

Projet Analyse et Conception Logiciel

M1 - Info



Dungeon Crypt

BOHAN Thomas

JACQUES Louis

SERRATORE Alexandre

SUCK Hugo

Table des matières :

1) Vision du produit	3
2) BackLogs	4
Sprint 0 : Lancement	4
Sprint 0 : Review	5
Sprint 0 : Retrospective	5
Sprint 1: Lancement	6
Sprint 2	10
Sprint 3	11
Liste des fonctionnalités	12
Salles	12
1) Créer la salle de départ du jeu et son affichage (Version terminal)	12
2) Créer un générateur de salle simple	12
3) Créer une sortie vers la salle suivante	12
4) Certaines cases du labyrinthe sont spéciales :	12
• Créer des trésors :	12
• Créer des pièges :	12
5) Améliorer la génération de salle	12
Héros	13
6) Créer, placer, déplacer le héros	13
7) Le héros peut infliger des dégâts aux monstres	13
8) Le héros peut utiliser des bonus	13
Monstres	13
9) Créer un monstre simple, se déplace aléatoirement	13
10) Les monstres infligent des dégâts aux joueurs	13
11) Créer différent type de monstre au comportement différents	13
• Squelette	13
• Zombie	13
• Fantôme	13
• Boss	13
Collisions	14
12) Gérer les collisions entre le héros, les monstres et les murs	14
Interface Graphique & divers	14
13) Adapter le projet a l'utilisation de LibGDX	14
14) Afficher les éléments d'une salle graphiquement (écran)	14
15) Actualiser l'affichage des entités et modifier leur déplacement	14
16) Ajouter un menu principale	14
17) Afficher et gérer le score	14
18) Afficher et gérer la barre de point de vie	14
19) Gérer la fin du jeu (mort du héros + écran Game Over)	14
20) Pouvoir sauvegarder / Charger une partie	15
21) Ajouter du son / bruitage / musique	15

BOHAN Thomas

JACQUES Louis

SERRATORE Alexandre

SUCK Hugo

BOHAN Thomas

JACQUES Louis

SERRATORE Alexandre

SUCK Hugo

1) Vision du produit

Dungeon Crawler. Thème Halloween/Horreur.

Un personnage traverse différentes salles avec des monstres à combattre et récupère de l'équipement. Le joueur peut consommer certains objets pour lui attribuer des avantages (exemple : gain de vie).

Lorsque le joueur atteint la sortie finale, il descend d'un étage et les monstres présents sont plus difficiles à combattre.

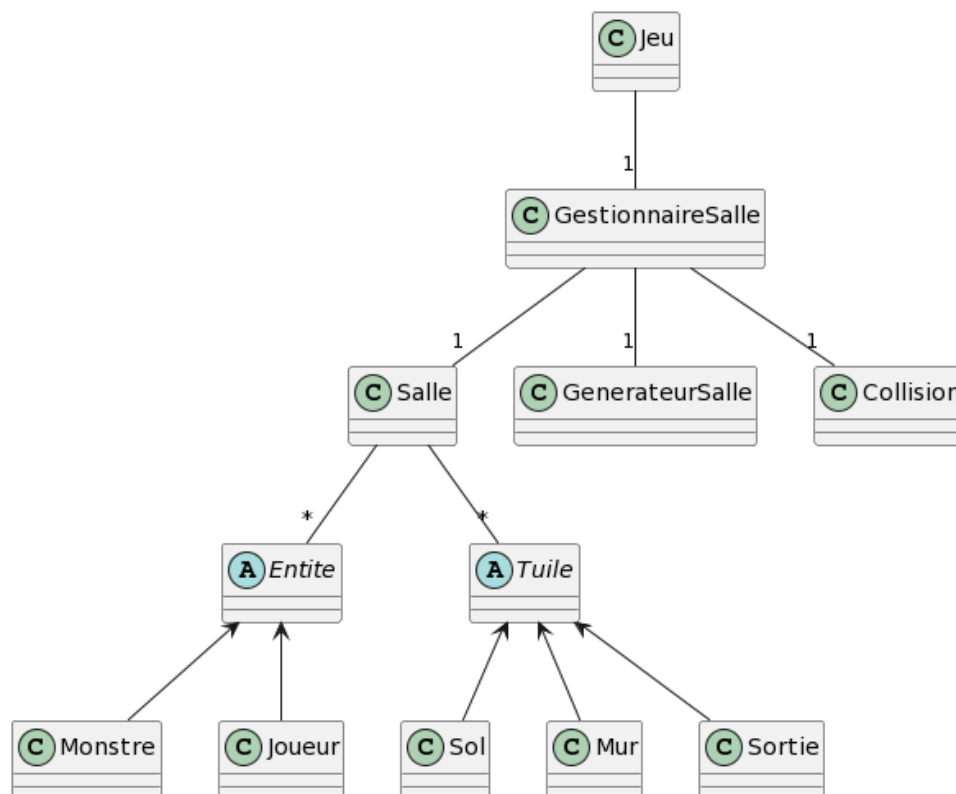
Des points lui sont attribués pour chaque monstre qu'il tue, certains équipements qu'il récupère et chaque étage qu'il descend.

2) BackLogs

Sprint 0 : Lancement

Liste des tâches	Responsable
Créer la salle de départ du jeu et son affichage (Version terminal)	Hugo
Créer un générateur de salle simple (Version terminal)	Hugo
Créer, placer, déplacer le héros	Thomas
Créer un monstre simple, se déplace aléatoirement	Thomas
Gérer les collisions entre le héros, les monstres et les murs	Alexandre
Créer une sortie vers la salle suivante	Louis

Diagramme de classe



BOHAN Thomas

JACQUES Louis

SERRATORE Alexandre

SUCK Hugo

Sprint 0 : Review

Liste des tâches	Validé	Commentaires
Créer la salle de départ du jeu et son affichage (Version terminal)	Validé	Réalisé sans soucis
Créer un générateur de salle simple (Version terminal)	Validé	Implémenté pour faciliter l'ajout de patterns et de forme de génération
Créer, placer, déplacer le héros	Validé	Réalisé sans soucis
Créer un monstre simple, se déplace aléatoirement	Validé	Les monstres n'ont aucun intérêt pour l'instant.
Gérer les collisions entre le héros, les monstres et les murs	Validé	Collisions simples pour l'affichage via terminal. A modifier dans le futur. Suppression de la classe Collision.
Créer une sortie vers la salle suivante	Validé	Réalisé sans soucis

Sprint 0 : Rétrospective

Ce qui s'est bien passé (ou pas) et les décisions pour le sprint suivant :

- Réalisation d'une architecture solide du projet pour faciliter l'implémentation graphique.
- Bonne organisation et communication via le git et trello. Chaque personne notifiait les autres membres du groupe de l'avancée du travail. Chaque membre était constamment au courant de quelle fonctionnalité est terminée et quelle fonctionnalité est en cours et par qui.
- Le travail a été commencé et terminé tôt, ce qui a permis de corriger les éventuels bugs rapidement.

Problèmes :

BOHAN Thomas

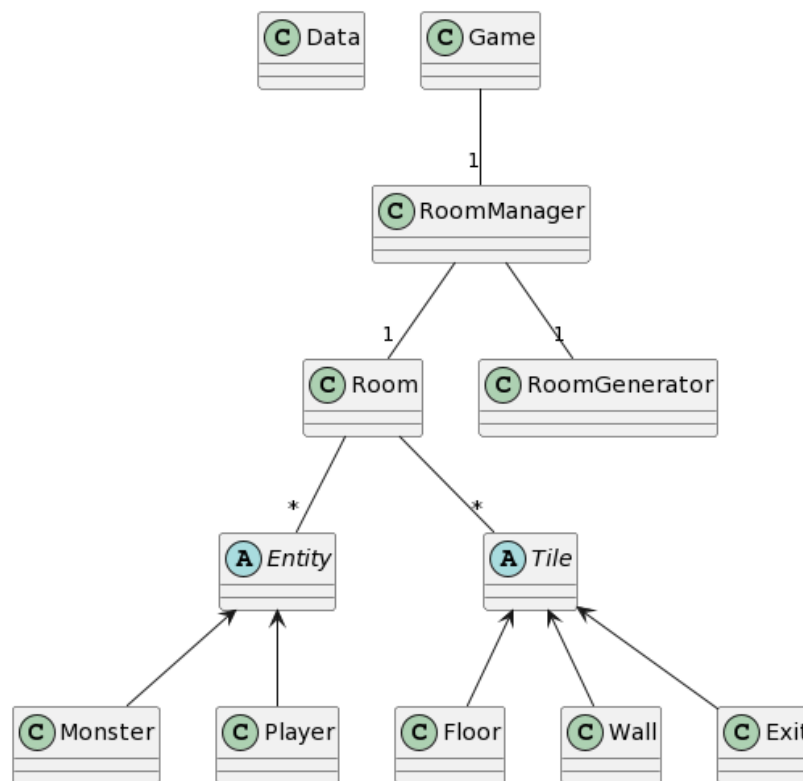
JACQUES Louis

SERRATORE Alexandre

SUCK Hugo

- Il y a cependant eu une mauvaise utilisation du système de branche sur git.
Décision prise : Créer et utiliser des branches pour chaque fonctionnalité lors du prochain sprint et utiliser les pull request afin de vérifier le code des autres intervenants avant la merge avec le reste du projet.
- Un problème a eu lieu au niveau de la compilation du projet pour générer le fichier JAR. Des problèmes causés par un souci dans l'ordonnancement du projet et une non détection du dossier de ressources. Pour ce cas exceptionnel, nous avons généré le fichier JAR via IntelliJ.
Décision prise : Pour le sprint suivant nous aurons les lignes de commandes claires et fonctionnelles pour générer et lancer le projet depuis les fichiers sources. Ceci sera effectué en début de sprint 1.

Diagramme de classe (actualisé) :



Sprint 1: Lancement

Liste des tâches	Responsable
Adapter le projet a l'utilisation de LibGDX	Tous
Utiliser Gradle pour l'installation et le lancement du jeu	Alexandre
Afficher les éléments d'une salle graphiquement (écran)	Thomas
Afficher et gérer la barre de point de vie	Louis
Les monstres infligent des dégâts aux joueurs	Alexandre
Gérer les collisions entre le héros, les monstres et les murs (Adapté à la librairie utilisée)	Alexandre
Le héros peut infliger des dégâts aux monstres	Louis
Actualiser l'affichage des entités et modifier leur déplacement	Thomas
Créer différent type de monstre au comportement différents (Zombie, Fantômes)	Fantôme : Louis Zombie: Hugo
Améliorer la génération de salle	Hugo
Certaines cases du labyrinthe sont spéciales :	Hugo

Tâches Bonus	
Afficher les éléments d'une salle graphiquement (Ajout des sprites)	Thomas
Ajouter un menu principale	Alexandre
Afficher et gérer le score	Thomas
Gérer la fin du jeu (mort du héros + écran Game Over)	Alexandre

BOHAN Thomas

JACQUES Louis

SERRATORE Alexandre

SUCK Hugo

Diagramme de classe :

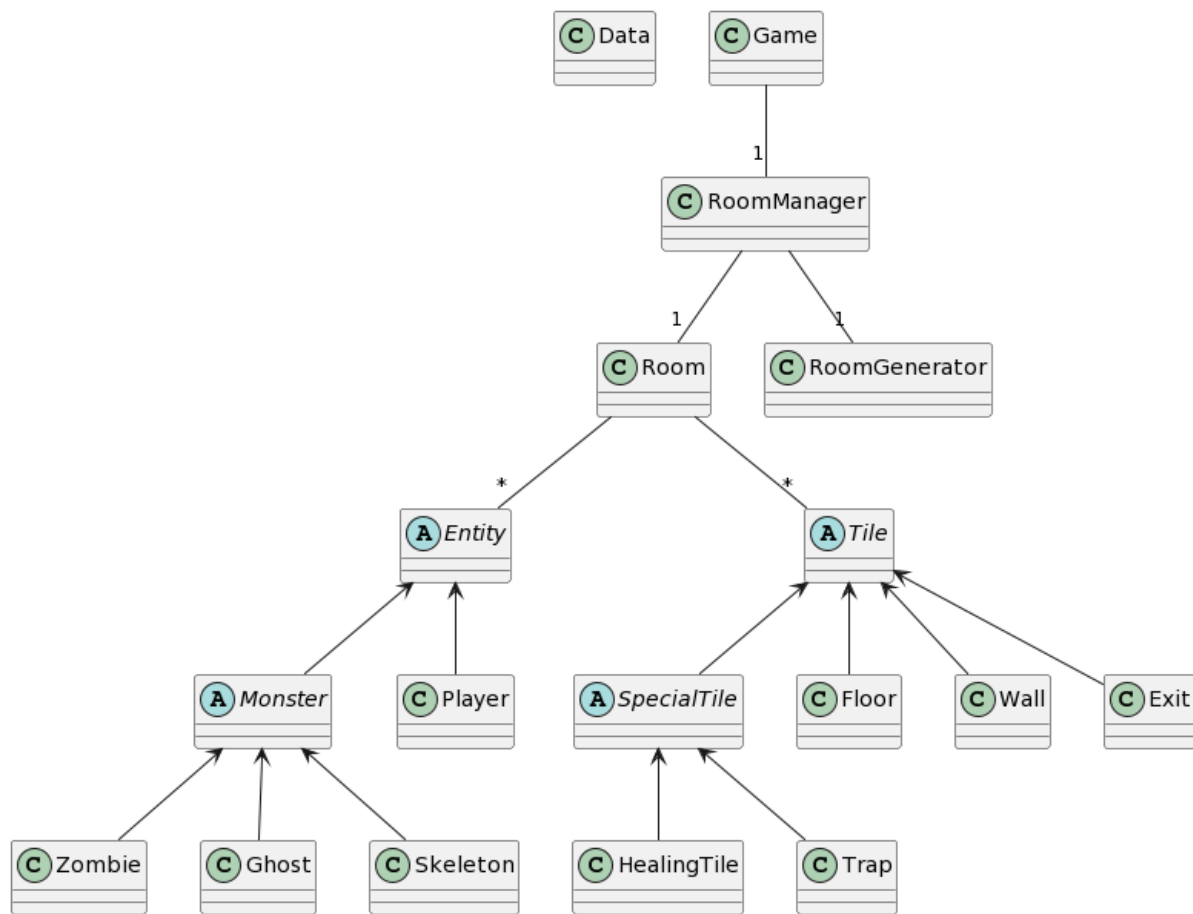
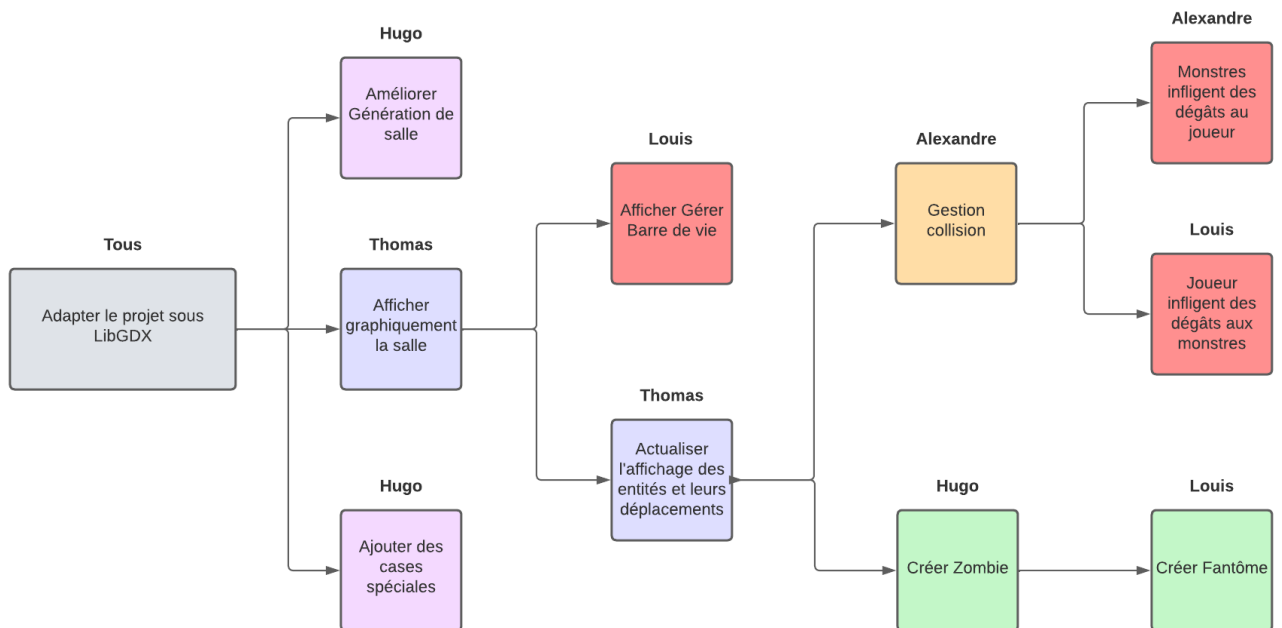


Diagramme d'ordonnancement des tâches :



Sprint 1 : Review

Liste des tâches	Validé	Commentaires
Adapter le projet a l'utilisation de LibGDX	Validé	Réalisé sans trop de soucis.
Utiliser Gradle pour l'installation et le lancement du jeu	Validé	Création d'un script pour faciliter l'installation et le lancement du jeu.
Afficher les éléments d'une salle graphiquement (écran)	Validé	Réalisé sans soucis.
Afficher et gérer la barre de point de vie	Validé	Réalisé sans soucis.
Les monstres infligent des dégâts aux joueurs	Validé	Réalisé sans soucis.
Gérer les collisions entre le héros, les monstres et les murs (Adapté à la librairie utilisée)	Validé	Problème d'accès aux bodies des entités, réécriture de certains constructeurs pour faciliter la tâche.
Le héros peut infliger des dégâts aux monstres	En cours	Problème lié au body de l'arme qui permet d'infliger des dégâts.
Actualiser l'affichage des entités et modifier leur déplacement	Validé	Petit blocage lié à un oubli sur le type d'entité d'un body qui n'était pas dynamique.
Créer différent type de monstre au comportement différents (Zombie, Fantômes)	En cours	Implémentation plus longue que prévu pour l'algorithme de pathfinding du zombie. Fantôme implémenté sans soucis.
Améliorer la génération de salle	Validé	Réalisé sans soucis et facilement évolutif (ajout ou suppression de pattern/génération).
Certaines cases du labyrinthe sont spéciales :	Validé	Création d'une classe abstraite pour les cases spéciales pour simplifier l'évolutivité.

Tâches Bonus		Commentaires
Afficher les éléments d'une salle graphiquement (Ajout des sprites)	Validé	Problème de caméra rencontré. Nous avons dû revoir totalement la gestion de la caméra.
Ajouter un menu principale	Validé	Réalisé sans soucis.
Afficher et gérer le score	Validé	Réalisé sans soucis.
Gérer la fin du jeu (mort du héros + écran Game Over)	Validé	Réalisé sans soucis.

Sprint 1 : Rétrospective

Résolution des problèmes du Sprint 0 :

- Mauvaise utilisation des branches GIT :
L'utilisation des branches sur Git a été mise en place, ce qui a facilité le travail en parallèle de plusieurs tâches. Des revues de code ont pu être effectuées avant les merge afin d'éviter d'éventuel problème.
- Problème de la génération du fichier JAR :
Création d'un script BASH pour générer automatiquement le fichier JAR. Une seule commande suffit désormais pour générer le fichier exécutable JAR et de lancer l'archive.

Ce qui s'est bien passé (ou pas) et les décisions pour le sprint suivant :

- **L'implémentation de la bibliothèque graphique** s'est faite sans trop de problème. Nous avons dû adapter quelque petit détail à la configuration gradle fournie avec Libgdx, mais le reste a été grandement facilité de part l'architecture créée lors du Sprint 0.
- Des problèmes d'accès à certaines fonctions/variables ont été rencontrés lors de **la gestion des collisions**. L'écouteur de Libgdx ne permettait pas de remonter jusqu'au **body** et à l'**objet** lié aux **fixtures**. Une légère réécriture de certains constructeurs a dû être effectuée.
- **Le diagramme d'ordonnancement des tâches** à faciliter l'organisation de l'équipe. Nous utiliserons de nouveau ce genre de diagramme pour les sprints suivants.
- **Le comportement des différents types de monstres** n'est pas terminé :
 - Le fantôme fonctionne (traverse les murs en direction du joueur)
 - Le zombie ne possède pas son comportement de déplacement (l'implémentation du pathfinding est plus longue que prévu).

Décision prise : La tâche sera donc **prolongée sur le prochain sprint**.

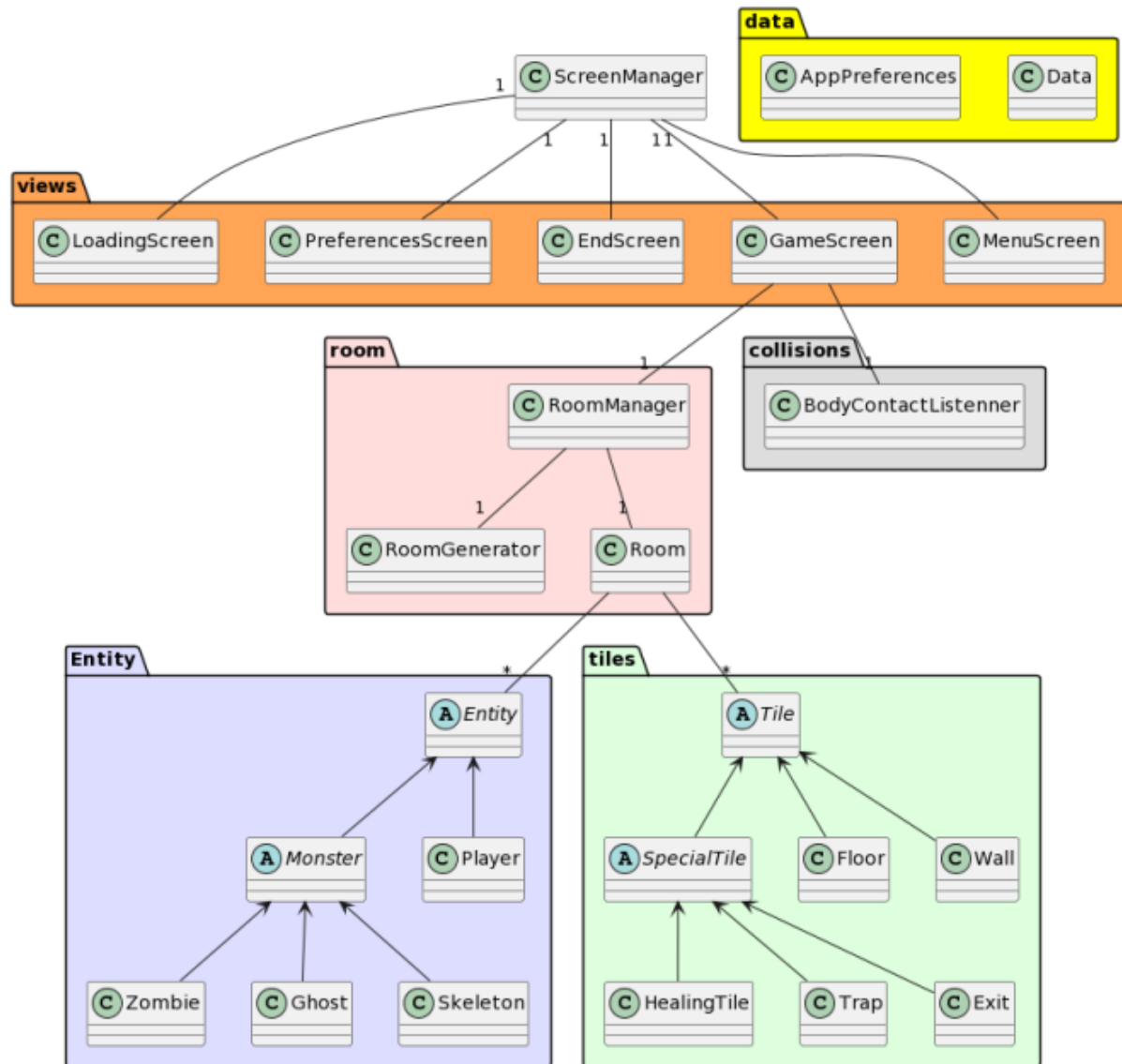
- **La création de l'arme du joueur** lui permettant d'infliger des dégâts n'est pas terminée. Un problème sur la position du body de l'objet a été rencontré. Le sprite quant à lui est placé au bon endroit.

Décision prise : La tâche sera donc **prolongée sur le prochain sprint**.

- **Le redimensionnement et la position de la caméra** posent des problèmes en fonction de la taille de la fenêtre.

Décision prise : Le bug sera corrigé sur le **prochain sprint**.

Diagramme de classe (actualisé) :



Sprint 2 : Lancement

Liste des tâches prolongées du Sprint 1	Responsable
Créer différent type de monstre au comportement différents (Zombie)	Hugo
Le héros peut infliger des dégâts aux monstres	Louis (Créer et afficher l'arme) Alexandre (Gérer la collision entre l'arme et les monstres)

Liste des tâches	Responsable
Problème de redimensionnement de caméra	Louis
Ajouter du son / bruitage / musique	Thomas
Pouvoir sauvegarder / Charger une partie	Alexandre
Recul lorsque le joueur subit des dégâts	Thomas / Louis
Recul lorsque les monstres subissent des dégâts	
Créer une sortie vers la salle suivante	Hugo

Tâches Bonus	Responsable
Ajouter des tuiles	
Ajouter des monstres	

Diagramme de classe :

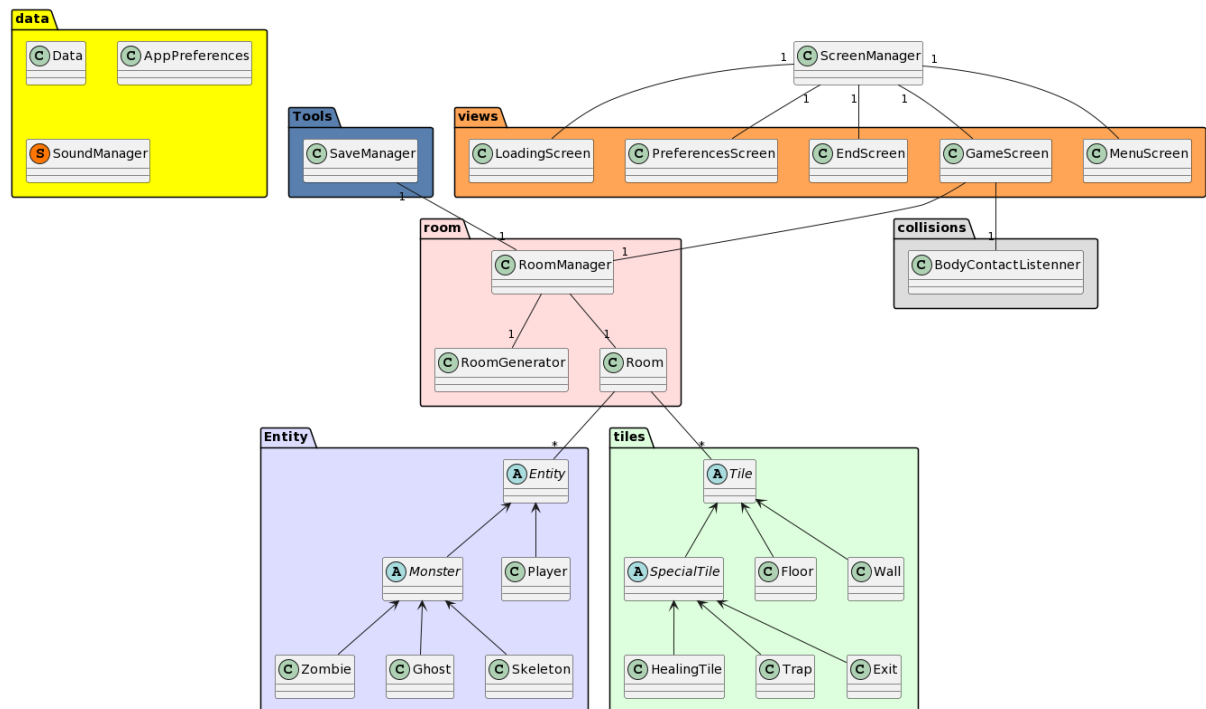
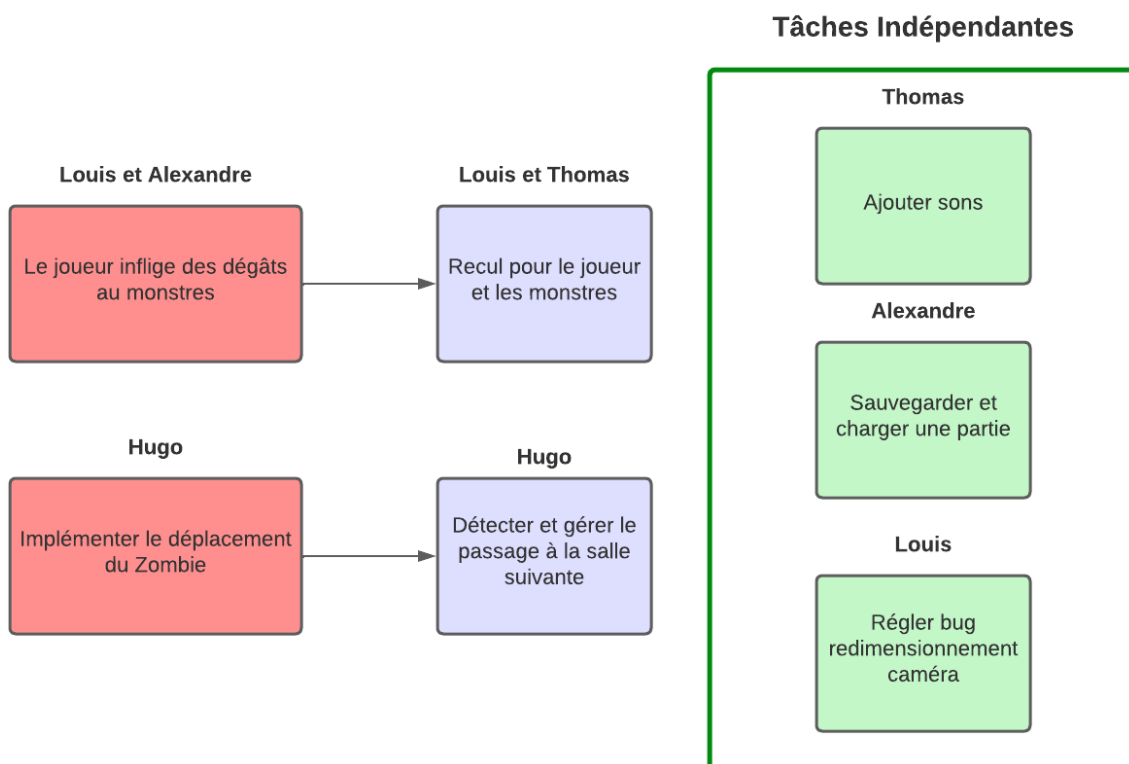


Diagramme d'ordonnancement des tâches :



BOHAN Thomas

JACQUES Louis

SERRATORE Alexandre

SUCK Hugo

Sprint 2 : Review

Sprint 3

Liste fonctionnalité	Responsable

Liste des fonctionnalités

Salles

1) Créer la salle de départ du jeu et son affichage (Version terminal)

Créer une salle à génération fixe. Une salle possède des casses de sols et des cases de murs.

2) Créer un générateur de salle simple

Créer des salles à génération variable. Se baser sur des patterns fixes écrits sur un fichier texte externe.

3) Créer une sortie vers la salle suivante

Lorsque le héros atteint la sortie de la salle, il est téléporté dans la salle suivante. La sortie est disponible lorsque tous les monstres ont été tués.

4) Certaines cases du labyrinthe sont spéciales :

• Créer des trésors :

Si le héros arrive sur la case, il gagne un bonus.

• Créer des pièges :

Si le héros arrive sur la case, il subit des dégâts.

5) Améliorer la génération de salle

- Ajouter des paternes
- Ajouter des générations
- Améliorer les générations
- [Placement de la sortie modulable]

Héros

6) Créer, placer, déplacer le héros

Le héros se déplace en fonction des entrées clavier. Il ne peut pas traverser les murs. Le héros possède des points de vie.

7) Le héros peut infliger des dégâts aux monstres

- Ajouter une barre de vie aux monstres
- Afficher une épée lorsque le joueur frappe le monstre

8) Le héros peut utiliser des bonus

Consommer des potions, armes, bouclier, etc

9) Recul lorsque le joueur subit des dégâts

Monstres

10) Créer un monstre simple, se déplace aléatoirement

11) Les monstres infligent des dégâts aux joueurs

Les dégâts sont infligés lors du contact avec le joueur. Lorsque le joueur est touché, il ne peut plus être touché pendant un court instant (iFrame)

12) Créer différent type de monstre au comportement différents

- Squelette
 - Se déplace de manière aléatoire
- Zombie
 - Se déplace vers le joueur (Algo A* pour le pathfinding)
- Fantôme
 - Se déplace à travers les murs vers le joueur (Algo A* pour le pathfinding)
- Boss
 - Apparaît tout les X étages

Collisions

- 13) Gérer les collisions entre le héros, les monstres et les murs

Utiliser les body de libgdx pour gérer les collisions.

- 14) Recul lorsque les monstres subissent des dégâts

Interface Graphique & divers

- 15) Adapter le projet a l'utilisation de LibGDX

Passage au mode graphique avec la librairie LibGDX.

- 16) Utiliser Gradle pour l'installation et le lancement du jeu

Créer un script pour faciliter l'installation et le lancement du jeu (Mode Desktop).

- 17) Afficher les éléments d'une salle graphiquement (écran)

- Placer les body (mode debug)
- Ajouter les textures sur les body

- 18) Actualiser l'affichage des entités et modifier leur déplacement

Avoir un rendu en temps réel et non plus action par action.
Gérer les déplacements via libgdx.

- 19) Ajouter un menu principale

Avant d'entrer en jeu, un menu principale s'affiche avec:

- Nouvelle partie
- Charger une partie
- Options
- Quitter

20) Afficher et gérer le score

21) Afficher et gérer la barre de point de vie

Le joueur possède une barre de PV visible sur l'écran. Lorsqu'il subit des dégâts ou lorsqu'il est soigné, la barre de PV s'actualise.

22) Gérer la fin du jeu (mort du héros + écran Game Over)

Si les PV du joueur atteignent 0, un écran de Game Over s'affichera.

23) Pouvoir sauvegarder / Charger une partie

Le jeu possède UNE sauvegarde automatique qui est écrasée à chaque changement de salle.

Il est possible de sauvegarder manuellement lorsque tous les monstres de la salle sont morts. Le joueur choisit alors le nom de sa sauvegarde.

24) Ajouter du son / bruitage / musique

Correction de bug

25) Problème de redimensionnement de caméra

Le redimensionnement de la caméra pose problème lorsque l'on modifie la taille de la fenêtre dans les écrans hors jeu. L'écran de jeu est complètement déréglé.

26)

Idées

Ajout de boss

Ajout de monstres

Ajout de tuiles (trou, lave, eau, buff)/

Ajout d'arme/équipement (arme à distance,)

Ajout d'un bouclier/armure

Système d'inventaire avec objets (potions,)

BOHAN Thomas

JACQUES Louis

SERRATORE Alexandre

SUCK Hugo

Utiliser une manette

Améliorer les statistiques du joueur (vitesse de déplacement, vie max, dégâts, etc)