

Université de Cergy-Pontoise

RAPPORT

pour le projet Génie Logiciel
Licence d'Informatique deuxième année

sur le sujet

AVENTURE

rédigé par

DAMODARANE Jean-Baptiste ELUMALAI Sriguru
ZHANG Victor



Mai 2022

Table des matières

1	Introduction	5
1.1	Contexte du projet	5
1.1.1	Synopsis	5
1.1.2	Description générale du jeu	5
1.2	Objectif du projet	5
1.3	Organisation du rapport	6
2	Spécification du projet	7
2.1	Outils de développement	7
2.2	Notions de base et contraintes du projet	7
2.2.1	Fonctionnement général du logiciel	7
2.2.2	Notions et Terminologies de base	7
2.2.3	Contraintes et Limitations connues	9
2.3	Fonctionnalités attendues du projet	9
3	Conception et réalisation du projet	10
3.1	Architecture globale du logiciel	10
3.2	Conception des classes de données	11
3.2.1	Les classes les plus importantes et leurs fonctionnalités	11
3.2.2	Diagramme des classes des données	12
3.3	Conception des traitements (processus)	13
3.3.1	Création de la map	13
3.3.2	Changement de la map	13
3.3.3	Collisions	15
3.3.4	Combats	16
3.4	Conception de l'IHM graphique	16
4	Manuel utilisateur	18
4.1	Touches manipulables	18
4.2	Interface utilisateur	19
5	Déroulement du projet	20
5.1	Réalisation du projet par étapes	20
5.1.1	Diagramme de Gantt	20
5.1.2	Répartitions des tâches	21
5.2	Répartitions des tâches entre membres de l'équipe	21
6	Conclusion et perspectives	22
6.1	Résumé du travail réalisé	22
6.1.1	Résumé du jeu	22
6.1.2	Défis rencontrés	22
6.2	Améliorations possibles du projet	22

Table des figures

1	Image illustrant les sprites utilisés dans le jeu	8
2	Diagrammes illustrant l'architecture du jeu	10
3	Diagramme les classes de données	12
4	Diagramme illustrant des classes pour la création des fenêtres et leurs contenus et ainsi que les paramètres du jeu	12
5	Image illustrant la création et l'affichage de la carte	13
6	Image illustrant 2 parties de la map entière	14
7	Image illustrant le changement de la map en fonction de la position du joueur fictif	14
8	Image illustrant les NPCs	15
9	Image illustrant les objets	15
10	Image illustrant les monstres/ennemis	15
11	Figure illustrant la conception de l'ihm graphique	16
12	Figure illustrant la conception de la phase de combat	17
13	Les touches du clavier pour le déplacement du joueur	18
14	La touche de la souris pour la sélection et validation des boutons	18
15	Image illustrant la page principale et la zone de combat avec leurs options . . .	19
16	Diagramme de Gantt	20
17	Image illustrant la répartitions des tâches	21
18	Image illustrant la répartition des tâches entre membres de l'équipe	21

Liste des tableaux

1	Tableau illustrant les équipements et les sorts	7
2	Tableau des classes les plus importantes et leurs responsabilités	11

Remerciements

Tout d'abord, nous tenons à remercier très sincèrement et à témoigner toute notre reconnaissance à notre professeur Mr.Liu TianXiao et Mr.Marc Lemaire, nos professeurs qui nous ont enseigné lors des CM et des TD, les bases du langage Java lors du semestre 3, qui nous ont permis de nous introduire dans la programmation orientée objet.

Nous remercions encore une fois, Mr.Liu TianXiao notre professeur de Génie Logiciel, qui nous a enseigné pendant le semestre 4, le POO, et le professionnalisme que nous aurons besoin dans le milieu professionnel quand nous travaillerons pour la société.

Remercier une troisième fois Mr.Liu TianXiao notre professeur de CM et chargé de TD, qui nous a guider et conseiller lorsque nous avons rencontré des problèmes pendant que nous développons notre projet

1 Introduction

Dans le cadre de notre deuxième année de Licence en Informatique à CY Cergy Paris Université, nous avons eu à réaliser, un projet de 4 mois au sein du module Génie Logiciel et Projet (GLP). Notre projet GLP consiste à développer un logiciel de jeu d'aventure, nommé "Shiva's Quests" mettant en place une série de quêtes prépondérantes. Nous détaillerons dans ce dossier les fonctionnalités contenues dans le jeu et l'architecture du logiciel. Toutefois, l'objectif primordial de notre projet est de mettre en avant les grands principes de la programmation objet.

1.1 Contexte du projet

Le projet se base sur la conception d'un jeu vidéo mettant en avant un personnage principal et des ennemis.

Le joueur est chargé de sauver le monde en fermant un portail ouvert dans l'espace temps. Pour y parvenir, il devra traverser des obstacles et achever un certain nombre de missions et de combats avec de différents ennemis. Chaque zone de combats met en place différents ennemis avec différents points de vie et d'endommagements. Le joueur ne peut aller plus loin, à l'étape suivante, que s'il sort de la zone actuelle dans laquelle il peut entrer et sortir pour battre l'ennemi.

1.1.1 Synopsis

Shiva, une incarnation du divin, apparaît sur le monde des Hommes dans le but de les sauver. Pour cela, il devra accomplir cinq niveaux. Shiva pourra-t-il sauver ce monde ?

1.1.2 Description générale du jeu

Le jeu met en avant un personnage principal qui va parcourir un long voyage où il devra réussir une série de cinq niveaux principales (missions). Ces quêtes à accomplir seront principalement des combats contre des asuras ou des monstres qu'il devra affronter et battre. Les quêtes ont pour objectif de rassembler trois fragments de Crystal possédant le pouvoir de réparer le portail et de cette manière, Shiva pourra empêcher son ouverture pour l'éternité.

Le joueur pourra également changer d'armes lors des combats. Nous avons conçu des zones de combats : une pour des combats avec des mini-monstres/gardiens et une autre pour le combat final avec le Boss final.

De plus, le jeu met en place des NPCs qui donneront des conseils au joueur et seront un moyen idéal d'interaction et rend le jeu plus vivant.

1.2 Objectif du projet

Les principaux buts de notre projet Aventure sont :

- Concevoir un jeu vidéo en implémentant les fonctionnalités fondamentales d'un jeu d'aventure classique,
- Implémenter des armes/sorts avec des pouvoirs unique,
- Mise en place des combats avec manipulations de différents armes,
- Mise en place de différents monstres avec des points de vie différents et qui ont un seul objectif : battre le héros,
- Implémenter un système de niveaux,

- Implémenter des effets d'animations sur les déplacements des personnages et des objets (armes),
- Créer une interface conviviale afin d'attirer l'attention du joueur.

1.3 Organisation du rapport

Afin de mener à bien ce rapport, nous commencerons par présenter les spécificités de notre jeu d'aventure avec des notions notables, les grandes fonctionnalités attendues du projet et les contraintes. Nous aborderons ensuite la partie technique basée sur l'architecture du logiciel, la conception et la réalisation du projet. Nous avons également une section présentant le manuel d'utilisateur avec les options fonctionnelles présentes dans notre jeu. Nous présenterons ensuite les différentes tâches que nous avons pu identifier ainsi que leurs répartitions au sein de l'équipe.

Nous terminerons par faire un bilan de notre travail en portant un regard critique notamment sur l'aboutissement final de notre jeu.

2 Spécification du projet

Nous avons présenté l'objectif du projet dans la section 1. Dans cette section, nous présentons la spécification de notre logiciel réalisé. Ceci correspond principalement au document de spécification du projet.

2.1 Outils de développement

- Système d'exploitation : ce jeu peut tourner sous les systèmes d'exploitation de Windows et GNU/Linux,
- Langage de programmation : programmé en Java,
- version du langage : JAVASE-11,
- Plateforme de développement : Eclipse IDE,
- Editeur d'images et animation : Pix2d (logiciel) et pixilart(site internet),
- Gestion des versions : github¹.
- Conception du rapport de projet : Latex.

2.2 Notions de base et contraintes du projet

2.2.1 Fonctionnement général du logiciel

Nous avons implémenté les fonctionnalités courantes et classiques d'un jeu d'aventure : le joueur peut se déplacer librement et possède un certains nombres d'armes dans le but de réaliser des tâches. Notre logiciel offre également une possibilité de changer d'armes durant les attaques. Le jeu est jouable avec des touches de clavier et de souris.

2.2.2 Notions et Terminologies de base

1. Equipements et Sorts prévues pour le rendu final :

Equipements et Sorts
" Pinaka ", l'arc
" Trishul ", le trident
"Ushnisha", le Casque
True fire of Samadhi (un sort permettant de faire une attaque spéciale)

TABLE 1 – Tableau illustrant les équipements et les sorts

2. Carte :

La carte sera constituée de différents lieux qui devront être explorés par le joueur en fonction des quêtes qu'il aura à réaliser. Ces différents lieux seront reliés par le village principal du héros, l'endroit où il pourra s'équiper d'armes via un magasin.

1. <https://github.com/JeanBaptiste02/MONPROJETJAVAS4>

3. Quêtes :

Le joueur devra effectuer trois quêtes principales du plus facile et au plus difficile et chaque quête principale sera composée d'une série de missions (quêtes secondaires à accomplir pour progresser dans le jeu) qui pourront être en lien avec les NPCs rencontrés en chemin par le héros. Chaque fin de quête principale mettra en place un combat mortel entre Shiva et un monstre.

4. Animations :

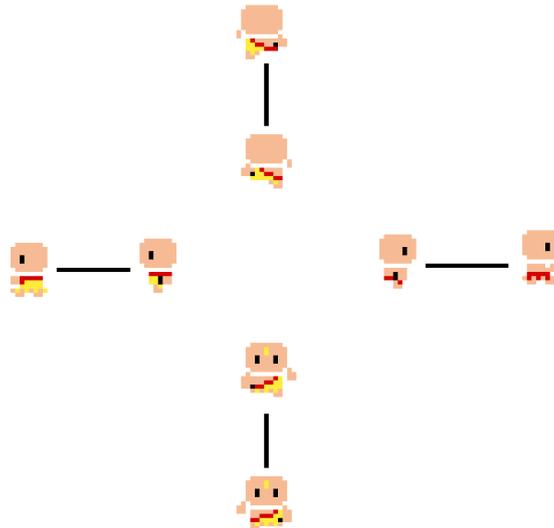


FIGURE 1 – Image illustrant les sprites utilisés dans le jeu

L'animation est une notion prépondérante dans le cadre de notre projet, pour les déplacements des éléments du jeu, les décors, etc. L'animation consiste, en fait, principalement à dessiner sur l'écran des figures en respectant des intervalles de temps régulier et permet de donner vie aux personnages ou aux objets, et enfin les animer. Toutefois, il faut que ces effets d'animation soient cohérents par rapport à l'environnement dans lequel se trouve le joueur.

5. Entités :

- Héros : un avatar de Shiva possédant de grandes forces et chargé de détruire le mal et fermer le portail et il sera le seul personnage qu'on peut contrôler avec le clavier.
- NPC : Les NPCs sont des personnages non contrôlables, qui seront déjà présents dans le décor avec des déplacements aléatoire ou non et permettront au joueur d'interagir avec eux pour pouvoir avancer dans le jeu.
- Monstres : Le jeu met en avant 3 types de monstres à battre :
 - **les mini-monstre** permettent débloquent une partie de la map.
 - **les gardiens (asuras)** permettent de récupérer les cristaux.
 - **Un Boss final** apparaîtra à la fin de l'achèvement de tous les quêtes principales.

6. Or :

Chaque quête principale effectuée rapporte un certain nombre de pièces d'or.

7. Magasins :

Un magasin (shop) sera présent dans le village principal du héros. Dans ce magasin, le héros pourrait s'équiper de nouvelles armes et des équipements en échange d'or qu'il a récolté dans ses parcours.

8. Niveau :

Au fil de l'avancement des quêtes, le héros montera en niveau en gagnant de l'expérience (XP) et lors qu'il aura atteint le dernier niveau, le héros pourra utiliser un sort.

2.2.3 Contraintes et Limitations connues

L'une des principales contraintes que nous avons rencontré pendant le déroulement du projet est la durée. Un planning très précis a été mis en place avec des dates limites correspondantes aux étapes précises du projet. Le respect de ce planning a été très important car cela nous a permis de se repérer dans l'avancement du projet.

Les actions des ennemis, le contrôle des arrières plans et la manipulation des niveaux ont été d'importantes sources de difficultés. La qualité du projet est importante dans notre sujet et l'application devait respecter certaines règles artistiques pour avoir une meilleure apparence.

En étant conscient du manque de temps pour notre projet Universitaire, nous avons décidé de ne travailler que sur les principales attentes de l'utilisateur.

Pour avancer davantage dans le développement de notre jeu vidéo, nous avons dû comprendre les attentes de divertissements des utilisateurs. Nous avons implémenté les fonctionnalités les plus nécessaire dans le jeu afin de permettre aux joueurs d'utiliser le logiciel facilement.

2.3 Fonctionnalités attendues du projet

Les jeux d'aventure, en général, insistent sur la mise en place sur l'exploration, des échanges de dialogues avec d'autres personnages, etc. Le principal but est de créer une histoire, au sein duquel, le joueur peut agir.

En effet, ils s'agissent des fonctionnalités les plus essentielles d'un jeu d'aventure.

Les fonctionnalités attendues de notre projet Aventure sont :

- Donner un sentiment d'aventure à l'utilisateur,
- Avoir une histoire intéressante afin d'attirer l'attention de l'utilisateur,
- Avoir un seul joueur principal,
- Possibilité de se déplacer librement sur la carte du jeu,
- Possibilité d'interagir avec d'autres personnages du jeu,
- Possibilité de visiter des différents endroits de la carte,
- Implémentation d'un système de niveaux et de quêtes,
- Possibilité de faire des combats contre d'autres personnages.

3 Conception et réalisation du projet

Ayant parlé des spécifications du projet dans la section 2, cette section se base principalement sur la partie technique de notre logiciel.

3.1 Architecture globale du logiciel

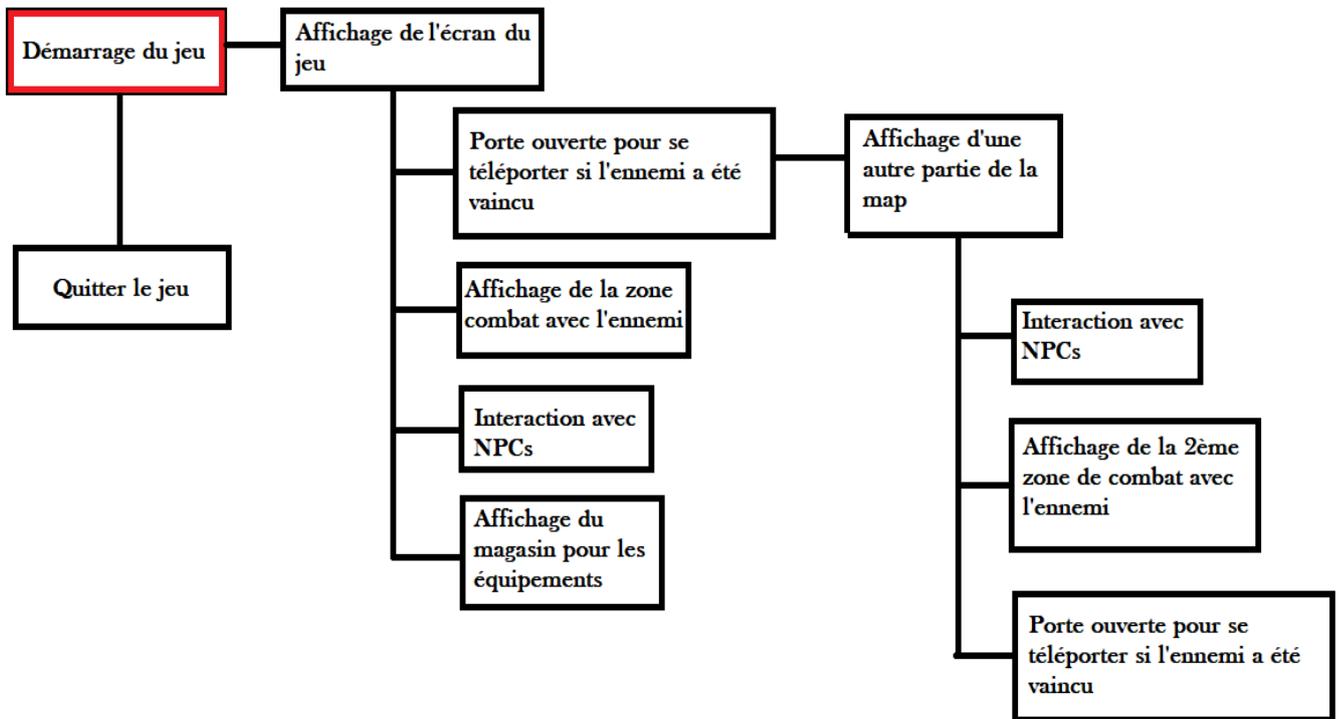


FIGURE 2 – Diagrammes illustrant l'architecture du jeu

Lorsque le joueur lancera le jeu, il se retrouvera dans le village principal du héros. Et on y retrouve un magasin où il pourra décider soit d'y entrer, soit d'interagir avec les Npcs présents, ou soit de combattre le monstre dans une zone dédiée pour le combat. En sachant que pour pouvoir passer dans la zone suivante, il sera obligé de combattre un mini-monstre. Le même processus se répète jusqu'au dernier niveau.

3.2 Conception des classes de données

Cette sous section se base sur les classes les plus prépondérantes et ainsi l'ensemble des diagrammes UMLs (sans attributs et sans méthodes).

3.2.1 Les classes les plus importantes et leurs fonctionnalités

Numéro	Classes	Responsabilités
1	GameConfiguration.java	Classe gérant les blocks du jeu, la taille de fenêtre et vitesse du joueur
2	MainGUI.java	Gère l'affichage du jeu
3	GameDisplay.java	Permet d'afficher les composants sur la fenêtre du jeu
4	PaintStrategy.java	Récupère les images nécessaires pour la map qui change compte tenu de la position du héros. Permet de dessiner les composants (objets). Permet l'affichage des monstres et des NPCs
5	OptionsWindow.java	Fenêtre affichant les missions et les équipements durant le déroulement du jeu. Le joueur peut également y créer un profil avec un pseudo.
6	Block.java	Gère les caractéristiques générales d'une case (largeur, longueur).
7	BlockManager.java	Permet de lire un fichier image.
8	Map.java	Récupère la position de chaque block
9	MapBuilder.java	Permet de lire un fichier texte et stocke les données dans une arrayList
10	Tile.java	Responsable d'ajouter une image dans un tableau d'images en fonction des indices qui seront appelés dans d'autres classes.
11	MouvElements.java	Gère les mouvements du héros, des ennemis
12	Joueur.java	Permet de manipuler les mouvements du joueur.
13	JoueurFictif.java	Permet de manipuler les mouvements du joueur 'fictif', l'état de santé de joueur (mort/vivant=gagnant).
14	GameBuilder.java	Permet d'initialiser le joueur (et le joueur fictif), les monstres et les objets
15	MobileElementManager.java	Le noyau du jeu

TABLE 2 – Tableau des classes les plus importantes et leurs responsabilités

3.2.2 Diagramme des classes des données

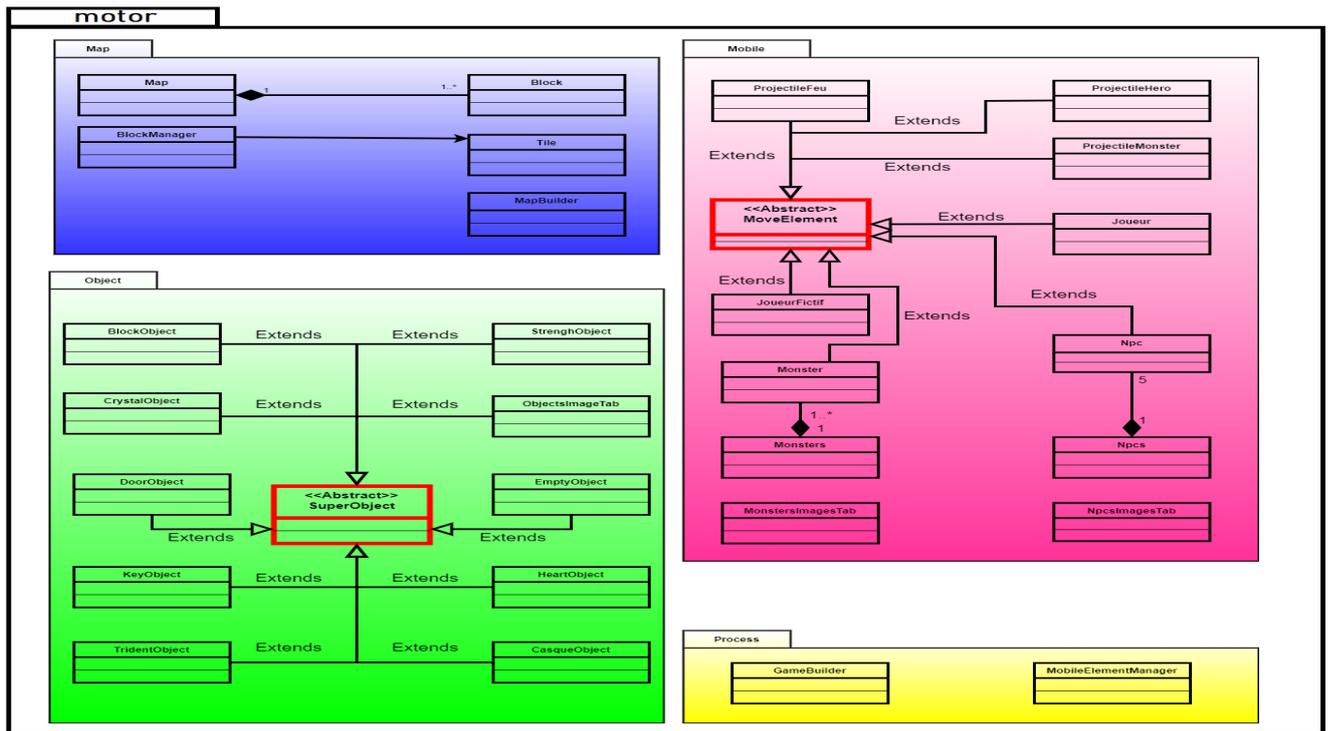


FIGURE 3 – Diagramme les classes de données

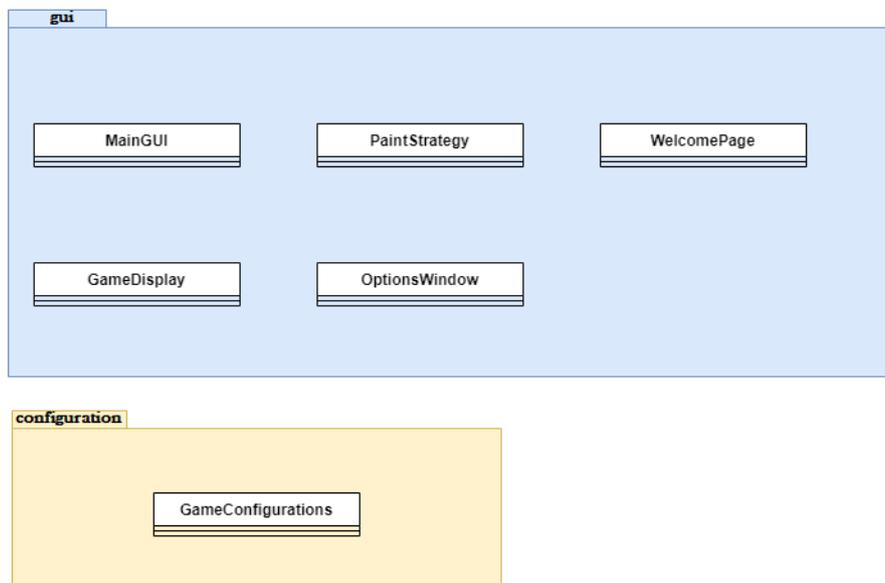


FIGURE 4 – Diagramme illustrant des classes pour la création des fenêtres et leurs contenus et ainsi que les paramètres du jeu

3.3 Conception des traitements (processus)

Cette sous-section porte sur les principaux éléments fonctionnels du logiciel tels que la création et le changement de la carte et puis les différents entités et objets qui y sont présents.

3.3.1 Création de la map

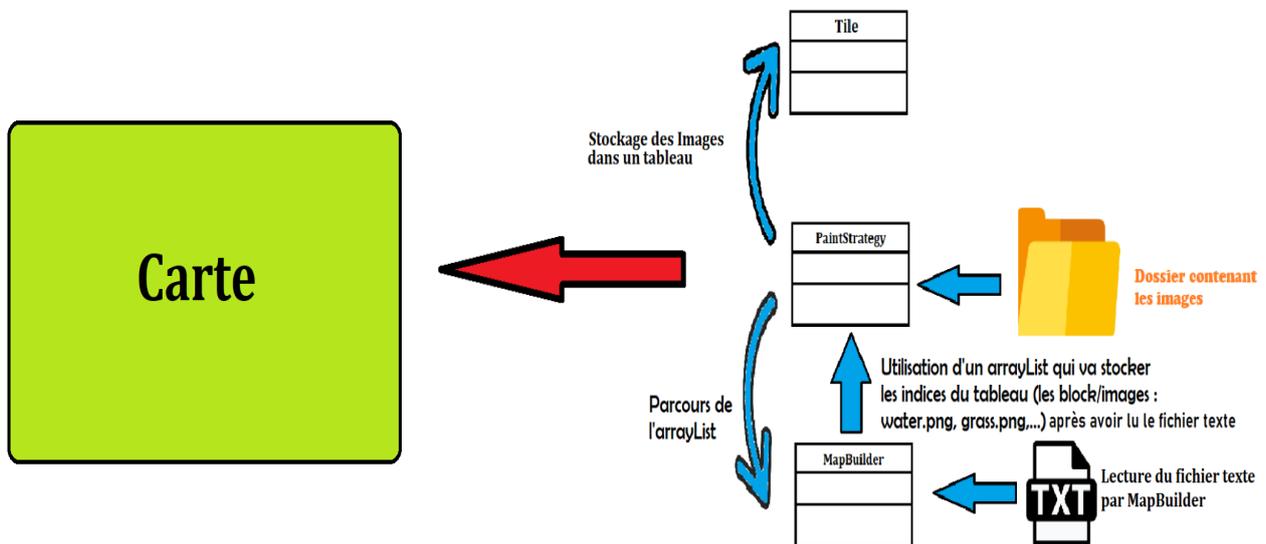


FIGURE 5 – Image illustrant la création et l’affichage de la carte

La carte du jeu occupe une partie très importante et un processus particulier précède sa création. Nous avons créé une classe Tile dans laquelle, un tableau est créé avec une taille correspondante aux nombres d’images des blocks de la map. La classe PaintStrategy permet de récupérer les images png à partir d’un dossier qui sont lues par la classe BlockManager et les ajoute dans le tableau de la classe Tile. Dans le but de créer l’environnement du jeu, les indices représentant les différentes images seront entrées dans un fichier texte dans un ordre précis comme une suite de chiffres qui sera lu et sauvegardé dans une ArrayList par la classe MapBuilder. Cette ArrayList sera parcourue par un Iterator et ainsi une partie de la map entière sera affichée en fonction de la position du joueur.

3.3.2 Changement de la map

Etant donné que notre joueur doit rester visible sur l’écran de notre jeu pendant toute la durée du programme, on a décidé d’implémenter un joueur fictif qui est représenté par un carré jaune (sur la figure 7), il va simuler le déplacement de notre joueur original, sur toute la map du jeu. Ces 2 joueurs auront exactement le même déplacement, par exemple, lorsque l’utilisateur décide d’aller à droite en appuyant sur la touche d, les deux joueurs iront à droite en même temps. Et grâce à ce processus, lorsque notre joueur a atteint le bord droit de l’écran, par exemple dans la figure 7, sa position sera réinitialisée à gauche de l’écran car il doit être visible, et le joueur fictif se retrouvera dans l’autre partie de la map et l’endroit où se trouve ce joueur fictif, sera redessiné à l’endroit où se trouve notre joueur a nous, ce qui donnera cet effet de changement de map.

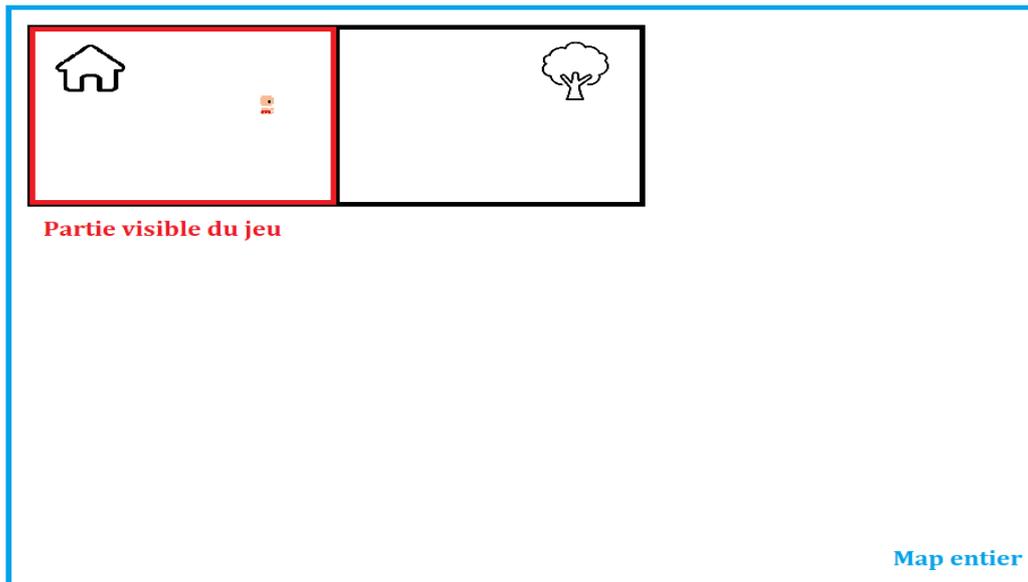


FIGURE 6 – Image illustrant 2 parties de la map entière



FIGURE 7 – Image illustrant le changement de la map en fonction de la position du joueur fictif

3.3.3 Collisions

Nous avons mis en place trois types de collisions :

NPCs : Lorsque le joueur entre en collision avec les NPCs, une boite de dialogue s'affiche à gauche de la fenêtre et disparaît si le joueur s'éloigne des NPCs.



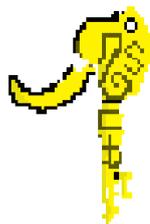
FIGURE 8 – Image illustrant les NPCs

Objets : nous avons différents objets qui apparaissent ou qui disparaissent dans le jeu en fonction de l'avancement du joueur et parmi ces objets, nous avons :

- "le poing" : qui apparaît quand le joueur a vaincu un mini-monstre et permet de débloquent le passage vers la partie suivante de la map.
- la clé : qui apparaît lorsque le joueur a vaincu un gardien et permet d'ouvrir une porte afin de récolter un crystal.
- les cristaux : permettent d'ouvrir le portail final pour accéder au combat final.



Le poing



La clé



Le crystal

FIGURE 9 – Image illustrant les objets

Monstres : Lorsque le joueur entre en contact avec un monstre, l'environnement de jeu change, le héros et le monstre se retrouvent dans une zone de combat.



Mini-monstre



Gardien(asura)



Boss

FIGURE 10 – Image illustrant les monstres/ennemis

3.3.4 Combats

Lorsque le joueur entre en collision avec les monstres, ils se retrouvent dans une zone de combat Afin d'équilibrer le jeu, Nous avons fait un système de combat tout à tour : une fois que nous avons attaqué, ce sera à l'adversaire d'attaquer à son tour. Le héros aura un certain nombre d'armes à sa disposition :

- "Pinkaka", l'arc : pouvant être utilisé autant de fois souhaité et retire un point de vie au monstre à chaque attaque..
- "Trishula", le trident : utilisable uniquement deux fois et retire deux points de vie à chaque attaque.
- le casque : une fois que le joueur atteint un nombre de point de vie faible, cet item lui sera proposé et permettra d'en gagner deux de plus

Le trident et le casque sont à récupérer dans le magasin lorsque le joueur aura assez de pièces d'or.

3.4 Conception de l'IHM graphique

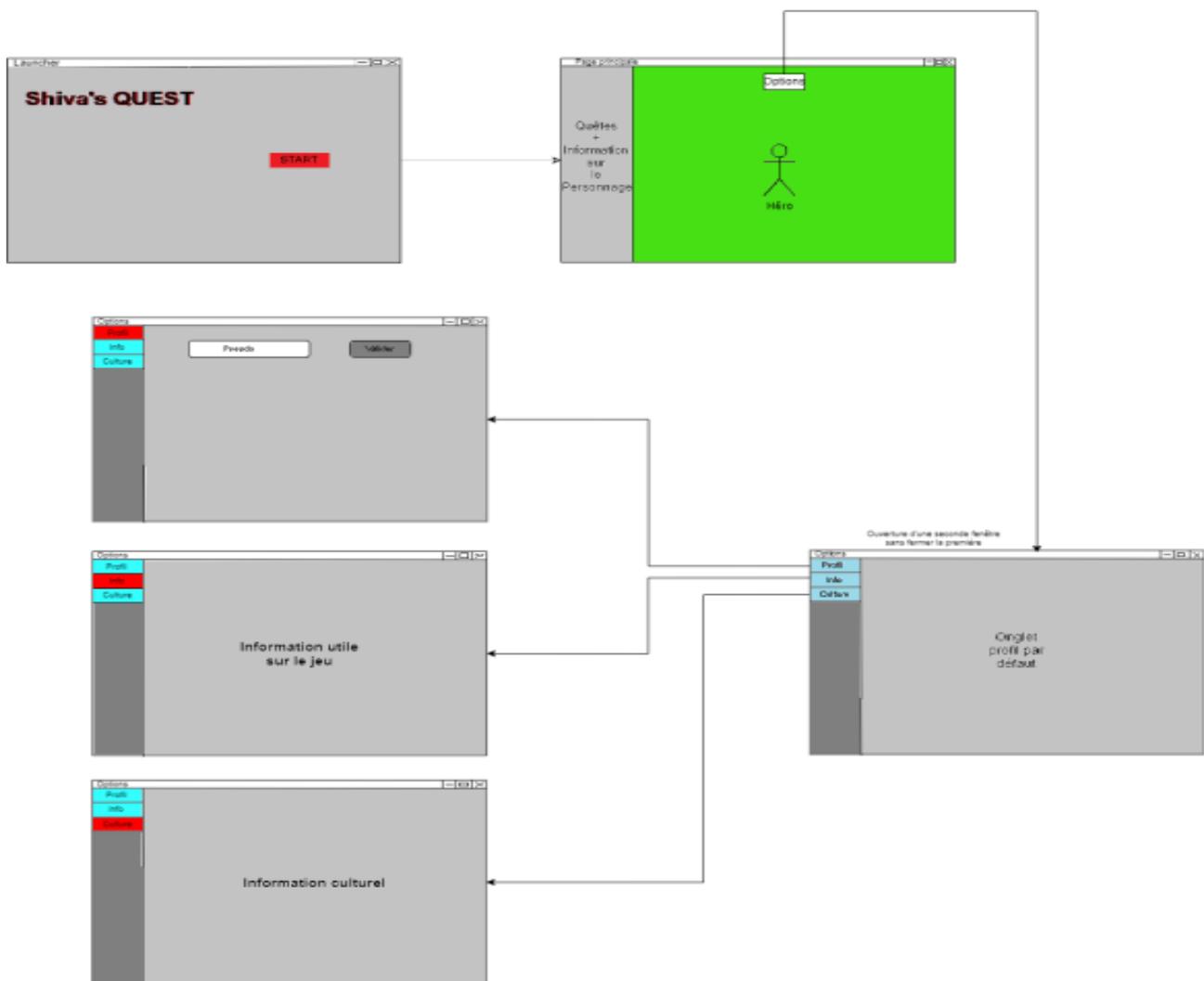


FIGURE 11 – Figure illustrant la conception de l'ihm graphique

La figure 11 montre la page principale du jeu et les principales fonctionnalités de la fenêtre Options contenant des informations pouvant être consultées durant le jeu.

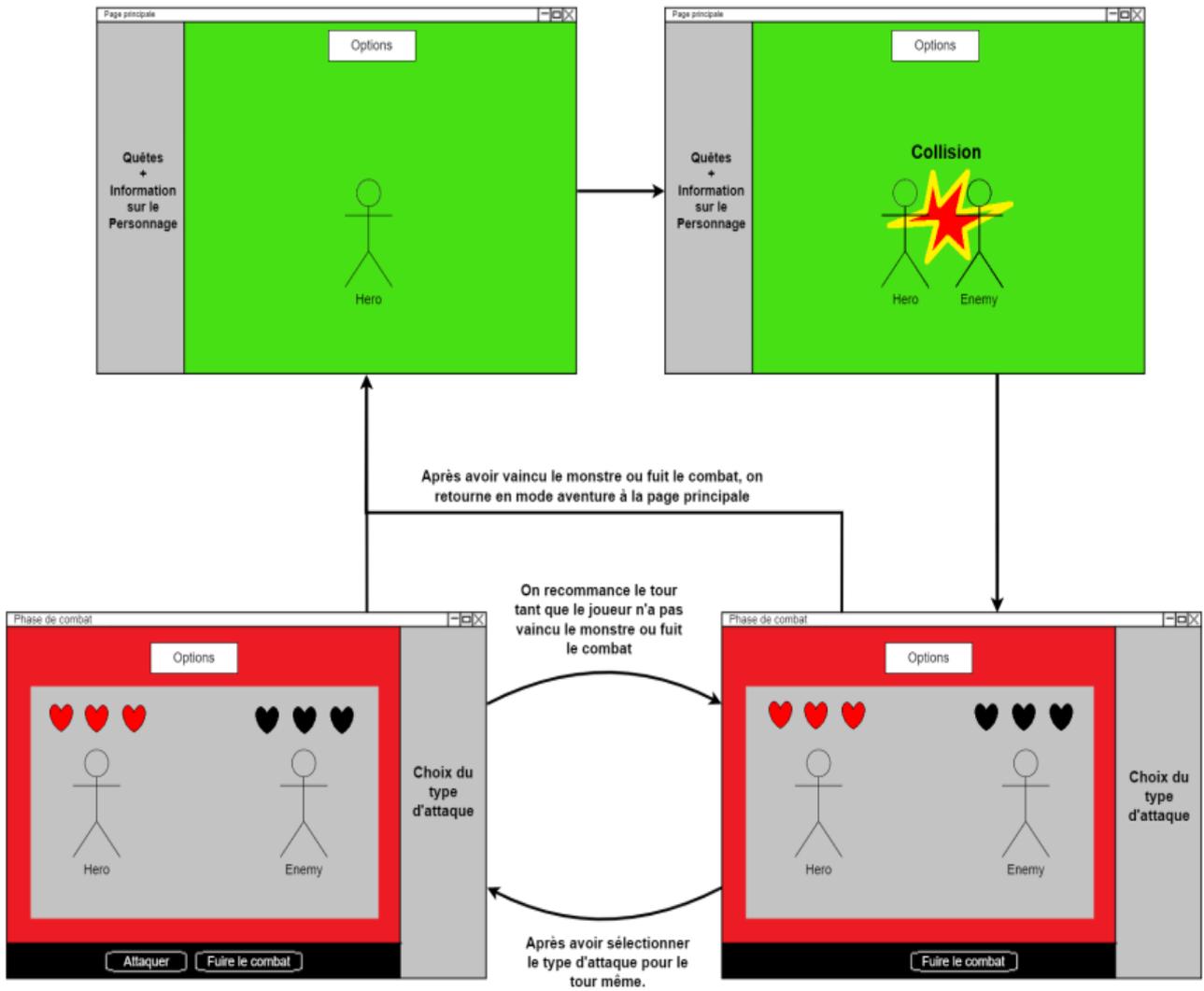


FIGURE 12 – Figure illustrant la conception de la phase de combat

La figure 12 montre les principales fonctionnalités de la phase de combat.

4 Manuel utilisateur

Cette section est entièrement dédiée au mode d'emploi du logiciel.

4.1 Touches manipulables

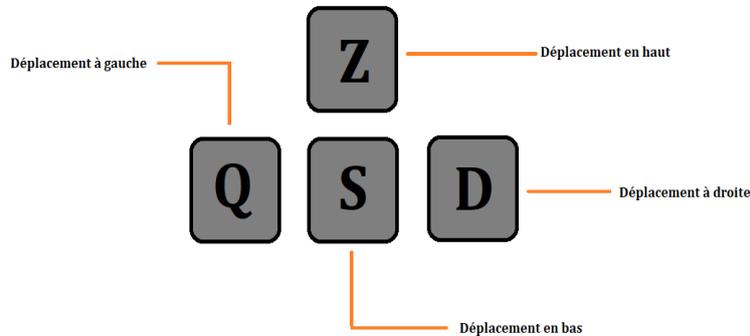


FIGURE 13 – Les touches du clavier pour le déplacement du joueur

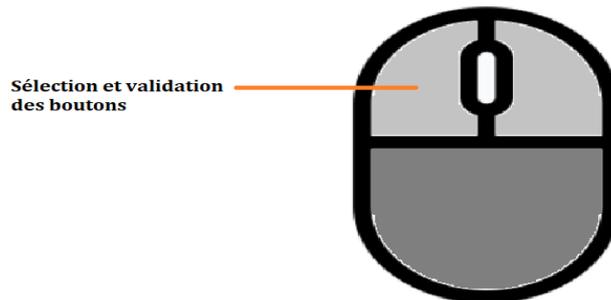


FIGURE 14 – La touche de la souris pour la sélection et validation des boutons

Ce jeu jouable sur PC fonctionne avec deux outils essentiels : clavier et la souris.

Les déplacements du personnage principal peuvent être contrôlés par des touches du clavier (Z, Q, S et D). Les sélections d'armes pendant les combats et les attaques sont manipulables par la souris.

4.2 Interface utilisateur

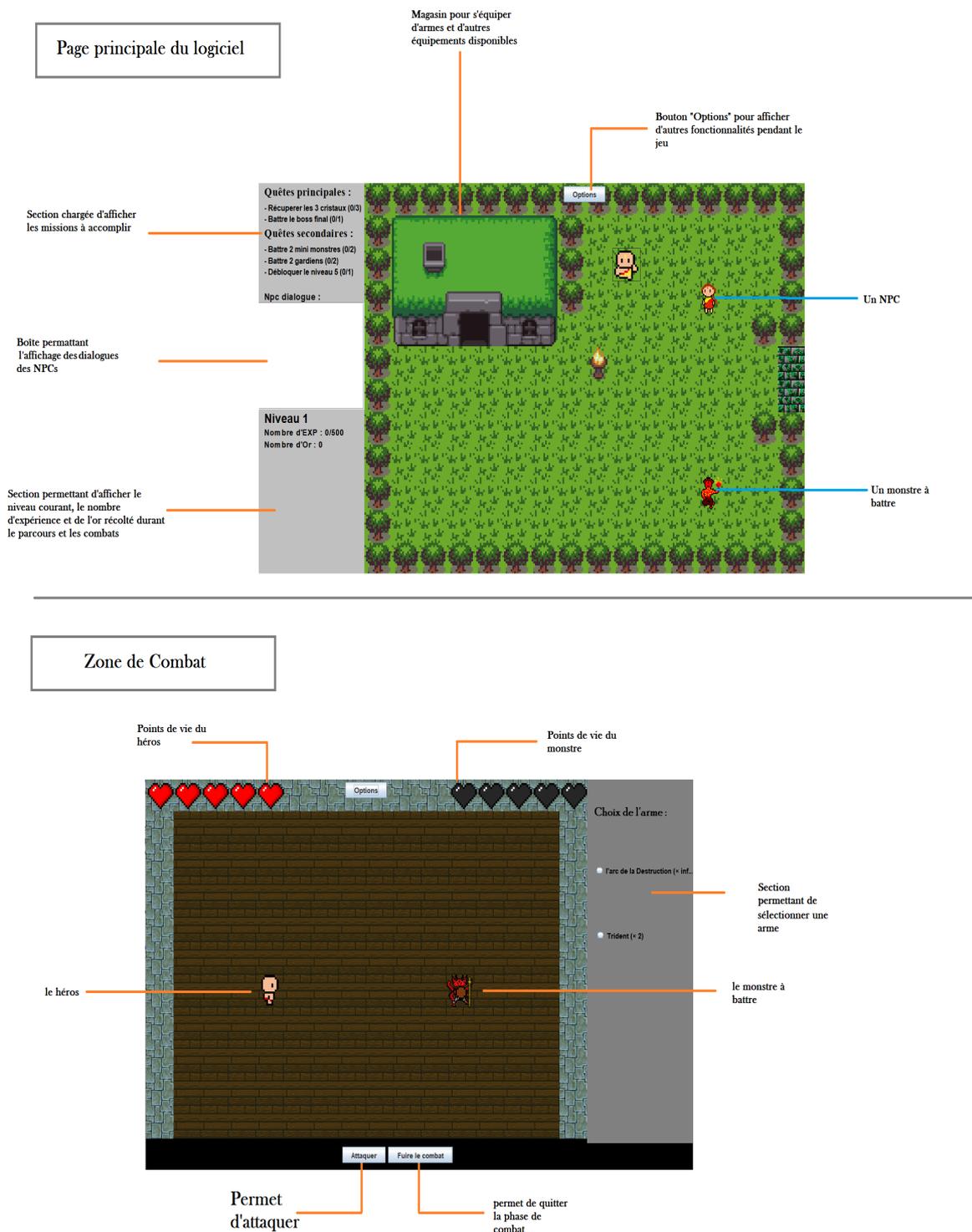


FIGURE 15 – Image illustrant la page principale et la zone de combat avec leurs options

La figure 15 présente les options de la page principale du logiciel et ainsi la zone de combat et ses fonctionnalités.

5 Dérroulement du projet

Dans cette section, nous décrivons comment le projet a été réalisé en équipe : la répartition des tâches, la synchronisation du travail en membres de l'équipe, etc.

5.1 Réalisation du projet par étapes

5.1.1 Diagramme de Gantt

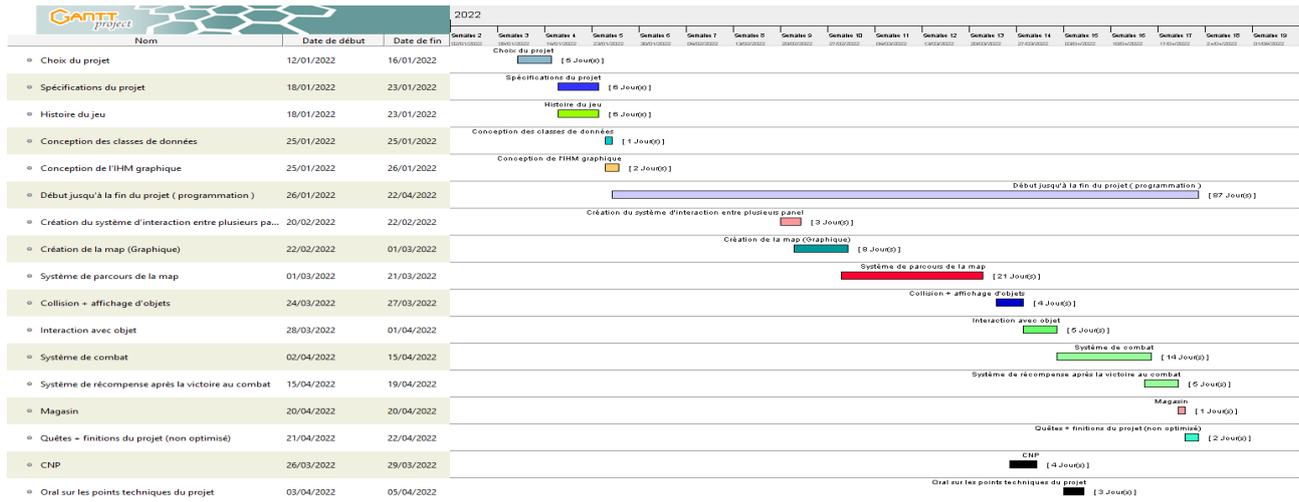


FIGURE 16 – Diagramme de Gantt

Le diagramme de Gantt permet d'avoir une vision plus globale et claire des objectifs du projet, leurs responsabilités et leurs niveaux d'importance. Nous avons fait un diagramme de Gantt après avoir étudié les niveaux de faisabilités, pour ne pas avoir du retard.

5.1.2 Répartitions des tâches

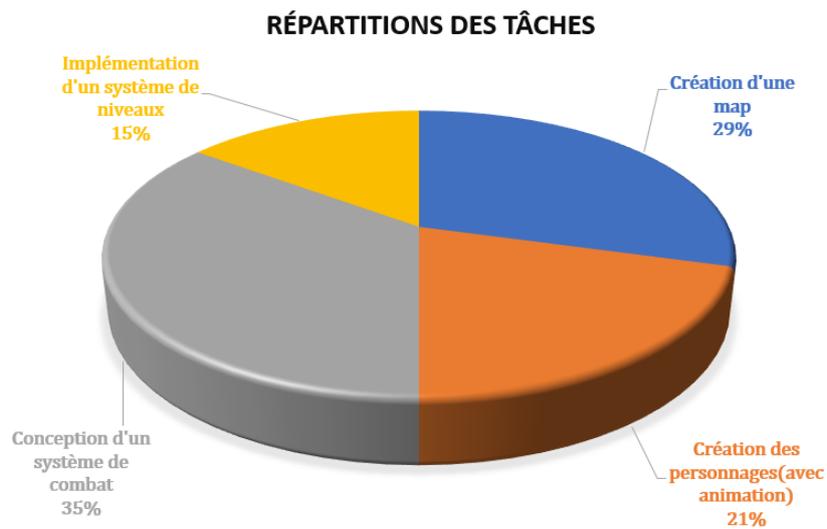


FIGURE 17 – Image illustrant la répartition des tâches

La figure 17 représente la répartition des tâches de chacune des grandes étapes du projet. Il s'agit d'une conclusion graphique résultant de notre gestion de projet.

5.2 Répartitions des tâches entre membres de l'équipe

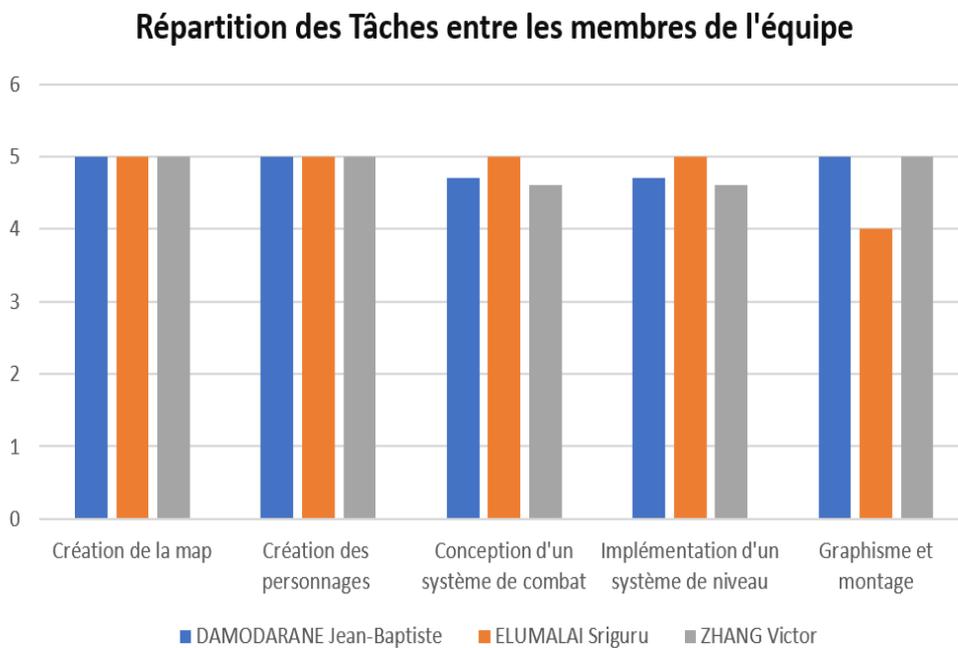


FIGURE 18 – Image illustrant la répartition des tâches entre membres de l'équipe

La Figure 18 représente les taux travaux effectués par les membres du groupe.

6 Conclusion et perspectives

Dans cette section, nous résumons la réalisation du projet et nous présentons également les extensions et améliorations possibles du projet.

6.1 Résumé du travail réalisé

6.1.1 Résumé du jeu

Notre jeu vidéo permet au joueur d'explorer un monde dans lequel il est libre d'effectuer les quêtes qu'il souhaite.

Pour divertir les utilisateurs, nous avons implémenté un système de niveau où le joueur pourra sélectionner des armes en fonction de son avancement afin de battre ses ennemis dans le but d'accomplir ses quêtes.

Différents types d'ennemis avec différents types de pouvoirs y sont présents et peuvent entraîner des dégâts plus ou moins forts pour complexifier l'achèvement des quêtes.

6.1.2 Défis rencontrés

La partie la plus complexe est celle du changement de la map car nous avons dû gérer la position de notre joueur, des NPCs, des monstres et ainsi des objets, en même temps sur la map, au fil des événements rencontrés dans le jeu.

Concernant les combats, nous avons fait face à une difficulté que nous avons su surmonter.

Il s'agit en fait, de la mise en position de l'ennemi, des objets et du joueur lui-même à la fin du combat. En plus de cela, sachant que chaque ennemi vaincu entraîne l'obtention d'un objet, nous avons été amené à faire apparaître ces objets à la même position que ces ennemis vaincus en sortant de la phase de combat.

6.2 Améliorations possibles du projet

- Amélioration des niveaux en rendant les quêtes plus difficiles.
- Utilisation de beaucoup plus d'armes.
- Rendre les mouvements plus fluides.
- Concevoir une option "mode tutoriel".
- Possibilité de changer l'apparence du joueur.
- Ajout des NPCs ayant pour objectif de donner des quêtes secondaires.
- Ajout de beaucoup plus d'ennemis.
- Ajout des effets d'animations pour les monstres et les NPCs et des objets.
- Ajout des effets sonores.

Références

- [Bro03] Barek Bronstring. Perspective pour les jeux d'aventure. *Planète Aventure*, 2003.
- [Seb17] Sebgal. Les bases d'un jeu d'aventure. *Medium.com*, 2017.
- [Tan04] David Adrien Tanguay. Que serait un jeu d'aventure idéal? *Planète Aventure*, 2004.