RAPPORT DU PROJET ACL 2018

Réalisé par
LINDAMULAGE Olivier, MANICOM Sandyla, JEBROUN Youssef, SAINT-JORE Amaury, IMAGOUREN Ilhame

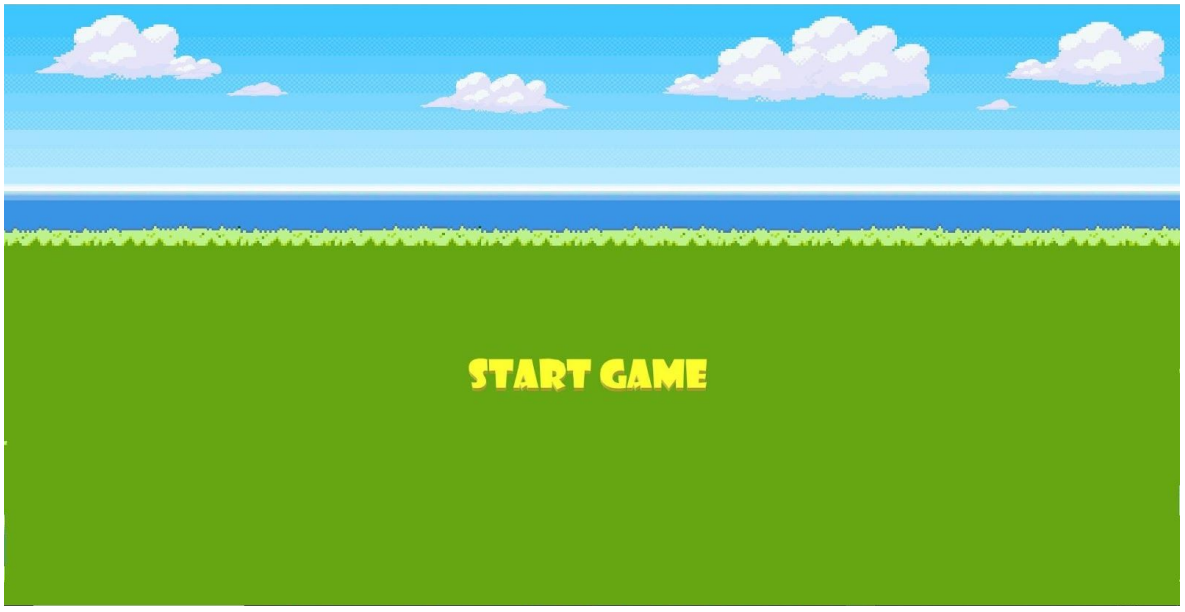

Image 1 : interface du début du jeu

N.B : Ce rapport détaille notre projet de conception d'un jeu vidéo réalisé dans le cadre du module "Analyse et conception de logiciels".

PLAN

Ci-après se trouve le concept game du jeu vidéo qui explique les grandes lignes de ce projet.

I. Game concept

- Fiche signalétique
  - Genre : arcade
  - Support : ordinateur
  - Cible : 3+
- Concept

Le jeu consiste à déplacer un personnage dans un labyrinthe dans le but de trouver un trésor. Dans le labyrinthe, il peut y avoir des monstres de différents types qui essaient d'attaquer le personnage.

- Conditions de victoire/ défaite

L'objectif pour le héros est d'arriver au trésor en vie. La partie est perdue si le héros n'a plus de points de vie. Le héros perd des points de vie lorsque qu'il tombe sur les cases pièges ou rencontre des monstres.

- Gameplay principal

Environnement : L'histoire se déroule dans un labyrinthe. Le héros doit essayer de s'en sortir à travers une dédale de murs.

Contrôles du jeu : Le jeu se déroule en vue de haut, de façon objective en 2 dimensions. Le joueur incarne le personnage. Il peut choisir le niveau de difficulté souhaité entre trois disponibles (facile, normal, difficile). Le héros peut se déplacer dans le labyrinthe et interagir avec divers éléments contextuels. Si le personnage arrive sur une case piège, il subit des dégâts. Il peut également arriver sur des cases magiques qui lui donnent des effets ou accéder à des passages qui permettent de le téléporter.

- Narration
  - Scénario normal

Le scénario normal serait que le héros puisse se déplacer dans un plateau de jeu de type labyrinthe et du niveau sélectionné par le joueur.
Le joueur doit pouvoir aussi choisir le niveau de difficulté du jeu (en trois niveaux) qui influe sur la complexité du labyrinthe ainsi que sur le nombre de monstres.
A l'instant initial, le héros les monstres, les pièges, les cases magiques et le trésor sont positionné aléatoirement dans des espaces admissibles de jeu.
Les caractères présents dans le jeu ne peuvent traverser les murs à l'exception d'un type de monstre : les fantômes.

Chaque caractère présent sur le plateau présente des points de vie, le héros est le caractère dans le jeu possédant le plus de point de vie. Celui-ci peut attaquer les monstres seulement sur des cases adjacentes, une attaque d'un monstre provoque sa mort au détriment d'un point de vie. Le héros meurt s'il ne possède plus de points de vie.

- Scénario exceptionnel

Il peut arriver que le fichier qui permet la génération du labyrinthe soit défectueux ou que le joueur soit placé à l'instant initial du jeu sur une place non admissible (mur, extérieur du plateau etc..).
Un autre possibilité serait que le graphisme ne suive pas le fil du jeu (déplacement du héros non désiré).

II. Diagramme de use-case

❖ Schéma

Le diagramme de use-case permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système. Il permet aussi de délimiter le système.
L'acteur représente un élément externe qui interagit avec le système. Dans notre cas, comme on peut le voir sur l'image 2 ci-après, l'utilisateur est l'acteur. Il est représenté par un bonhomme.

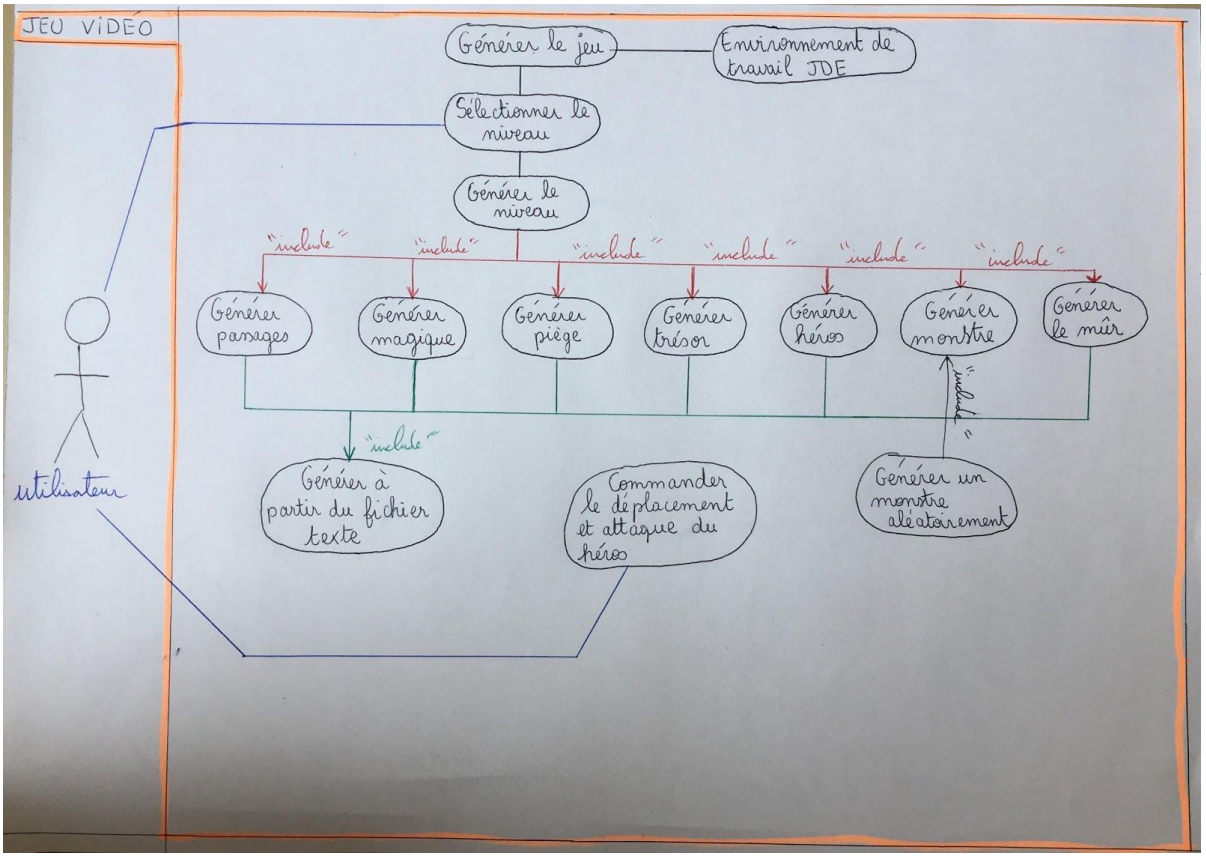

Image 2 : première version du diagramme use-case

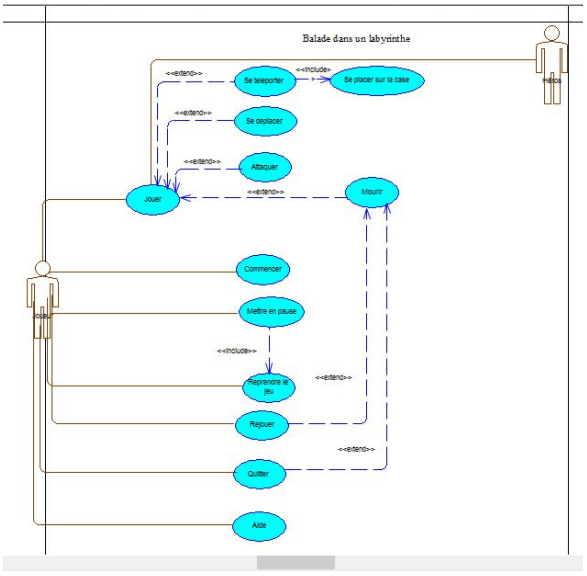

Image 3 : deuxième version du diagramme use-case (version numérique)

❖ Détails des fonctionnalités

Maintenant, nous devons définir les objectifs en détaillant les fonctions. Ci-après les backlogs :

- Fonctionnalités du labyrinthe (plateau de jeu) :
  - se génère à partir d'un plateau de base et rajout de manière aléatoire des murs sur le plateau
  - chaque niveau contient les cases spéciales (pièges, passages, magiques, trésor)

- Fonctionnalités du piège :
  - si le joueur arrive sur la case piège, il subit des dégâts

- Fonctionnalités du passage :
  - si le joueur arrive sur une case magique de type téléportation, le héros est téléporté à un autre endroit

- Fonctionnalités de la case magique :
  - si le joueur arrive sur la case magique, un effet bonus aléatoire se déclenche

- Fonctionnalités du trésor :
  - case de fin de niveau, le joueur gagne s'il l'atteint. Cela termine le niveau.

- Fonctionnalités du héros :
  - le placer de manière aléatoire
  - se déplacer dans le labyrinthe
  - attaquer les monstres de la case adjacente
  - attaquer les monstres en cas de contact
  - perdre des points de vie
  - mourir en cas d'attaque de monstre

- Fonctionnalités des monstres :

  Monstres de type normaux :
  - sont placés de manière aléatoire
  - certains se déplacent aléatoirement
  - certains essaient d'attraper le héros de manière intelligente

  Monstres de type fantôme :
  - sont placés de manière aléatoire
  - certains se déplacent aléatoirement
  - certains essayent d'attraper le héros de manière intelligente

- peuvent traverser les murs (ne prennent pas en compte la collision)

- Fonctionnalités de l'affichage (moteur graphique) :
  - affiche dans une fenêtre les sprites des objets

❖ Développement des scénarios

On peut développer le scénario de chacune de ces fonctionnalités :

Scénario pour "Déplacer le héros"

Nom : Déplacer le héros
Résumé : Permet au héros de se déplacer selon la volonté de l'utilisateur
Date : 10/10/2018
Auteur : Manicom Sandyla
Pré-condition : Le héros doit être placé dans le labyrinthe
Post-condition : le héros peut se déplacer sur le plateau
Scénario nominal :

1. Positionner le héros à l'entrée du niveau
2. L'utilisateur définit la direction souhaitée sur le clavier : S'il appuie sur la touche 7, le héros va vers le bas à gauche, sur la touche 8, vers le bas, sur la touche 9, vers le bas à droite, sur la touche 4, vers la gauche, sur la touche 6, vers la droite, sur la touche 1, vers le haut à gauche, sur la touche 3, vers le haut à droite.
3. Le héros se déplace sur le plateau si déplacement possible (gestion de collisions)

Scénario alternatif :
A1. L'utilisateur dirige le héros vers un mur alors qu'il est sur une case adjacente à ce mur -> retourne vers 3
A2. Le héros est sur une case piège : celui-ci perd 1 point de vie
A3. Le héros est sur une case magique: le héros récupère la propriété magique associé
Scénario exceptionnel :

E3. Il y a un bug au niveau des commandes et le héros ne va pas à droite quand le joueur appuie sur la flèche de droite.

Scénario pour "se téléporter"

Nom : se téléporter
Résumé : permet au héros de se déplacer dans un plateau de jeu de type labyrinthe
Date : 10/10/2018
Auteur : IGOURRAMEN Ilhame
Pré-condition : Le héros a récupéré une case magique contenant l'attribut téléporter
Post-condition : Le héros est sur une case différente après avoir taper sur une touche
Scénario nominal :

1 Le héros est sur une case Magique
2. La case magique est de type téléportation
3 une icone indiquant que le héros peut se téléporter s'affiche à l'écran
4. Après que le héros ai essayé de ce déplacer, celui-ci est replacé sur une autre case aléatoire du plateau.

Scénarios alternatifs :

4a. Les coordonnées (x,y) ne sont pas valides correspondent à un mur ou ne sont pas sur le plateau= Retourner à l'étape 3

Scénario pour "Générer un plateau aléatoire"

Nom : Générer un plateau aléatoire
Résumé : Afin de dynamiser le jeu ,un plateau différent sera généré à chaque jeu avec des mur placé de manière aléatoire sur le plateau
Date : 11/10/2018
Auteur : Lindamulage Olivier
Pré-condition : Le plateau de base sans mur est générer
Post-condition : Un plateau de jeu exploitable est généré
Scénario nominal :

1. Un plateau de n lignes et n colonnes est généré contenant seulement les murs extérieurs
2. Imposer Nbmurs murs de manière aléatoire sur le plateau
3. Ajuster le plateau de manière à ce que toute case ai au moins 3 case adjacentes de type chemin
4. retourner la matrice correspondant au nouveau plateau de jeu

Les scénarios alternatifs :

1. Les coordonnées (x,y) des murs générer de manière aléatoire ne sont pas valides. Retourner à l'étape 2.
2. Une zone du plateau n'est pas accessible par le héros . Retourner en 3

Les scénarios Exceptionnels :

1. Les murs générer de manière aléatoires ne sont pas sur le plateau : retourner en 3

Scénario pour "Affichage graphique"

Nom : Affichage graphique
Résumé : Permet l'affichage des objets dans une fenêtre
Date : 11/10/2018
Auteur : Saint-Jore Amaury
Pré-condition : Le fichier du labyrinthe doit être généré.
Post-condition : Visualiser le jeu et tous ses objets.
Scénario nominal :

1. Considère les coordonnées de chaque objet coder sur une matrice
2. Ouvre une fenêtre.
3. Affiche chaque objet codé dans la matrice aux bonnes coordonnées figurant sur la matrice avec la bonne image associé
4. Met à jour l'affichage constamment pour permettre la visualisation des déplacements.

Scénarios alternatifs :

5. Un personnage (monstre ou héros) s'est déplacé = retourne en 3

Scénarios Exceptionnels :

6. La fenêtre ne s'est pas ouverte , retourne en 2
7. Deux affichage se superpose et engendre une trace retourne en 3

Scénario pour "Lire le fichier texte"

Nom : Lire le fichier texte
Résumé : Permet d'obtenir le labyrinthe à partir du fichier texte
Date : 11/10/2018
Auteur : JEBROUN Youssef
Pré-condition : Le fichier texte doit être créé.
Post-condition : Visualiser le labyrinthe.
Le scénario nominal:

1. Lire le fichier texte à l'aide d'un code en java .
2. Préciser le type de chaque contenu existant dans le fichier.
3. Utiliser le contenu de fichier pour créer le labyrinthe.

Les scénarios alternatifs:

4. Le contenu de fichier texte n'est pas utilisé correctement .Relire le fichier texte .

Les scénarios Exceptionnels:

5. La lecture du fichier donne un labyrinthe mal organisé, Retourne 3.

III. Organisation des sprints

- **Sprint 1**

Voici ce qui a été développé pendant le sprint 1 :

Responsable	Amaury	Sandyla	Olivier	Youssef	Ilhame
Titre	Affichage graphique	Déplacer le héros	Générer un plateau aléatoire	Lire le fichier texte	Fonction téléporter
Résumé	Permet d'afficher les objets dans une fenêtre.	Le joueur pourra déplacer le héros à travers le labyrinthe	Afin de dynamiser le jeu un plateau différent sera généré à chaque jeu	permet d'obtenir le labyrinthe à partir d'un fichier texte	permet au héros de se déplacer dans un plateau de jeu de type labyrinthe

- **Sprint 2**

Voici ce qui a été développé pendant le sprint 2 :

Responsable	Amaury	Sandyla	Olivier	Youssef	Ilhame
Titre	Créer piège, case magique	Fonctionnalités des monstres	Corriger les bugs de l'interface	Fonctionnalités des monstres	Créer trésor
Résumé	-création des cases pièges, des cases magiques -tester trésor	-création de monstres - placer les monstres aléatoirement -héritage personnages, monstres, héros -tester interface	-réduire la zone d'entrée clavier - régler le problème de double affichage du héros -Problème de suppression du labyrinthe -tester classe monstres	-création de monstres - placer les monstres aléatoirement	-placer aléatoirement le trésor -fin du jeu

❖ **Sprint 3**

Voici ce qui a été développé pendant le sprint 3 :

Responsable	Amaury	Sandyla	Olivier	Youssef	Ilhame
Titre	-Création des images	- Classe Fantome	-implémentation de fantome dans la classe	-Classe Fantome	-Création des images
Résumé	- développement de sprits	-developpement de la classe -méthodes traverserlesmurs()	- développement graphique	developpement de la classe -méthodes traverserlesmurs()	- développement de sprits

❖ **Sprint 4**

Voici ce qui a été développé pendant le sprint 4 :

Responsable	Amaury	Sandyla	Olivier	Youssef	Ilhame
Titre	Affichage graphique	Développement	Développement	Développement	Affichage graphique
Résumé	Gérer animation du jeu.	Fantome intelligent	Fantôme intelligent, Bug du jeu	Fantome intelligent	Gérer animation du jeu

IV. Diagramme de classes et séquentiel

Le diagramme de classes représente les classes intervenants dans le système. Il s'agit d'une représentation statique des éléments qui composent un système et de leurs relations. La classe est définie par son nom, ses attributs et ses opérations comme on peut le voir sur l'image 4 ci-dessous.

Ci-après le diagramme de classes :

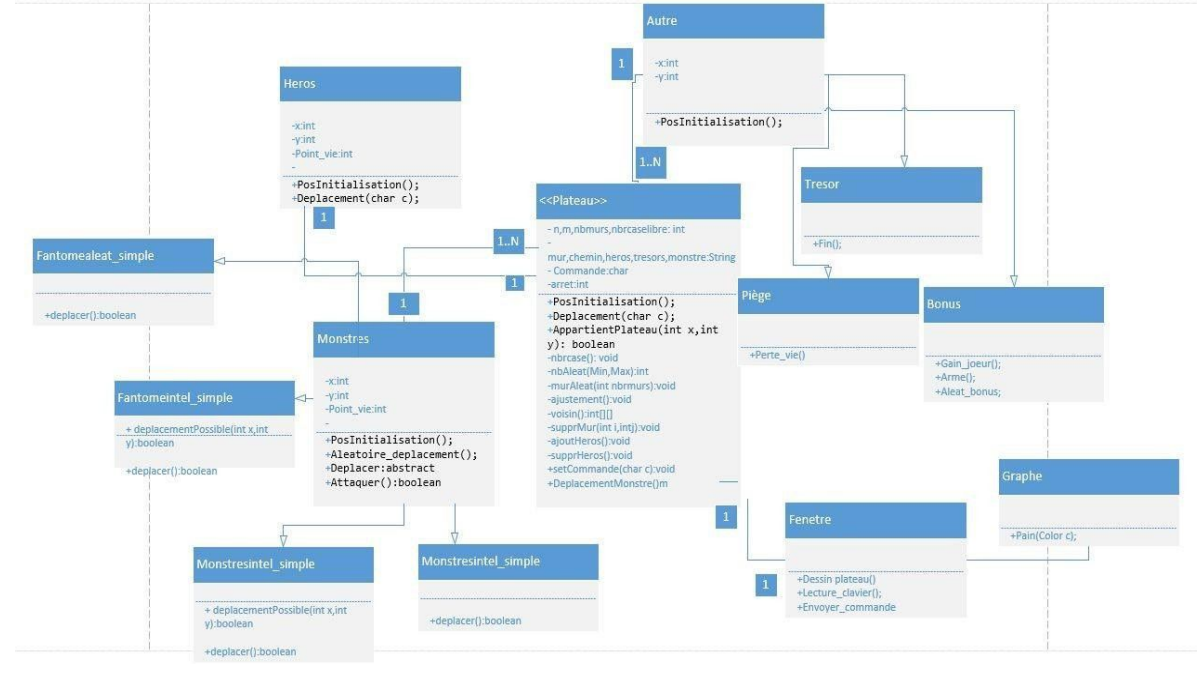

Image 4 : diagramme de classes

Un diagramme séquentiel contient toutes les informations échangées entre les lignes de vie, le tout présenté dans un ordre chronologique. Une ligne de vie se représente par un rectangle, auquel est accroché une ligne verticale pointillée.

Ci-après le diagramme séquentiel :

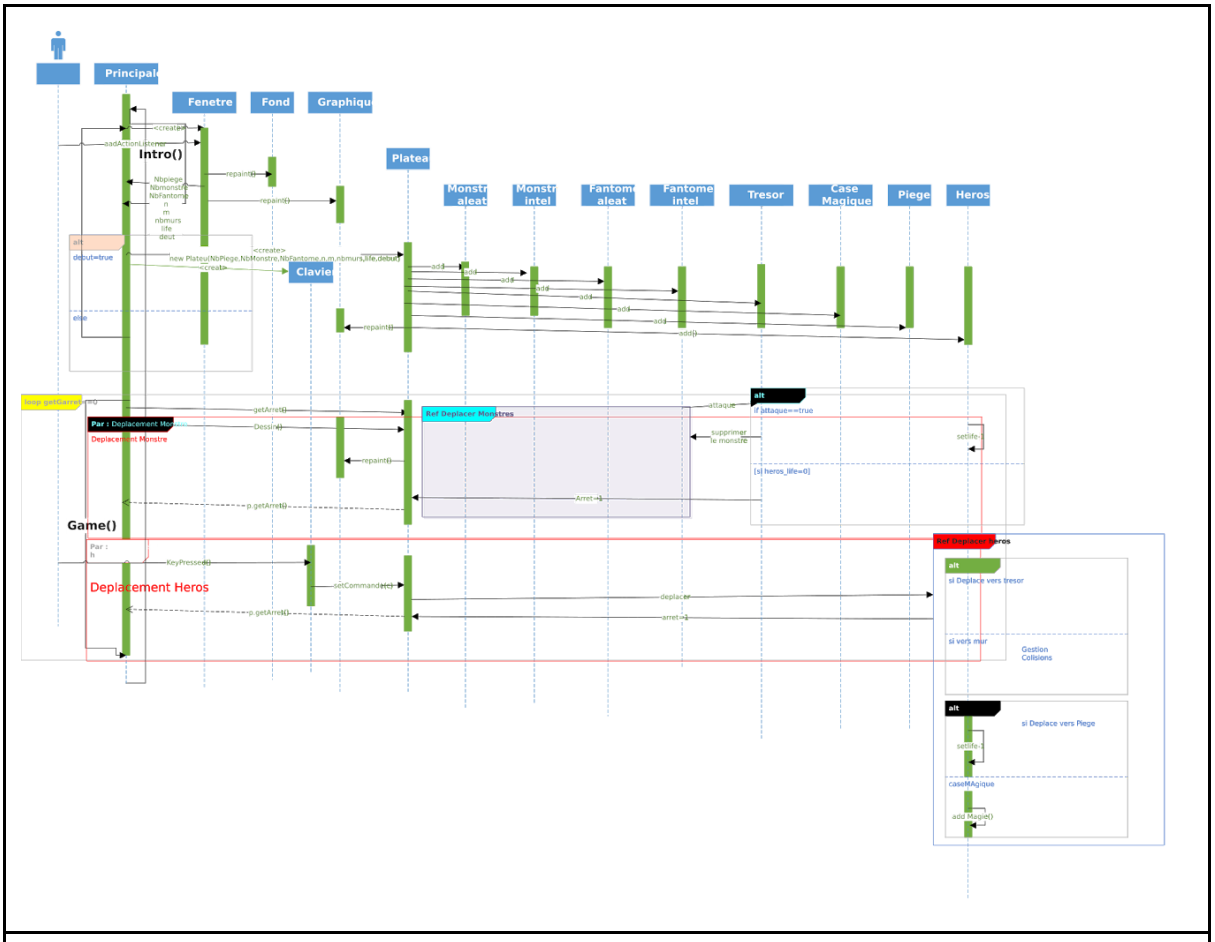

Image 5 : diagramme séquentiel

V. Stratégie d'affichage

Lors de ce projet, nous avons dû faire face à un choix algorithmique. Afin d'envoyer un rendu visuel à chaque fois que la matrice codant le plateau de jeu ait été actualisée, nous pouvons soit réactualiser l'affichage du plateau au complet en parcourant la matrice de codage, ou soit seulement afficher les objets ayant subi un déplacement, en reprenant les vecteurs où sont contenus ces objets.

L'avantage de la deuxième solution est sa rapidité d'exécution, cependant afin de réaliser un tel affichage, le programme doit réaliser 2 affichages pour 1 déplacement. On considère un monstre seul contenu sur le plateau à l'instant n et à la position (x,y) puis ce monstre subit un déplacement, pour afficher le plateau à l'instant n+1. Nous pouvons soit

- Illustration procéder 1 :

Parcourir toute la matrice et afficher un par un l'objet coder à l'indice (i,j) de la matrice et le placer dans la fenêtre au pixel (x,y)

ou bien

- Illustration procédé 2 :

Afficher initialement le plateau sans monstre et récupérer continuellement le vecteur dans la classe plateau contenant le monstre puis récupérer ses coordonnées avant et après l'appel de la méthode déplacer monstre et enfin remplacer le pixel dans la fenêtre correspondant à la position du monstre avant déplacement par une image de type chemin et remplacer le pixel correspondant à la position du monstre après déplacement par une image de type monstre.
De plus du fait que l'affichage se fait sur un Thread disjoint de celui du code principal, nous devons mettre des temps de pose lors des affichages afin de ne pas observer des marques de traces de monstre sur la fenêtre d'affichage.

Le rendu visuel de la procédure visuelle n°2 entraîne un clignotement du monstre lors de chaque déplacement, alors que celui de la procédure N°1 ne présente aucun inconvénient visuel mais présente un inconvénient algorithme du fait de parcourir entièrement la matrice de codage et de réactualiser chaque pixel d'affichage sachant que certains pixels restent invariants.

La solution 1 a été retenue, et nous devons optimiser la complexité de l'algorithme du jeu afin de rendre un rendu visuel et une ergonomie de jeu la plus agréable possible à l'utilisateur.

VI. Bilan

Voici la version finale du jeu, on peut y voir l'implémentation du héros, des fantômes, des monstres, des pièges, des cases magiques (représentés par des points) et du trésor. Également sur le côté droit, on peut voir les points de vie du héros qui sont représentés.

Image 6 : visualisation de la version finale du jeu

Ce projet nous a également permis de développer de nombreuses compétences comme celles listées sur l'image 7 ci-après :

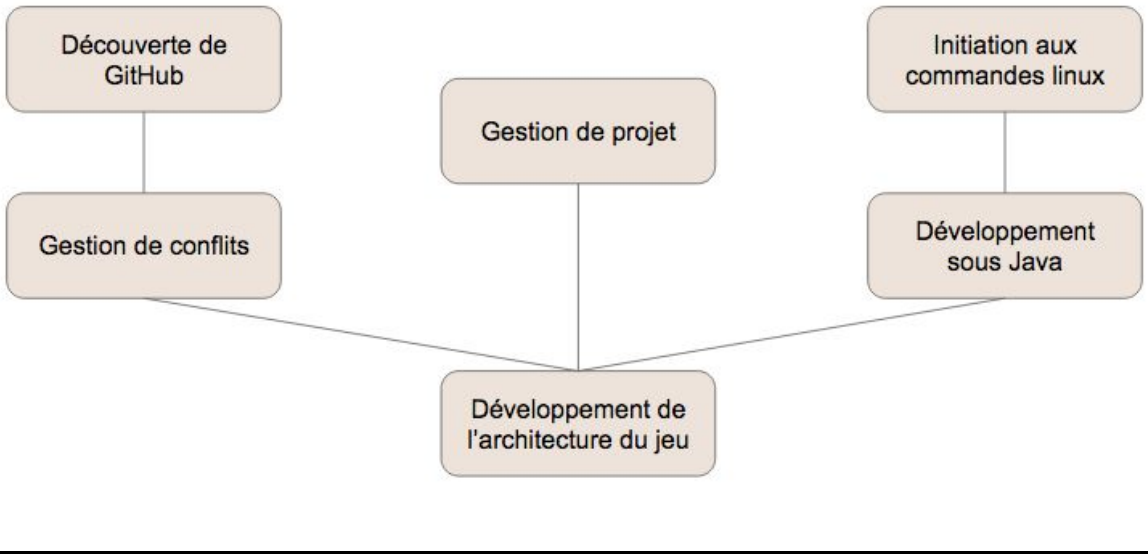

Image 7 : compétences acquises