

Ransan
Randrianantoandro
Saussard
Lagauche

Bilan projet ACL

Fonctionnalités du jeux:

- Demande le niveau auquelle le joueur veut jouer (dans la console)
- Le héros peut se déplacer sur toutes les cases qui ne sont pas un mur
- Le héros peut donner des coup au monstre s'ils sont sur la même case ou s'ils ont une case de différence
- Les monstres se déplacent de manière aléatoire
- Le fantôme passe à travers les murs
- Le héros peut prendre des potions qui se vide et ne sont plus utilisable, cela lui rajoute 50 point de vie
- Le héros peut tomber dans un trou et meurt
- Le héros perd 100 de vie s'il marche sur un pic
- Lorsque le Héros arrive sur la case fin une fenêtre de fin s'ouvre
- Lorsque le héros meurt une fenêtre de "Game over" apparaît
- Lorsqu'un monstre est tué son visuel disparais
- Dans la console, il est possible de suivre si le héro a une quelconque interaction avec des objets ou avec un monstre

Bilan des sprints:

Sprint 1 :

- Discussion des fonctionnalité du jeux	Tout le monde
- Prévisualisation des classes par le groupe	
-Création des classes attaques, main vérification	Romane
- Crédit de la classe LabyrintheGameGui,main de vérification	Apolline
- Crédit des classes Personnage, Héro et monstre	Guilian
-Création de la classe Labyrinthe	Thomas

Sprint 2 :

- Au niveau de l'interface graphique, le labyrinthe (niveau facile) s'affiche ainsi que le héros.	
- Le héros se déplace sur l'interface graphique lorsque les touches fléchées sont utilisées.	Apolline
- L'interface graphique utilise la classe Labyrinthe pour récupérer le labyrinthe.	
- Le héros se déplace sur l'interface graphique lorsque les touches fléchées sont utilisées.	Guilian
- Les monstres se déplacent de manière aléatoire.	
- Dans la classe attaque, le héros meurt au contact des monstres, peut donner un coup d'épée en appuyant sur la touche entrée, il peut entrer en contact et donner un coup d'épée s'ils sont sur la même case.	
- Main commun entre classe personnage et attaque.	
- Des objets spéciaux ont été codés : pic, trou et potions.	Romane
- Création de la classe spéciale avec l'utilisation des objets spéciaux et interaction entre les objets spéciaux et le héros	
- La classe labyrinthe peut charger des labyrinthes issus de fichiers txt différents selon le niveau choisi.	Thomas

Sprint 3 :

-- Le labyrinthe affiche les objets spéciaux ainsi que le monstre	
- L'utilisateur peut choisir le niveau de labyrinthe qu'il veut jouer.	
- Réalisation de test Junit (vérifier que le héros ne peut pas aller sur des briques ou en dehors du labyrinthe + le héros meurt en cas de contact avec le monstre + le héros ne peut pas avoir de PV négatifs).	Apolline
- -Rectification de l'erreur après test : pv négatif manque d'une condition	
- Crédit des classes Monstre_normal et Fantômes avec des méthodes différentes	Guilian
- Transformation de LabyrintheGameGui en manager qui contrôle le héros et les interactions	Romane
- Main commun à toutes les classes	
- Le labyrinthe possède des cases pour les objets spéciaux	Thomas

Sprint 4:

Arrêt du jeu lorsque le héros meurt ou lorsqu'il gagne (arrive à la case de Fin) et si possible afficher une fenêtre à message	Apolline et Romane
- relier la position des objets spéciaux dans le labyrinthe à la classe objets spéciaux pour que le héros puissent interagir avec les objets	Thomas et Romane
-Disparition des monstres de l'interface graphiques lors de la mort de l'un d'eux	Romane
- faire bouger les monstres (normaux et fantôme) sur le labyrinthe	Guillian et Apolline
réaliser une partie jouable du début à la fin	Tout le monde

Diagrammes de classes :

Diagramme 1:

Pour le sprint 1, nous n'avions pas encore créé beaucoup de classes, il est donc peu fourni.

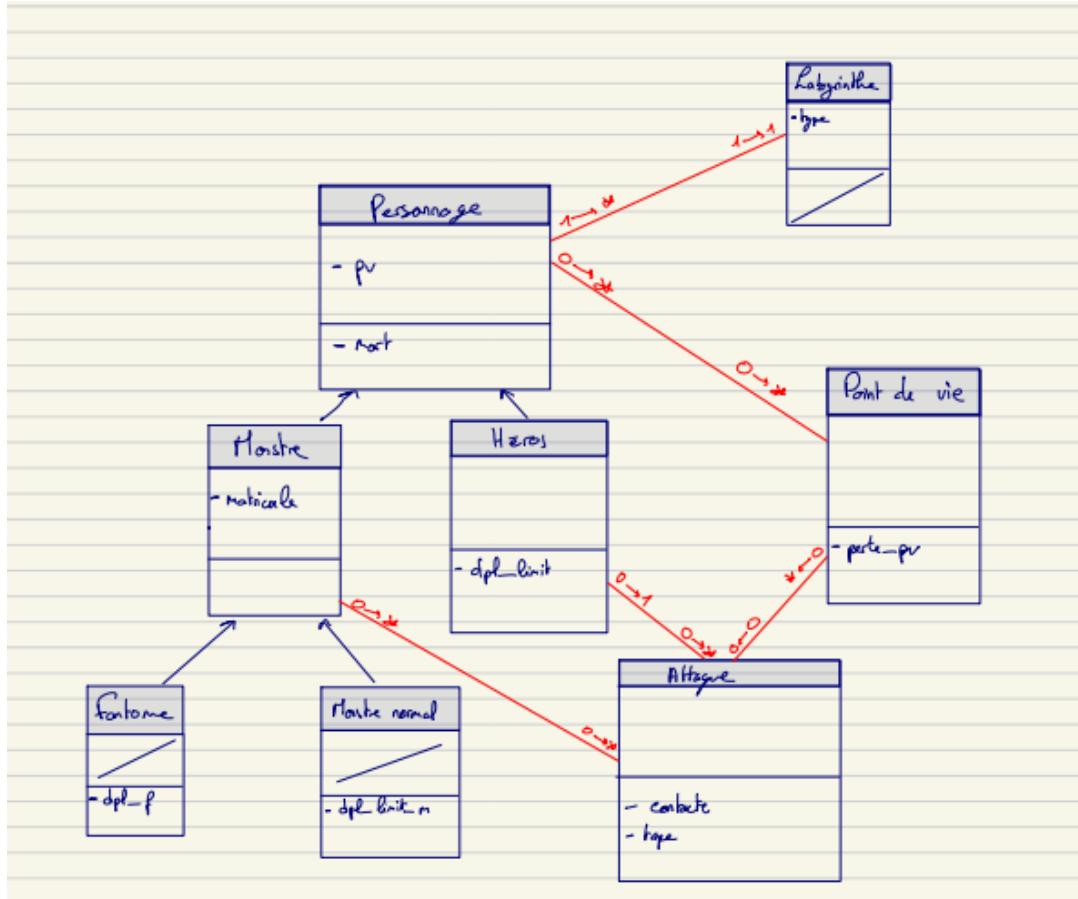


Diagramme 2 :

Pour le sprint 2, nous avons tous commencé à faire nos classes qui apparaissent donc sur le diagramme. Cependant, comme elles n'étaient pas toutes finies, certaines sont vides.

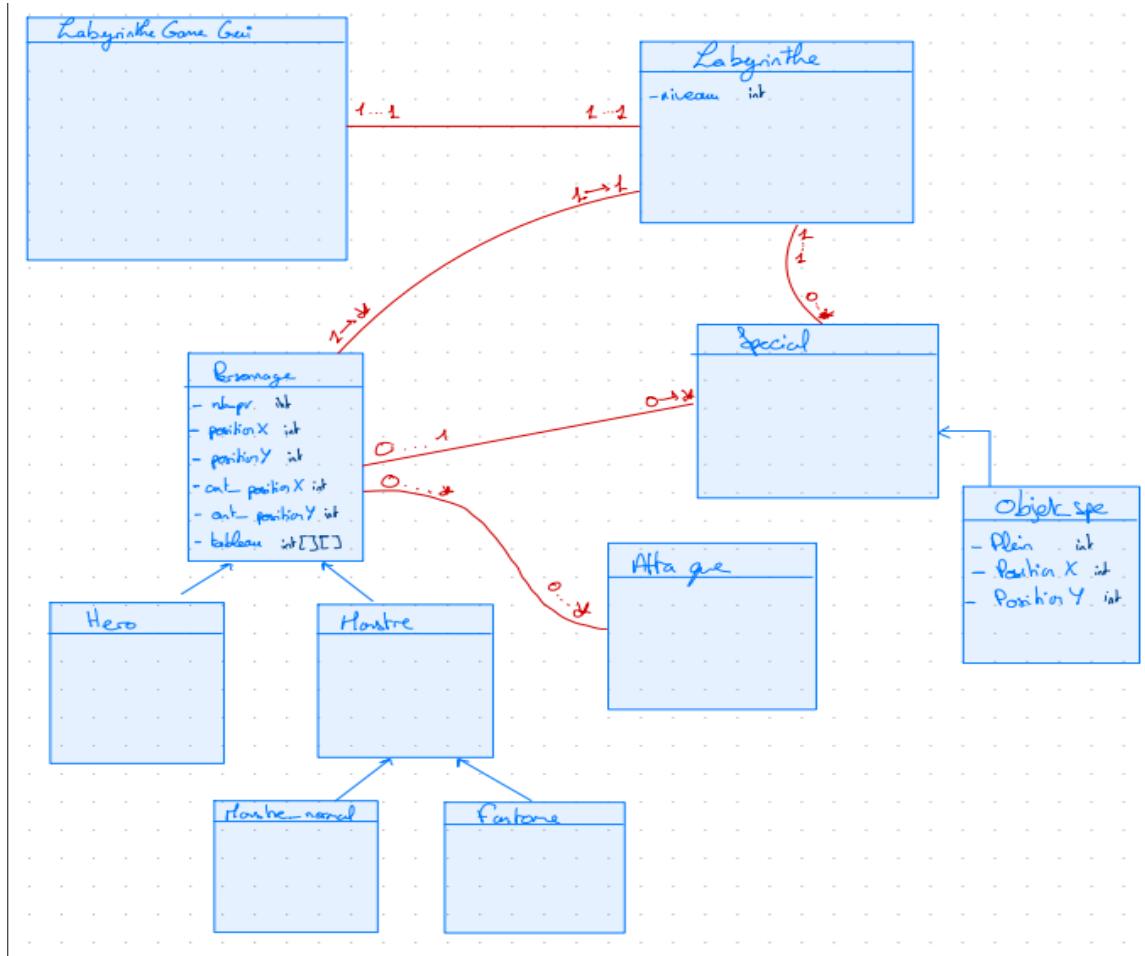


Diagramme 3 :

Pour le sprint 3, nous avons avancé sur nos classes et nous les avons remplies.

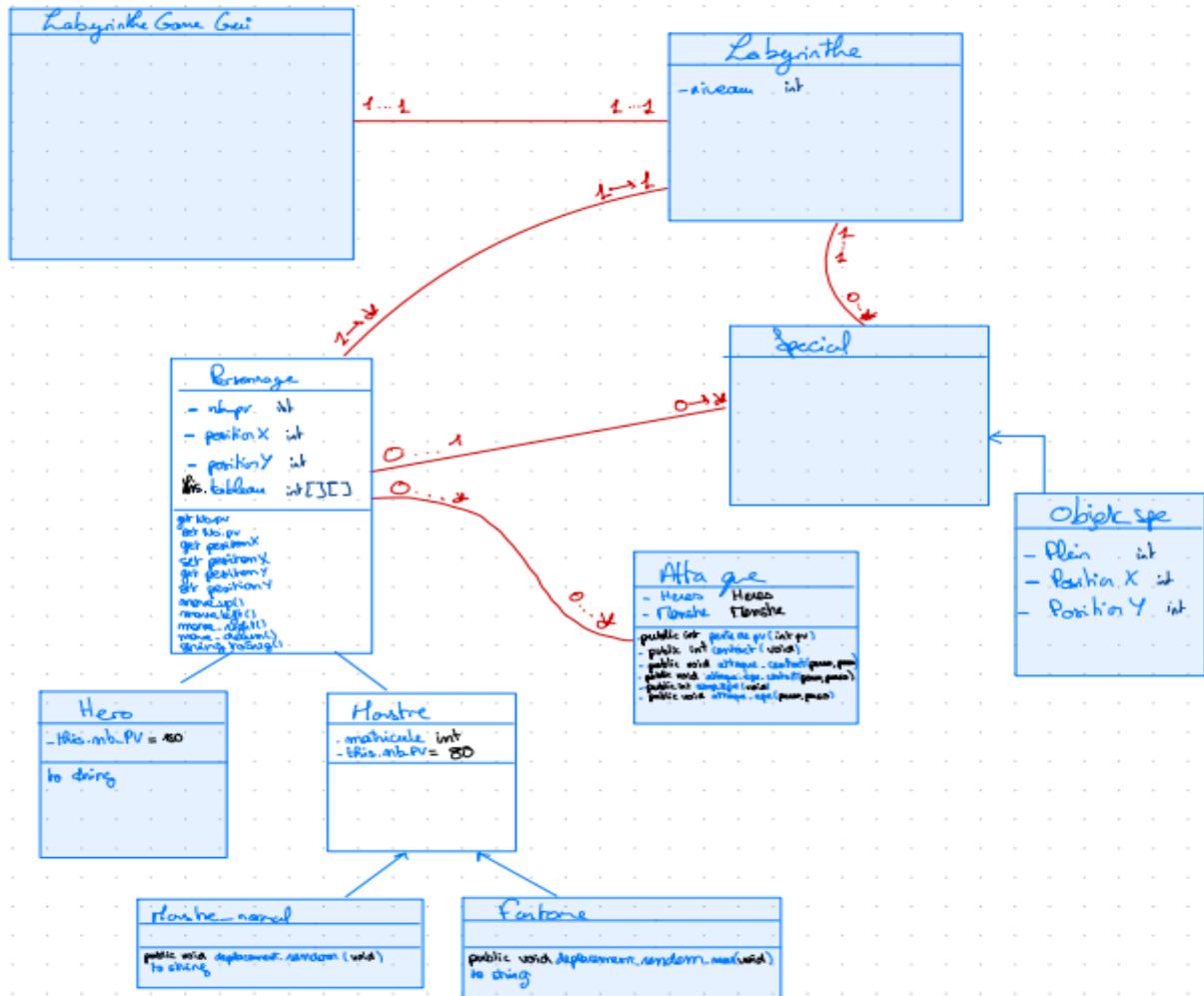
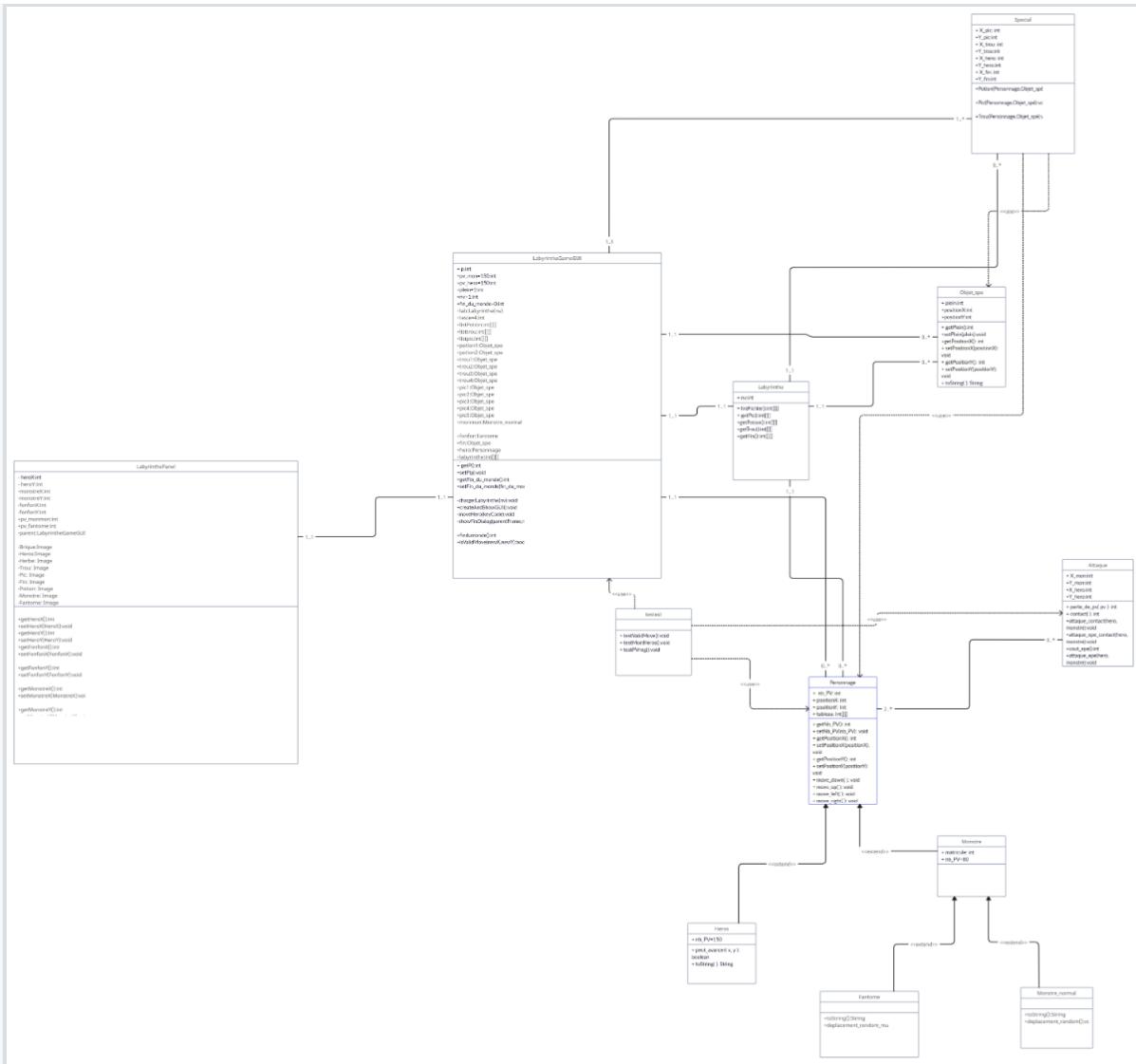


Diagramme 4 :

Pour le sprint 4, nous avons décidé de faire le diagramme en ligne pour avoir un rendu plus propre. Nous avons également rempli correctement toutes les cases, maintenant que nous avons finalisé les classes.



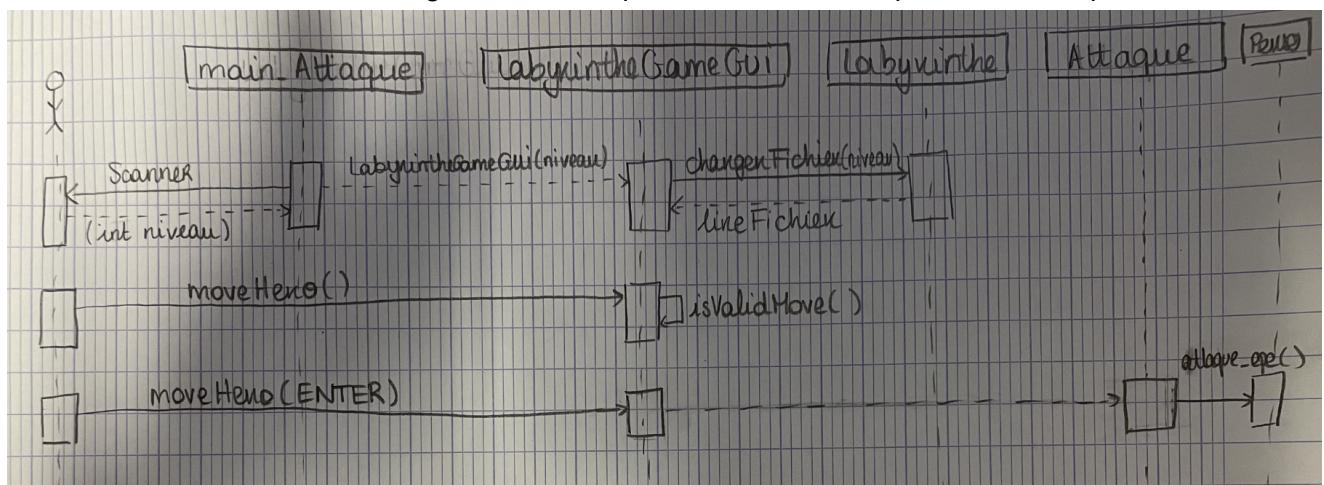
Diagrammes de séquences :

Sprint 1 et 2 :

Nous n'avions pas fait de diagramme de séquence pour ce sprint car nous n'étions pas encore assez avancés dans nos classes.

Sprint 3 :

Nous avons tenté de faire un diagramme de séquence mais il n'était pas assez complet.



Sprint 4 :

Nous avons tenté de faire un diagramme de séquence plus complet.

