UPEC
Connaissance - Action

UNIVERSITÉ
PARIS-EST CRÉTEIL
VAL DE MARNE

Projet Démineur

Pierre Villiers /Steven Renault

Iut Sénart Fontainebleau

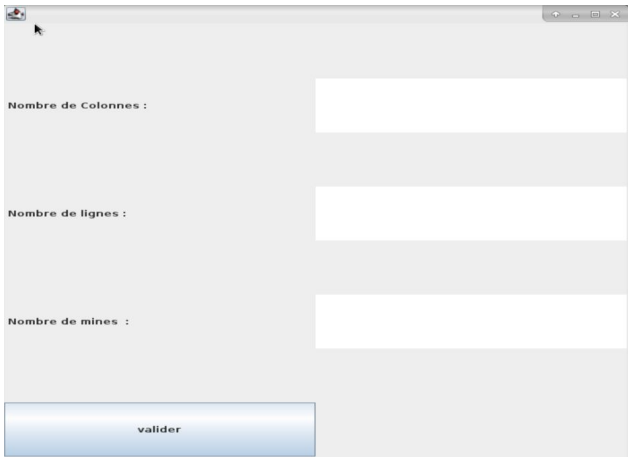
Sommaire

Présentation du projet :

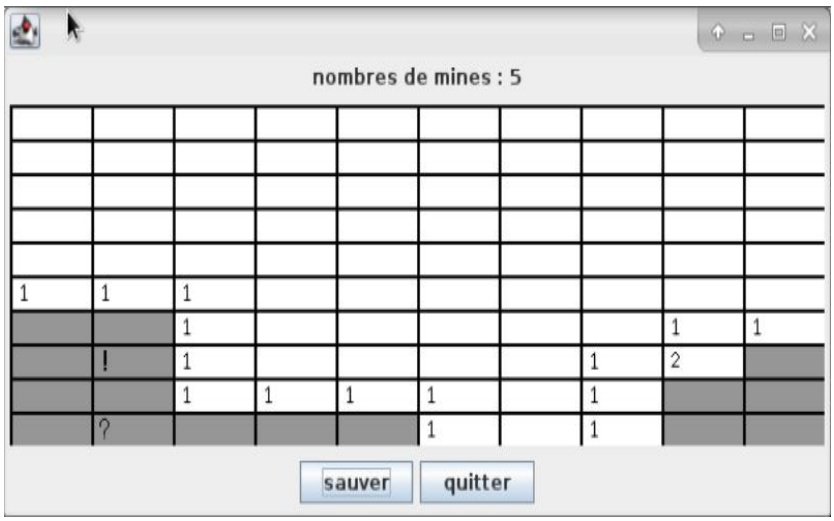
Notre projet consiste à créer un jeu du style demineur, ou des cases pouvant contenir des mines sont cachés, et dont le joueur devra les découvrir sans tomber sur des mines. Et pour gagner, le joueur devra découvrir toutes les cases cachés qui ne sont pas des mines.

Fonctionnalités du programme :

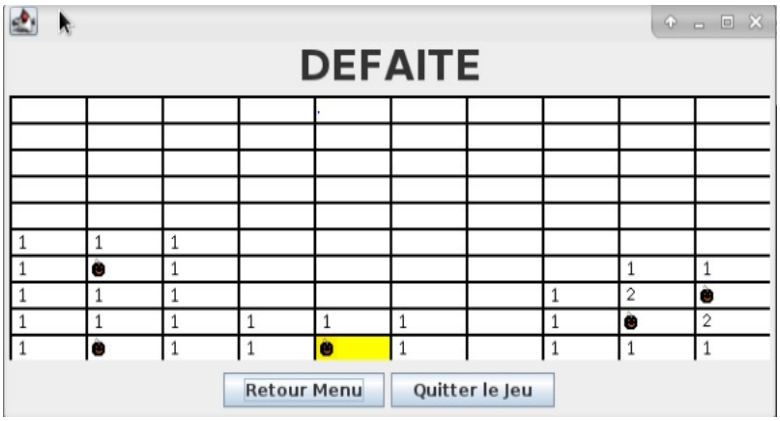
- Lors du lancement du programme nous tombons sur le Menu, ou on peut trouver deux boutons (voir trois si le joueur a déjà fait une partie sauvegardé).



- En cliquant sur le bouton "Nouvelle Partie", vous arrivez sur une fenêtre où vous devrez rentrer le nombre de case en largeur, en longueur et le nombre de mine (sachant qu'il ne peut pas avoir plus de mine que de case).



- Après avoir bien rentré les données correctement et en ayant confirmées, vous pourrez commencer votre partie. En haut de la fenêtre vous trouverez le nombre de mines restante (le nombre peut diminuer si le joueur place des marqueurs ("!") sur les cases croyant que c'est une mine, mais pas avec le marqueur du doute " ?").Vous pouvez à tout moment sauvegarder votre partie, comme la quitter.



- Si vous révélez une case minée qui n'est pas marque, alors vous tomberez sur la fenêtre de défaite ou si vous découvrez toutes les cases sauf ceux miné, alors vous tomberez sur la fenêtre de victoire. Vous verrez toutes les cases, ainsi que si vous avez perdu, la mine sur laquelle vous avez révélé (qui a exploser).Pour quitter cette fenêtre vous avez deux choix, soit quitter le jeu soit revenir au menu.

Structure interne du programme :

-La classe Main contient le main qui va initialiser le lancement du jeu en appelant la classe menu.

-La classe Menu permet d'afficher la première interaction avec l'utilisateur en utilisant la souris.

-La classe DecorMenu permet de gérer les différentes caractéristiques du titre et l'emplacement de l'image sur le menu du jeu principal.

- La classe ControleurMenu permet d'observer sur quel bouton le joueur clique sur le menu du jeu pour soit démarrer, reprendre une partie ou bien la quitter.

-La classe Fichier permet de charger ou de sauvegarder une partie.

-La classe Dimension permet d'afficher les différents champs de texte ou le joueur devra rentrer le nombre de case en largeur, en longueur et le nombre de mines dans les champs correspondants. Cette classe permet également de récupérer les valeurs inscrites dans chacun des champs de texte.

-La classe Erreur est utilisée pour gérer les cas d'erreur avec une phrase détaillée.

-La classe ControleurDimension permet de convertir les données inscrites dans les champs de texte et de vérifier si elles remplissent les conditions pour créer la grille et le plateau du jeu dans le cas contraire en fonction de l'erreur un message d'erreur apparait.

-La classe Case est utilisée comme structure et composant graphique, il représente une case de la grille

-La classe Grille est utilisée pour créer une grille de case qui sera la référence logique du jeu, elle permettra également de déterminer le nombre de mines adjacents qui seront adjacents aux cases.

-La classe Plateau est utilisée pour créer le plateau du jeu , elle permettra d'afficher le nombre de mines d'une partie en cours et les boutons « sauver » si l'utilisateur souhaite sauvegarder sa partie en cours ou bien quitter avec le bouton « quitter ».Elle aura la capacité de supporter également la grille du jeu.

-La classe ControleurPlateau permet de contrôler sur quel bouton a été appuyé dans le cas du bouton « sauver » nous sauvegardons le différent paramètre de la partie en cours (la grille, le nombre de colonne, le nombre de ligne et le nombre de mines sinon en appuyant sur le bouton « quitter» la partie en cours prend fin.

-La classe ControleurFenetre sauvegarde les données de la fenêtre du jeu avant la fin de la partie.

-La classe ControleurSouris est utilisée pour récupérer les informations sur la souris. En fonction des commandes on modifiera les paramètres de la case sélectionnée dans la grille du plateau de jeu durant une partie en cours.

-La classe ActionClick est utilisée pour contenir toute les actions qui vont être entrainés par un clic et changer l'état d'une case et des autres sous certaines conditions.

-La classe Resultat signale si le joueur à gagné ou perdu la partie en laissant le temps au joueur de contempler la grille avant de revenir au menu du jeu principal.

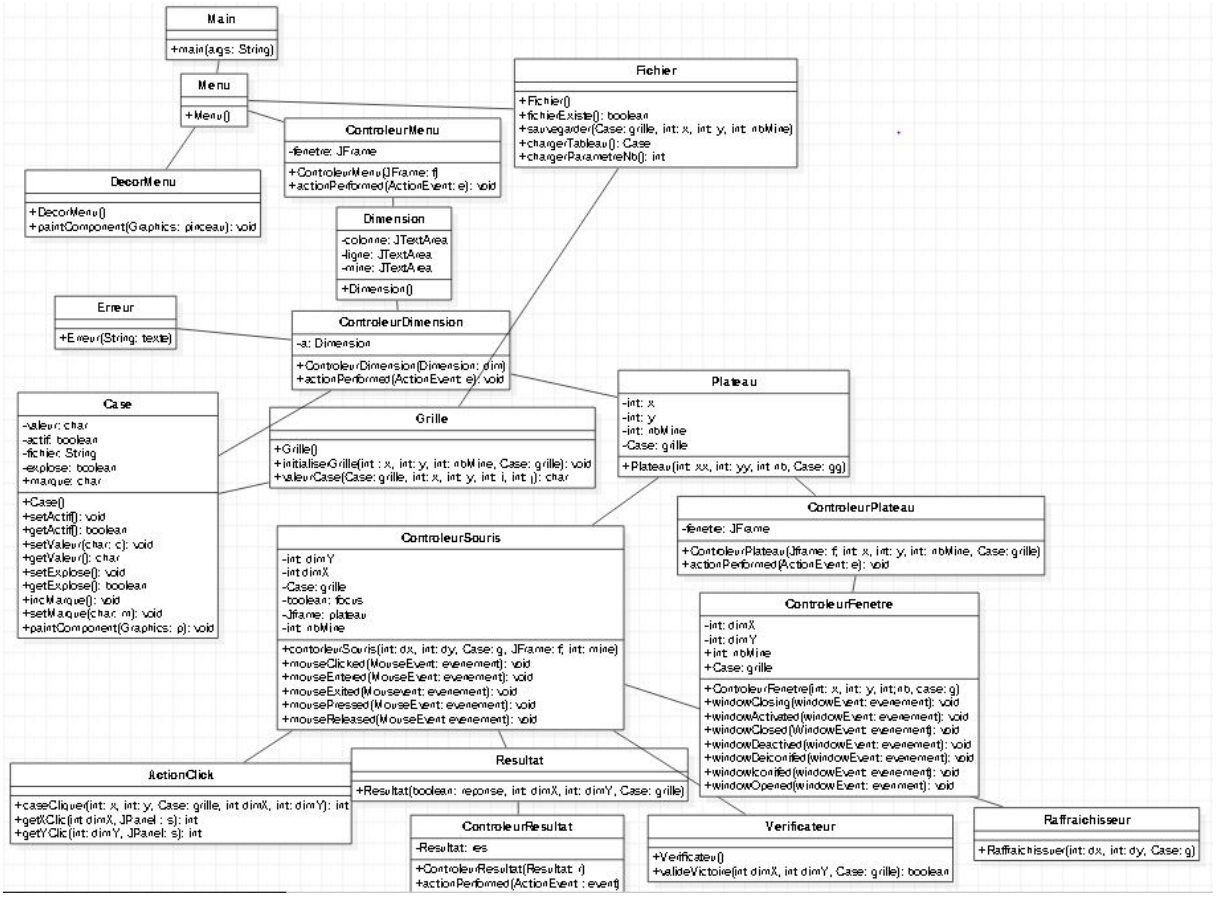
- La classe Verificateur est utilisée pour exécuter la méthode valideVictoire en validant si toutes les cases ont été activées sauf les mines.

La classe Raffraichisseur permet de repeindre les différentes cases de notre tableau.

-La classe ControleurResultat est utilisée pour définir les différentes actions des boutons de la classe Resultat soit « retour au menu » ou « quitter le jeu ».

Diagramme de classe :

Le programme se décompose comme ceci en plusieurs classes s'appelant entre elles. Le diagramme de classe ci-dessous (de haut en bas) représente la structure des classes avec les flèches d'association en dessous d'elles qu'elles utilisent.



Mécanisme de sauvegarde d'une partie

Lorsque vous quitter la fenêtre de partie ou vous cliquez sur le bouton sauvegarder, vous déclencherez ce mécanisme dont le but est d'enregistrer les paramètres et la grille de la partie sur un fichier. Tout d'abord, les données sont enregistrées en ascii pour simplifier la réaffectation des valeurs. On inscrit sur la première ligne le nombre de case de largeur, puis on continue avec une nouvelle ligne ou l'on trouve le nombre de case en longueur, on rajoute encore une ligne pour cette fois le nombre de mine dans la partie. Enfin on commence à rentrer les données de la grille, une ligne contient 3 caractères par case en largeur et on doit écrire autant de ligne que le nombre de case en longueur. Le premier caractère est la valeur de la case (sachant "*" correspond à une mine et "0" a une case vide), la seconde définie si cette case a été révélé ("0" caché et "1" révélé), et la troisième représente le type marqueur sur la case ("0" vide "1" danger "2" doute).Après avoir finit la grille, l'écriture dans le fichier est terminé et donc la sauvegarde est faite.

Algorithme d'activation de cases multiple

C'est une partie de la méthode d'activation de la case cliquée. Lorsqu'une case est cliquée par le joueur, on appelle cette méthode qui va récupérer cinq arguments. Le premier est le numéro de la case en largeur, puis le numéro de la case en longueur, la grille, puis la dimension en largeur de la grille et la longueur. Ensuite on active cette case en changeant sa variable "actif" à true (variable booléen qui est dans la case), on vérifie la valeur de cette case si c'est une mine on retourne –1 à cette méthode ce qui l'arrête, si elle vaut 0 alors on doit révéler les cases adjacentes qui respectent les conditions (que ce ne soit pas une case qui est hors de la grille),on vérifie aussi leur valeurs mais si une case vaut aussi 0 alors on rappelle la méthode ainsi on crée un effet de cascade (cette méthode peut être récurrente) donc chaque case vide révélé pourra être traité.

Conclusion

Pierre Villiers :

Le projet fut attractif pour moi car il propose de créer un jeu dont je n'avais jamais bien compris le mécanisme et qui a un côté stratégique qui m'intéresse. Suite à ce projet, j'ai pu bien confirmer mes acquis en java, même si les interfaces graphiques sont toujours assez difficiles à utiliser lorsqu'on veut approfondir l'esthétique de l'interface graphique. Il m'a permis de mieux voir comment dans un projet, on distingue le rôle d'une classe ou d'une méthode (vue, contrôleur, modèle) pour la structure totale du projet.

Steven Renault :

Ce projet m'a permis d'affirmer les différentes connaissances que j'ai pu emmagasiner sur ces deux années avec le langage java et de comprendre l'organisation d'un projet oriente objet et de bien assimiler les différentes structure d'un projet (vue, contrôleur, modèle).Ce projet m'a également permis de jauger la difficulté d'un projet en fenêtre et de mettre en pratique l'utilisation de git Hub et StartUML dans le cadre d'un projet concret.