

DEMINEUR

Projet APL

Jürgen Hablainville | Nathan Bertholier

Description du sujet

Le projet demande a pour but de réaliser un Démineur à partir du Java.

Le jeu se lance sur un menu qui permet de choisir entre jouer ou quitter (et charger si une partie a été sauvegardée).

Quand on lance une partie on arrive sur un menu qui permet de choisir le nombre de lignes et de colonne ainsi que le nombre de mines.

Les mines sont alors réparties de manière aléatoire sur la grille et cachée a l'utilisateur. La grille est entièrement cachée, l'utilisateur doit alors cliquer sur une case, ce qui révèle la case.

Si la case est minée alors la partie est terminée et l'utilisateur a perdu.

Si la case n'est pas minée alors on affiche dans la case le nombre de mines colée à la case (angles compris).

Si le nombre de mines colée à la case est égal à 0 alors on révèle les cases aux alentours si elles sont égales à 0 (donne un effet récursif).

L'utilisateur peut poser des marqueurs sur les cases avec le clique droit. Le premier clique pose un drapeau, le second pose un "?" et le dernier remet la case à vide. Seul la case à vide peut être cliqué pour révéler la case.

L'utilisateur gagne lorsqu'il a révélé toutes les cases non minées.

Un second menu apparaît en même temps que la grille principale et affiche le nombre total de mines ainsi que le nombre de marqueur, il a la possibilité de sauvegarder ou de quitter.

L'action de fermer la fenêtre de la grille permet de sauvegarder.

Quand l'utilisateur perd alors on affiche sur la grille les mines non explosées et la mine explosée (de manières différentes).

Description des fonctionnalités

Le jeu se lance sur un menu qui permet de choisir entre commencer une nouvelle partie ou quitter.



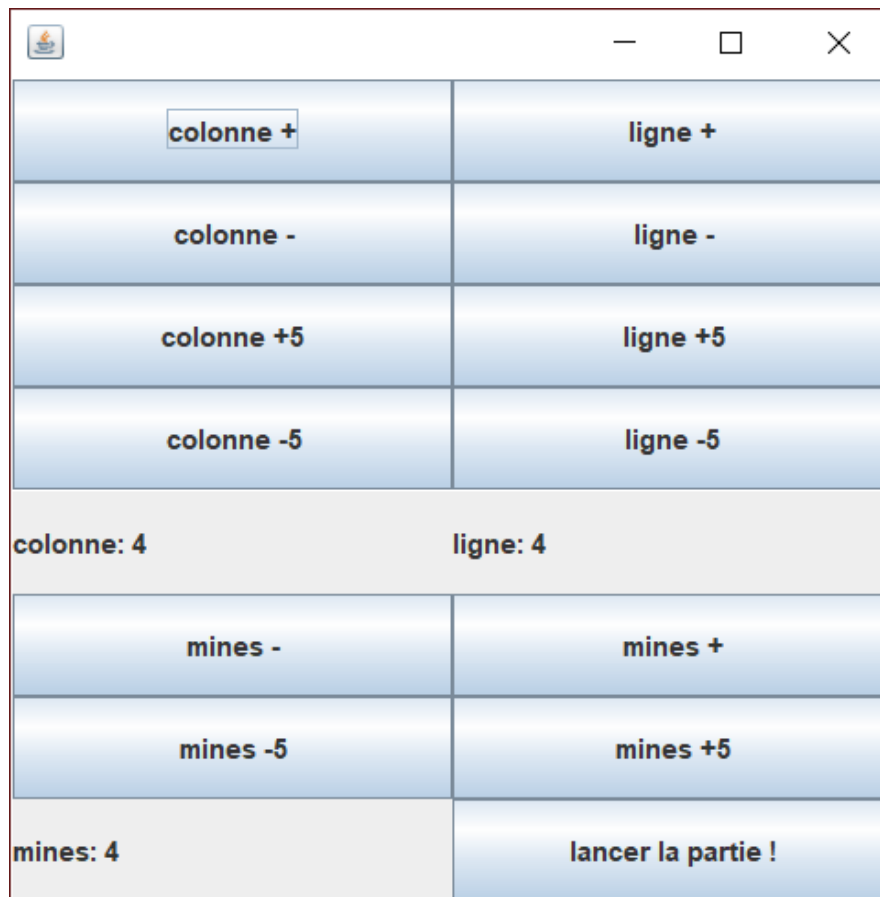
Menu de base

Si une partie a été sauvegarder, alors une nouvelle option charger apparait.

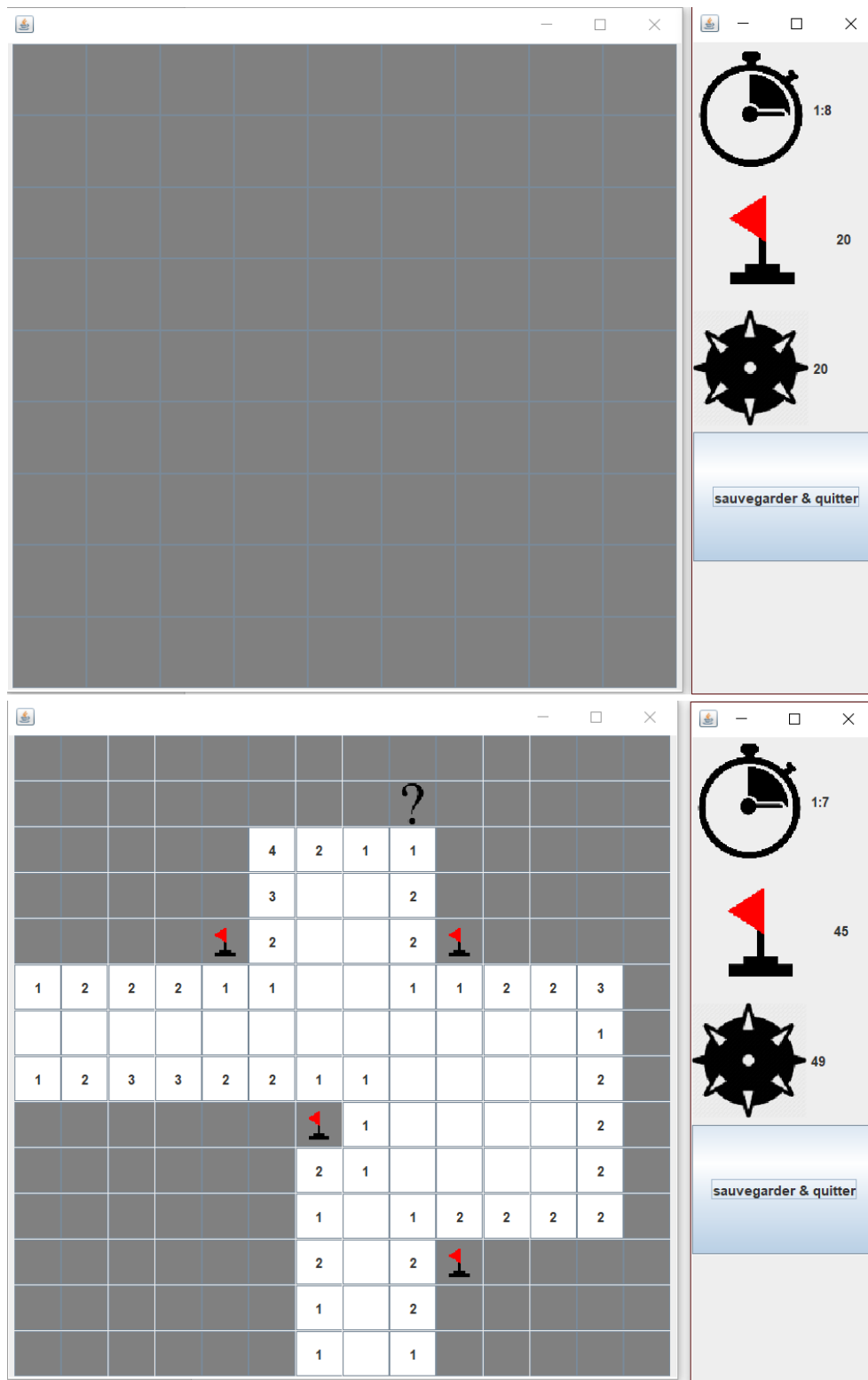


Menu quand une partie a été sauvegardé

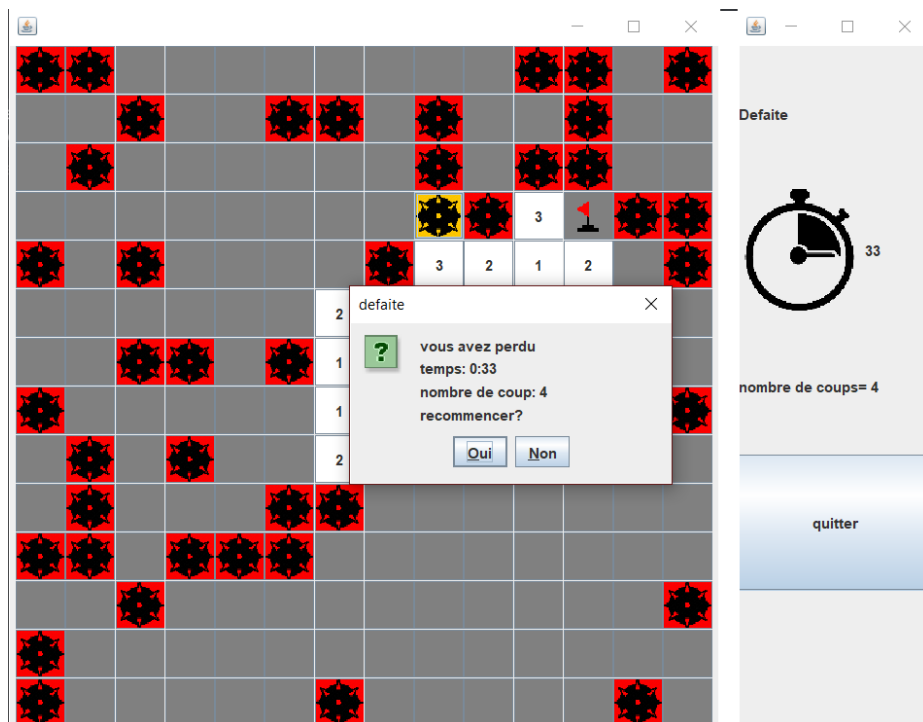
Lorsqu'on lance une partie on arrive sur un menu qui permet de choisir le nombre de lignes, le nombre de colonnes ainsi que le nombre de mines.



Les mines sont alors réparties de manière aléatoire sur la grille et cachées à l'utilisateur. La grille est entièrement cachée, l'utilisateur doit alors cliquer sur une case, ce qui la révèle.

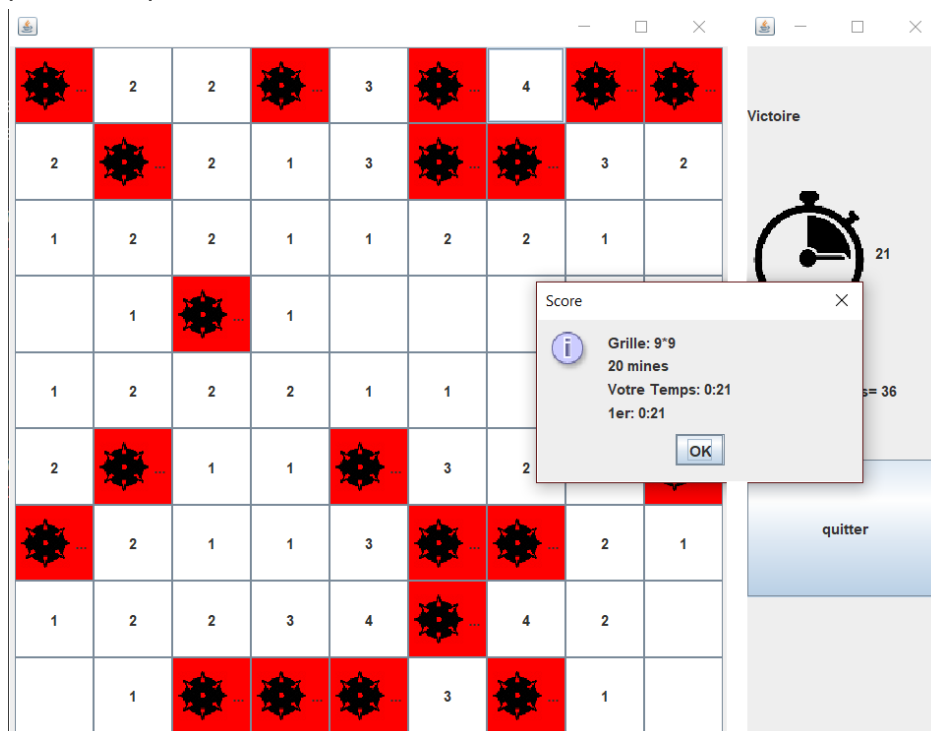


Si la case est minée alors la partie est terminée et l'utilisateur a alors le choix entre quitter ou recommencer la partie avec les mêmes paramètres.

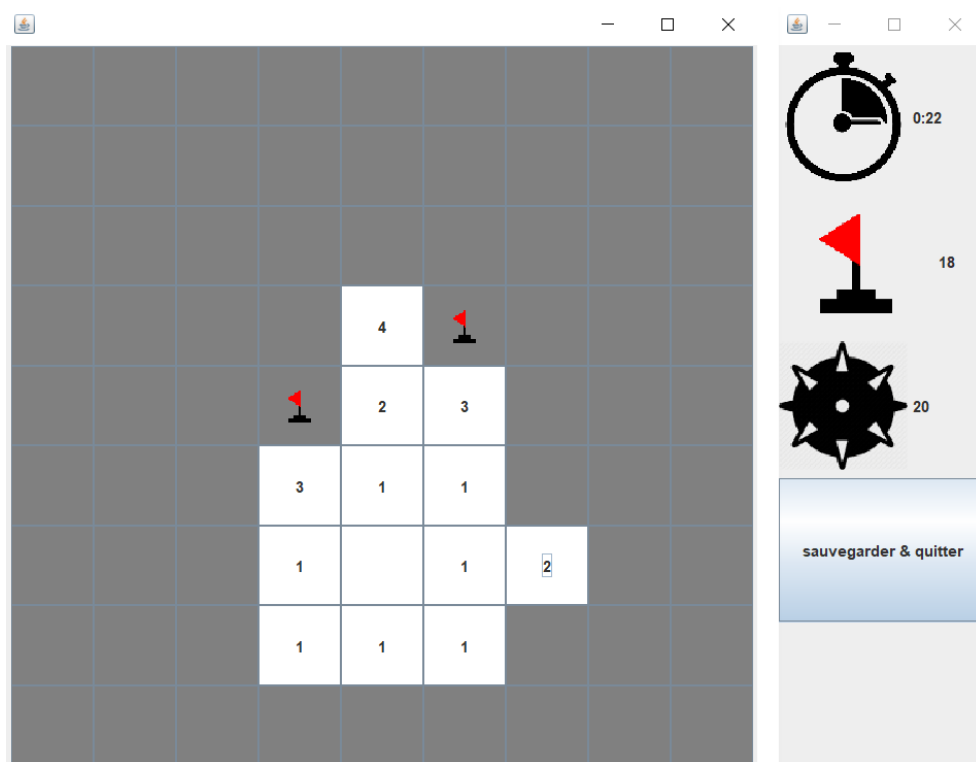


Si la case n'est pas minée alors on affiche dans la case le nombre de mines adjacents à cette case. Si le nombre de mines colée à la case est égal à 0, alors on révèle les cases aux alentours si elles sont également égale à 0 (effet récursif).

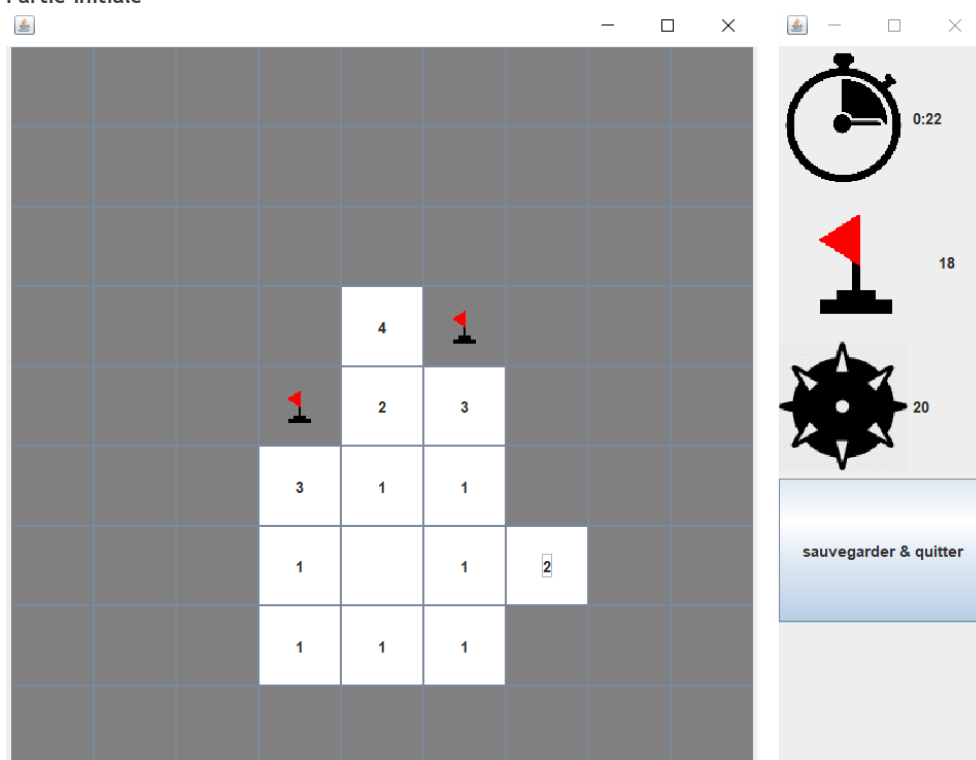
L'utilisateur a la possibilité de poser des marqueurs sur les cases avec le clic droit. Le premier clique pose un drapeau, le second pose un « ? » et le dernier remet la case vide. Seul une case vide peut être cliqué pour révéler la case. L'utilisateur gagne la partie lorsqu'il a révélé toutes les cases non minées.



Un second menu apparaît en même temps que la grille principale et affiche le nombre total de mines ainsi que le nombre de marqueur, il a la possibilité de sauvegarder ou de quitter. L'action de fermer la fenêtre de la grille permet de sauvegarder.

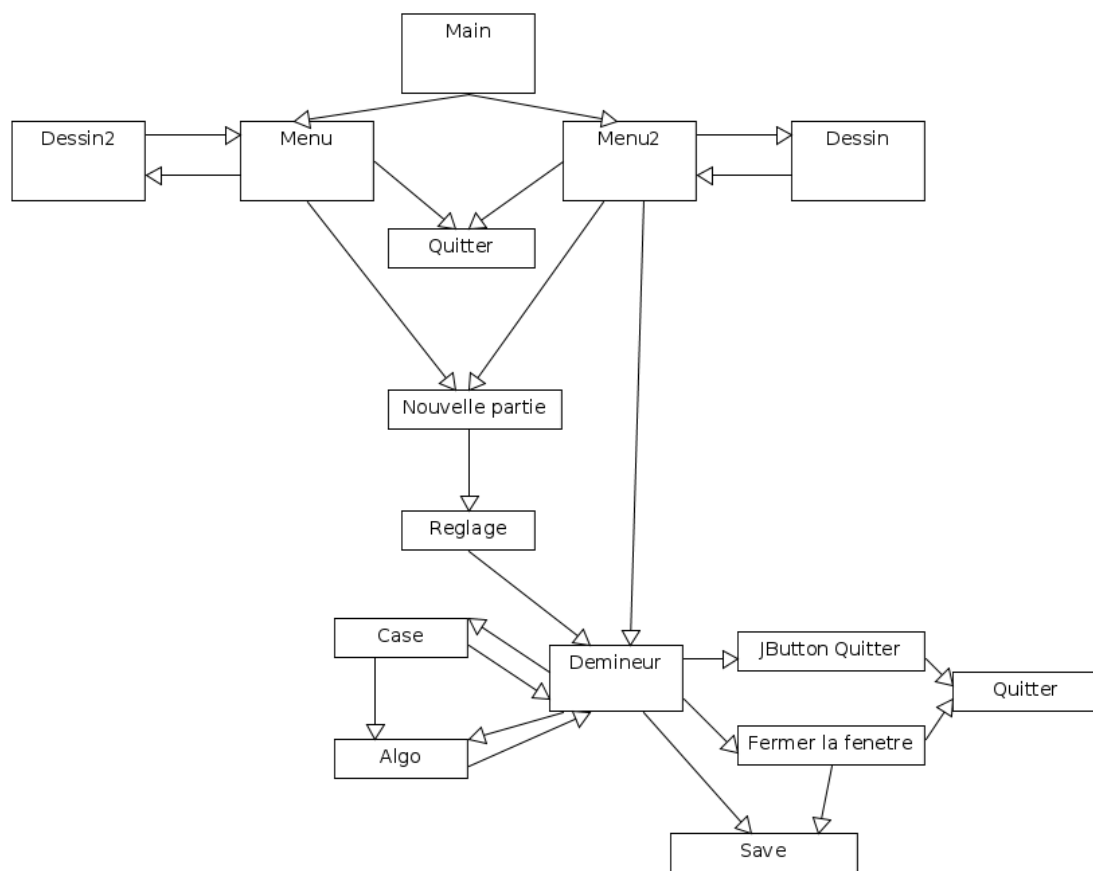


Partie initiale



Partie chargée

Quand l'utilisateur perd, toutes les mines non explosées s'affichent d'une couleur, la mine qui a explosé est affichée d'une couleur différente.



Structure du programme

Le programme se lance par le Main.

Si aucune sauvegarde n'est présente, alors on lance Menu, si une sauvegarde est présente alors on lance Menu2.

Menu permet de lancer une partie ou de quitter tandis que Menu2 permet de lancer une partie, charger une partie ou quitter.

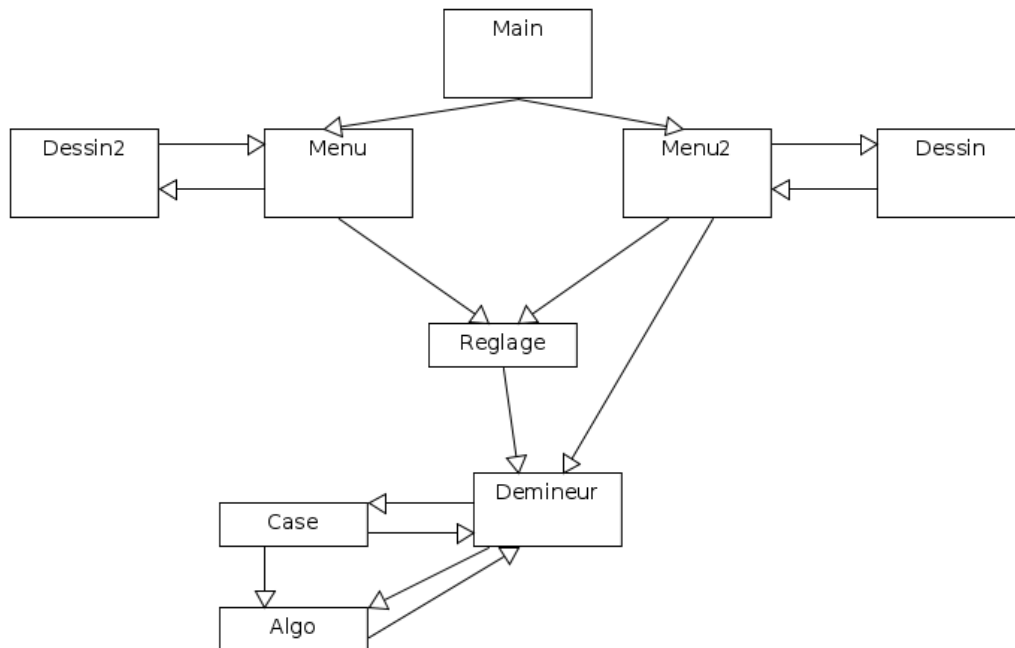
Quand une partie est lancée, on ferme le menu et on lance Réglage qui permet de choisir le nombre de lignes, de colonnes, et de mines.

Quand la sélection est confirmée on lance Démineur qui répartit de manière aléatoire les mines sur un héritage de JButton (Case) et permet le fonctionnement du jeu.

Si on charge une partie, alors la répartition des mines est la même que dans le fichier de sauvegarde.

Quand une case est cliquée on utilise Algo qui permet de calculer le nombre de mines aux alentours ainsi que le calcul récursif des cases vides.

La sauvegarde est effectuée par Case qui stocke dans « save.txt » les informations de la partie



System de sauvegarde

Lors de la sauvegarde, le nombre de ligne, de colonnes, de mines, le temps, le nombre de marqueurs, ainsi que la position des mines est enregistré dans le fichier « save.txt ». Si on perd la partie, alors le fichier sauvegarde est vidée. Un indicateur est présent afin de savoir si une sauvegarde est présente ou non. Dans le cas où une sauvegarde est présente, le menu propose une option pour charger cette sauvegarde. L'écriture des données est effectuée dans un ordre précis afin de pouvoir les lire aisément, et pour pouvoir les utiliser facilement lors du chargement d'une sauvegarde. Toutes les informations stockées sont lues lors du chargement de la sauvegarde. La partie est également sauvegarder lorsque le joueur quitte la partie en utilisant la croix, ainsi il n'y a pas de risque de perdre sa progression en cliquant sur la croix par inadvertance.

Algorithme de révélation

L'algorithme de révélation est construit de manière récursive. Dans un premier temps l'utilisateur clique sur une case, si cette case n'est pas entourée par une mine alors on rentre dans une boucle récursive, la case est considérée comme visitée alors on ne peut plus revenir sur cette case, alors on regarde si la case du dessous est vide (pas de mine détectée aux alentours), si c'est le cas alors on passe dans un nouveau tour de boucle

Conclusion personnelle

Nathan Bertholier

Pour conclure, je peux dire que la réalisation du premier projet d'APL (Taquin) nous a permis de mieux nous préparer en réalisant un UML ainsi que d'être plus à l'aise avec les tableaux bidimensionnels. Nous avons donc pu nous répartir les tâches et gagner du temps en commençant le jour de la sortie du projet. Si je devais améliorer une partie, je pense que je me focaliserai sur l'aspect graphique pour un rendu de meilleure qualité. La plus grosse difficulté étant l'algorithme de découverte des cases vides, je pense que nous avons trouvé une solution adaptée en passant par la récursivité.

Jürgen Hablainville

En conclusion, je dirais que ce projet nous a permis de pousser plus loin et d'approfondir nos connaissances et notre maîtrise du java. En nous y prenant de bonne heure, nous avons pu nous organiser de façons à gagner du temps sur certaines parties afin de passer plus de temps sur les parties plus complexes tel que l'algorithme de révélation des cases vides.