NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément

Rapport de Projet - S2.01 Développement d'une application

Table des Matières

1. Introduction

Objectifs du projet

L'objectif de ce projet consistait à construire un Jeu de labyrinthe en lui implémentant des fonctionnalités.

Brève description du système développé

Nous avons développé cette application en Java et JavaFX sur le logiciel IntelliJ IDEA.

Nous avons eu l'occasion d'écrire des tests pour vérifier le bon fonctionnement des fonctionnalités ainsi que des diagrammes de classes et de séquences pour faciliter la compréhension et l'organisation des différentes fonctionnalités.

2. Répartition du Travail

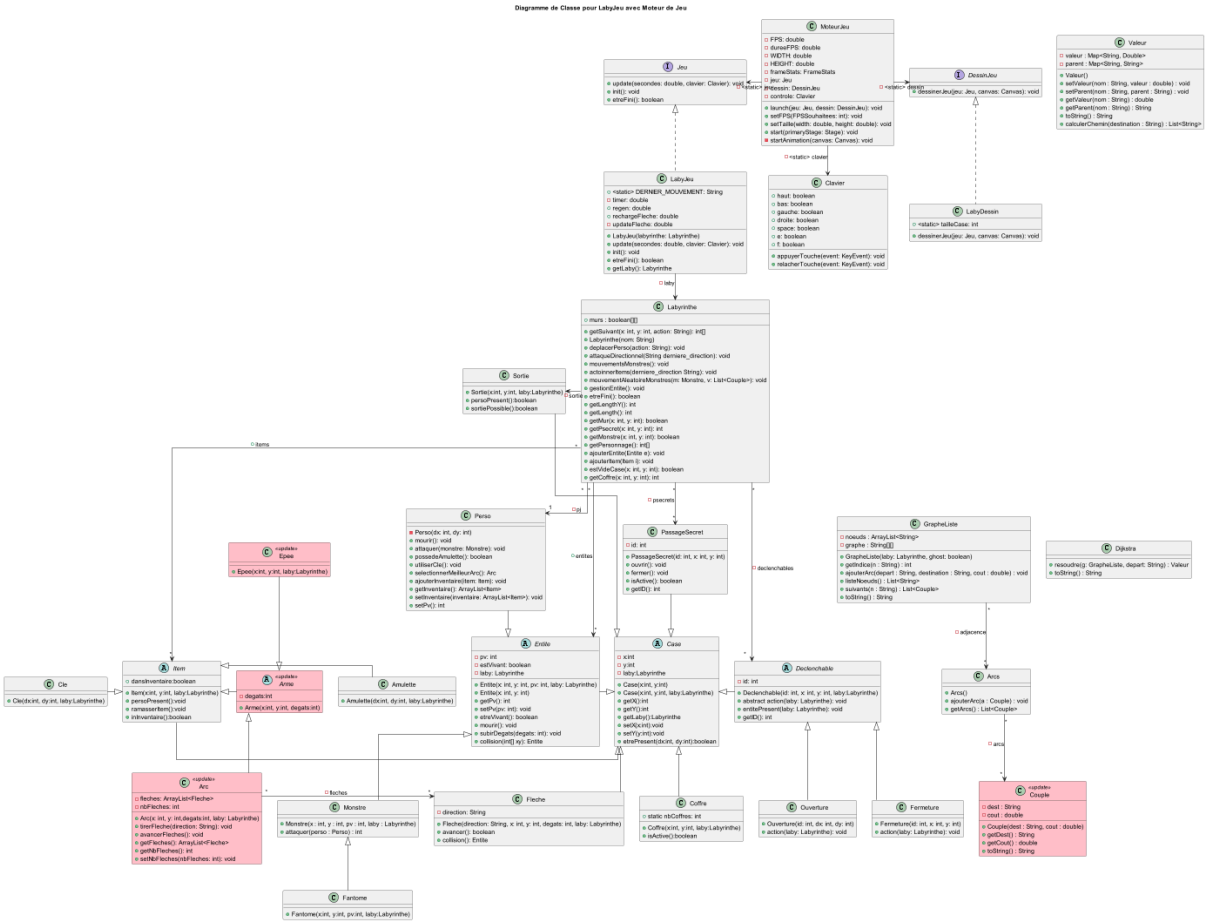
NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément

2.1. Diagrammes

Julien a majoritairement travaillé sur les diagrammes de classes en PlantUml, mais tout le monde a apporté sa participation, notamment pour les diagrammes de séquence

Version 7 - Diagramme de classes

2.2. Code



Nous avons essayé de factoriser le code au maximum en créant des Classes Abstraites globales desquelles hériteront d'autres classes. Ces classes globales et leurs enfants sont dans des packages pour faciliter la lisibilité et la compréhension du code.

Le principaux packages sont :

NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément

Entites qui regroupe la classe abstraite entités et ses enfants Perso et Monstre ainsi que Fantôme qui hérite de Monstre. Cela permet de factoriser le comportement des entités selon qu'elle est un monstre ou un personnage.

Le package Graphe qui regroupe la partie qui transforme le labyrinthe en graphe et permet à terme de gérer le comportement intelligent des monstres.

Le package Interactif qui regroupe tous les éléments avec le joueur peut avoir une interaction (hors items). On peut donc retrouver les coffres que le joueur peut ouvrir, la sortie qui permet au joueur de quitter le labyrinthe. On retrouve la classe abstraite Déclenchable qui regroupe les cases Ouverture et Fermeture qui se déclenche lorsque le personnage se trouve dessus. Enfin, la classe PassageSecret est la classe qui se fait déclencher par ces déclenchables.

Le package Laby regroupe les classes de gestion du labyrinthe, son dessin, la logique du jeu ainsi que la classe abstraite Case qui permet une importante factorisation de part ses méthodes et attributs.

Le package Objet qui regroupe tous les items ramassables du Labyrinthe. Il regroupe donc la classe abstraite Item avec tous ses enfants : les classes Amulette et Clé qui servent pour un coffre et la sortie, la classe Fleche qui gère les différentes flèches sur le labyrinthe et la classe Abstraite Arme. Cette dernière a également deux enfants : les classes Epee et Arc.

La package moteur jeu n'a pas changé, seul Clavier a été modifié pour intégrer de nouvelles touches.

Version 2

1. Donner une position initiale au monstre
- Descriptif :
  - Le monstre débute sur une case décrite dans le fichier labyrinthe.
- Critères de validation :
  - Le monstre doit avoir une position initiale.
  - Le monstre se trouve sur la case indiquée dans le fichier labyrinthe.
  - Le monstre est représenté par le caractère 'M' dans le fichier labyrinthe.
  - Le monstre ne se trouve pas sur la même case que le personnage.

2. Afficher le monstre
- Descriptif :
  - Le jeu doit afficher le monstre à sa position.
- Critères de validation :

NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément

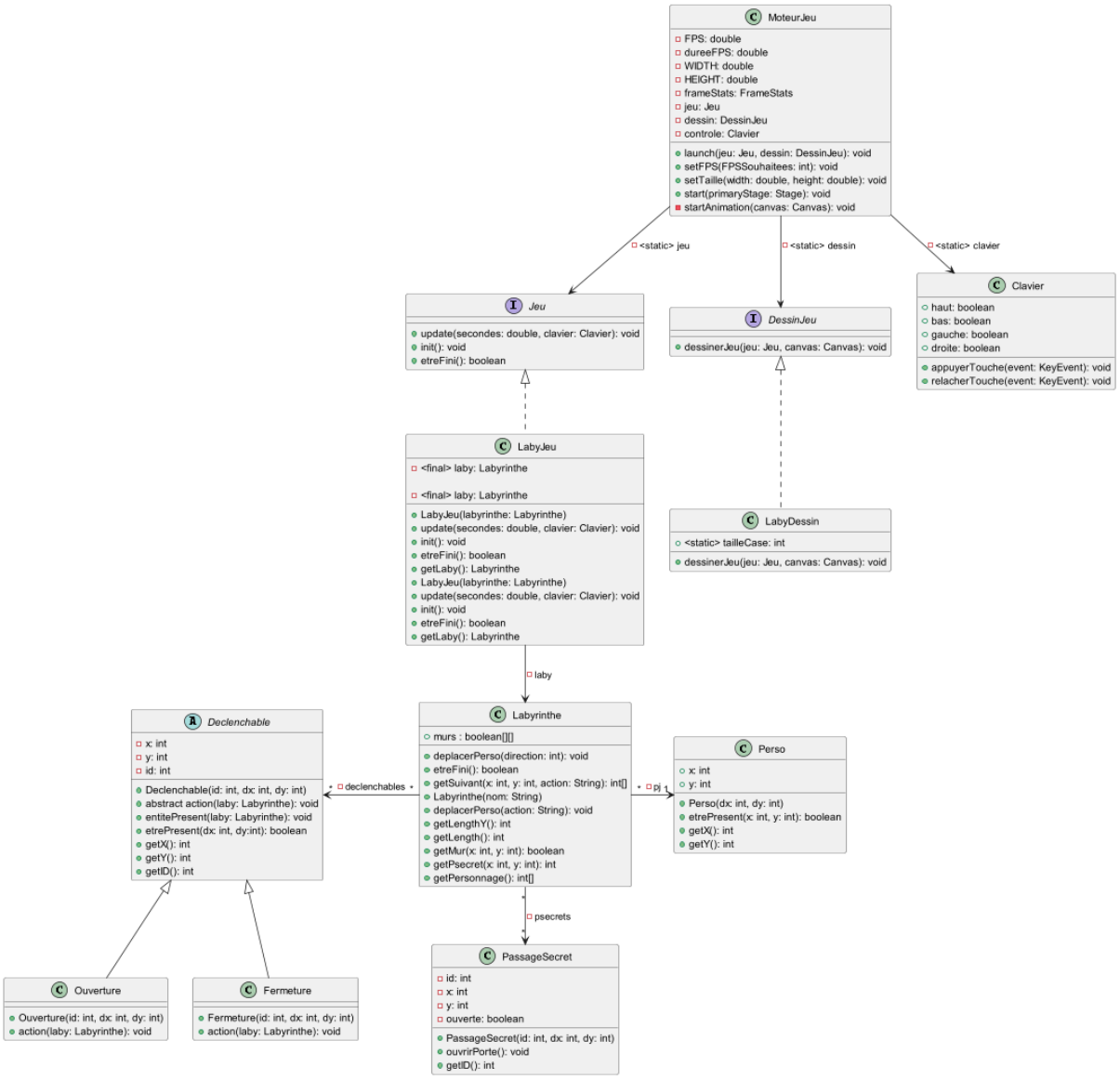
- Le monstre doit être affiché à la bonne position dans le labyrinthe.
- Le monstre sera représenté sous la forme d'un cercle violet de la taille du personnage.

3. Considérer le monstre dans les déplacements du personnage
- Descriptif :
  - Lorsque le jeu évolue, le personnage ne peut pas se déplacer sur la case du monstre.
- Critères de validation :
  - Le monstre constitue un obstacle pour le personnage.
  - Le monstre et le personnage ne peuvent pas se trouver sur la même case.
  - Le personnage ne peut pas traverser la case du monstre

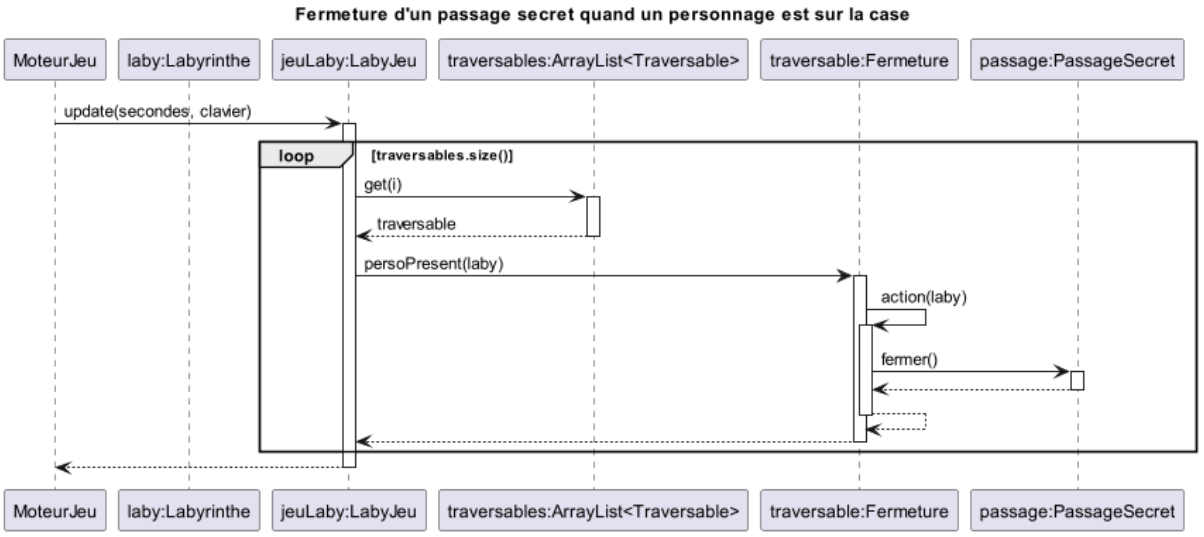
4. Déplacer le monstre (optionnel en fonction avancée)
- Descriptif :
  - Lorsque le jeu évolue, le monstre choisit une case adjacente de manière aléatoire et tente de s'y déplacer.
- Critères de validation :
  - Le monstre doit se déplacer sur une case adjacente. Il considère les 4 directions de déplacement possibles.
  - Le monstre ne peut pas se déplacer sur mur. S'il tente de se déplacer sur cette case, il ne bouge pas.
  - Le monstre ne peut pas se trouver sur la même case que le personnage.

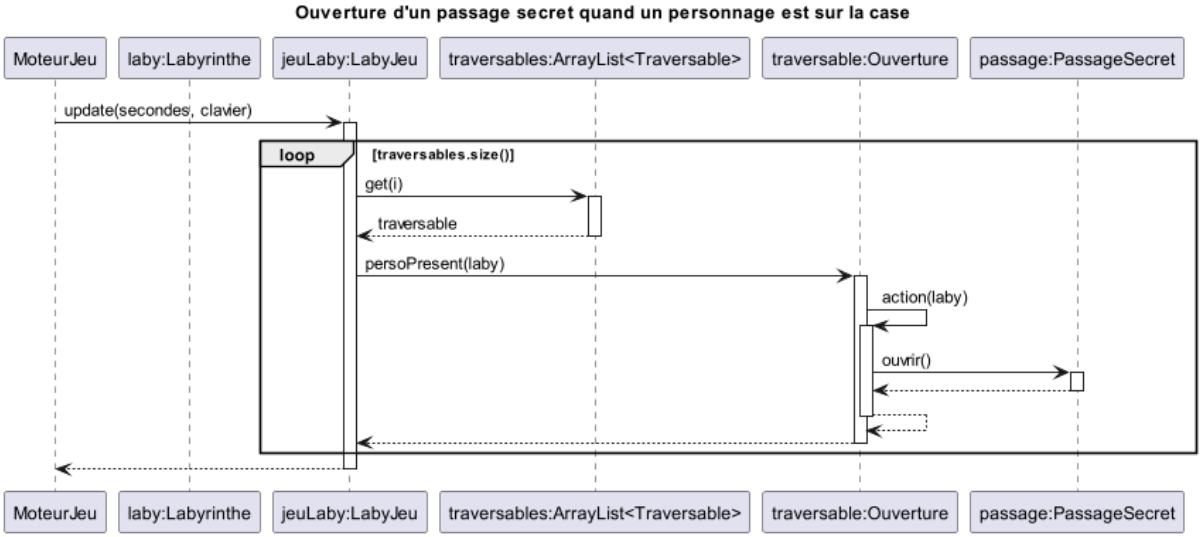
NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément

Diagramme de Classe pour LabyJeu avec Moteur de Jeu



NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément





Version 3

Répartition du travail :

NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément

1 diagramme chacun : Julien (de classe), léo, clément adrien : un diagramme de séquence chacun

Léo Fontaine : implémentation de Dijkstra

Adrien : implémentation de la classe abstraite entite et la classe monstre

Clément implémentation de tests sur les monstres

Julien : fix sur les versions précédentes

quelques difficultés d'implémentation sur les déplacements intelligent : règle de collision entre monstre, prise en compte par les monstres des autres monstres.

Monstres au comportement intelligent
Collision avec un monstre
Déplacement aléatoire des monstres

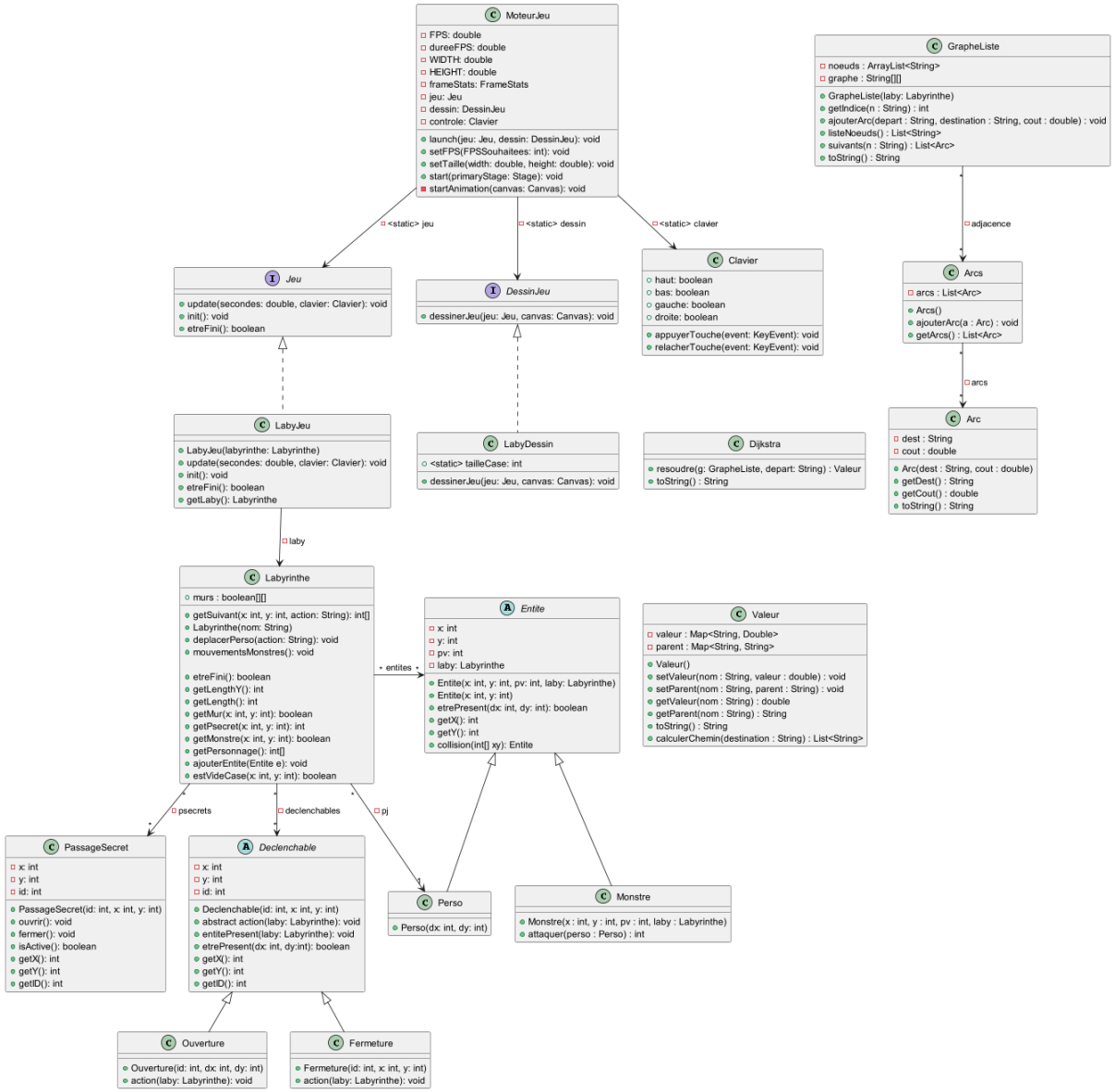
1. Monstres au comportement intelligent
Difficulté : 4

- Descriptif :
A chaque fois que les monstres se déplacent, ils se rapprochent dans la direction du héros en prenant en compte la présence des murs.

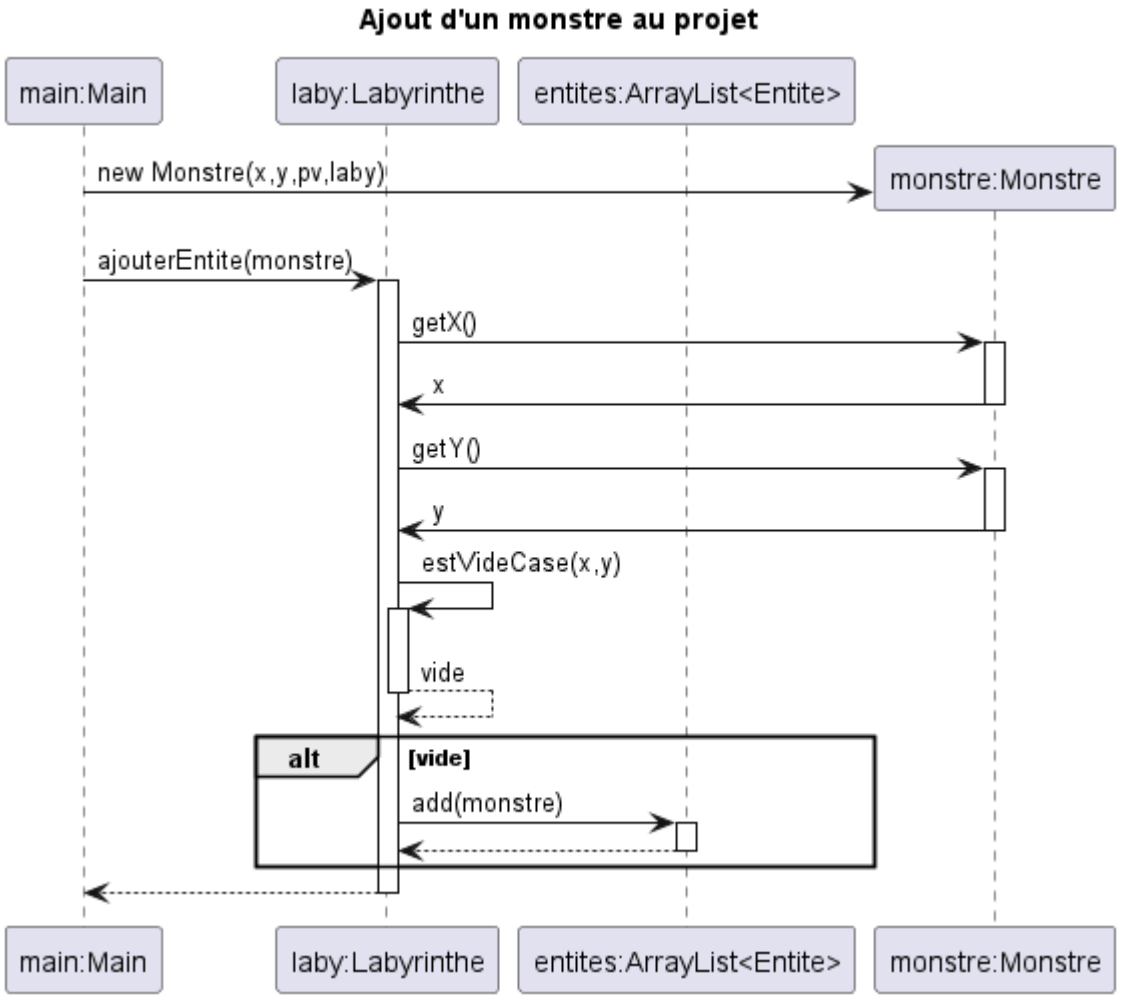
- Critères de validation :
    - Avant son déplacement, le monstre détermine le meilleur chemin pour atteindre le héros.
    - Le monstre prend en compte les obstacles qui le concernent pour trouver le meilleur chemin.
    - lorsqu'il se déplace, le monstre suit eectivement le meilleur chemin et parvient jusqu'au héros même s'il y a des obstacles.
    - vous pourrez vous inspirer (a) des transparents "IA dans le jeu vidéo" (transparents 112-130) ou (b) (c) du site redblob (avec de nombreuses démonstrations autour du jeu vidéo)
    (a) https://members.loria.fr/VThomas/mediation/JV_IUT_2016/#ia
    (b) http://www.redblobgames.com/pathfinding/a-star/introduction.html
    (c) http://www.redblobgames.com/pathfinding/tower-defense/

NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément

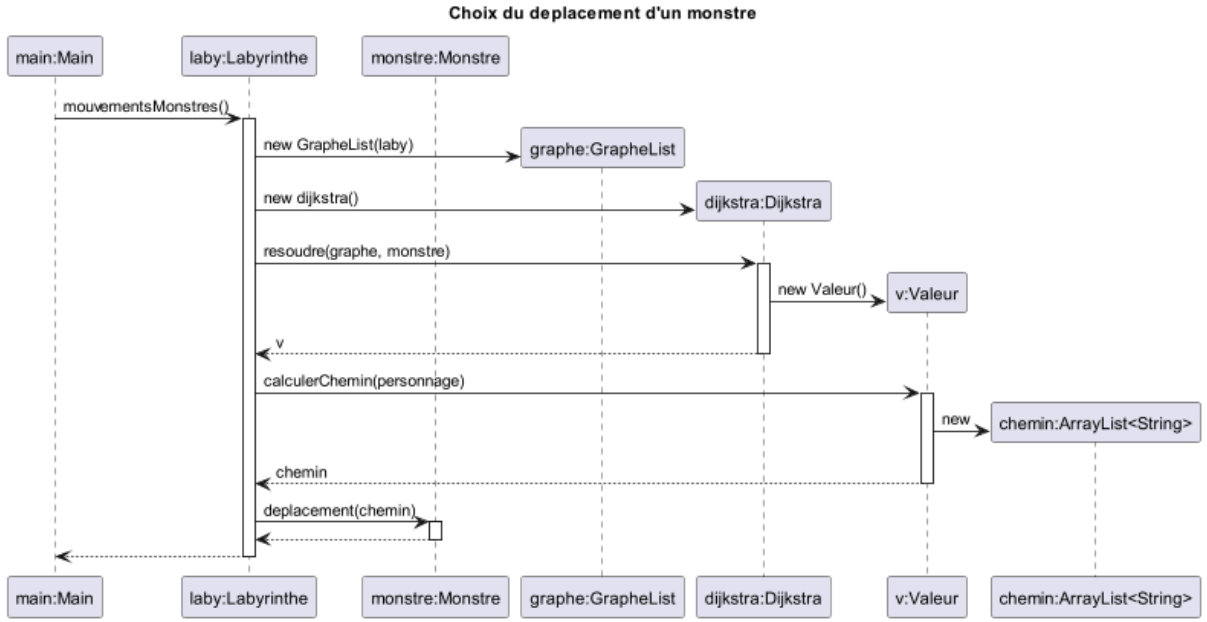
Diagramme de Classe pour LabyJeu avec Moteur de Jeu

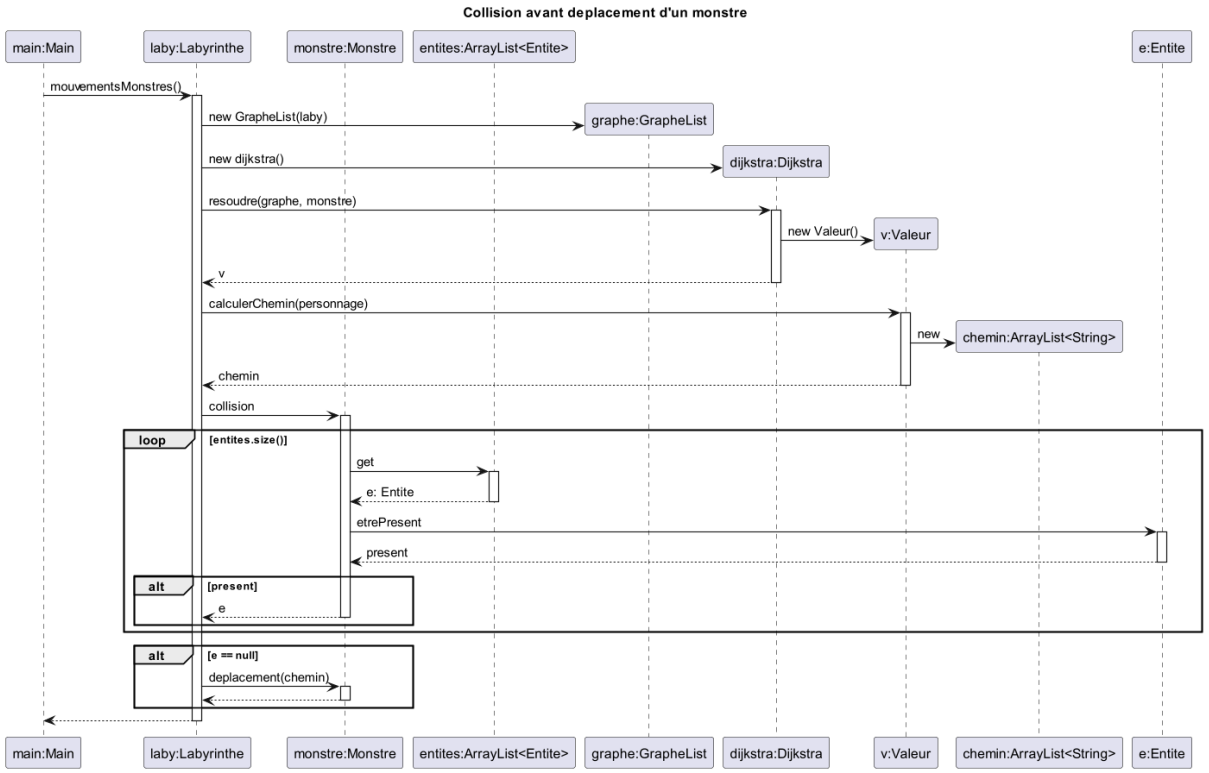


NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément



NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément





Version 4

Répartition du travail :

1 diagramme chacun :
- Julien (de classe) Fonctionnement_Diag.puml
- léo: Changement_labyrinthe.puml
- clément: Attaque_directionnel.puml

NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément

- adrien: Attaque_directionnel.puml

Léo Fontaine : méthode changement labyrinthe

Adrien : gestion de la mort + attaque directionnelle

Clément : avancement conception version 5

Julien : Tests sur les nouvelles méthodes

quelques difficultés sur les gestion de tests : Nous avons dû ajouter des getters spécifiques aux méthodes de Test dans le Main

Voici les fonctionnalités ajoutées :

- Fin du jeu : mort du héros

- Attaque du joueur

- Attaque directionnelle

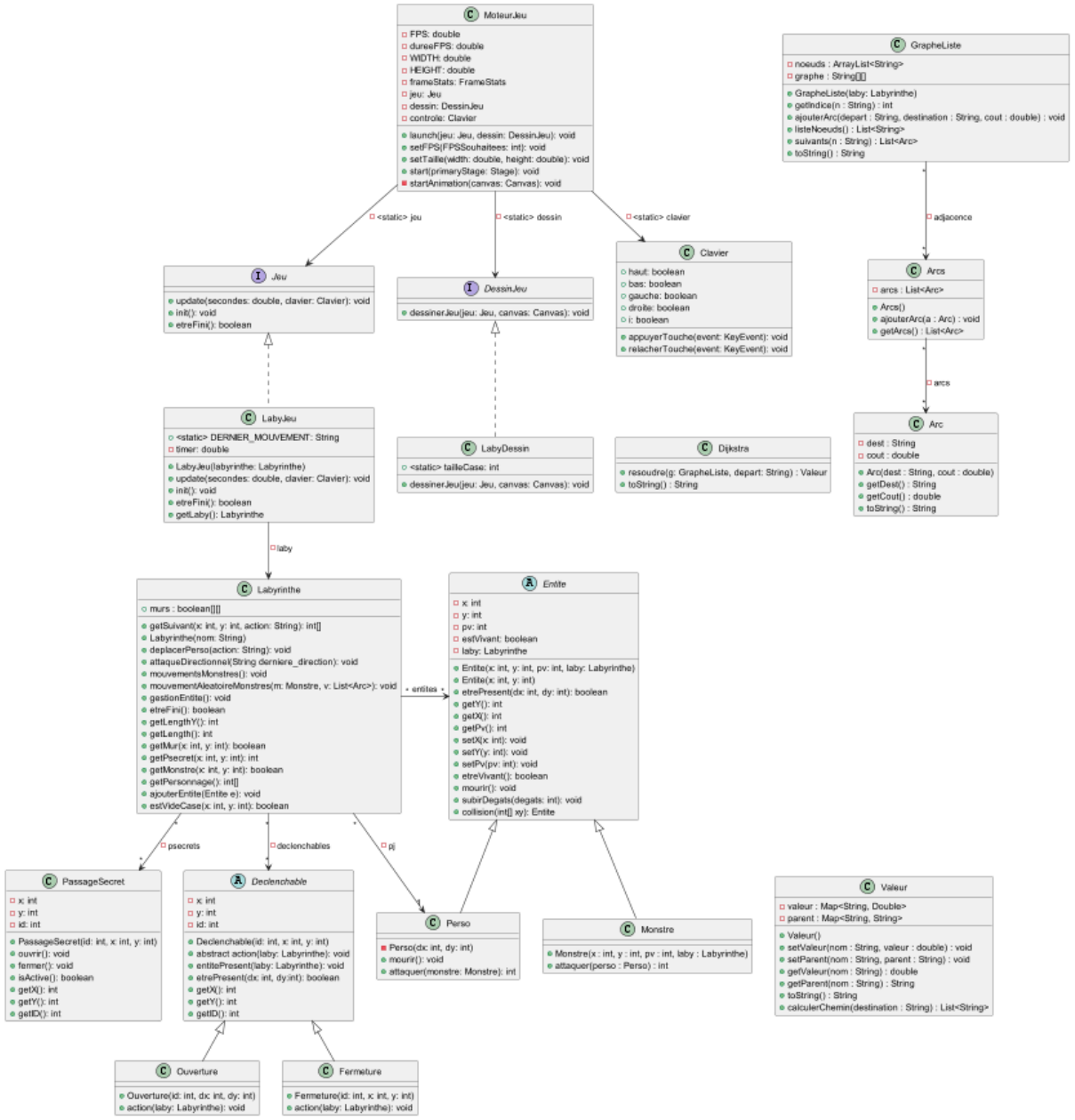
- Fin du jeu : mort du héros

- Changement de labyrinthe

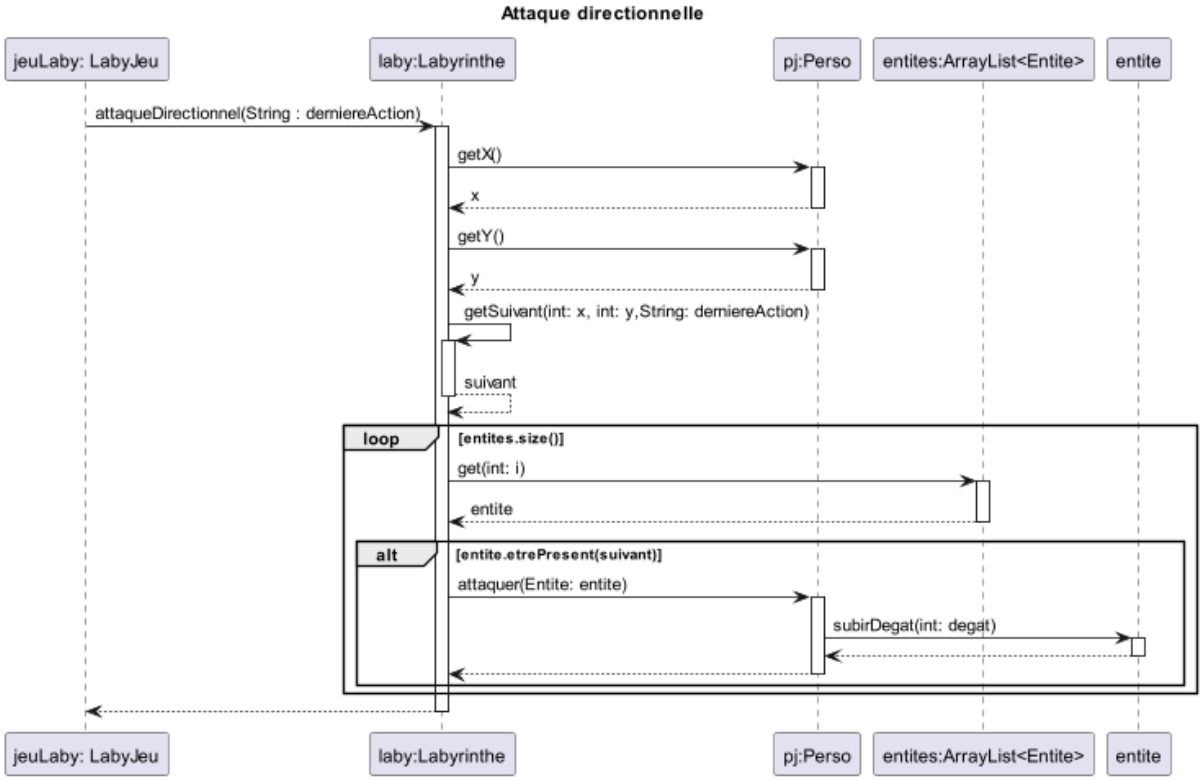
    - création d'une matrice de labyrinthe
    - changement de labyrithe lorsque le personnage atteint un bord "ouvert"

NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément

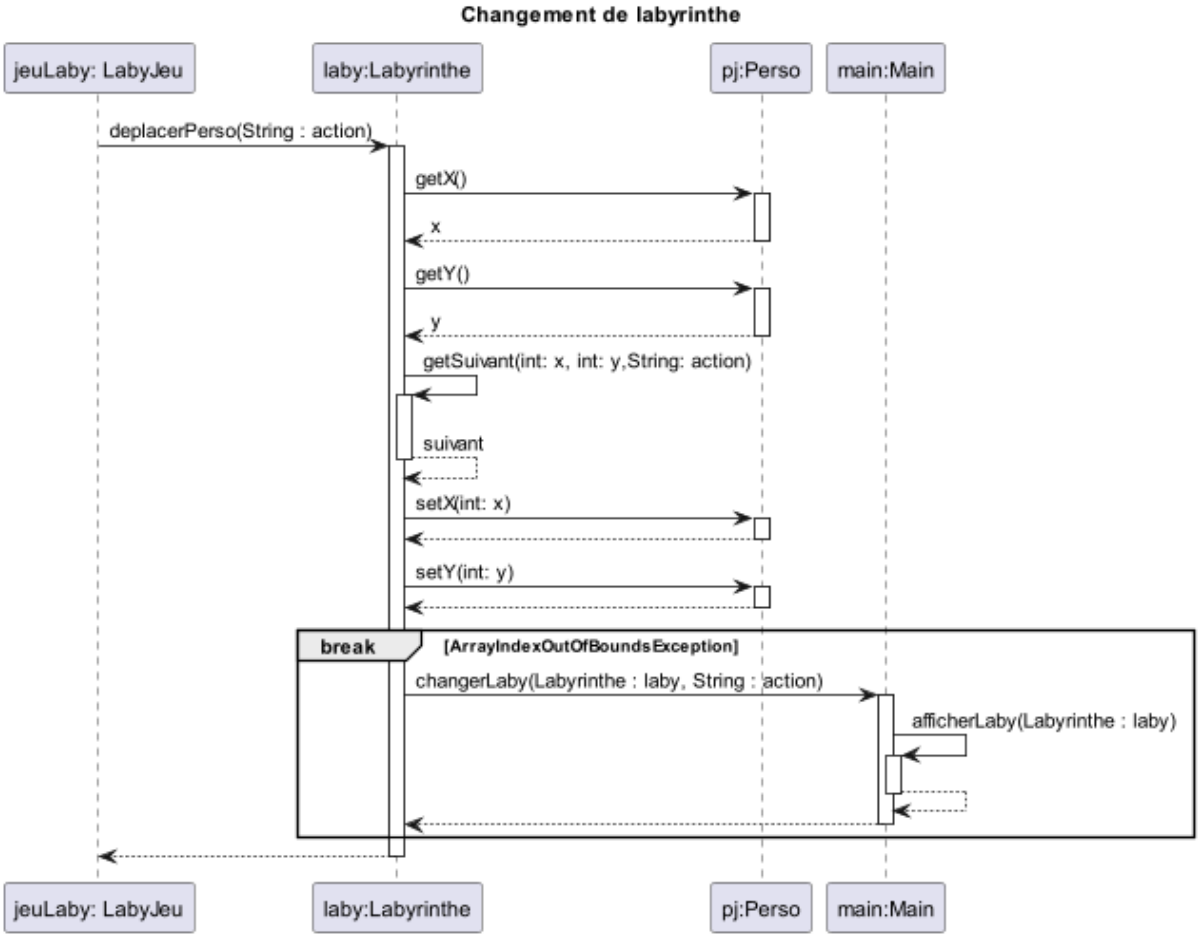
Diagramme de Classe pour LabyJeu avec Moteur de Jeu



NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément



NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément



Version 5

Mise en place de l'amulette

Au lancement du jeu, une amulette est placée sur une case vide du labyrinthe. Le placement de l'amulette est toujours le même et dépend du niveau.

Critères de validation

• L'amulette ne peut être placée que sur une case vide. : Ok
• Les monstres et le héros peuvent se situer sur la case de l'amulette. :Ok
• L'amulette est affichédans le jeu sous la forme d'un cercle jaune sur la case vide où elle se trouve. : Ok

Acquisition de l'amulette

Dès que le joueur demande à déplacer le héros sur l'amulette, le héros se déplace et prend l'amulette. : Ok

Critères de validation

• L'amulette n'est plus sur le plateau (et n'est plus affichée). : Ok
• Le Héros possède l'amulette. : Ok
• Un monstre ne récupère pas l'amulette. : Ok

NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément

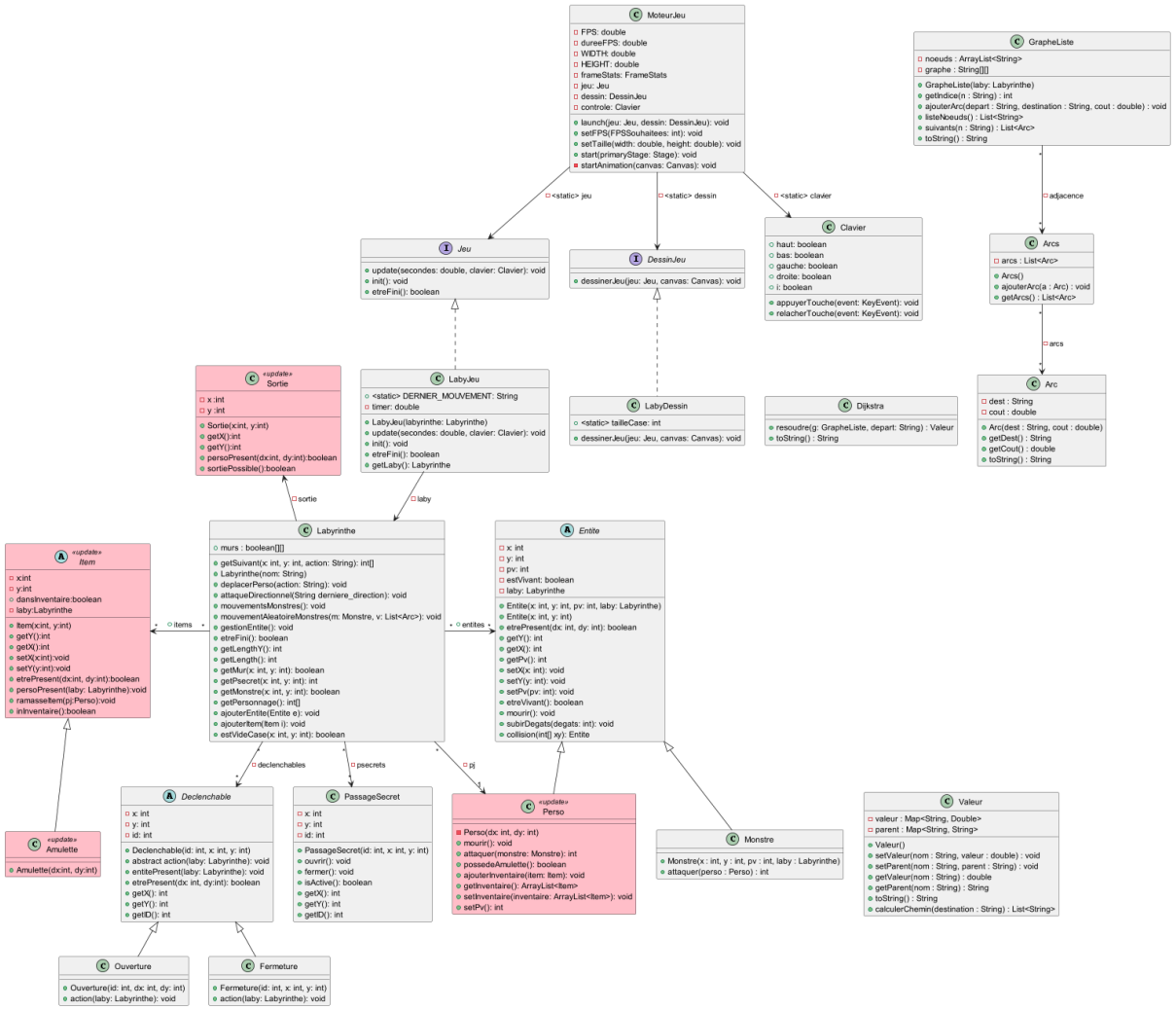
Fin du jeu : victoire du héros
Une fois que le héros a pris l'amulette, il peut retourner à l'entrée du labyrinthe et remporter le jeu. : Ok

Inventaire
Des objets (sans utilité pour le moment) sont disposés dans des cases vides du labyrinthe, lorsque le héros appuie sur la touche d'utilisation ("E" par défaut), il récupère les objets de la case où il se trouve et les ajoute dans son inventaire. : Ramassage automatique amulette
Critères de validation
• L'inventaire est de taille infinie. : Ok
• Lorsque le joueur cherche à prendre un objet alors qu'il n'y a rien sur la case où il se trouve, rien ne se passe. : ramassage auto
• Lorsqu'un joueur prend un objet, l'objet disparaît du labyrinthe (et n'est plus affiché) mais apparaît dans son inventaire.
• À chaque évolution du jeu, le jeu affiche dans la console l'inventaire du héros.
• Les objets sont affichés dans le labyrinthe tant qu'ils sont présents (sous la forme d'un cercle noir) : Utilisation de sprite

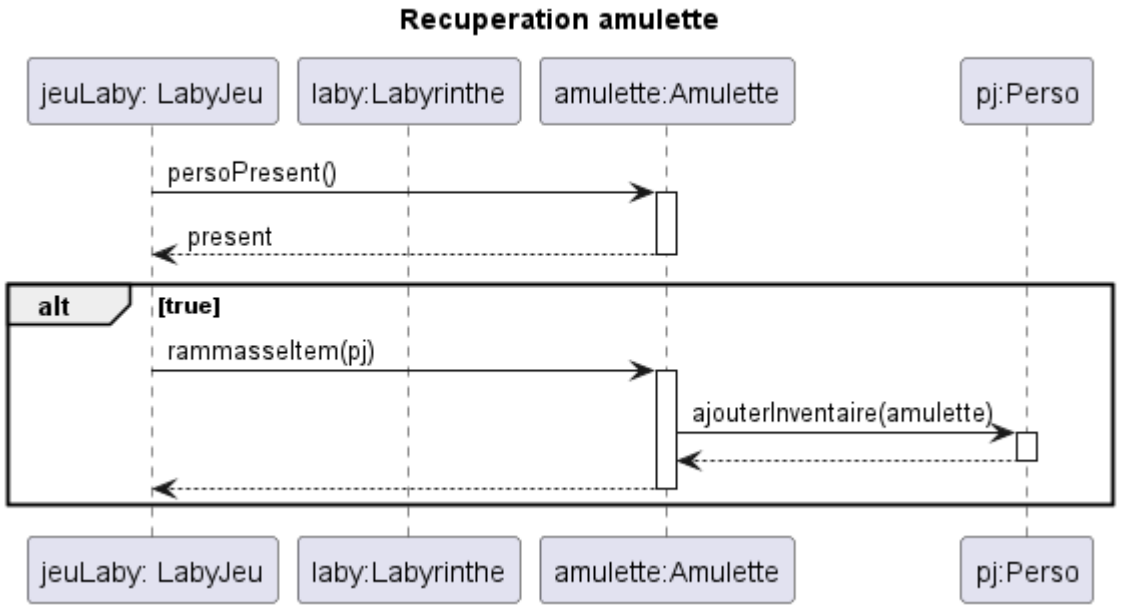
Affichage de l'inventaire
L'inventaire est affiché dans la fenêtre de jeu.
Critères de validation
• L'inventaire est représenté par 6 cases carrées en bas de la fenêtre de jeu : Ok
• A chaque fois qu'un joueur prend un objet, en plus de disparaitre dans le labyrinthe, celui-ci s'ache dans l'inventaire : à voir
• Les objets sélectionnés (meilleure arme et meilleur bouclier) sont d'une couleur différente.
• Les objets affichés dans l'inventaire ont des formes spécifiques (rond pour un bouclier et un trait vertical pour une arme). Modification : utilisation d'image
• Lorsqu'un bouclier est détruit, il disparait de l'inventaire. : Pas de bouclier implémenté

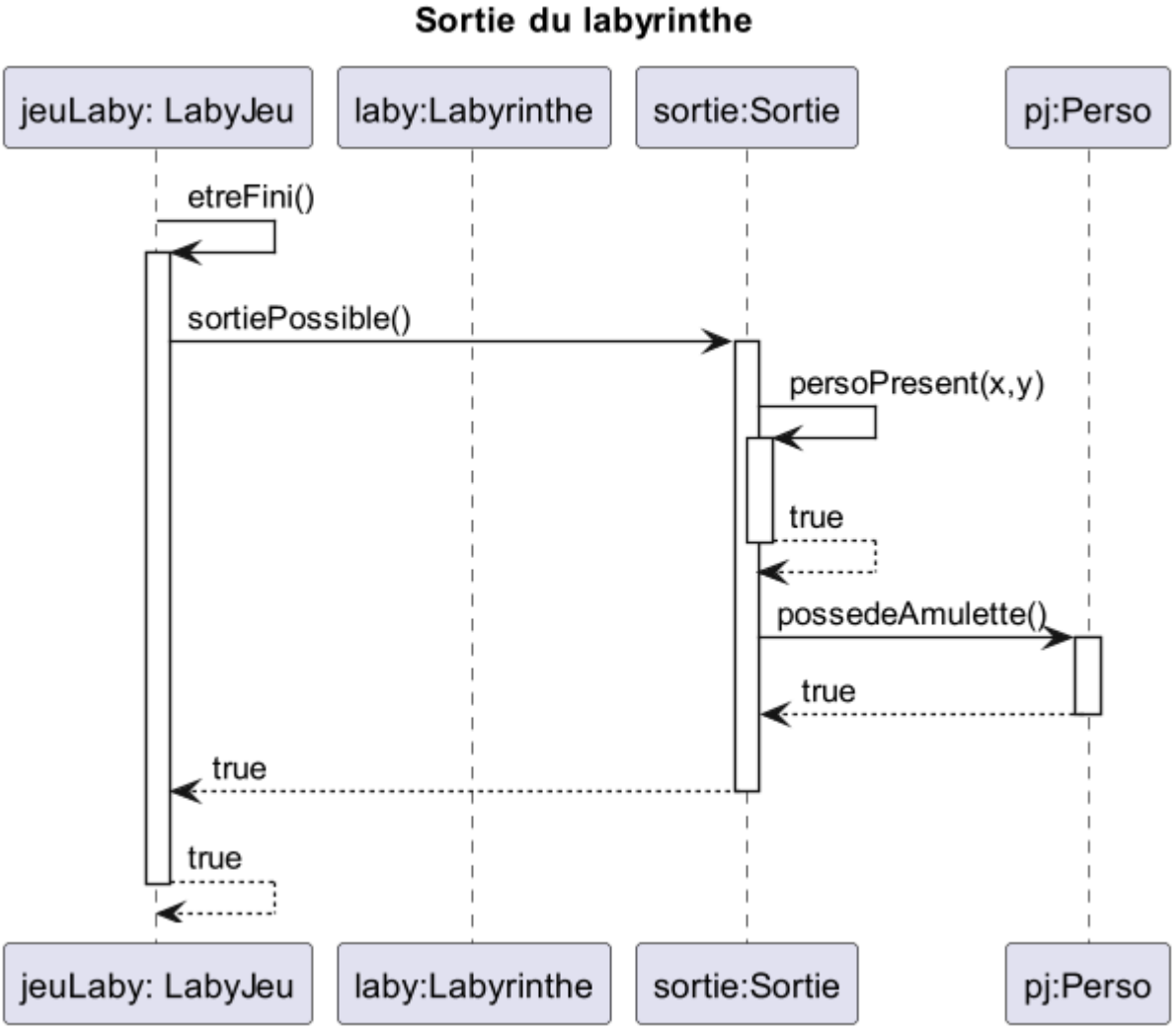
NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément

Diagramme de Classe pour LabyJeu avec Moteur de Jeu



NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément





NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément

Version 6

Fantômes

Le labyrinthe dispose d'un nouveau type de monstre : les fantômes.
Les fantômes peuvent traverser des murs mais pas les autres personnages (les autres monstres ou le héros)

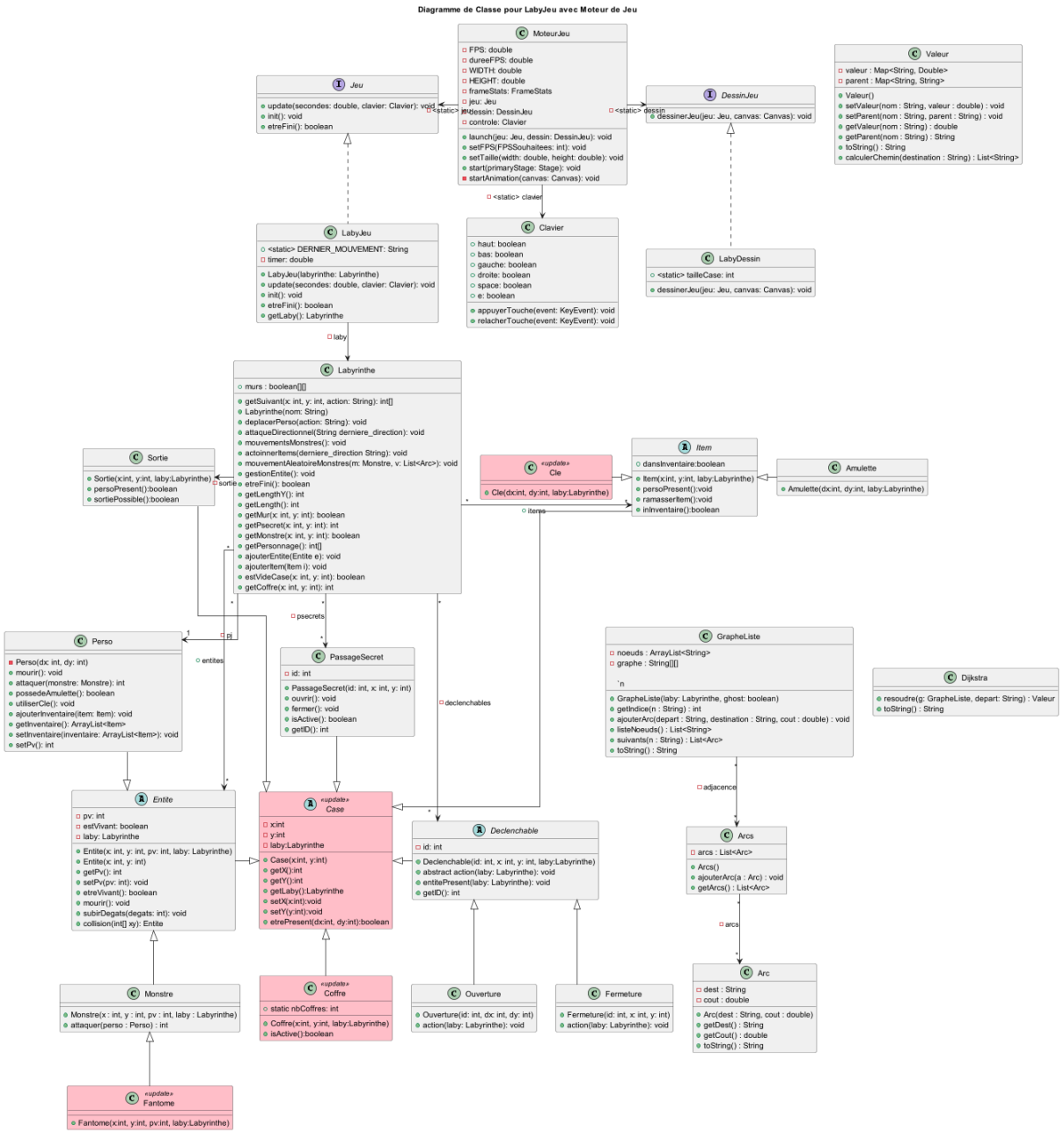
• Les autres monstres continuent à être bloqués par des murs. : OK
• Les attaques des fantômes fonctionnent de la même manière que les attaques des monstres de base. : OK

Implémentation d'un coffre et d'une clef

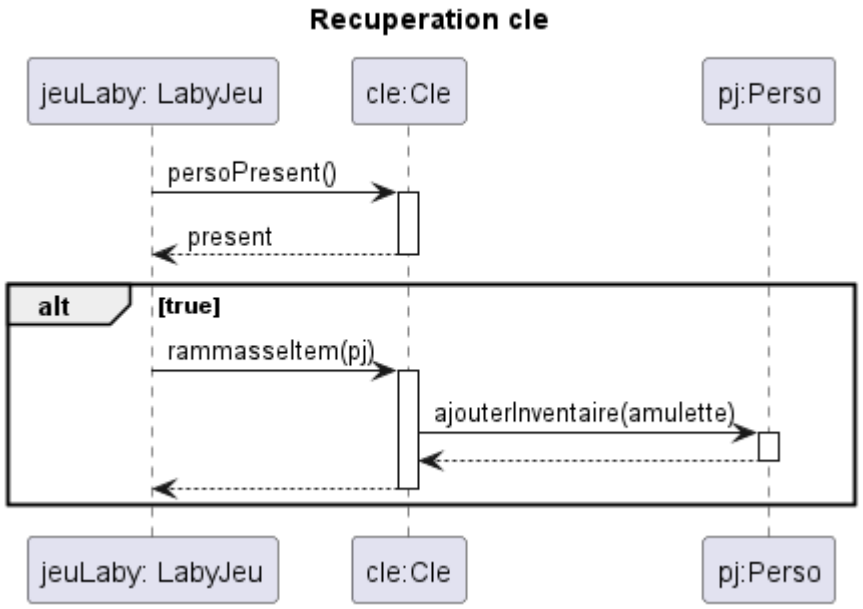
-une clef apparait sur le labyrinthe et permet d'ouvir un coffre : OK
-donne de manière aléatoire : un arc ou une épée : OK

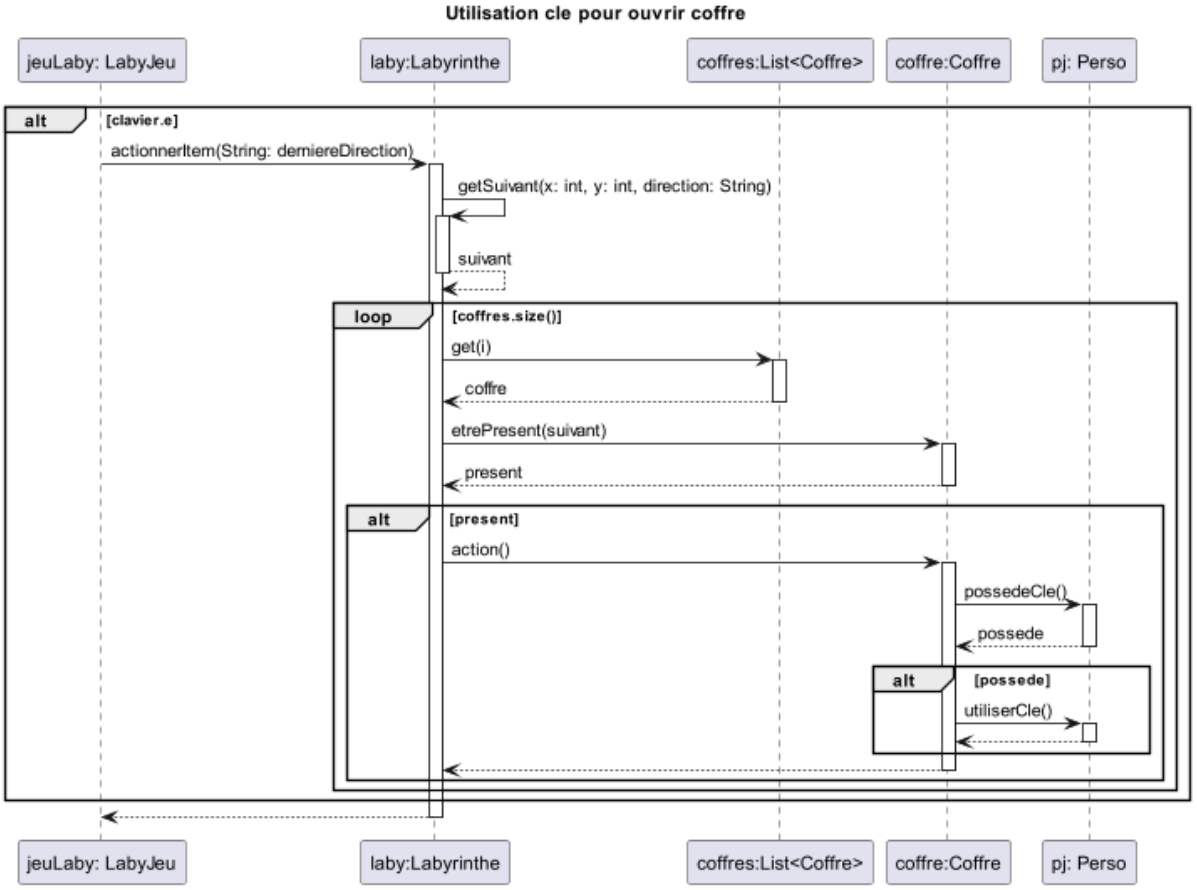
NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément

Diagramme de Classe pour LabyJeu avec Moteur de Jeu

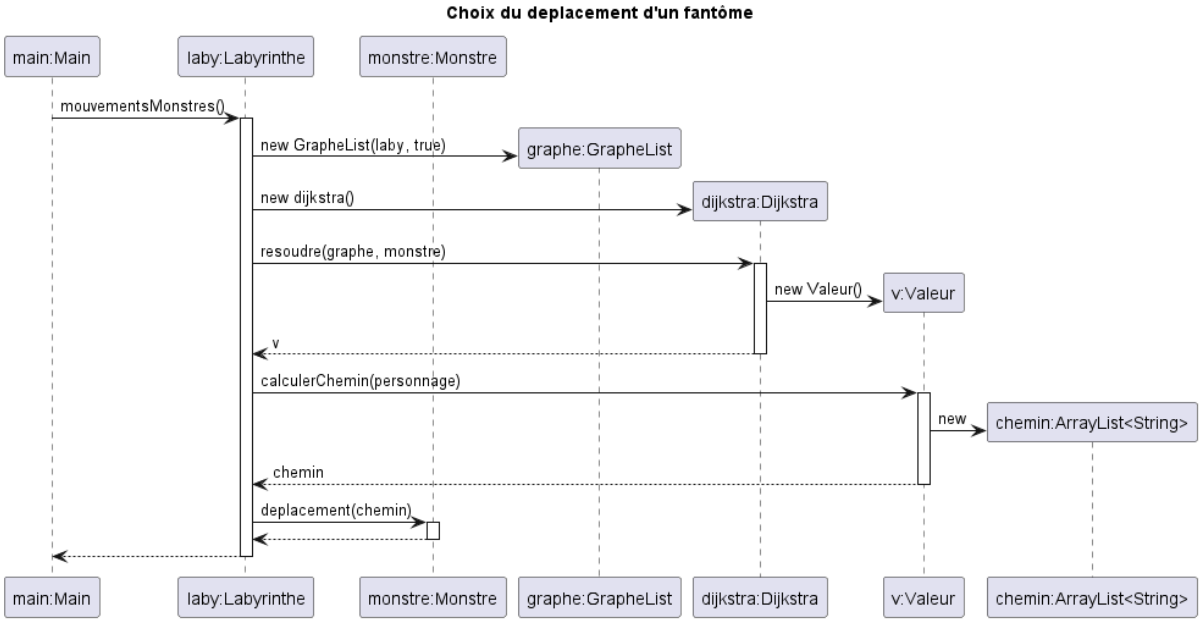


NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément





NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément



Version 7

Armes

Les armes sont des objets que le héros peut prendre. Une arme augmente d'un certain nombre les dégâts du héros.

Critères de validation

• Une arme est définie par ses dégâts.

• Lorsque le héros prend une arme, l'arme est ajoutée à son inventaire et disparait du labyrinthe. : OK

• Lorsqu'un héros possède une arme et attaque un monstre, son attaque est augmentée des dégâts de l'arme. : OK

• Un héros ne peut utiliser qu'une seule arme à la fois. Lorsqu'un héros possède plusieurs armes dans son inventaire, il s'équipe automatiquement de l'arme la plus forte. : OK : OK

Affichage de l'inventaire

L'inventaire est affiché dans la fenêtre de jeu.

Critères de validation

• L'inventaire est représenté par 6 cases carrées en haut de la fenêtre de jeu : OK

• A chaque fois qu'un joueur prend un objet, en plus de disparaitre dans le labyrinthe, celui-ci s'affiche dans l'inventaire : OK

• Les objets sélectionnés (meilleure arme et meilleur bouclier) sont d'une couleur diérente. : NON

NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément

• Les objets affichés dans l'inventaire ont des formes spéciques (rond pour un : Sprite bouclier et un trait vertical pour une arme).
• Lorsqu'un bouclier est détruit, il disparait de l'inventaire. : PAS DE BOUCLIER

Attaque à distance du joueur

Attaque à distance du joueur
Diculté
Le joueur dispose d'une touche particulière pour tirer à l'arc. Lorsqu'il appuie sur cette touche ("f" par défaut), le héros lance une flèche qui traverse l'écran dans la direction suivie par le héros.
Critères de validation
• La èche se déplace de manière rectiligne à partir du héros et selon la direction : ok de son dernier déplacement.
• la èche se déplace à la vitesse des monstres et du héros (cf tâche suivante) : ok
• Lorsqu'elle rencontre un obstacle (monstre ou mur), la èche disparait et fait 1 point de dégât si c'est un monstre. : ok
• Tant qu'elle n'est pas détruite, la èche se déplace d'une case par évolution du jeu. : ok
• La èche est achée dans la fenêtre de jeu sous la forme d'un petit point rouge centré dans la case où la èche se trouve. : ok

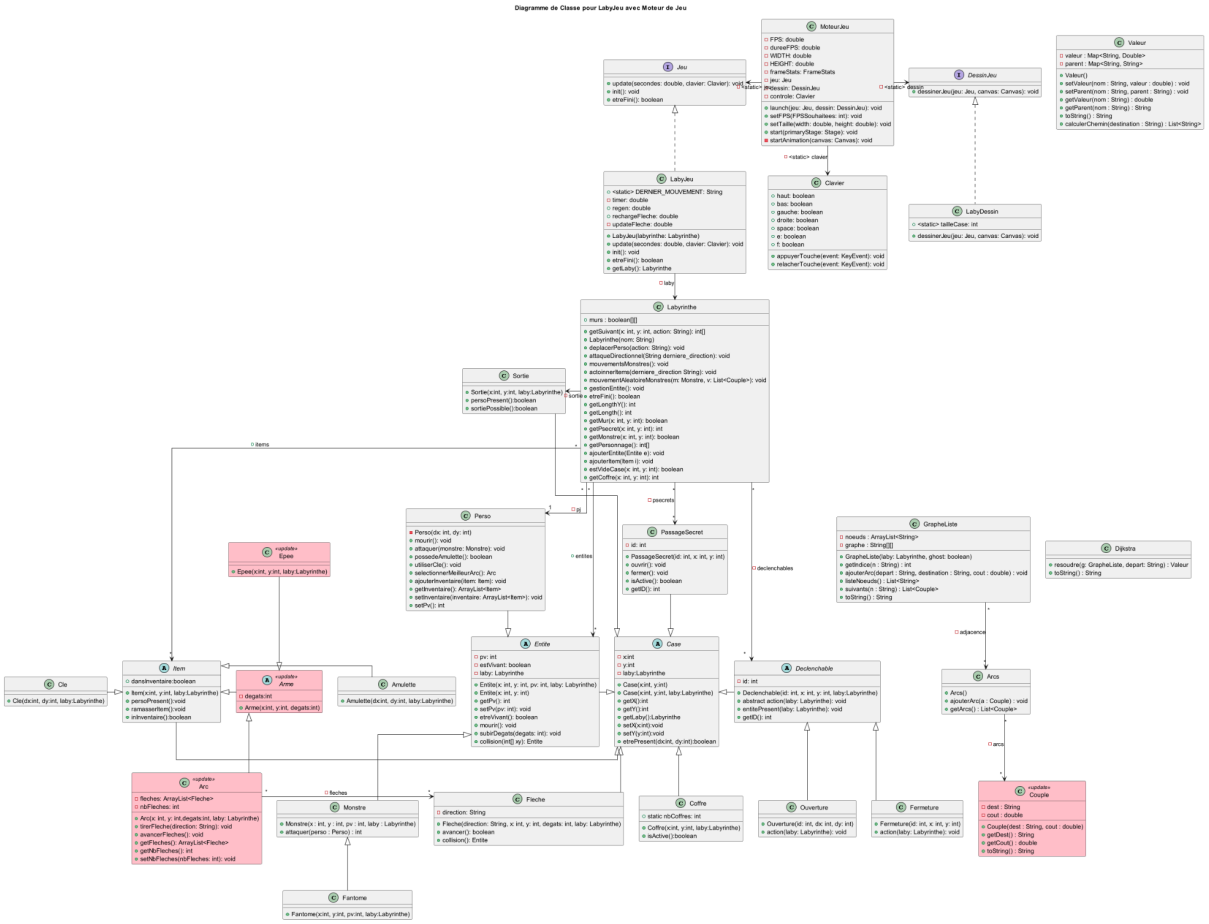
Vitesse des èches
Diculté
Les èches lancées par le joueur vont deux fois plus vite que les déplacements des monstres et du joueur.
Critères de validation
• A chaque évolution du jeu, la èche se déplace de deux cases. : ok
• La détection d'obstacles se fait dans toutes les cases traversées (pas juste la première et la dernière : ok

Retour sur Inventaire, objet ramasser lorsque le joueur passe dessus

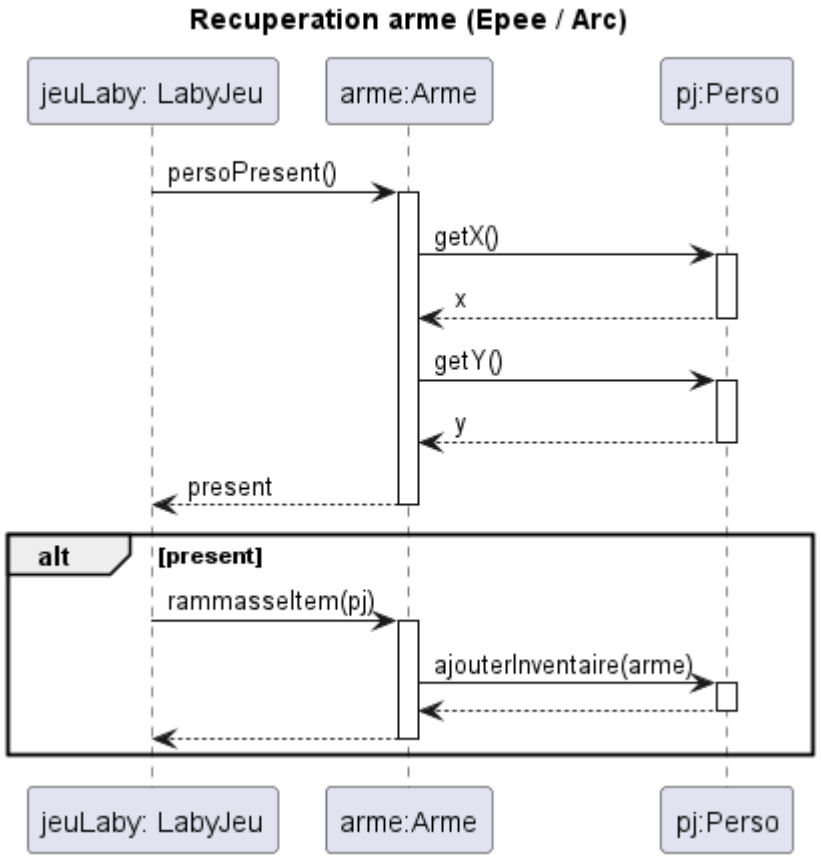
Inventaire
Diculté : 88999
Des objets (sans utilité pour le moment) sont disposés dans des cases vides du labyrinthe, lorsque le héros appuie sur la touche d'utilisation ("E" par défaut), il récupère les objets de la case où il se trouve et les ajoute dans son inventaire.
Critères de validation
• L'inventaire est de taille innie.
• Lorsque le joueur cherche à prendre un objet alors qu'il n'y a rien sur la case où il se trouve, rien ne se passe.

NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément

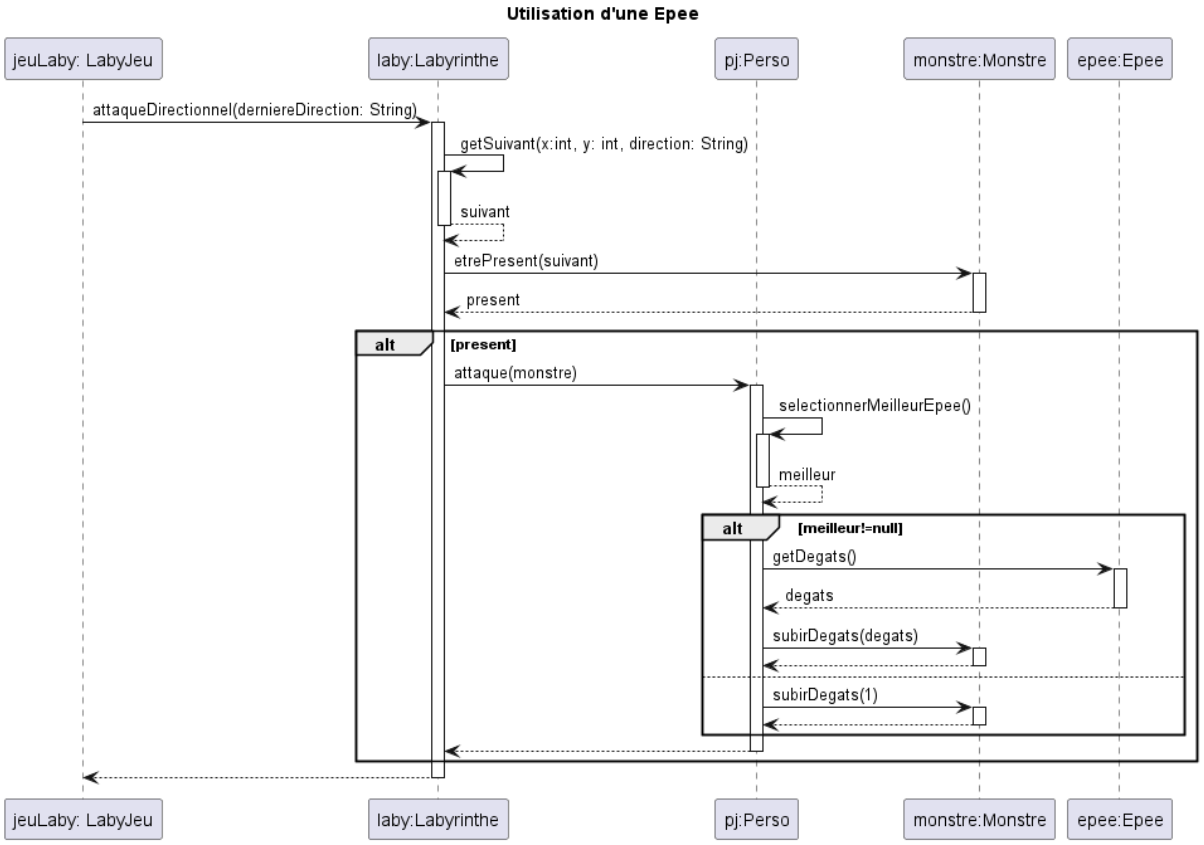
• Lorsqu'un joueur prend un objet, l'objet disparait du labyrinthe (et n'est plus affiché) mais apparait dans son inventaire.
• A chaque évolution du jeu, le jeu ache dans la console l'inventaire du héros.
• Les objets sont achés dans le labyrinthe tant qu'ils sont présents (sous la forme d'un cercle noir)

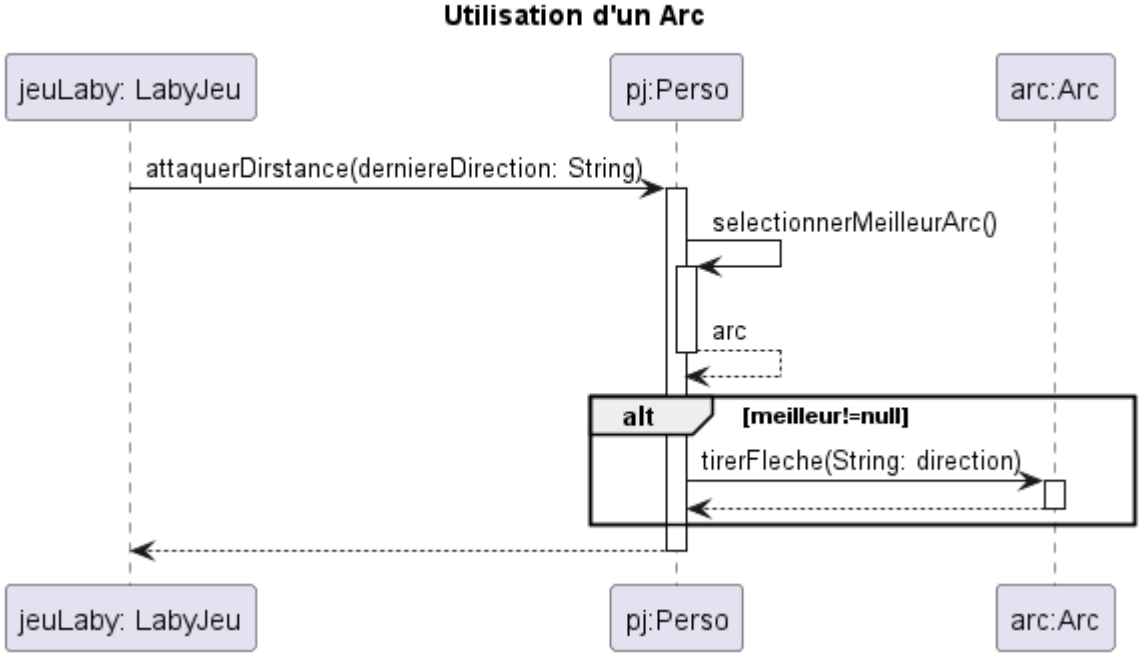


NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément

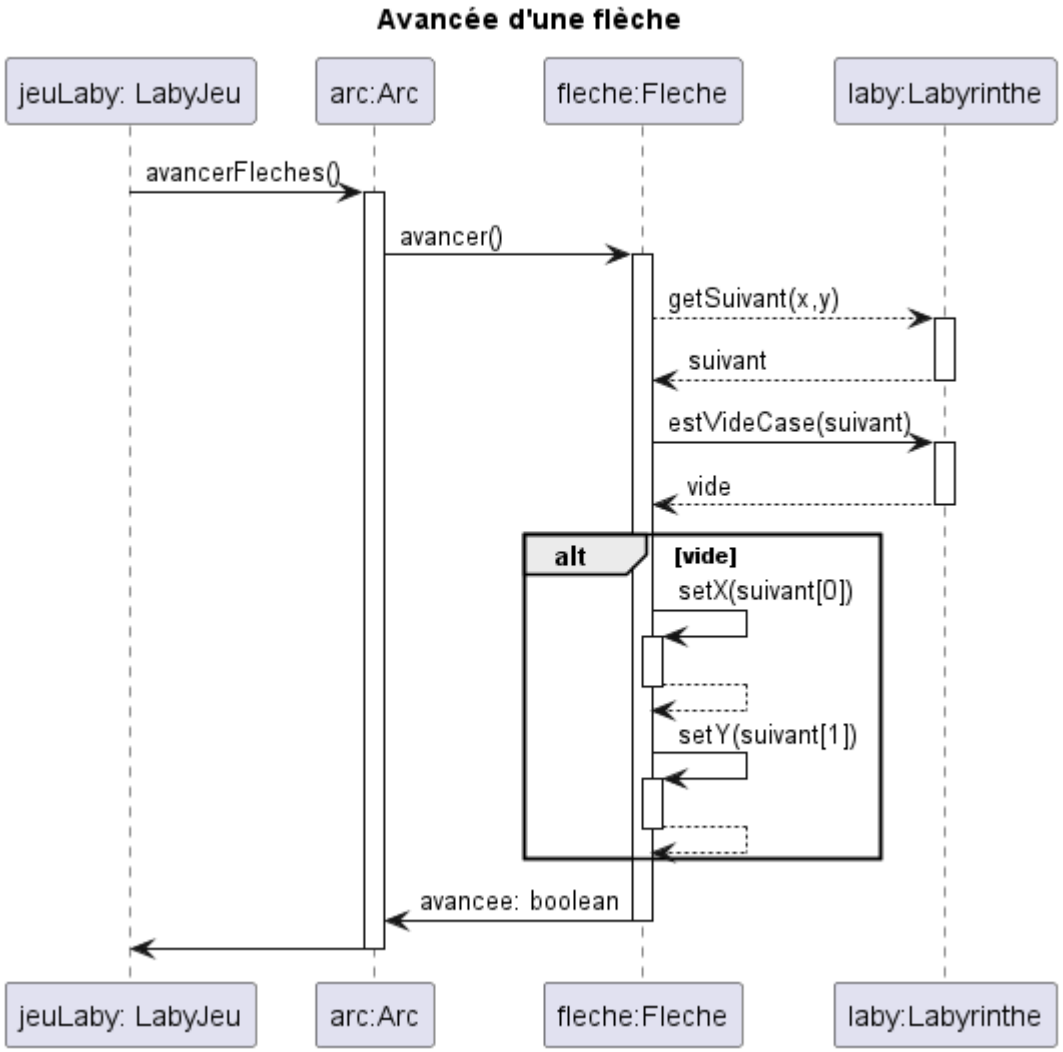


NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément





NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément



NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément

2.3. Répartition globale

Globalement, nous avons essayé de maintenir un équilibre de la charge de travail pour chaque membre du groupe dans les différents domaines du projet.

La répartition était différente à chaque fois, mais plutôt égalisée, de manière à ce que chacun puisse participer à la création des digrammes et l'écriture du code.

3. Développement du Projet

3.1. Implémentations

Léo Fontaine : Implémentation de l'algorithme de Dijkstra

Nous avons réutilisé l'algorithme de Dijkstra utilisé dans une SAÉ précédente.

Adrien : Implémentation de la classe abstraite Entité et la classe Monstre

Clément : Implémentation du Coffre, de l'Inventaire et de l'amulette ainsi que l'écriture de quelques Tests

Julien : Écriture des Tests et ajout de nouvelles fonctionnalités qui n'ont pas vu le jour

4. Problèmes rencontrés et Solutions

Difficultés d'implémentation des déplacements intelligents

Problèmes de collision entre monstres

Nous avons rencontré quelques problèmes de collision avec les monstres, notamment avec des monstres morts, qui n'étaient plus visibles, mais toujours présents sur le jeu.

Nous avons réglé ce problème après sa découverte.

NOEL Julien
MANGIN Adrien
FONTAINE Léo
DEWASCH Clément

Implémentation de l'algorithme de Dijkstra pour palier aux problèmes de déplacement des monstres qui n'étaient pas en mesure d'attaquer le joueur efficacement

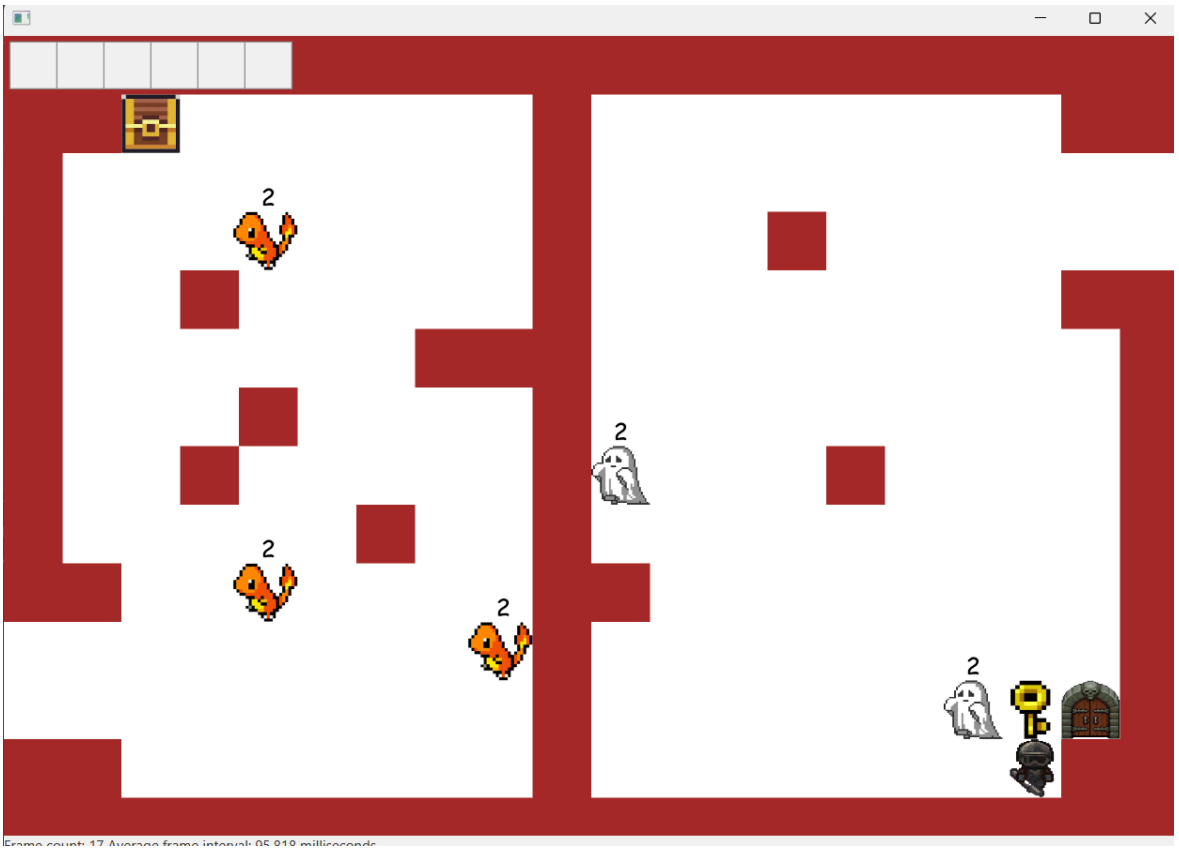
5. Conclusion

En conclusion, nous avons tous apprécié ce projet, qui nous a permis d'appliquer nos connaissances acquises durant l'année.

Toutefois, nous aurions pu mieux nous organiser pour éviter certains moments d'incompréhension et de confusion, notamment sur la conception des diagrammes de classes et de séquence en amont, qui nous aurait permis d'avancer plus rapidement et plus efficacement.

6. Annexes

Captures d'écran du système en fonctionnement



Références et ressources utilisées

Nous avons utilisé le langage Java ainsi que des diagrammes codés en PlantUML