# SᲾDA : Plateformeur 2D

Catillaz Dallas

Ruegger Yann

Rodrigues dos santos Fabio

18.11.2020

IFA-P3A

Table des matières

[SᲾDA : Plateformeur 2D 1](#_Toc59620351)

[1. Introduction 3](#_Toc59620352)

[1. Pourquoi ce sujet 3](#_Toc59620353)

[2. Quel IDE ? 3](#_Toc59620354)

[2. Cahier des charges 3](#_Toc59620355)

[3. But 3](#_Toc59620356)

[4. Spécifications 3](#_Toc59620357)

[5. Restrictions 3](#_Toc59620358)

[6. Environnement 4](#_Toc59620359)

[7. Planning prévisionnel 4](#_Toc59620360)

[8. Planning en temps réel 4](#_Toc59620361)

[3. Analyse fonctionnelle 4](#_Toc59620362)

[1. Fonctionnalités 4](#_Toc59620363)

[4. Analyse Organique 6](#_Toc59620364)

[1. Diagramme de classe 6](#_Toc59620365)

[2. Pseudo-code 6](#_Toc59620366)

[3. Quelques tests 8](#_Toc59620367)

[4. Conclusion 8](#_Toc59620368)

## Introduction

### Pourquoi ce sujet

Nous avons choisi de faire ce projet car ayant déjà créé d’autres jeux en 2D avec des objets et autre connaissances en C# nous avons trouvé cela judicieux d’essayer de transporter nos connaissances et de les agrandir en créant un jeu 2D sur un véritable environnement de développement de jeux tel que Unity.

### Quel IDE ?

Nous nous sommes décidés à utiliser Unity car n’ayant pas vraiment eu à faire à cet environnement de développement, nous avons donc pensé que ça serait une expérience enrichissante d’apprendre à l’utiliser et de faire un projet avec étant donné que c’est un des nombreux environnements qui sont conçus pour faire des programmes tels que des jeux 2D ou 3D.

## Cahier des charges

### But

Création d’un jeu solo dans lequel on contrôle un personnage 2D qui va progresser dans un niveau et affronter un certain nombre d’ennemis tout en évitant de succomber aux pièges du niveau afin d’affronter un boss à la fin du niveau.

### Spécifications

* Contrôler un personnage à la 2ème personne
* Attaquer des ennemis avec des armes
* Combat contre un boss possédant différentes attaques
* Exploration d’un/de niveau/x
* Esquiver des coups ou une autre mécanique permettant cela

### Restrictions

* Uniquement jouable avec un clavier et une souris.
* Fonctionne uniquement sur Windows.

### Environnement

* Ordinateur de type PC sous Windows 10
* Environnement de développement de jeu Unity
* Visual Studio 2019

### Planning prévisionnel

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Analyse |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Conception |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Réalisation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Validation + rapport de test |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Bilan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Livre de bord |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Documentation technique |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

### Planning en temps réel

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Analyse |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Conception |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Réalisation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Validation + rapport de test |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Bilan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Livre de bord |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Documentation technique |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## Analyse fonctionnelle

### Fonctionnalités

#### Déplacement joueur :

Le joueur peut se déplacer en appuyant sur les touches :

* A : Le joueur va effectuer un déplacement vers la gauche
* D : Le joueur va effecteur un déplacement vers la droite

De plus, en appuyant sur la barre espace le joueur va effectuer un saut. Plus cette touche est maintenue longtemps plus le joueur saute haut.

#### Attaque joueur :

Le joueur en appuyant sur le clic gauche de sa souris peut effectuer une attaque avec une portée définie au préalable et dans une certaine zone dans la direction vers laquelle il regarde et infliger des dégâts à un ennemi qui verra sa barre de vie réduite.

#### Blocage :

Le joueur en maintenant le clic droit de sa souris peut se mettre en position de blocage ce qui va nullifier tout dégât qui était censé lui causer des dommages. Cependant le bouclier ne peut résister qu’à 2 attaques tout type confondu et ensuite le bouclier se verra rendu inutilisable par le joueur pendant un délai de temps définis avant d’être à nouveau capable de le réutiliser.

#### Barre de vie

Toute entité dans le jeu possède une barre de vie qui se réduit en réponse aux dégâts que l’entité subit et si elle est vide l’entité meurt. Bien évidemment le nombre de points de vies dépendent de l’entité en question.

#### Attaque ennemi

Les ennemis dans le jeu pourront effectuer 2 types d’attaques :

* Tirer des projectiles
* Attaquer au corps à corps

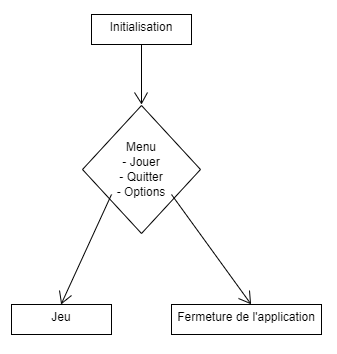
Ces attaques infligent des niveaux de dommages différents et avec une portée différente.

#### Détection du joueur par l’ennemi

Lorsqu’un ennemi a le joueur à une certaine portée de lui et dans son champ de vision il va passer en mode chasse et va se mettre à poursuivre le joueur. Si le joueur s’éloigne trop il va retourner en mode passif et continuer à se déplacer de gauche à droite en suivant un chemin prédéfinis.

## Analyse Organique

### Organigramme général



### Diagramme de classe

### Pseudo-code

##### PlayerMovement

Lorsqu'une des touches de déplacement est pressée, l'animation de déplacement va se lancer. Va être ensuite testé la direction du joueur et effectuer une rotation ce dernier en conséquence. Ensuite étant donné que le joueur est en déplacement sa vitesse de déplacement va être modifiée en conséquence pour que l'objet Joueur se déplace dans la bonne direction. Si aucune touche n'est pressée, l'animation de déplacement s'arrête. Lorsque la touche espace est pressée, le joueur va se mettre à sauter et l'animation de saut se lancera et lors qu'il touche le sol elle s'arrête.

##### ProjectileScript

Lorsqu'un projectile est créé et qu'il entre en collision avec un autre objet, on va tester que ce n'est pas un ennemi et que c'est bien le joueur et ensuite infliger des dommages si c'est un joueur et après le projectile va se détruire

##### golemScript

Le golem va se mettre à se déplacer dans une direction et lorsqu'il va entrer en contact avec un trigger "Turn" il va se mettre à se déplacer dans le sens opposé. Lorsque le joueur se trouve à une certaine portée du golem, le golem va accélérer et se mettre à attaquer le joueur. Lorsque les points de vie du golem sont à 0, l'animation de mort du golem est jouée et le golem disparait

##### planteScript

La plante se mets à tirer des projectiles lorsque le joueur entre dans sa zone de tir et si le joueur est trop proche de la plante elle va se mettre à faire des attaques de corps à corps. Lorsque les points de vie de la plante sont à 0, l'animation de mort de la plante est jouée et la plante disparait

##### GameHandler

Permet de faire en sorte que lorsque des entités prennent des dégâts, la barre de vie et la variable contenant leurs points de vie se réduisent en conséquence

##### PlayerAttack

Lorsque le joueur appuie sur clic gauche de la souris, que le temps de recharge entre chaque attaque est à 0, qu'il se trouve au sol et qu'il n'est pas en déplacement, le joueur va effectuer une attaque. Si des ennemis se trouvent à portée de l'attaque, ils vont subir un nombre fixe de dégâts et si un projectile se trouve à portée, il sera détruit. Lorsque le joueur appuie sur clic droit, qu'il se trouve au sol et que son bouclier est utilisable, le joueur va lever son bouclier. Lorsque son bouclier est levé il va pouvoir bloquer un nombre de dégâts égal à un nombre définit et après son bouclier sera rendu inutilisable et il ne pourra plus le lever tant que le temps de recharge du bouclier n'est pas à nouveau à 0. S’il y a un temps de recharge supérieur à 0, il va se réduire graduellement.

##### EnemyAttack

Lorsque l'ennemi peut attaquer et que son temps de recharge entre chaque attaque est à 0, l'ennemie en question va effectuer une attaque infligeant des dégâts uniquement au joueur s'il se trouve à portée. S'il y a un temps de recharge supérieur à 0, il va se réduire graduellement.

##### Ennemy

Ce script contient les méthodes nécessaires à la plante pour qu'elle puisse tirer ses projectiles ou pour que les ennemis puissent subir des dégâts

### Quelques tests

Nous avons testé le fait que le joueur ne puisse effectuer d’attaques ou lever son bouclier lorsqu’il est en l’air et le résultat est concluant.

Nous avons testé le fait que la plante tire bien des projectiles lorsque le joueur se trouve à portée de la plante et le résultat est concluant.

Nous avons testé le fait que le Golem se retourne bien en touchant les triggers « Turn » et le résultat est concluant.

Nous avons testé le fait que le Golem poursuive bien le joueur lorsqu’il est à portée et le résultat est concluant.

### Conclusion

Ce que nous avons constaté après la réalisation de ce projet est que nous avons été pris légèrement de court au niveau du planning et nous avons dû continuer de coder le jour même du rendu. Ce problème a été en partie causé par le fait qu’en plein milieu du temps qui nous a été alloué pour la réalisation du projet nous avons radicalement changé le type de jeu car ce dernier à la base était en 3D et maintenant il est en 2D ce qui nous a donc coûté énormément de temps ce qui a fait que nous avons dû dépasser le délai prévu à la base pour la réalisation du projet.

De ce fait nous estimons que le projet dans son état actuel est partiellement incomplet à cause de ce changement de projet en plein milieu ce qui a donc fait que nous avons dû réduire nos objectifs de moitié afin de pouvoir livrer une version un minimum fonctionnelle.

Lors de la réalisation de ce projet nous avons appris à un peu mieux prendre en main l’environnement Unity et ses différentes fonctions propres à ce dernier et nous nous estimons un peu plus à l’aise quand au développement sur Unity en C# malgré les difficultés rencontrées en chemin.

Ce projet était néanmoins une bonne expérience et était avant tout un bon exercice pédagogique pour agrandir nos compétences dans un environnement nouveau en compagnie de nos camarades ce qui a parfois été une bonne chose pour surmonter les quelques parties difficile lors de la réalisation du projet.