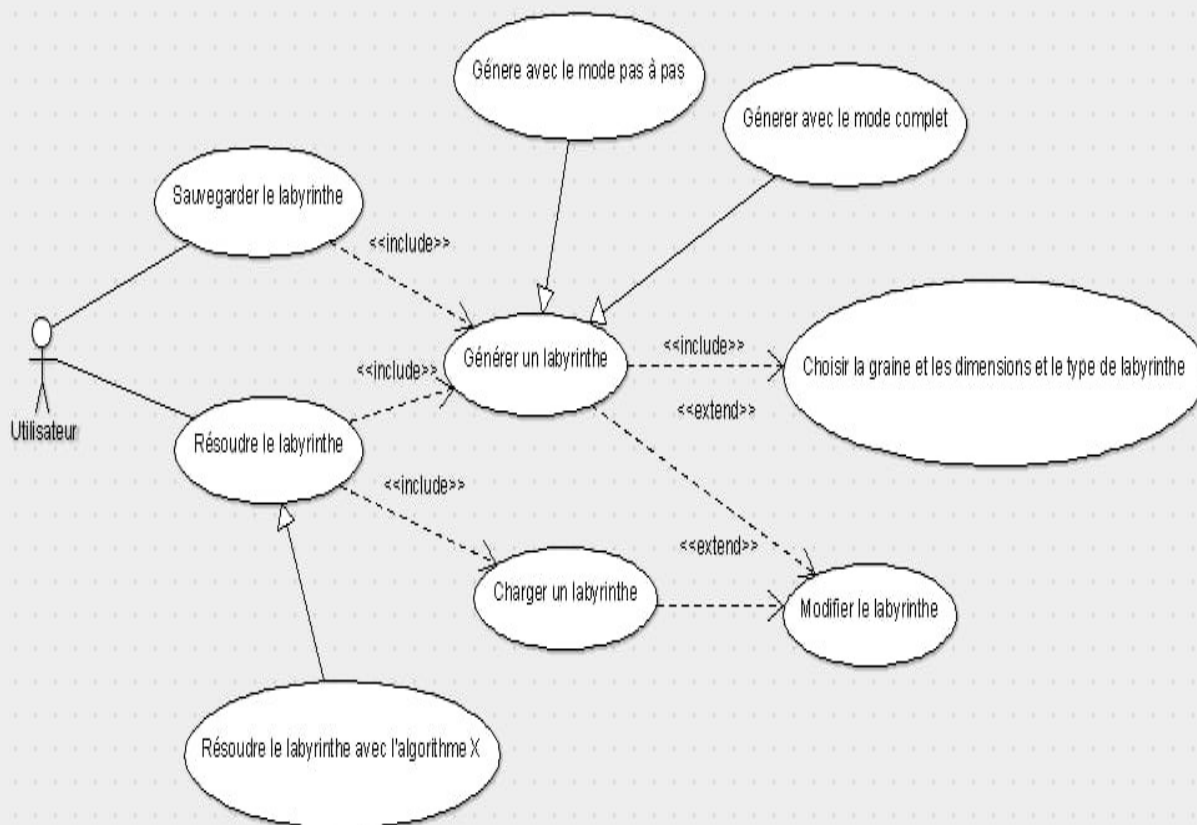


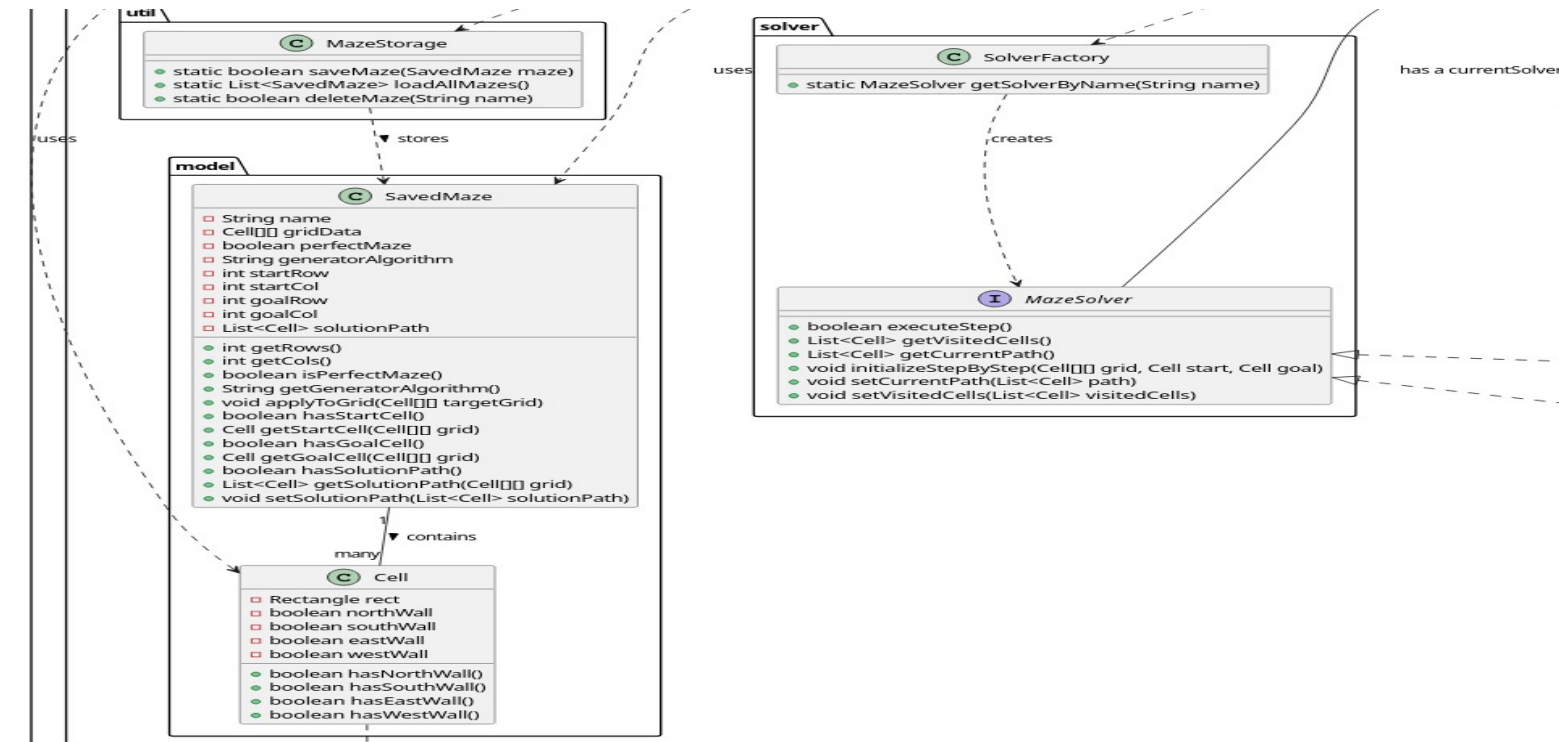
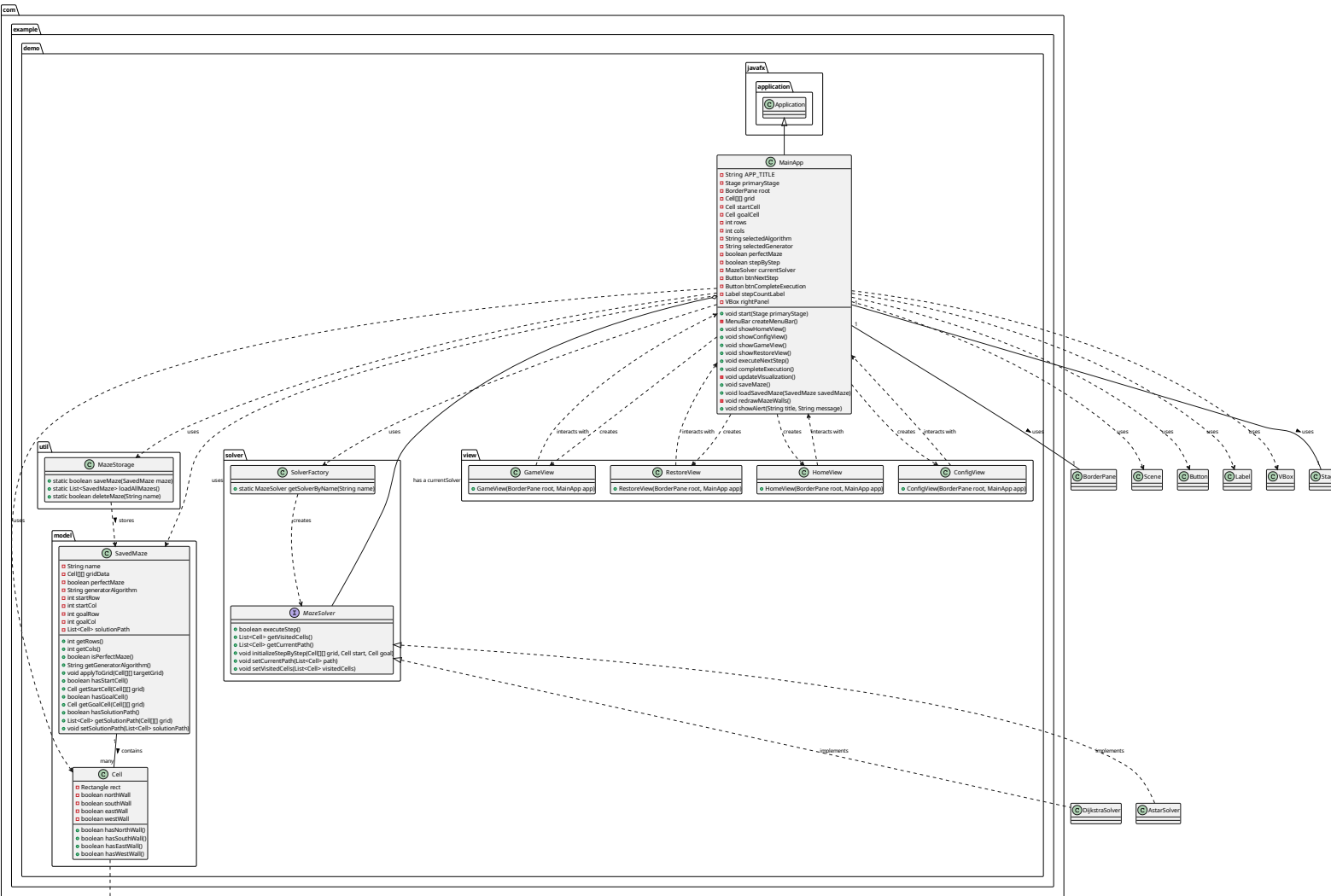
***Projet génie logiciel semestre 2:***

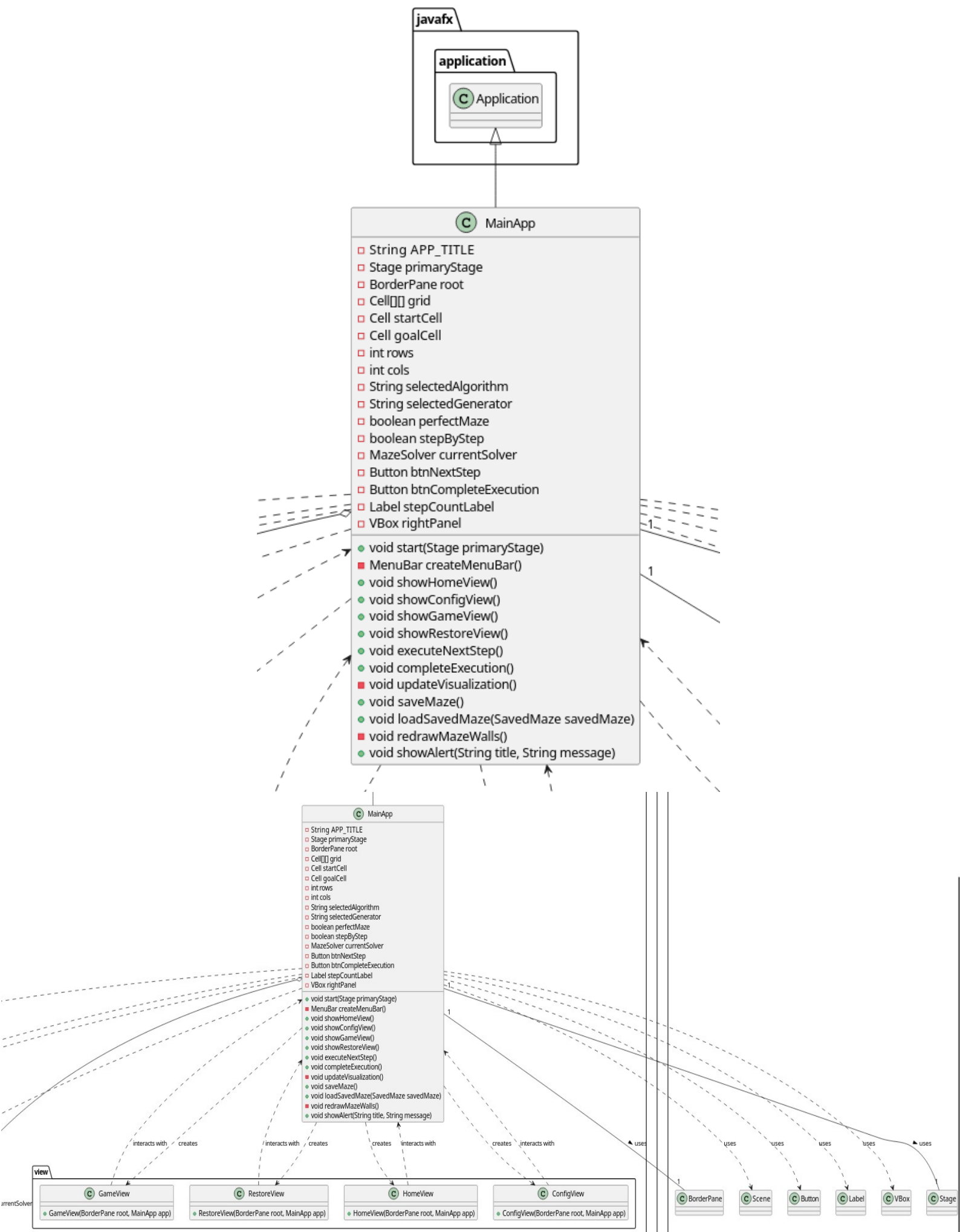
***Projet CYnapse***

## Diagramme de cas d'utilisation



# Diagramme de classe





# Maquette

 YBAY Maze


Menu principal

## Bienvenue sur YBAY Maze

Générateur et résolveur de labyrinthes

Générer un labyrinthe

Restaurer

 YBAY Maze

Menu principal

## Configuration du labyrinthe

Mode :

Mode Complet◀

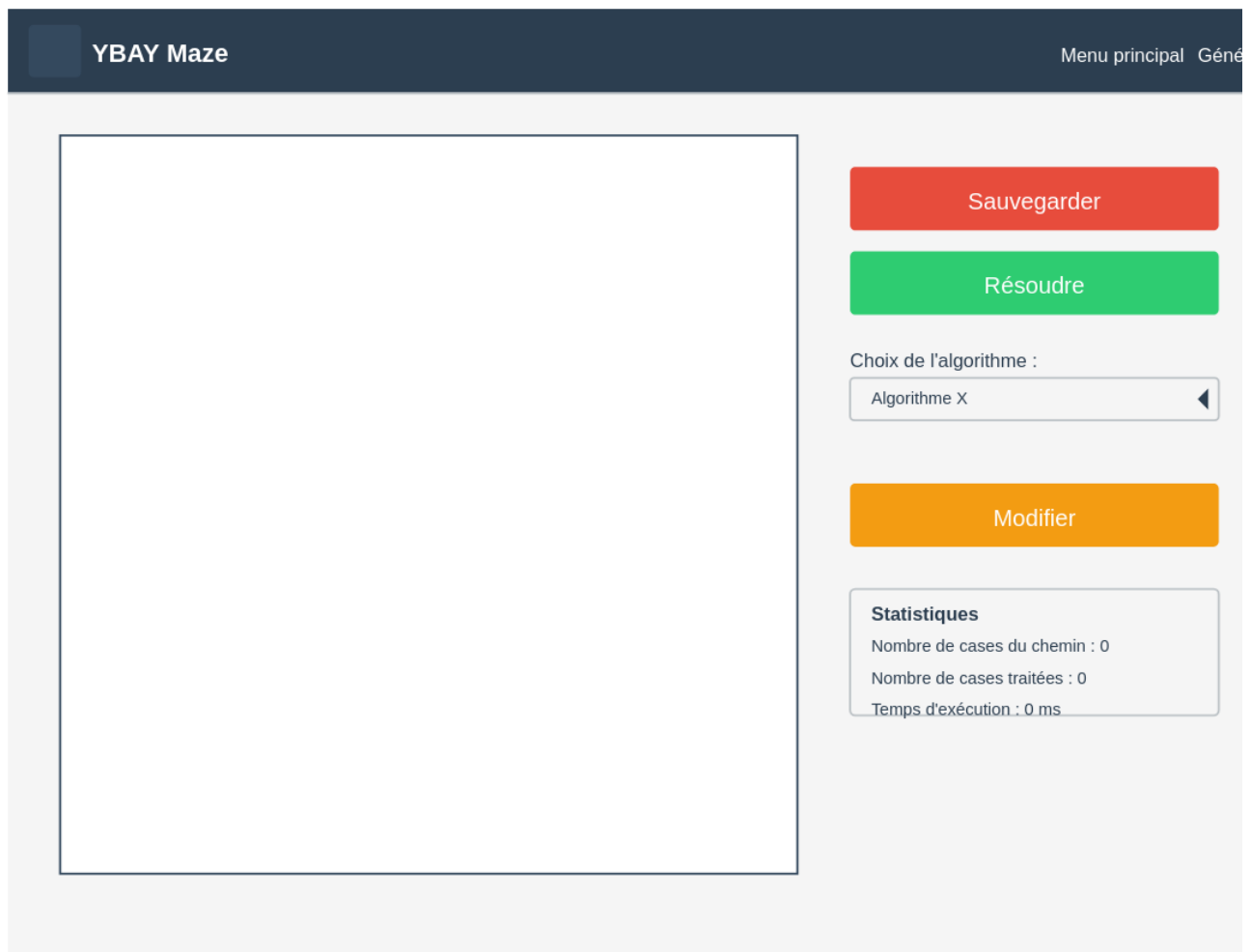
Mode Complet

Mode Pas à Pas

Nombre de cases horizontales :(Max 50)

Nombre de cases verticales :(Max 35)

Générer



## **Planning de réalisation initiale:**

**Semaine1** : Maquette + Diagramme de classe+ Diagramme de cas d'utilisation+Choix techniques

**Semaine2 (après les partiels)** : Tâches du backlog+ Projet et classes principaux prêts+avancement sur les écrans +Fonctionnalités prêtes : génération du labyrinthe avec Prim

**Semaine3** : Algorithmes de résolutions prêts: BFS,DFS et Dijkstra + Début de la programmation de A\*

**Semaine4** : Algo A\* prêts + Sauvegarde et Restauration+ Modification des murs + Mode pas à pas pour tous les algos

## **Planning de réalisation réelle:**

**Semaine1** : Maquette + Diagramme de classe + Diagramme de cas d'utilisation+ Choix techniques

**Semaine2** : Projet et classes principaux prêts + avancement sur les écrans + Fonctionnalités prêtes : génération du labyrinthe (parfaits seulement) avec Prim

**Semaine3** : Algo de résolutions BFS prêts seulement en mode complet et pour les labyrinthes parfaits + Début de DFS, Dijkstra et A\*

**Semaine4** : Algorithmes de résolutions prêts: BFS,DFS, Dijkstra et A\* en mode pas à pas ( pas automatique) et complet pour les labyrinthes parfaits et imparfaits + Sauvegarde et Restauration partielle

## **Organisation d'équipe**

**Makiese Bertony:** Diag de classe, Cell.java+ Maze.java, Algo de résolution DFS

**Rayanne El Saighi:** Diag de use case, les écrans, algos de génération avec Prim, Algo de résolution A\*, avancement sur la fonctionnalité de modifications des murs

**Khadjiev Amine:** Maquette, Algo de résolution BFS, Sauvegarde et Restauration

**Youssef El Haiti:** Diag de classe, les écrans, Algo de résolution Dijkstra, Sauvegarde et Restauration, Algos de génération avec Kruskal

## **Limites Fonctionnelles:**

### **Mode pas à pas non automatisé pour la résolution**

Bien que les algorithmes de résolution (BFS, DFS, Dijkstra, A\*) soient tous disponibles en mode pas à pas, ce mode n'est pas encore entièrement automatisé. L'utilisateur doit manuellement passer à l'étape suivante, car le réglage automatique de la vitesse de visualisation n'a pas été intégré. Il en résulte une interaction plus laborieuse et moins fluide pour l'utilisateur.

### **Mode pas à pas non disponible pour la génération**

Contrairement à la résolution, la génération du labyrinthe ne dispose pas de mode pas à pas. Seul le mode complet est actuellement implémenté, ce qui signifie que l'utilisateur ne peut pas observer la création progressive du labyrinthe. L'ensemble de la structure est généré en une seule étape, sans visualisation intermédiaire, ce qui limite la compréhension du fonctionnement de l'algorithme de génération.

### **Absence de modification du labyrinthe après sa génération**

L'une des fonctionnalités clés du cahier des charges – la possibilité de modifier localement la topologie du labyrinthe (ajout ou suppression de murs) – n'a pas été implémentée. Une fois le labyrinthe généré, il n'est donc pas possible de le modifier dynamiquement ni de relancer une résolution sur une structure modifiée.