

Rapport D'analyse : Jeu du démineur
Brice Labbé
v2 – 04/11/2024

L'objectif est de créer un jeu de démineur, potentiellement muni d'une interface graphique en python, en suivant un paradigme de programmation orienté-objet. Ce document va d'abord détailler la première analyse réalisée avant le début de la programmation puis donner une mise à jour à la fin de la programmation.

Tous les diagrammes peuvent être trouvés sur le dépôt git dans le dossier « diagram »
Certains ne sont pas très lisibles sur ce document.

Analyse Initiale : (03/10/2024)

On ne considère initialement pas d'interface graphique (GUI), qui pourra normalement être greffé aux code avec l'ajout d'une classe dédié car toutes les fonctions de gestion du jeu existeront déjà.

On omet aussi dans le diagramme la classe « Main » qui n'existe que pour initialiser le jeu.

Diagramme de Classe :

On part sur un modèle très simple basé sur trois classes :

- Minesweeper : Lors du démarrage du jeu, est appelé par la classe Main. Celle-ci gère l'initialisation du jeu, son déroulement, ainsi que la victoire ou défaite du joueur.

- Grille : Classe responsable de la grille du jeu. Elle comprend les fonctions nécessaires au placement des mines en fonction de la difficulté ainsi que le calcul des « numéros » des cases (qui correspondent au nombres de mines adjacentes à une case). Ces numéros sont stockés et ne seront pas recalculés.

- Case : Classe représentant une case unique de la grille. Elle peut être miné, révélé, ou marqué. Cette case contient les fonctions permettant de changer le statut des cases.

Enfin, un tableau de Case est une Grille qui elle-même est une partie du Minesweeper.

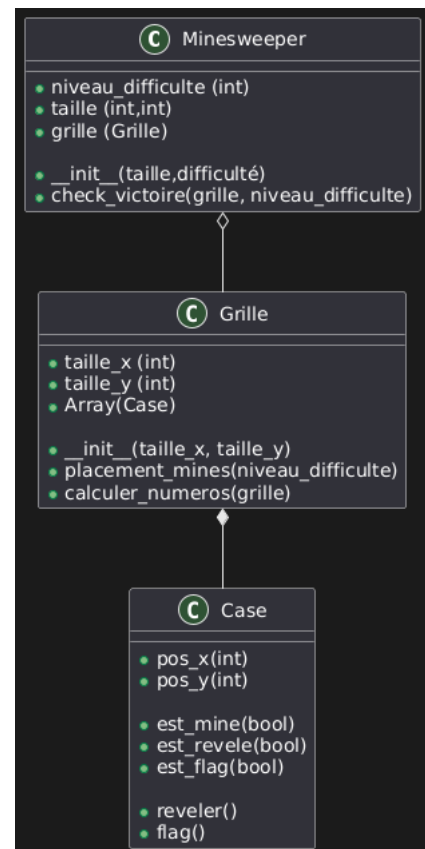


Diagramme de Cas d'Utilisation :

Pour les cas d'utilisation, on n'as qu'un seul utilisateur : le joueur.

Celui-ci peut commencer un jeu, puis révéler ou marquer des cases, ce qui cause soit sa victoire, soit sa défaite.

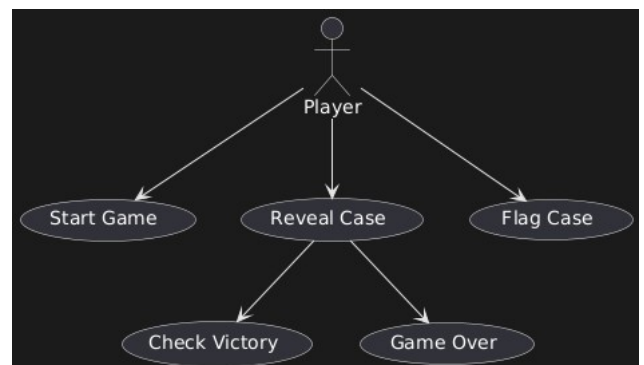
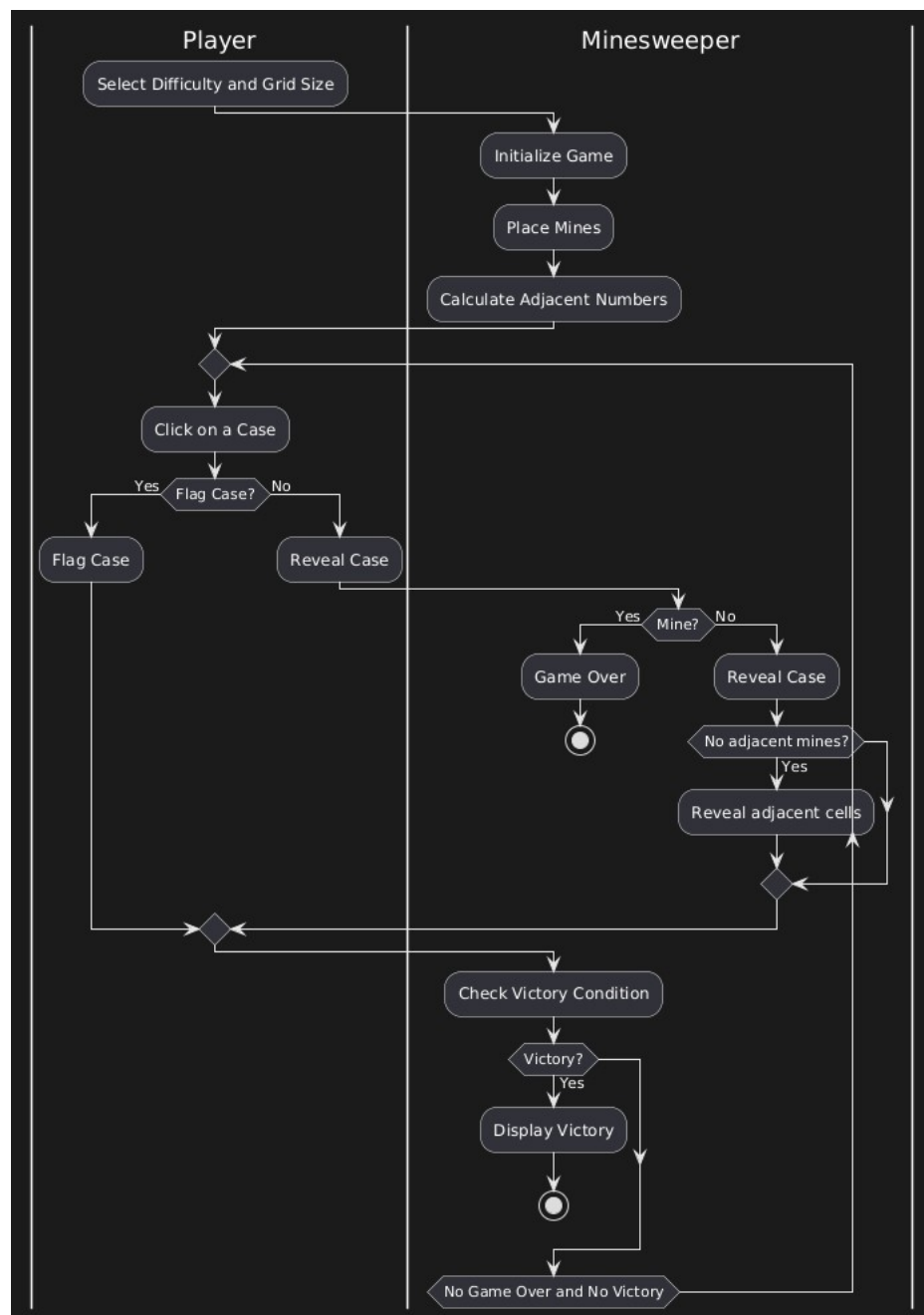


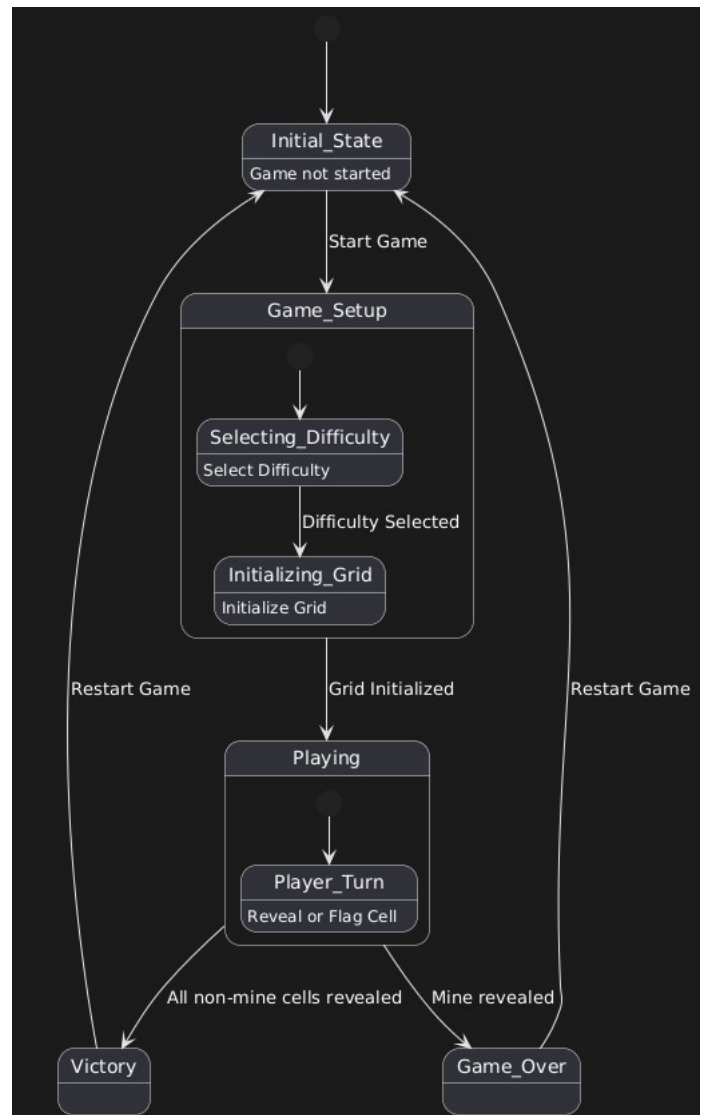
Diagramme d'Activité :



Dans ce diagramme, on distingue deux acteurs qui agisse sur l'État (/le déroulement) de la partie. Ces deux acteurs sont le joueur et le jeu. Le joueur va agir en commençant le jeu, en révélant ou en marquant certaines cases. Cela fait des appels au second acteur, le jeu qui va répondre en modifiant l'état du jeu et en renvoyant au joueur le jeu mis à jour et en vérifiant les conditions de victoire ou de défaite.

Diagramme d'État :

Ce diagramme décrit les états principaux du jeu lors de la création et du déroulement d'une partie.



Mise à jour de l'Analyse : (04/11/2024)

Lors du développement du code, il est devenu apparent que beaucoup de fonctions étaient manquantes pour assurer le bon déroulement du jeu. De plus, pour inclure une interface graphique, on va, comme prévu, ajouter une classe gérant le GUI (Graphical User Interface). Cette partie présente, à l'aide de diagrammes mis à jour, la structure du code ainsi que les différentes étapes de son exécution.

Diagramme de Classe :

Comme précédemment, on omet la classe Main.

La structure est très similaire à celle prédite par l'analyse initiale. Le principal changement est l'ajout de la classe MinesweeperGUI liée à la classe Minesweeper.

Ce faisant, la classe Main appelle non plus la classe Minesweeper mais la classe MinesweeperGUI qui ensuite appelle Minesweeper.

Certaines fonctions ont été ajoutées, notamment toutes celles qui permettent de gérer l'interface graphique. On peut aussi noter la fonction `reveal_adjacent_cells` qui permet de récursivement révéler les cases adjacentes aux cases ayant un « numéro » égal à 0. Cette fonction appelle un autre ajout, la fonction `calc_adj` de Grille qui permet de calculer le « numéro » d'une case.

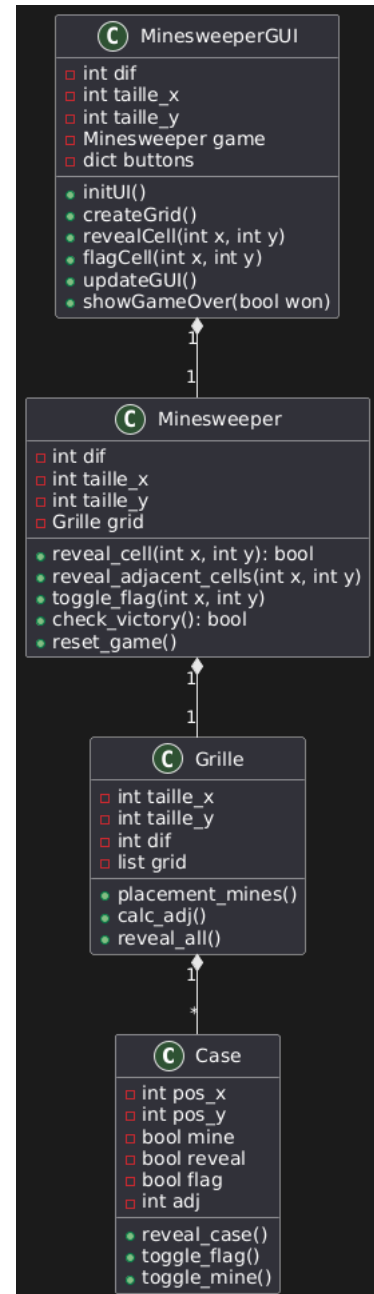


Diagramme de Cas d'Utilisation :

Ce diagramme reste globalement inchangé par rapport à sa première version puisque les actions possibles du joueur n'ont pas évoluées.

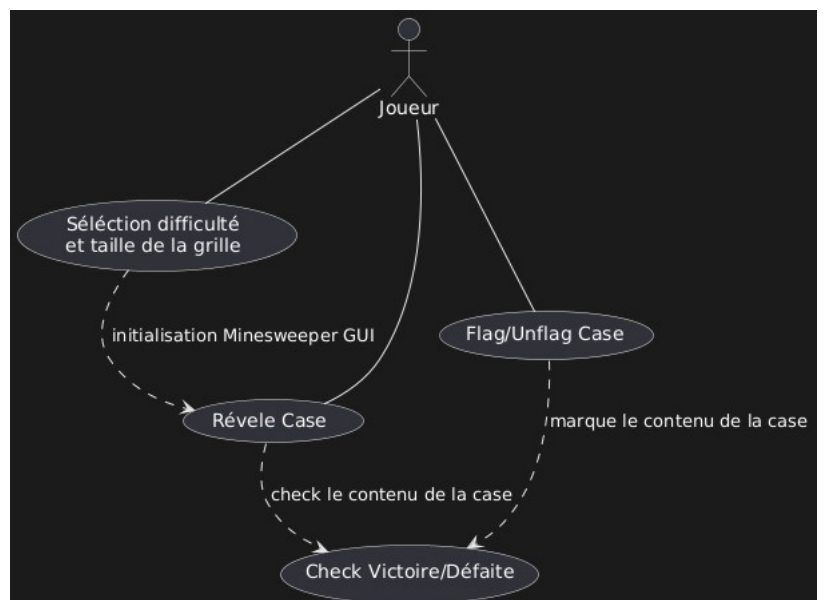
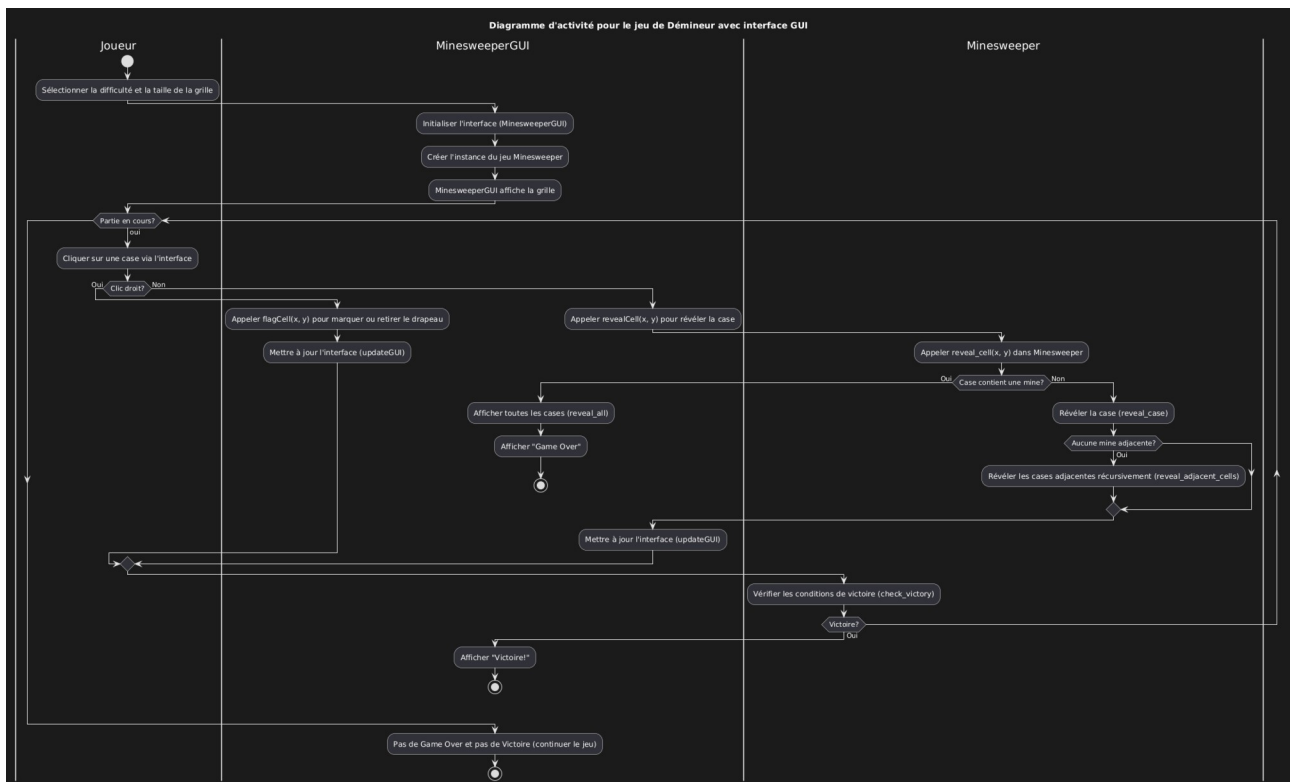


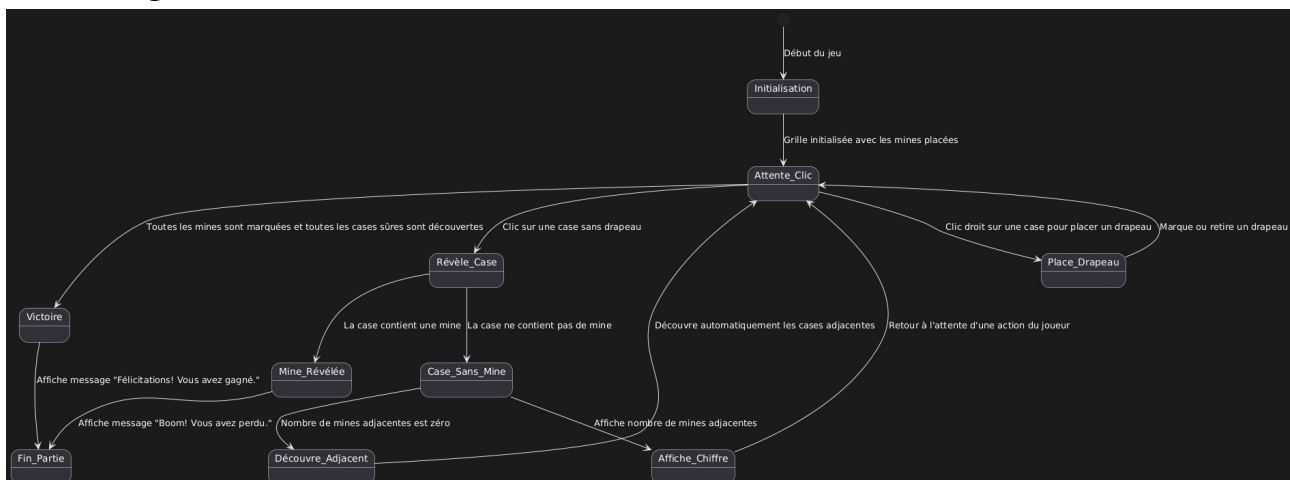
Diagramme d'Activités :



On reprends la séparation en plusieurs acteurs de la première version et on en rajoute un : le GUI.

Celui-ci va interpréter les actions et réponses du joueur et du jeu et afficher une version mise à jour du jeu.

Diagramme d'État :



Le diagramme d'État initial est toujours correct, la version actuelle est plus us ajout qui décrit avec plus de détails les états liés au action de l'utilisateur et au déroulement de la partie.