

Projet Tutoré Rapport

SUS-FIGHTER

Sommaire

I/Objectif du Projet.....	4
1)Modalité.....	4
2) Gestion du projet.....	5
3) Modification.....	6
III/Rapport Technique.....	7
1) Partie Visuel.....	7
2) Partie Objet.....	8
3)Difficultés rencontrées et Améliorations possibles.....	9
a)Difficultés rencontrées.....	9
b) Améliorations possibles.....	10
Bilan.....	11

Introduction

Pour l'option du semestre 5 de la licence informatique, nous avons choisis de faire un projet tutoré avec Mr Forax comme tuteur. On a donc réfléchi à quel type de projet nous voulions faire et avec quel langage de programmation. Nous nous sommes assez vite d'accord sur le fait de faire un projet en Java sur le jeu Street-Fighter. En effet, nous étions en DUT Informatique l'année dernière et avons fait du Java. Cependant, chaque projet étant des projets se déroulant tour par tour, nous voulions pour une fois changer de type de jeu. De plus, nous nous sommes dit que le jeu « Street-Fighter » était assez complexe afin qu'on réussisse à s'améliorer en Java.

Comme Mr Forax était notre professeur de Java, nous sommes à la fin de son cours aller le voir pour lui demander s'il pouvait devenir notre tuteur. Il a accepté et nous sommes mis d'accord pour une première réunion. Pendant cette réunion, nous avons parlé du projet et nous nous sommes donc mis d'accord sur le fait que ce projet allait être sur le jeu « Street-Fighter ».

I/Objectif du Projet

1)Modalité

Le projet est donc basé sur le jeu « Street-Fighter ». Un jeu de combat 2D en 1 contre 1. Cependant, afin de faire le projet, nous devons d'abord mettre sous écrit toutes les modalités du jeu que nous devons implanter. Les modalités du jeu étaient d'ajouter une mode duel permettant à deux joueurs sur un même ordinateur de s'affronter et un mode histoire.

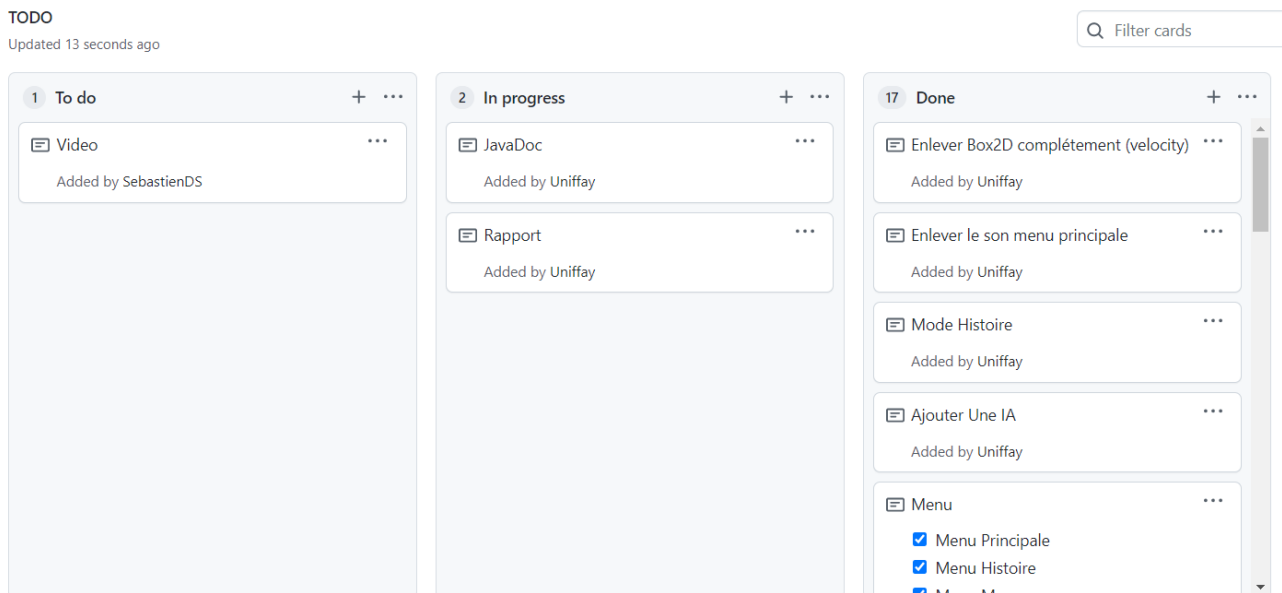
Pour le mode duel, l'utilisateur peut choisir une carte parmi les plusieurs cartes du jeu et peut choisir 2 personnages (un pour chaque joueur). Après avoir choisis, le joueur peut lancer un combat entre les deux personnages sur la carte choisie.

Pour le mode histoire, le joueur commence au premier palier du premier niveau. Il doit alors se battre contre l'ennemi du dernier palier afin de débloquent le palier suivant. Avant le combat, il peut choisir son personnage cependant la carte est en fonction de l'ennemi. Chaque palier est constitué d'un même nombre de sbire et se termine par un boss. Si le joueur bat le boss, il le débloquent alors dans la sélection des personnages et peut donc jouer avec.

Un combat dure 99 secondes et se termine lorsque l'un des deux joueurs est mort (sa barre de point de vie est vide) ou quand le chronomètre atteint 0 (le joueur avec le moins de point de vie meurt s'il ont le même nombre de point de vie, le premier qui attaque l'autre gagne). A la fin du combat, un écran s'affiche pour afficher le vainqueur. Pour un duel, le joueur peut appuyer sur un bouton pour recommencer le duel avec les mêmes options ou alors quitter et revenir au menu principale. Pour le mode histoire, le joueur peut appuyer sur un bouton pour recommencer s'il a perdu ou alors continuer en affrontant l'ennemi du palier suivant.

2) Gestion du projet

Pour effectuer le projet, nous avons décidé de créer un répertoire sur Github. Cela nous permet de sauvegarder notre avancée sur Github afin que chaque membre du groupe puisse l'obtenir facilement avec une commande du terminal. De plus, un lien permet au tuteur de voir le projet quand il le souhaite.



Pour que chaque membre du groupe sache les parties à faire pour avancer dans le projet, nous avons fait une to do liste sur Github.

Cette liste possède trois colonnes : A faire, En cours et Fait.

Durant la conception du projet, chaque membre pouvait faire un élément de la liste A faire. Pour cela, il devait juste mettre dans la liste En cours cette élément afin que l'autre membre du groupe ne fasse pas la même chose. Dès que cette élément était terminer, il était déplacer dans Fait et un autre élément était choisit.

Une fois par semaine, notre groupe faisait un point sur ce qui a été fait pendant la semaine afin de savoir si nous devions augmenter le rythme et si nous devions ajouter des éléments à la liste de mission à faire.

3) Modification

Afin de ne pas reproduire exactement le même jeu que « Street-Fighter », nous avons décidé de changer un peu les mécaniques et le thème du jeu.

Après avoir réfléchi à quel thème et nom le jeu pourrait avoir, un jeu connue et jouer par beaucoup de personne nous est venu en tête. En effet, c'est le jeu « Among us » que nous avons choisit. Ce jeu est un jeu où 4-10 joueurs apparaissent sur une carte et doivent faire un certain nombre de tâche chacun, cependant un nombre défini d'entre eux seront des imposteurs et devront tuer les autres joueurs sans se faire remarquer par les autres car il peut y avoir un vote pour tuer un joueur. Nous avons donc décidé de prendre ce thème pour les personnages du projet et avons nommé le jeu en fonction d'une phrase beaucoup dites par les joueurs lors du vote afin de dire qu'on a des suspicions contre un joueur : « Tu es sus » , d'où le nom Sus-Fighter. Nous avons aussi les personnages du jeu de League of Legends sous la forme des personnages d'Among us afin d'avoir des personnages varié. (Image de certains personnage en Annexe)

Pour ceux qui est des mécaniques, nous avons décidé d'ajouter une barre d'énergie qui augmente quand le joueur tape l'autre joueur. Lorsque la barre est pleine, le joueur peut lancer une attaque ultime sur l'autre joueur qui fait des dégâts normalement plus élevé. De plus, quand un joueur tape un autre joueur, au bout de 3 coups consécutifs, le joueur ennemie est projeté en arrière.

Les statistiques de chaque personnage du jeu de base de Among Us possèdent les mêmes statistiques et sont débloqués dès le début alors que

Quentin BOULET
Sébastien DOS SANTOS

les personnages en référence au jeu League of Legends sont à débloquent dans le mode histoire.

III/Rapport Technique

1) Partie Visuel

Choisir la bibliothèque graphique que nous allons utiliser a été un long périple car chaque bibliothèque graphique de jeu sur java n'était pas à jour et se limitait donc à Java 15. Cependant, une des contraintes du projet était d'utiliser Java 17. Nous avons donc du prendre une bibliothèque qui n'était pas forcément adapté pour les jeux. Notre premier choix se portait sur Swing qui est la bibliothèque graphique de base de java. Cependant, nous avons au final choisit JavaFX que nous avons déjà utilisé en DUT. Cette bibliothèque possède plusieurs avantage. En effet, les éléments de l'interface graphique peuvent être créer grâce à une application faites pour ça appelé Scène Builder. Cette application créer un fichier FXML que l'on peut ouvrir avec JavaFX afin d'afficher la fenêtre avec les éléments. Le fichier possède plusieurs information : le type du conteneur (fenêtre), le nom du contrôleur et les éléments ainsi que leur ID et la méthode a appelé en cas d'action (par exemple `mouseClick:startGame` indique que si l'utilisateur clique sur l'élément, la méthode `startGame` sera lancée.

L'inconvénient de cette bibliothèque, c'est qu'on ne peut pas utilisé de boucle qui gère l'affichage. On doit donc ajouté plusieurs classes de singletons qui sauvegardes en global les informations du jeu ainsi que des timelines qui lance des méthodes à chaque temps x spécifié afin de par exemple regarder si la partie est fini.

Pour les images, elles sont chargées et sauvegardées avant le lancement d'un duel dans une classe nommée `ImageManager`.

2) Partie Objet

Au niveau de la partie objet, nous avons décidé de faire une classe duel qui gère toutes les données du duel. Cette classe stocke donc le temps de la partie ainsi que les différents combattants et la carte du duel.

Pour stocker les Combattants, nous avons créés une interface Combattant qui est implémenter dans le Joueur et l'IA. Pour ensuite mettre toutes les donnée du combattant dans les classes qui l'implémente afin que le duel les sauvegardes dans une liste.

La classe Joueur est sauvegarde donc les coordonnées du personnage, ses statistiques, son nom ainsi que sa hit-box, son élément, ses commandes pour se déplacer, si le joueur est retourné et son nombre de saut maximum.

La classe IA sauvegarde une instance de joueur afin d'avoir les données du personnage mais permet à une IA de se battre sans l'interaction d'un joueur grâce au polymorphisme.

Pour stocker la carte, un record Carte permet de stocker son nom, sa gravité ainsi que ses rebords.

Une classe Commande permet de lié les touches du clavier à une commande (par exemple : z pour sauter) .

Pour sauvegarder les personnages débloquent ainsi que l'étape du mode histoire, une classe Sauvegarde crée un fichier binaire de la progression du joueur. Le fichier est en binaire pour éviter toutes tricheries.

Pour finir, la classe Statistique sauvegardes toutes les stats du joueurs (point de vie, énergie, vitesse) et la classe Vec2 sauvegarde des coordonnées.

3) Difficultés rencontrées et Améliorations possibles

a) Difficultés rencontrées

Pendant la conception de ce projet, plusieurs problèmes ont été rencontrés. En effet, le premier problème que nous avons rencontré est, comme expliqué plus tôt, pour le choix de la bibliothèque graphique.

Aucune bibliothèque graphique de jeu n'était à jour donc nous nous étions tournés vers Swing. Cependant, lors de la conception du menu, l'affichage d'une image de fond avec des boutons n'étaient pas possible avec ce que l'on avait trouvé et comme Swing ressemblaient fortement à JavaFX, nous avons décidé, d'au lieu de changer presque toute la partie Swing que nous avions faites, de changer pour JavaFX ce qui nous a fortement ralenti.

Le deuxième gros problèmes que l'on a rencontré est au niveau de la conception des hit-box. Nous avons, au début, décidé de faire des hit-box entre les personnages. Cependant, cela fonctionnait pour presque toutes les collisions sauf celle du blocage de saut quand un ennemi était sur la tête de l'autre. De plus, cela était mal fait et possédait donc plusieurs bug avec certaines interactions. Nous avons donc décidé d'utiliser Box2D, une bibliothèque fait pour gérer les collisions. Malheureusement, cela ne fonctionnait pas comme l'on voulait et après avoir testé pendant longtemps, nous avons décidé de changer notre façon de faire et donc d'enlever les collisions entre les personnages. Nous avons donc refait d'une meilleur façon les collisions des joueurs entre les bordures de la cartes et celle des poings du joueurs sur l'autre joueur. Les joueurs peuvent donc se traverser comme dans le jeu Brawlhalla (un autre jeu de combat 2D). Ce problème est celui qui nous a pris le plus de temps et nous a vraiment beaucoup retardé.

b) Améliorations possibles

Pour améliorer le jeu, beaucoup d'améliorations sont possibles. En effet, nous avons pensé à l'ajout d'équipement qui serait obtainable lors de l'affrontement des sbires et des boss de l'histoire et donnerait des statistiques supplémentaire au personnage et s'ajouterait donc au visuel du personnage. De plus, l'ajout d'une armes qui serait obtainable comme l'équipement et qui pourrait donner des capacités supplémentaires comme des dégâts passifs après avoir touché une attaque, une double attaque, un bouclier lorsque le joueur devait mourir (cela n'est que des exemples). Cependant ces armes auraient une durabilité descendante à chaque attaque et devront donc être changer lorsqu'elle se casse. Pour cela, plusieurs caisses d'armes pourraient apparaître de temps en temps sur le terrain afin que les joueurs puissent changer/récupérer une nouvelle arme. Cependant l'arme de la caisse est aléatoire.

Une autre amélioration possible serait d'ajouter des événements sur la carte. Par exemple, sur la carte du volcan, le volcan pourrait tomber en éruption et donc pleuvoir des rochers enflammés qui feront des dégâts s'il touche le joueur, pour le glacier, il pourrait y avoir une tempête de glace qui ralentie les joueurs et baisse la visibilité des personnages du jeu.

Une dernière amélioration que nous avons pensé et qui demanderait que du temps comme pour les deux améliorations précédentes serait de diversifié les personnages par autre chose que seulement leur statistique. Par exemple, l'ajout d'un système de distance. Les personnages pourrait donc utilisés des attaques de bases qui serait différentes que simplement une attaque au corps à corps. De plus, les personnages pourraient avoir des compétence unique comme une invisibilité, du soin, du vol de vie, etc... ainsi qu'un ultime différent.

Bilan

Ce projet tutoré a été une très bonne expérience. En effet, elle nous a permis de nous améliorer dans le langage Java ainsi que dans la gestion de projet. Cela nous a donc aussi permis de nous préparer à notre future. Effectivement, faire des projets en groupe nous prépare pour nos futures projets en entreprise.

De plus, ce projet était notre premier projet en temps réel au lieu d'un projet en tour par tour. Cela nous a donc permis de nous améliorer d'un façon globale et non seulement au langage Java.

Pour finir, le temps que nous avions étaient assez court au vue des différents projets que nous devions réaliser en amont et des différents problèmes rencontrés qui nous ont pris pas mal de temps. Cependant, nous avons réussis à respecter les délais imposés. Ces contraintes de temps et les différents problèmes rencontrés rapprochent encore plus se projet à un projet en entreprise.

Quentin BOULET
Sébastien DOS SANTOS

ANNEXE

Échantillons personnage du jeu :



Jeu de base



League of Legends