Projet tutoré #2
Démineur - Java

Année universitaire 2017-2018

Rapport de projet

Julien Hoeglund
& Valentin Lefebure
1ère année de DUT Informatique

Table des matières

Introduction

Le jeu du démineur est un jeu à un seul joueur dont le but est de révéler toutes les cases non minées d’une grille sans cliquer sur une mine, sinon la partie s'arrête.

Le titre est un jeu de mot entre "Minesweeper" et "my nerves" ("mes nerfs"en français) faisant référence à l'aspect stressant du jeu et il rappelle la déesse de la guerre, de la stratégie et de l'intelligence dans la mythologie romaine, Minerve (Minerva en latin), équivalente d'Athena chez les Grecs.

Description des fonctionnalités

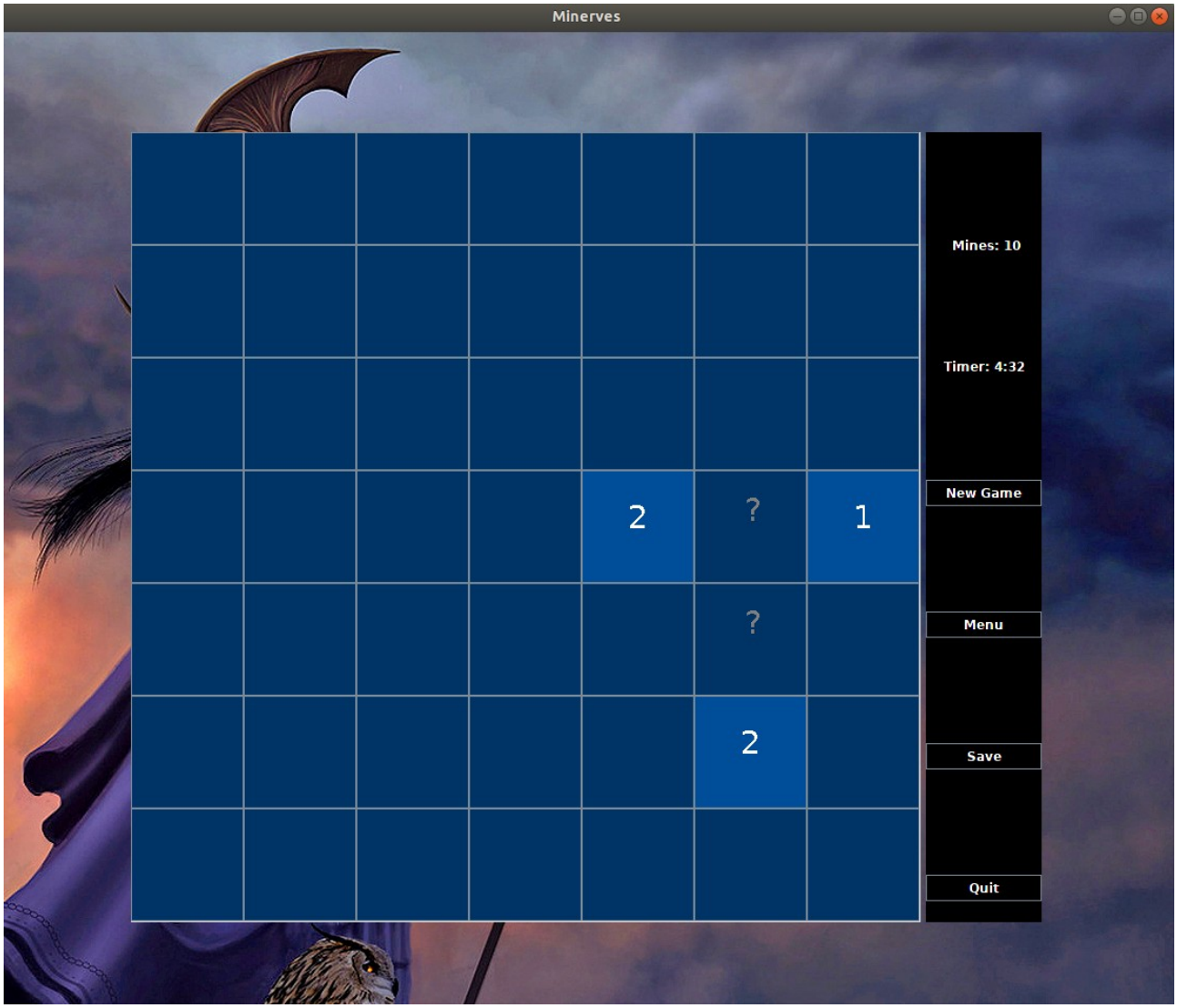
Menu principal

- Le bouton Resume : il n'est affiché que si une sauvegarde est disponible, si c'est le cas la partie reprend directement au clic.
- Le bouton New Game : amène le joueur au menu de configuration de la partie.
- Le bouton Story amène le joueur à un texte qui l'introduit au contexte du jeu.
- Le bouton Quit permet de quitter le jeu.


Minerves
Minerves
"Are the odds good ?" asked the legionary to the Oracle.
She turned her hooded face and set her old, maliceful eyes upon him.
"The last three sacrifices' guts were black. A bull and two horses."
The thought made the man shudder.
"Then she's coming..."
"The war will be terrible for you, legionary. Many losses."
He paused, thinking about his reasons for a moment.
"It will be worth the fight. Reason must prevail. Men will learn to live without the reassuring presence of their gods."
The Oracle turned away to feel the fire in the hearth made of cold marble.
"Men will have to learn to live without your presence.
She will surely have prepared her deadly game for you..."
"I shall fight the odds. I shall fight with the power of my mind."
Menu

Scénario

Menu de configuration de partie
Les boutons permettent d'augmenter ou réduire la taille de la grille en largeur et en hauteur, le nombre de mines et le chrono. Le nombre de mines est limité par la taille et cette dernière est limitée à 30x30. Le joueur est libre de créer la partie avec les paramètres choisis, revenir au menu ou quitter le jeu.



Partie

Le bouton Save crée un fichier de sauvegarde de la partie tout comme la croix en haut de la fenêtre.
Save et Quit sont séparés, pour donner plus de liberté au joueur.
La sauvegarde est supprimée si la partie a été terminée (victoire ou défaite).
Les flags du clic droit permettent au joueur de garder des repères.

Structure du programme

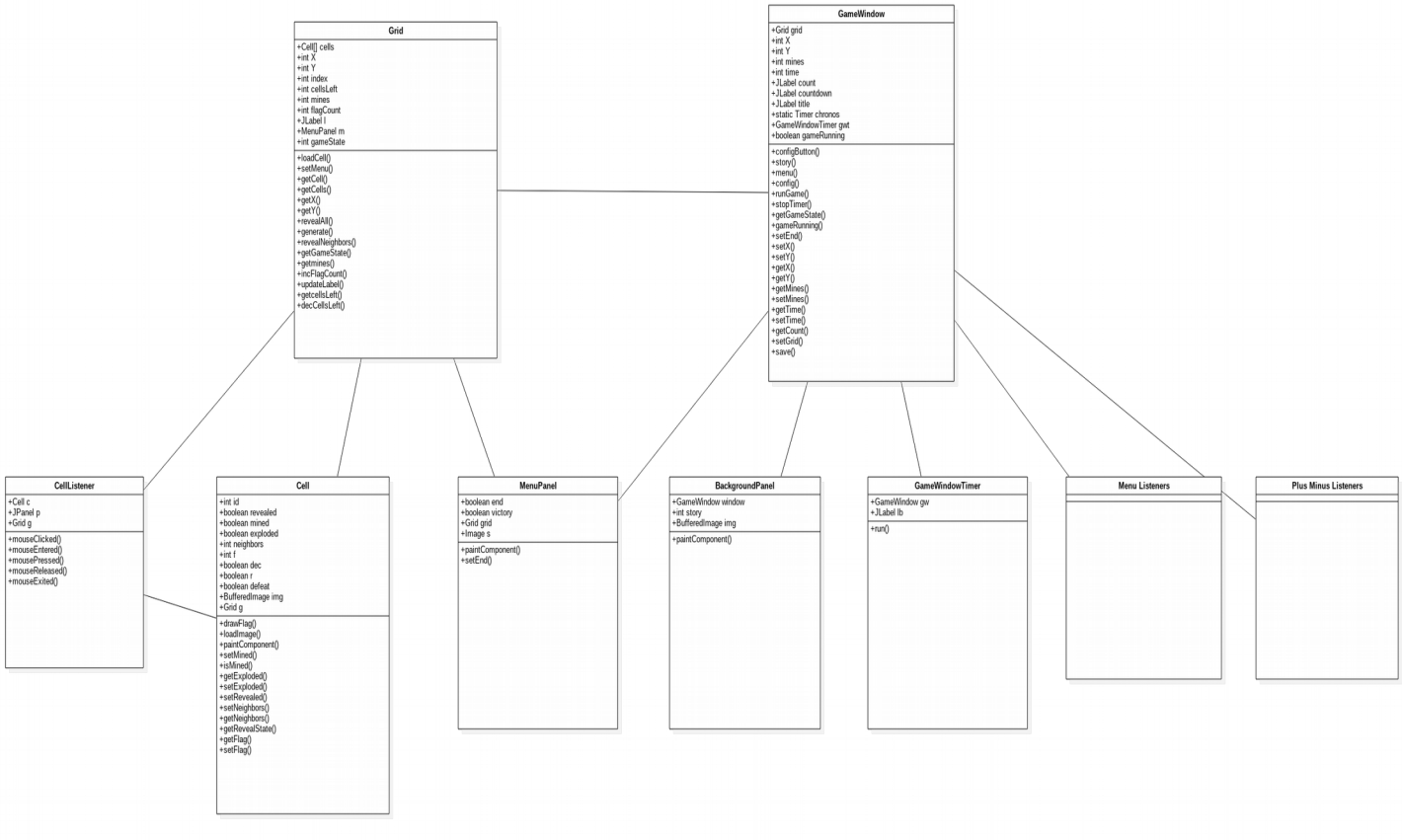


Diagramme de classes du projet

Les listeners ont été réunis et schématisés par la classe MenuListeners et Plus Minus Listeners. GameButton est absente car elle n’apporte rien au diagramme.

La classe principale est GameWindow. Elle reçoit les panneaux, boutons, labels et autres objets qui composent le jeu.
La classe Cell rassemble les différents états et propriétés d’une case en un objet. Elle hérite de GameButton qui est une petite modification esthétique de JButton.
La classe CellListener est l'actionListener utilisé par les instances de Cell. Elle gère les actions à réaliser selon l'état de la cellule et de la partie et le clic du joueur.
La classe Grid représente le plateau de cellules entier et gère des états globaux comme les flags mis par le joueur et le nombe de mines à afficher. Elle gère donc un tableau de cellules.
Un chronomètre à été ajouté au jeu par le biais d'un objet de la classe Timer située dans Java.util.Timer qui se sert d'un objet de la classe abstraite TimerTask, que nous avons override (GameWindowTimer) pour lui transmettre les indications que nous voulions donner à notre Timer.

Mécanisme de sauvegarde

La méthode save() écrit les dimensions de la grille puis l'état de chaque cellule dans l'ordre d'itération dans un fichier "save.mns".
La méthode actionPerformed() du bouton Resume fait exactement la même chose mais en lisant les entiers et les booléens puis en les chargeant dans la grille et dans les cellules.

Algorithme de révélation des cellules

Si une case révélée par le joueur n'a aucun voisine minée, alors on révèle ses voisines directes.
La methode revealAll() prend soin de ne pas créer de ArrayOutOfBoundsException en vérifiant que les cellules à révéler sont bien dans la grille (index supérieur ou égal à 0 ou inférieur à largeur*longueur).
Comme on ne veut pas que la révélation soit "circulaire" on ne révèle pas les cellules voisines d'une cellule en bord opposé de grille, donc on vérifie cela aussi.
Une case révélée de cette manière vérifie donc à son tour qu'elle n'a
aucun voisins grâce à la methode paintComponent().

Conclusion - Julien

Ce projet a été formateur pour moi. La programation événementielle est un paradigme différent de la programmation séquentielle et son classique Game Loop que l'on est écrit lorsque l'on programme un jeu dans d'autres langages tels le C ou C++. Travailler sur un exercice d'une ampleur comme celle-ci m'a vraiment fait réaliser cette différence.
Je pense que créer un jeu qui repose simplement sur une interface graphique est à la portée de tout le monde. En effet, les compétences techniques requises pour créer un jeu qui a un minimum de graphismes 2D ou 3D en temps réel ou une gestion du réseau (sans moteur de jeu) par exemple sont bien plus complexes. Cependant, ce type de projet est un pas en avant et il nous fait comprendre que des systèmes comme la sauvegarde ne sont pas si complexes qu'ils ne paraissent.

Conclusion – Valentin

J'ai beaucoup aimé travailler sur ce projet, tout d'abord car je n'avais jamais programmé quelque chose d'aussi développé en Java , le Python et le C# sont les seuls langages sur lesquels j'ai effectué des projets importants jusqu'ici, mais aussi parce que ce projet m'a aidé d'un côté à comprendre le fonctionnement du Java de façon plus poussée (en découvrant l'utilisation des images et des Timer par exemple), mais aussi à devoir décomposer un projet et ainsi percevoir les particularités de chaque fonctionnalité du jeu de manière isolée.
Ce projet en plus de m'avoir vraiment donné une idée plus précise de ce qu'est le java m'a totalement décidé à redémarrer des projets personnels de programmation, notamment d'applications ou de jeux vidéo avec une vision nouvelle de la façon d'aborder certaines fonctionnalités de ces-dits projets.