.

— **Ressources :** Chaque joueur possède un certain nombre de pierre qu'il peut utiliser pour construire des tuiles.

3.2.1 Mécaniques de Jeu

3.2.2 Interactions Utilisateur

3.3 Gestion des Ressources

Images et Textures

— **Chargement des Images :** Les images et textures du jeu sont stockées dans des fichiers séparés.

— Le jeu les charge dynamiquement lors de son exécution.

— **Format des Images :** Les images sont au format PNG pour assurer une qualité graphique optimale tout en minimisant la taille des fichiers.

Fichiers Sonores

— **Musique de Fond :** La musique de fond du jeu est également gérée en tant que ressource sonore, contribuant à l'ambiance globale.

3.4 Fichiers de Configuration

3.5 Logique de Jeu

4 Difficultés Rencontrées

5 Conclusion

6 Annexes

Table des matières