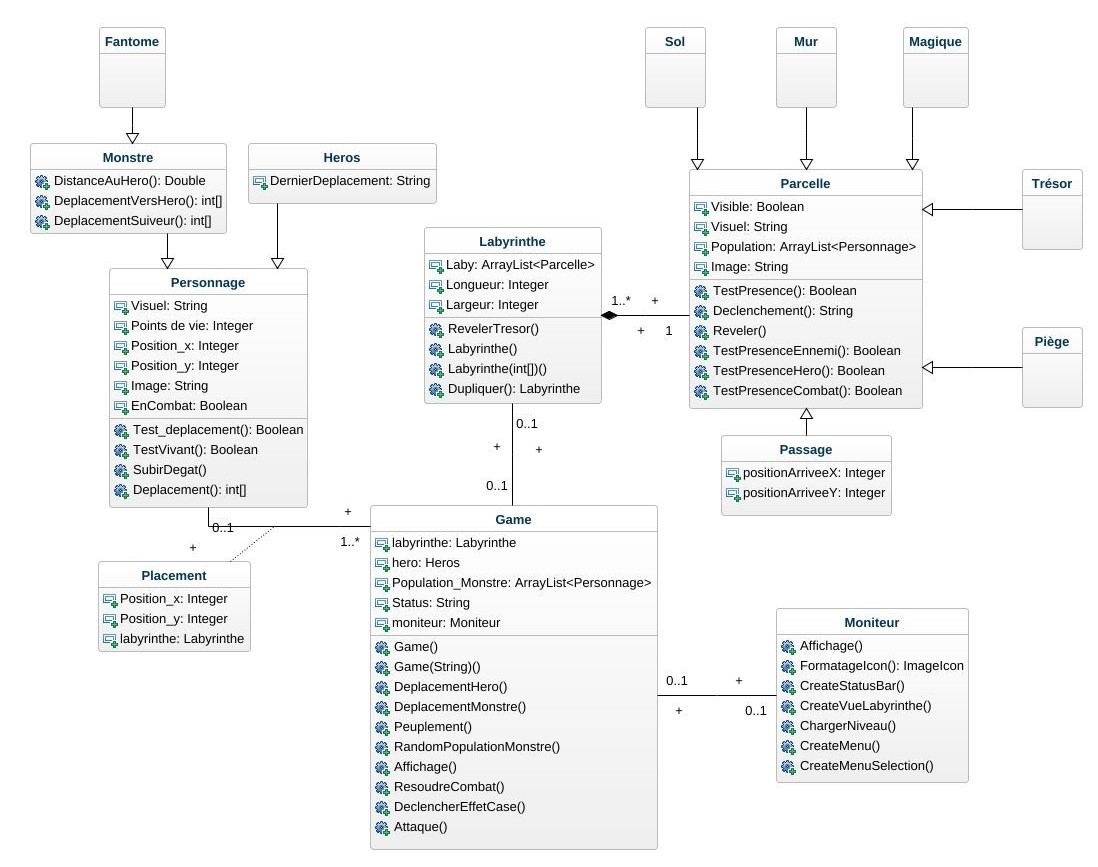
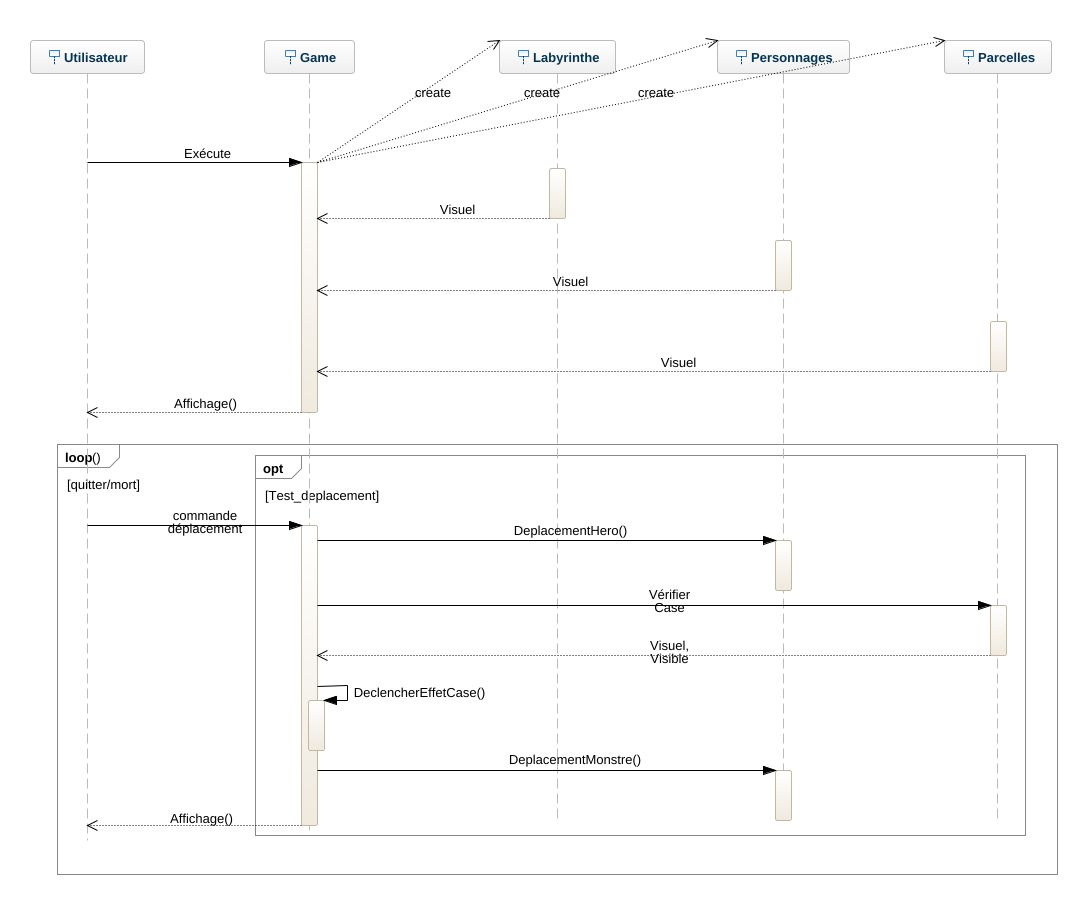
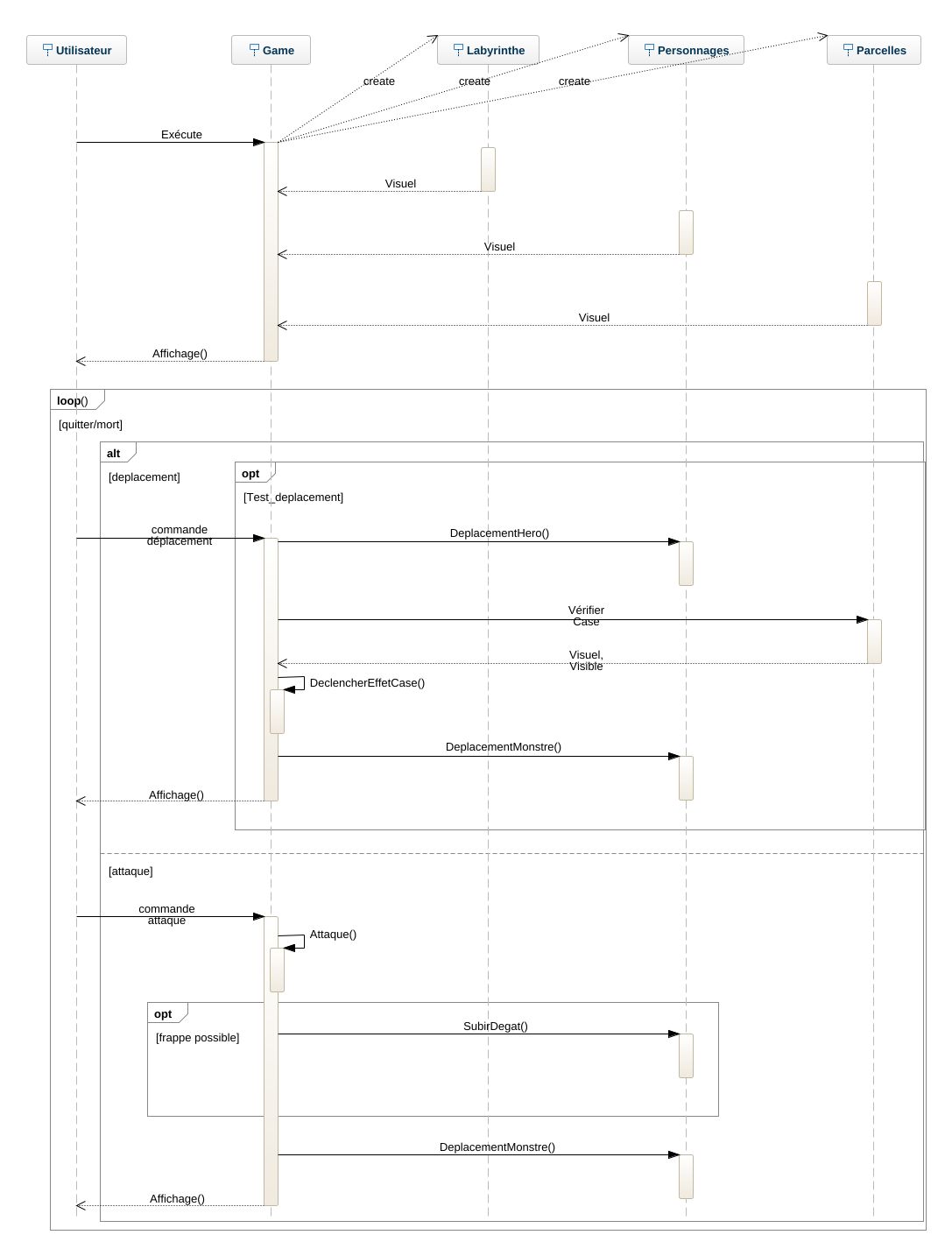
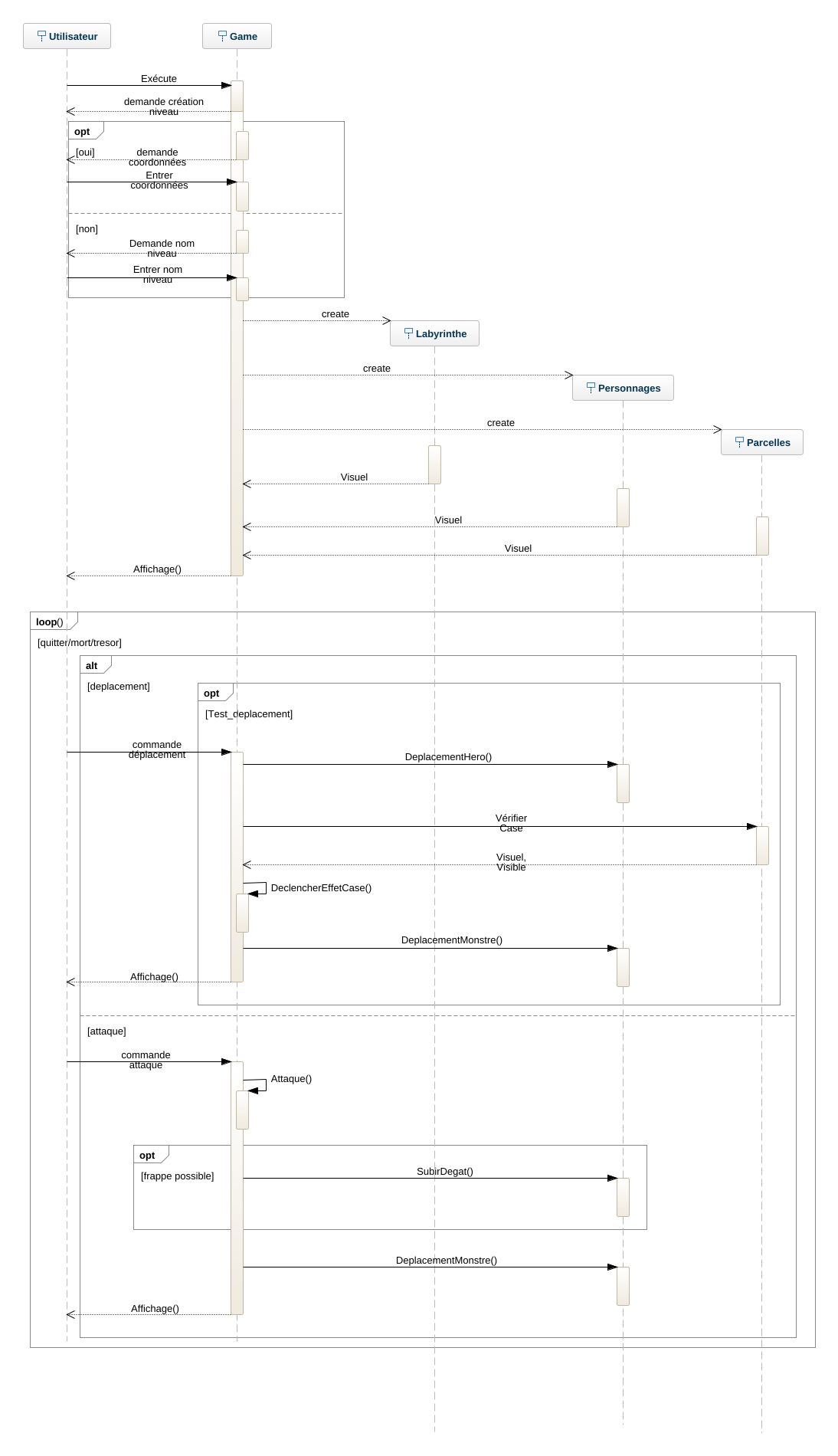
Usecase Diagram :

Class Diagram :

Sequence Diagram Sprint 3 :

Sequence Diagram Sprint 4 :

Sequence Diagram Sprint 5



Titre : Déplacement

Résumé : permettant à l’utilisateur de déplacer le héros dans le labyrinthe.

Acteur : Joueur

Responsable : Stan-Loïs Riviere

Date de création : 21/11/2020

Date de modification : 01/01/2021

Version : 3

Pré-condition : le héros est sur une case valide (position initiale)

Scénario nominal :

1. Le joueur choisit la direction (z,q,s,d)
2. Le choix du joueur est analysé et traité par le programme.
3. Le héros se déplace à sa nouvelle position

Scénarios alternatifs :

A – Le joueur ne choisit pas de direction.

1 – Il ne se passe rien jusqu’à sa prochaine action. Retour au point 1.

B – Le joueur choisit d’appuyer sur deux boutons de direction opposé en même temps

1 – Le héros se déplace dans une direction puis l’autre quel que soit l’ordre. Il revient donc à son point de départ.

C – Si le joueur appuie sur une ou plusieurs autres touches que les touches de directions.

1 - Ils ne se passe rien jusqu’à la prochaine action du joueur. Retour au point 1.

D – Le joueur appuie sur 3 ou plus touches de direction en même temps.

1 – Le héros se déplace dans l’ordre des touches qui sont appuyées. Retour au point 1.

Scénarios exceptionnels :

1. Le joueur décide de quitter le jeu.
2. Problèmes d’affichage : Il y a eu un bug ou une erreur d’affichage du programme.

Post-condition : La position du héros est changée

Titre : Générer Labyrinthe

Résumé : permettre de générer graphiquement le labyrinthe souhaité

Auteur : ORBAN-CHENOT David

Modifié le : 03.01.2021

Acteur : Système

Précondition : Le labyrinthe créé par le joueur ou depuis un fichier doit être valide

Scénario nominal :

1. Sauvegarde des différentes cases et de leur emplacement
2. Lecture de la visibilité des cases

2.1) Si la case est dite « visible », son visuel sera affiché par la suite (comme les murs, sols…)

2.2) Si la case est dite « cachée », son visuel ne sera pas affiché par la suite (case trésor, piège…)

1. Affichage du labyrinthe avec les visuels de chacune des cases

Scénario alternatif :

1. L’affichage d’un sprite/personnage indiqué n’est pas renseigné
2. Affichage d’un carré blanc à la place
3. Retour étape 3

Scénarios exceptionnels :

* Problème de génération : 2 cases se superposent alors que ce n’est pas prévu (par exemple un mur et le trésor)

Titre : Générer un labyrinthe à partir d’un fichier

Résumé : permettre de générer le labyrinthe souhaité à partir d’un fichier

Auteur : COLLIN Pierre-Louis

Modifié le : 30.12.2020

Acteur : Joueur

Précondition : L’utilisateur a choisi de générer un labyrinthe préexistant sous la forme d’un fichier.

Scénario nominal :

1. L’utilisateur indique le nom du niveau.
2. Lecture du fichier.
3. Affichage du labyrinthe avec les visuels de chacune des cases et des personnages.
4. Le jeu démarre. Le process attend la prochaine instruction de l’utilisateur.

Scénarios alternatifs :

1. Ce scénario commence à l’étape 1 si l’utilisateur rentre un nom de fichier inexistant.
   1. L’interface affiche que le fichier est introuvable.
   2. Retour étape 1
2. Ce scénario commence à l’étape 1 si le fichier existe mais qu’il ne contient pas les informations nécessaires à la création d’un niveau.
   1. L’interface affiche que le contenu du fichier est invalide.
   2. Retour étape 1

Titre : Le joueur crée son labyrinthe

Auteur : Guillaume LIORET

Modifié le :30/12/2020

Résumé : permettre au joueur de créer son propre labyrinthe

Acteur : Joueur

Prérequis : L’utilisateur a choisi de créer son propre labyrinthe.

Scénario nominal :

1. L’utilisateur indique les dimensions du labyrinthe
2. L’utilisateur place les murs
3. L’utilisateur tape fin pour passer à la suite
4. L’utilisateur choisit son point de départ
5. L’utilisateur choisit le nombre de monstres présent dans le labyrinthe
6. L’utilisateur choisit l’emplacement de l’arrivée
7. L’utilisateur choisit l’emplacement de la case magique
8. L’utilisateur choisit l’emplacement de la case piège
9. L’utilisateur choisit l’emplacement de l’entrée et de la sortie du passage.

Scénario alternatif :

A - l’utilisateur indique des dimensions non valides pour le labyrinthe (entier négatif voire pas un entier du tout)

1-l’interface affiche que les données entrées sont invalides

2-retour étape 1

B -Ce scénario commence si l’utilisateur indique des coordonnées non valides pour l’emplacement d’un mur (entier négatif voire pas un entier du tout)

1-l’interface affiche que les données entrées sont invalides

2-retour étape 2

C -Ce scénario commence si l’utilisateur indique des coordonnées non valides pour le point de départ

1-l’interface affiche que les données entrées sont invalides

2-retour étape 4

D -Ce scénario commence si l’utilisateur indique des coordonnées non valides pour le nombre de monstres présents( entier négatif, pas un entier, ou un entier supérieur au nombre de places disponibles)

1-l’interface affiche que les données entrées sont invalides

2-retour étape 5

E-Ce scénario commence si l’utilisateur indique des coordonnées non valides pour l’arrivée (entier négatif ou pas un entier)

1-l’interface affiche que les données entrées sont invalides

2-retour étape 6

F -Ce scénario commence si l’utilisateur indique des coordonnées non valides pour la case magique (entier négatif ou pas un entier)

1-l’interface affiche que les données entrées sont invalides

2-retour étape 7

G -Ce scénario commence si l’utilisateur indique des coordonnées non valides pour la case piège (entier négatif ou pas un entier)

1-l’interface affiche que les données entrées sont invalides

2-retour étape 8

H -Ce scénario commence si l’utilisateur indique des coordonnées non valides pour l’entrée ou la sortie du passage (entier négatif ou pas un entier)

1-l’interface affiche que les données entrées sont invalides

2-retour étape 9

Scénario exceptionnel :

Ce scénario commence après l’étape 7 dans le cas où le labyrinthe est plein

1-Les étapes 7 et 8 sont sautées

Post-condition : le jeu commence

Répartition des tâches :

Sprint 1 :

* Stan : Fonction déplacement du héros + Scénario
* Pierre-Louis : Monstres placés aléatoirement + Héros subit des dégâts s’il est sur la même case qu’un monstre + Scénario
* Guillaume : Créer un labyrinthe vide avec bordure + Scénario
* David : Use-Case + Scénario + Classe Héros avec points de vie

Sprint 2 :

* Stan : Création des différentes cases spéciales + maj Scénario
* Pierre-Louis : Créer un labyrinthe à partir d’un fichier + maj Scénario
* Guillaume : Permettre à l'utilisateur de créer son propre labyrinthe + maj Scénario
* David : Diagramme de classe + maj Scénario + Permettre à l'utilisateur de créer son propre labyrinthe

Sprint 3 :

* Stan : Tests JUnit
* Pierre-Louis : Création interface graphique
* Guillaume : Monstres se déplacent aléatoirement
* David : Diagrammes de classe et séquence + Tests

Sprint 4 :

* Stan : Ajout de nouveaux tests et correction d’anciens
* Pierre-Louis : Animation des combats + Le héros peut attaquer les monstres adjacents
* Guillaume : Fantômes + Tests
* David : Fantômes + maj diagrammes

Sprint 5 :

* Stan : Ajout d’autres tests
* Pierre-Louis : Les monstres essayent de suivre le héros
* Guillaume : Tests + Compilation projet
* David : maj diagrammes + Powerpoint