



Martins Clement, Monin Tom

Rapport java

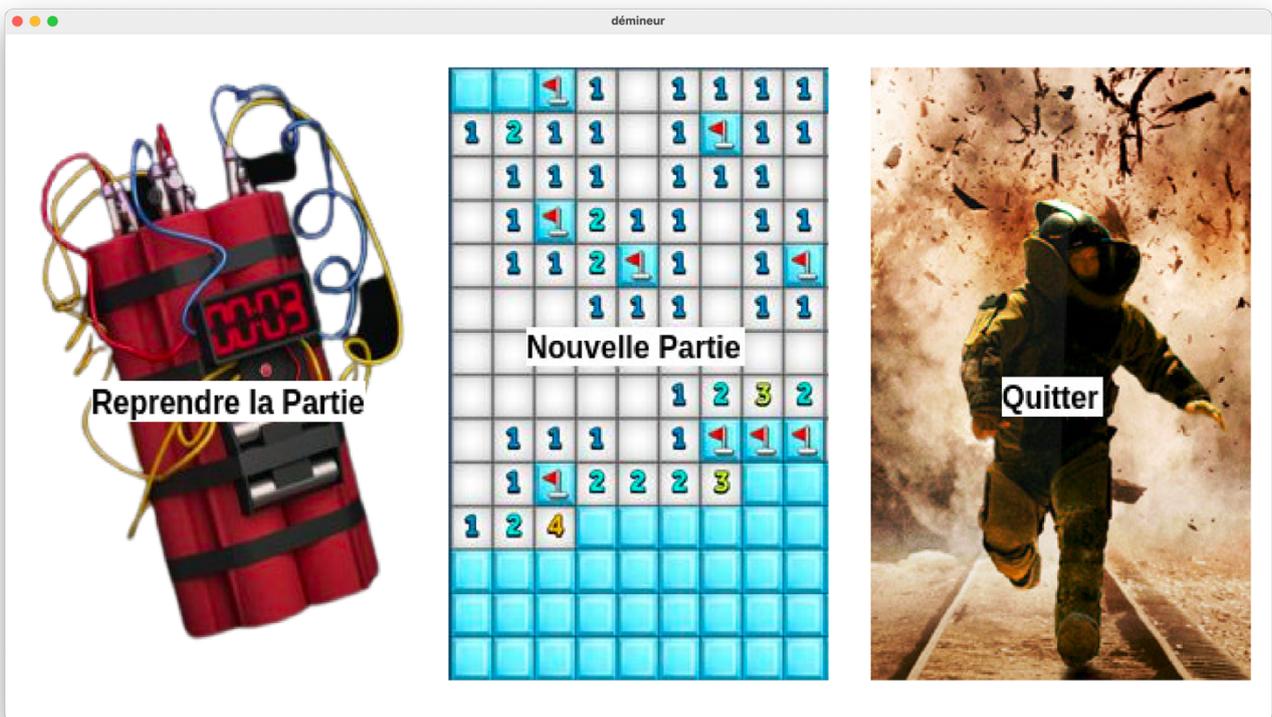
Introduction :

Ici le projet consisté a créé un jeu déjà bien connu qui est le démineur. Les règles du jeu sont assez simples mais ne rendent pas le jeu plus facile pour autant. On se retrouve devant une grille avec plusieurs cases qui peuvent comporter une mine ou non le but étant de trouver toutes les cases adjacentes aux mines sans toucher celle-ci, ce qui demande de la réflexion. Pour nous aider dans ce jeu les case indique le nombre de bombe voisines à celle-ci. Pour ce projet quelques petites options son demandé comme la sauvegarde de la partie du joueur, il peut choisir le nombre de ligne, de colonne ainsi que de bombe. La possibilité pour un joueur de marquer les cases sans les révéler, il a aussi le fais que si une case et révélé mais que celle-ci n'est adjacente à aucune mine alors toute les cases aux

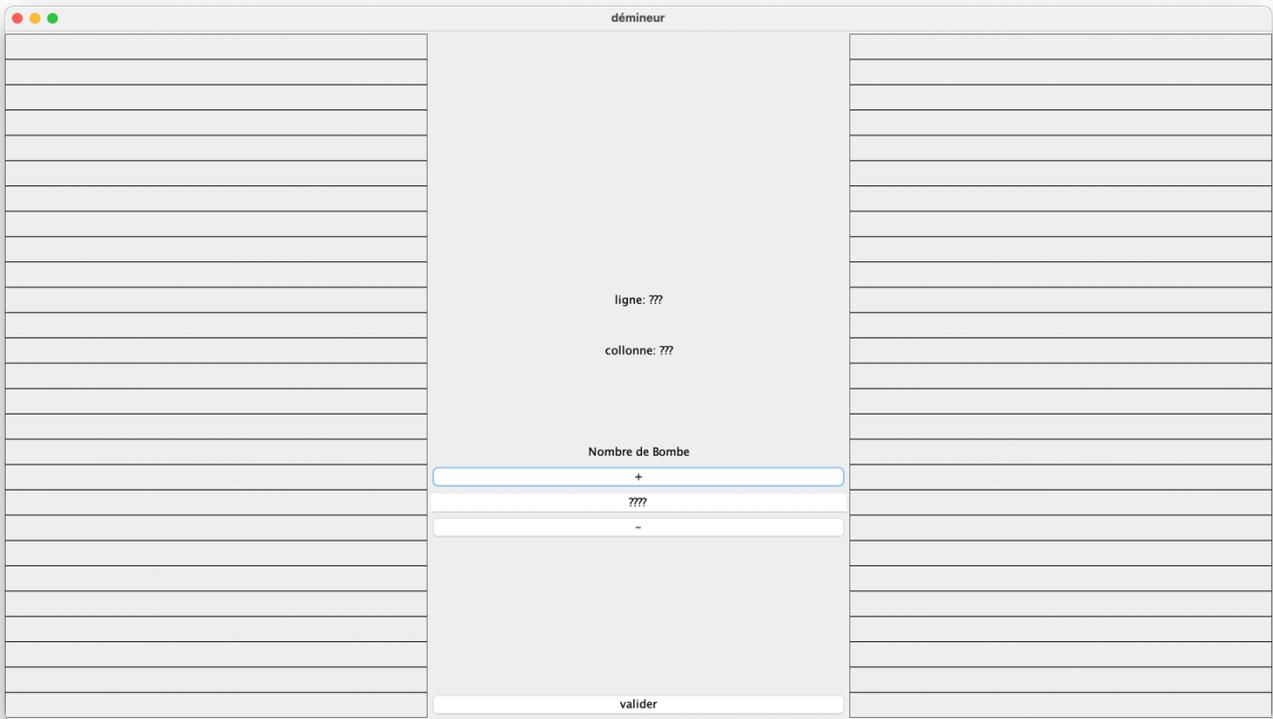
alentours sont révélées. Enfin, la partie est gagnée si toutes les cases non minées sont révélées. Pour jouer le joueur a deux options : clic gauche pour révéler une case et clic droit : pour suspecter celle-ci ou enlever cette suspicion. Les case suspecter ne peuvent plus être révéler.



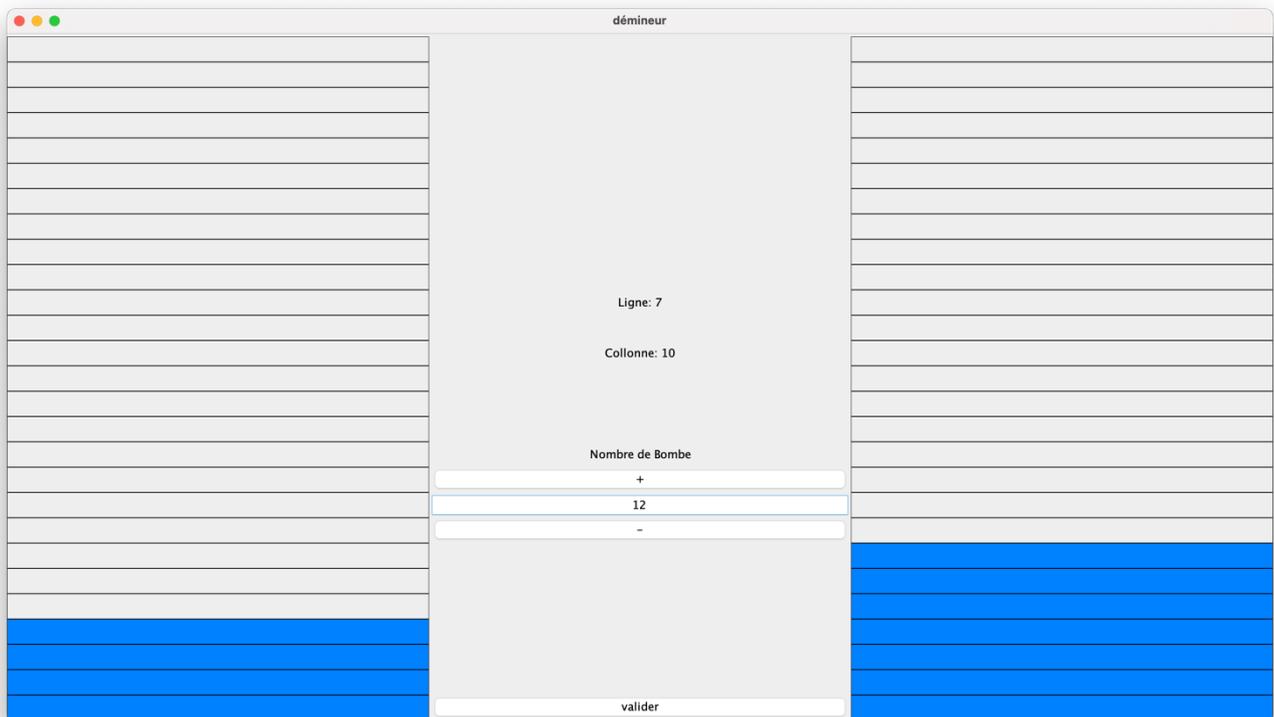
Dans un premier temps nous allons parler du debut du programme quand nous commençons nous pouvons voir cela :



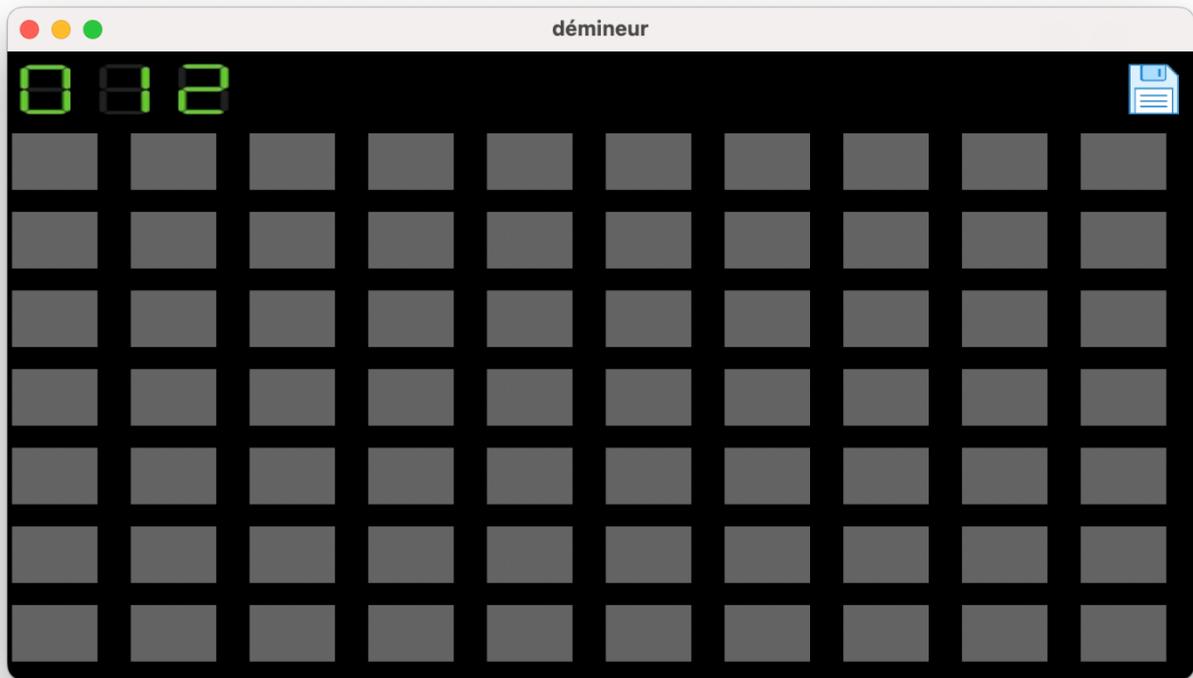
Ceci est la page d'accueil du jeu qui permet de reprendre la partie qui avait été effectuée avant par le joueur ou de rejouer une nouvelle partie ou de quitter tout simplement le programme. Si il n'y a pas de sauvegarde ou que le fichier de sauvegarde comporte un problème « reprendre la partie » affichera juste un message d'erreur et ramènera le joueur sur cette page. Si nous cliquons sur nouvelle partie nous avons une nouvelle page qui s'affiche et qui nous permet de choisir le nombre de bombe, de ligne et de colonne.



Ici nous avons pas mal de fonctionnalités qui sont pour la plupart dédié à la sécurité, et quand nous parlons de sécurité c'est le fait d'éviter que le programme plante avec des valeurs qui ne sont pas adéquate ou avec des action non désirer. Nous avons par exemple, un choix de ligne et de colonne qui est limité, et un nombre de bombes aussi qui est limité. Nous pouvons pas avoir par exemple un nombre de bombes qui est inferieure a 0 ou un nombre de bombe superireure au nombre de case demander. Si une lettre ou un caractère est saisie au lieu d'un nombre compréhensible il ne sera pas non plus possible de lancer la partie.

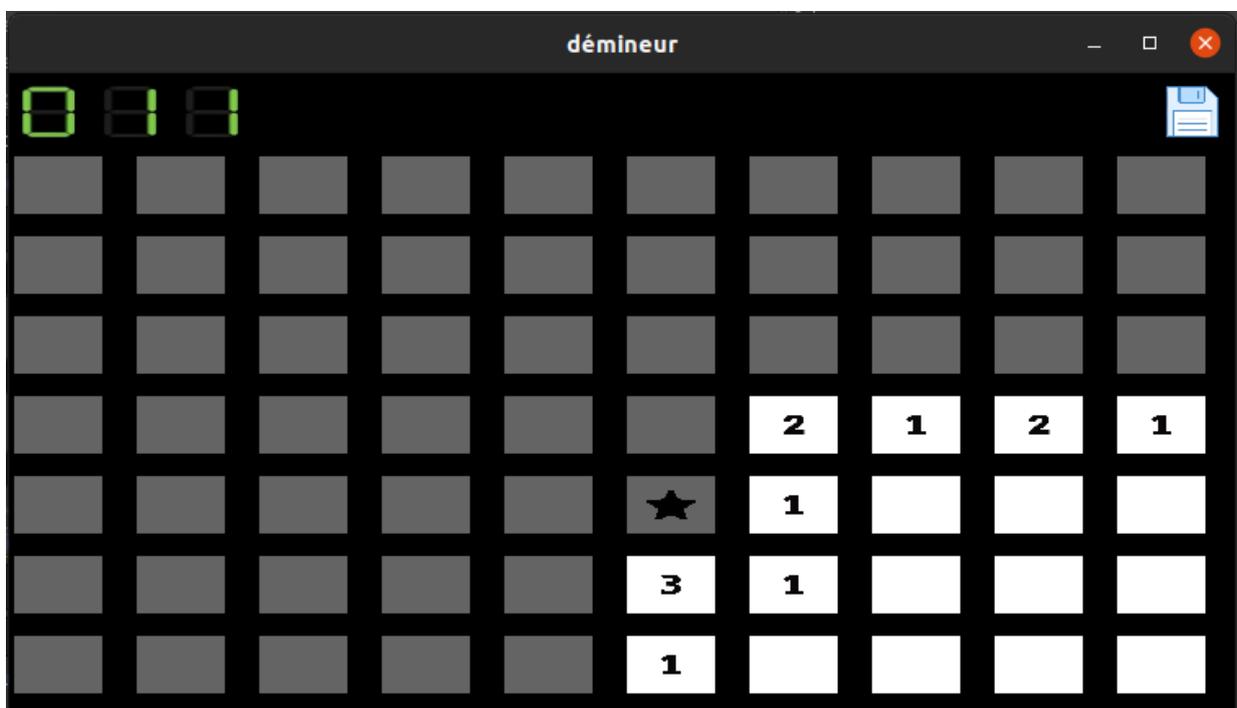


Une fois que lignes et colonnes et que le nombre de bombe et choisis nous pouvons commencer la partie. Alors une nouvelle fenêtre se génère.

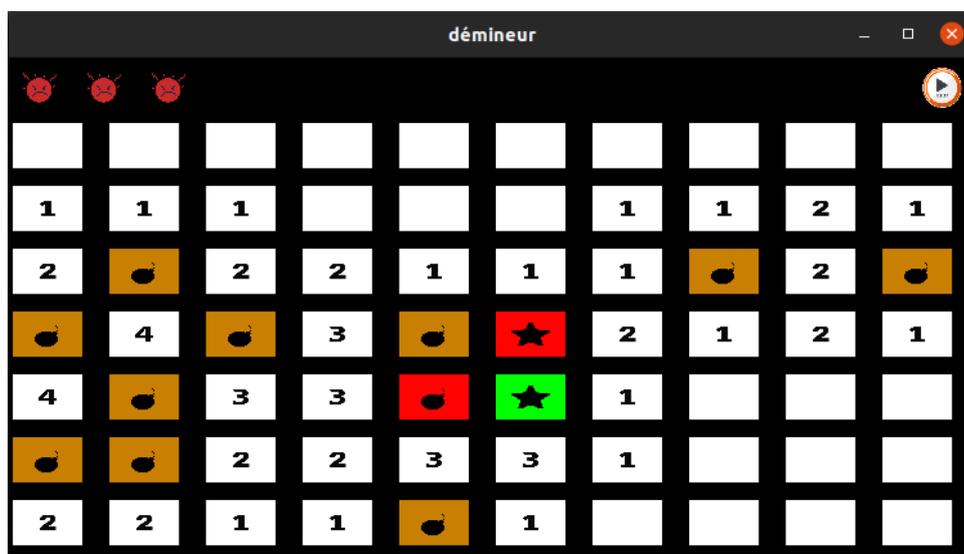


Nous avons maintenant la fenêtre du jeu. On peut apercevoir ici que le nombre de colonnes et de lignes a été respecté il y a bien ici 7 lignes pour 10 colonnes et on peut voir que le compteur en haut à gauche indique 12 qui est le nombre de bombes restante à trouver (le nombre de bombe – le nombre de case suspecter en tant que telle). En haut à droite nous pouvons voir un bouton qui représente une sauvegarde qui permet aux joueurs de sauvegarder sa partie.

Quand les cases sont générées par le programme les bombes sont posées aléatoirement dans la grille, bien sûr les cases numérotées qui indique la présence ou non d'une bombe dans les cases adjacentes sont aussi généré après que les bombes soit posée.



Dans l'image ci-dessus nous pouvons voir qu'une partie a commencé et que plusieurs cases sont visibles, cette fonctionnalité est rendue possible grâce au clic gauche de la souris, par contre si nous utilisons le clic droit, nous pouvons avoir l'étoile qui permet de marquer une case non révélée ce sont donc des marqueurs qui sont que des aides à la réflexion et ne fonctionnent pas différemment que la case soit réellement minée ou pas. Il suffit de cliquer à nouveau avec le clic droit sur la case muni déjà d'une étoile pour voir apparaître le « ? » qui représente un doute et non une certitude sur le fait que la case soit une bombe.



Nous avons ici aussi pas mal de petites fonctionnalités. Une fois que le joueur joue et touche une bombe il perd alors toutes les cases sont rendus visible. La bombe qui a fait perdre le joueur est mis sur fond rouge, toutes les autres sont sur fond orange, ensuite pour les marqueurs quand-t-ils sont évalués à faux ils sont marqués aussi d'un fond rouge en revanche quand-t-ils sont évalués à vrais ils sont marqués d'un fond vert. Pour la dernière fonctionnalité de cette image en haut à droite nous avons un bouton qui nous permet de tout simplement revenir à la toute première page pour recommencer une partie. L'ancien nombre de bombe restant est quant à lui remplacé par des images représentant la victoire ou la défaite du joueur.



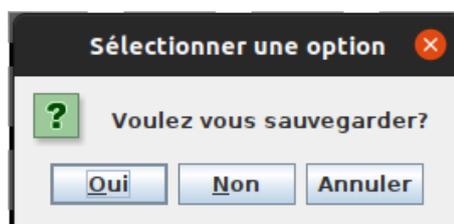
Si le joueur a gagné alors nous calculons son score de cette manière :

$$(nb \text{ de bombe} * 10\ 000 / nb \text{ de case}) / nb \text{ de minute passer à le résoudre}$$

Donc plus le joueur a choisi un nombre de bombe élevé par rapport à son nombre de case et plus il a été rapide mieux sera son score. Le score est ensuite affiché au joueur par un pop-up.

Le principe de sauvegarde dans notre jeu :

Tout d'abord avant de sauvegarder ou non on affiche au joueur un pop-up avec une question et 3 choix possible :



Si le joueur choisit annuler il peut alors continuer sa partie s'il choisit non le jeu est quitter sans sauvegarder. Si oui la partie est sauvegarder de cette manière :

Pour sauvegarder nous avons tout simplement adopté une méthode d'écriture, lecture dans un fichier. Quand le joueur demande une sauvegarde de la partie sur laquelle il joue on commence dans un premier temps à sauvegarder les lignes et colonnes pris par le joueur pour indiquer lors de la reprise de la partie les dimensions de celle-ci. Ensuite on parcourt le tableau de case du jeu en sérialisant chaque objet à leur tour pour stocker l'état exact de leurs attributs lors de la sauvegarde. Pour finir, on stocke le nombre de bombe puis le score à l'instant de la sauvegarde ainsi que le nombre de minute passer lors de ce jeu à essayer de résoudre le démineur. Une fonctionnalité aussi qui a été rajouté et que si la partie est finie le joueur ne peut pas sauvegarder.

Une fois que la partie est sauvegarder le fichier de sauvegarde est rempli de donnée. Maintenant on lit juste les données écrit dans le fichier de sauvegarde, donc on reprend lignes et colonnes. Puis, on crée le tableau avec les bonnes dimensions et on le parcourt pour remettre les cases dans l'état de la sauvegarde. Nous réinscrivons aussi le nombre de bombe est le score à l'instant de la sauvegarde et initialisons le nombre de minute.

Principe de l'algorithme qui permet de révéler plusieurs cases :

C'est algorithme est relativement simple à comprendre c'est-à-dire qu'une fois que le joueur clique, si c'est une case n'ayant aucune bombe comme voisin alors on va faire appel à cette fonction récursive :

Elle prend en argument la ligne et la colonne de celle-ci, elle rend ensuite celle-ci visible et appel une fonction ayant pour but de révéler toutes les cases avec au moins une bombe adjacente autour de la case choisit. La fonction regarde ensuite autour de la case s'il n'y a pas d'autre case ayant elle aussi 0 bombe voisines pas encore visible. Dans ce cas elle appellera elle-même sa fonction avec comme ligne et colonne les coordonnées de celle-ci. Toutes les cases ayant 0 bombe voisines ainsi que celles autour d'eux ayant plus d'une bombe voisine seront révéler.

Schéma Star UML :

Schéma des relations sans les détails des classes :

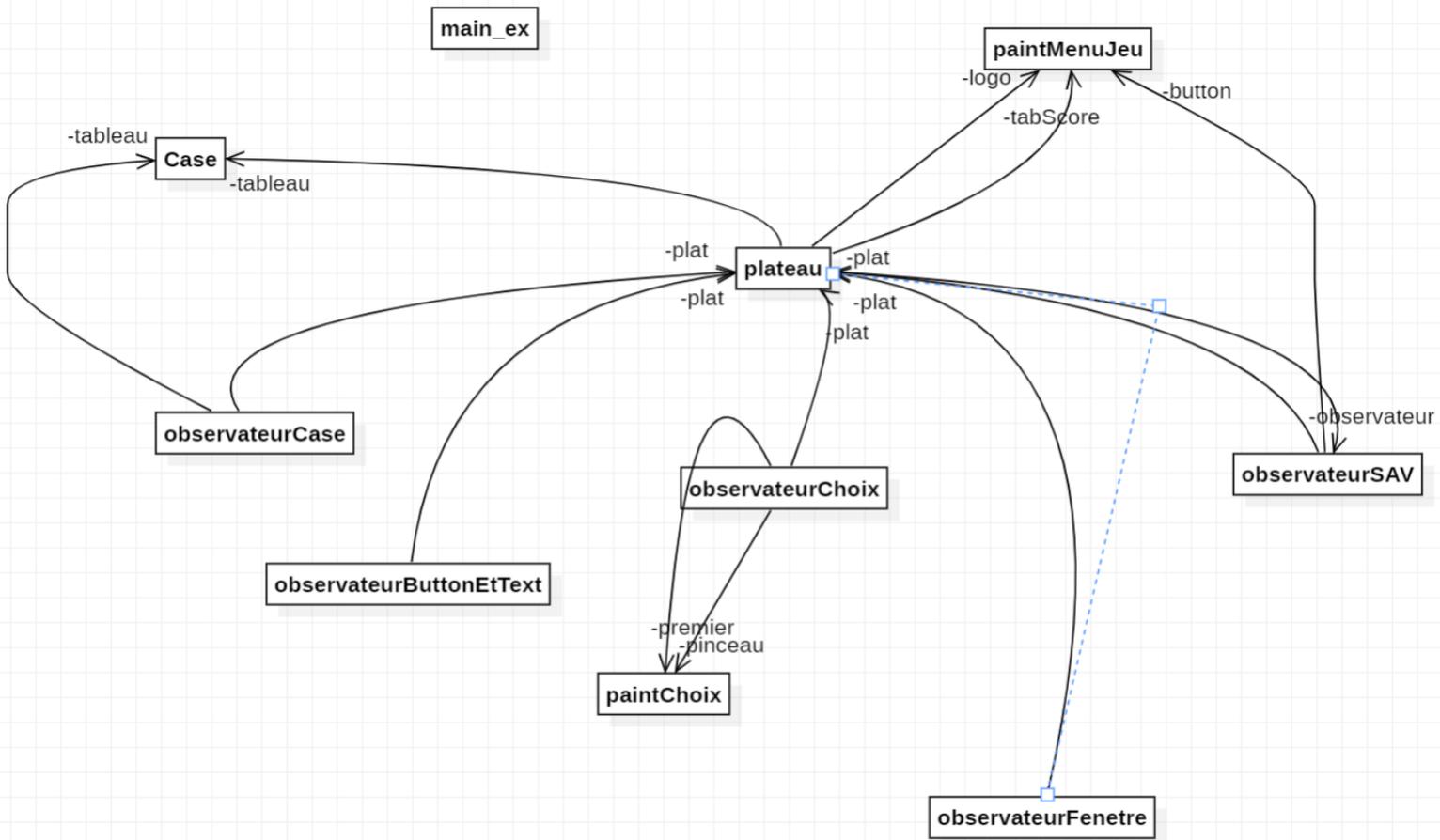
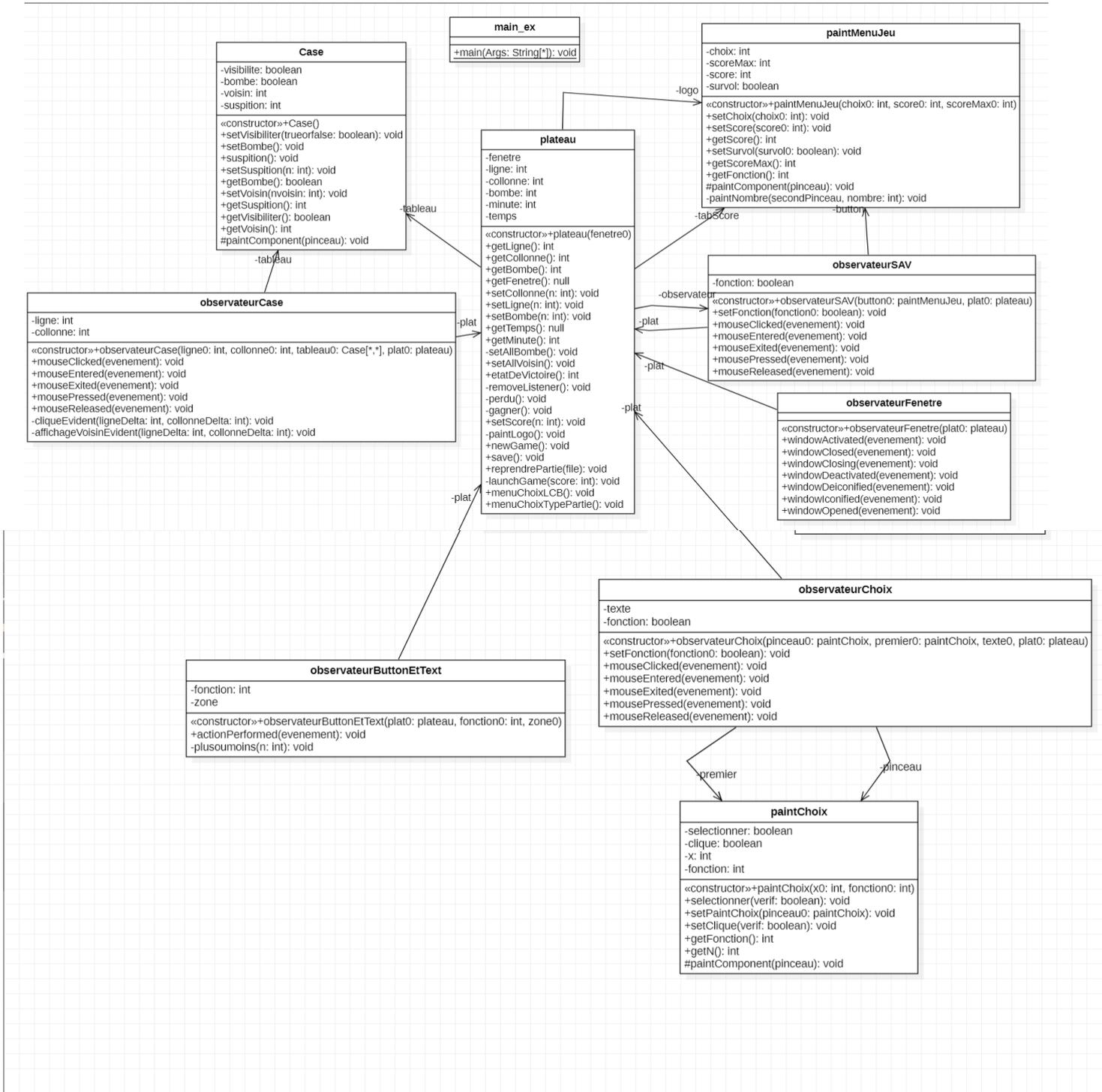


Schéma avec détails des classes :



Conclusion personnelle :

Tom :

J'ai bien aimé ce projet parce-que même si le JAVA est un langage pas facile il reste un langage que j'apprécie c'est pourquoi j'ai apprécié coder le démineur. Je trouve que pour ce projet nous nous sommes bien organisés avec mon binôme pour parvenir à ce résultat. Les tâches ont été définis précisément, nous avons fini chaque tâche dans les temps ce qui nous a permis de finir dans le temps imparti. Je tire de ce projet encore une nouvelle expérience en matière de développement.

Clément :

Personnellement j'ai adoré ce projet, je me suis senti à l'aise en java grâce à ce projet et j'ai adoré mettre en œuvre mes idées et l'avancer très vite du projet. Petite complication pour essayer de faire un démineur beau visuellement malgré de nombreux essaies je ne le trouve toujours pas très beau dans certaines proportions de ligne et de colonne ou il aurait avec plus de temps adapter le format des Case et du Menu. A part ça je suis content de notre travail d'équipe qui c'est très bien dérouler.