

Rapport de soutenance finale:

## **Zorestone Realm**

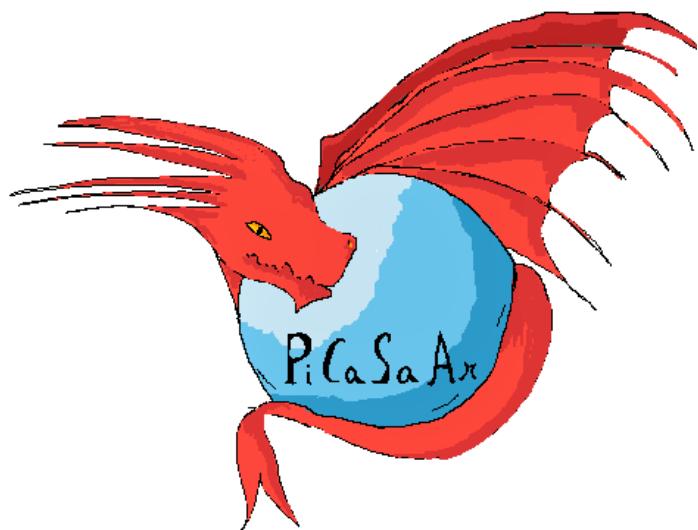
### **PiCaSaAr**

Camille Michon

Pierre Fouché

Samy Yacef

Arthur Le Flem



## Sommaire :

<b>I/: Introduction :</b>	<b>2</b>
<b>A/: Création du groupe :</b>	<b>2</b>
<b>B/: Le Jeu :</b>	<b>3</b>
a/: Histoire :	3
b/: Protagonistes :	4
<b>C/: Inspiration Artistique :</b>	<b>6</b>
a/: L'environnement de Jeu :	6
b/: L'état de l'art :	7
<b>II/: Réalisation :</b>	<b>8</b>
<b>A/: Prise en main des outils :</b>	<b>8</b>
<b>B/: Mise en place du projet :</b>	<b>13</b>
a/: Gameplay :	13
b/: Multijoueur :	19
c/: Intelligence artificielle :	23
d/: Site web :	27
e/: Effets sonores :	27
<b>III/: Expériences personnelles :</b>	<b>29</b>
<b>A/: Camille Michon :</b>	<b>29</b>
<b>B/: Pierre Fouché :</b>	<b>30</b>
<b>C/: Arthur Le Flem :</b>	<b>31</b>
<b>D/: Samy Yacef :</b>	<b>32</b>
<b>IV/: Conclusion :</b>	<b>34</b>

## I/: Introduction :

Ce document relate la création de notre projet, depuis l'idée originale jusqu'au développement final de celui-ci. Il sera ici mentionné la manière dont nous avons implémenté les différents éléments informatiques du projet, de même que les problèmes que nous avons pu rencontrer tout au long de cette aventure.

## A/: Crédit du groupe :

Notre groupe s'est formé à l'annonce du lancement du projet du semestre deux. Cela s'est fait de manière presque naturelle puisque Camille et Pierre se connaissaient déjà avant de venir à l'Epita. De plus, lors du séminaire de mathématique, attirés par leur intérêt à propos des jeux vidéos, Samy et Arthur les ont rejoints pour partager leur vision de ce domaine. Nous nous sommes alors aperçus que nous partagions beaucoup de points communs.

Nous avons donc décidé de nous rassembler pour créer un jeu qui correspond à notre vision de joueur, et ce projet du semestre deux était l'opportunité parfaite. Notre groupe était donc déjà presque formé.

Nous avons donc choisi comme nom de groupe, PiCaSaAr, en allusion aux premières lettres de chacun des prénoms des membres de groupe afin que chacun puisse laisser sa marque.

Depuis, il n'y a jamais eu de problème majeur de coopération ou de dissonance dans la coordination entre les membres du groupe. En effet, ce projet nous a appris à travailler en équipe, et à faire abstraction de détails désagréables qui pouvaient influencer notre comportement.

L'entente entre les membres du groupe a été permis, entre autres, grâce à une cohésion que nous avons entretenue à l'aide de réunions hebdomadaires à propos de l'évolution du projet, de ce qui avait été implémenté et de ce que nous souhaitions faire ensuite.

Ainsi, encore aujourd'hui, PiCaSaAr est efficace, et prospérera peut-être en réalisant d'autres projets avec cette même coordination et organisation qui nous a permis de réaliser Zorestone Realm.

## B/: Le Jeu :

L'idée principale nous est venue lors de la première réunion à propos de ce projet. En tant que joueur, nous voulions un jeu qui motiverait les utilisateurs à y jouer. C'est pour cette raison que nous avons développé un jeu se basant sur l'histoire. Ainsi, les joueurs s'impliqueront dans cette aventure pour en apprendre plus sur le royaume de Zores.

Nous avons pensé que pour transmettre cette histoire sur laquelle se basait notre projet, le meilleur format était un jeu en deux dimensions, où le joueur doit traverser les différents niveaux de gauche à droite. Ce format nous permet ainsi de raconter l'histoire du royaume entre les niveaux, sans perturber la progression du joueur.



## a/: Histoire :

L'histoire se déroule dans le royaume fantastique de Zorestone Realm. Alors que la paix et l'harmonie perdurent depuis des millénaires même si quelques conflits mineurs ont entraîné la transmission des traditions militaires, un personnage démoniaque, accompagné de son armée aux soldats innombrables, détruit alors le royaume et défait tous ses habitants lors d'une nuit de sang et d'horreurs. Il parvient à s'emparer de la pierre de Zoré puis s'enfuit laissant

derrière lui un royaume brisé et en flamme. Certains survivent et parmi eux, 4 protagonistes aux destins liés et hors du commun.

**b/: Protagonistes:**

Ainsi ces quatres aventuriers se retrouvent après avoir expérimenté les pires moments de leur existence :

- La prêtresse était autrefois la Grande Prêtresse de l'Ordre des Défenseurs de l'Aube dont le titre qui revêtait une grande importance n'est aujourd'hui plus qu'honorifique. Son Ordre étant détruit et son temple en ruine, elle n'avait plus d'autre choix que de se lancer à la poursuite de celui qui lui avait tout pris. Armée de son bouclier et de sa promesse de protéger ses alliées, elle inflige certes moins de dégâts mais possède une résistance bien supérieure aux autres.



- Le mage était quant à lui Doyen de l'Académie des Mages. Entré dans cette académie à l'âge de 15 ans, elle représentait sa vie entière, détruite en une nuit de cauchemar. A présent sa détermination n'est possible que par son désir de vengeance, ancré en lui par un serment de magie noire. Malgré une grande faiblesse physique, son savoir astronomique lui permet de détruire ses ennemis à grand renfort de sorts de feu, une force de frappe non négligeable.



- La princesse était aimée de ses sujets et c'était son peuple qui formait son royaume. Celui-ci détruit et ses citoyens annihilés, sa fonction n'est plus que poudre aux yeux. Ainsi, ayant troqué sa robe richement brodée pour un manteau d'assassin, elle parcourt ses terres avec comme espoir de soigner son royaume du mal qui le ronge. Equipée de sa dague aussi tranchante que ses paroles, elle est mobile et violente, mais reste faible face aux monstres importants et ceux tirant à distance.



- L'elfe vivait en paix dans sa forêt avec ses semblables lors de cette catastrophe. Fraîchement promu Capitaine de Chasse et des Gardes Frontières dû à ses compétences au tir à l'arc, il s'est retrouvé impuissant devant la déferlante de monstres qui a rasé son village. Seul survivant de son espèce, il souhaite recréer son village grâce aux légendes de son enfance à propos d'autres tribus d'elfes

sylvestre. Habile avec son arc, ses tirs sont précis et peuvent faire un carnage dans le rang des monstres, même si celui-ci reste peu utile au corps à corps.



Les 4 héros jouables.

### C/: Inspiration Artistique :

#### a/: L'environnement de Jeu :

Nous nous sommes donc concertés sur le choix artistique. Nous voulions tout d'abord faire un jeu fonctionnel, certes, mais aussi visuellement correct. C'est pourquoi nous avons décidé de faire un jeu en 2D. Mais quoi de mieux que d'inviter le joueur dans notre univers, pour profiter pleinement du jeu ? Il était donc logique d'avoir une histoire cohérente et évolutive au fil de l'aventure. Nous nous sommes donc inspirés des nombreux univers fantastiques que nous côtoyons depuis notre enfance. Ainsi, l'univers est créé de toutes pièces, mais avec cette touche fantastique que nous retrouvons dans Le Seigneur des anneaux, Eragon, mais aussi Skyrim et beaucoup d'autres œuvres de fiction. Nous voulons ainsi que le joueur ne se sente pas étranger à cet univers, dans lequel il découvrira l'histoire des différents personnages.

**b/: L'état de l'art :**

Ce jeu s'inscrit dans l'univers des platformers et plus précisément platformers 2D. Le joueur peut déplacer le personnage qu'il contrôle et le faire sauter sur des plateformes qui sont omniprésentes dans ce type de jeu. Ce type de jeu est apparu au milieu des années 80 grâce à deux jeux qui sont considérés comme les prémisses du genre : « Space Panic » et « Crazy Climber ». Depuis les platformers ont connu certaines améliorations comme le scrolling qui caractérise le déplacement horizontal et vertical de l'écran. Ce type de jeu est encore incontournable aujourd'hui grâce notamment au personnage de « Mario » qui est une véritable icône du genre. Nous pouvons encore citer les premiers Castlevania qui sont représentatifs depuis 1986 du type platformer.

En tant que joueurs, nous avons pu apprécier ces jeux, mais aussi souffrir de certains aspects que nous pouvons considérer comme négatifs. C'est pourquoi nous voulons pour ce projet les éviter tout en conservant les points positifs de ces jeux. Ainsi nous avons voulu développer l'univers dans lequel le joueur évolue, en créant une véritable histoire derrière chaque personnage afin de ne pas être limité dans notre narration. Par exemple, le scénario de Mario reste limité à sauver la princesse capturée par Bowser à chaque opus de la série. Dans Castlevania, même si les personnages changent, le scénario reste le même et l'histoire du “méchant” à combattre évolue peu.



## II/: Réalisation :

### A/: Prise en main des outils :

Pour réaliser notre idée, nous avons utilisé de nombreux outils, le premier étant Unity. Nous avons été initiés au premier semestre, en cours de NTS, à l'outil Unity, moteur de jeu, ou application multi plateforme, à travers une application utilisant la réalité augmentée. Il supporte le langage de programmation orienté objet C#, que nous avons étudié. Cette première initiation nous a confrontés au monde de la création de jeux vidéo où rigueur et sens de l'organisation sont maîtres.

Au-delà de la facilité d'utilisation du logiciel ainsi qu'une documentation riche et complète, les possibilités infinies que proposent Unity nous ont tout de suite séduits et nous ont donc amené à utiliser ce logiciel pour la conception de notre jeu.



La documentation très fiable et profuse en ligne nous a permis de réaliser ce que nous souhaitions, c'est-à-dire un jeu qui nous plaisait en tant que joueurs mais aussi en tant que créateurs. Ainsi, grâce aux nombreux créateurs sur Unity, nous avons pu comprendre comment ce logiciel fonctionnait, mais aussi dû adapter de nombreux éléments qui n'existaient pas sous la forme que nous voulions.

Au-delà de la compréhension du logiciel, nous avons dû établir les bases de notre projet. Nous avons donc commencé, comme il était prévu, par dessiner les tiles qui constitueront nos niveaux. Cette partie a été relativement longue et s'est finalement étalée sur le projet en entier. Néanmoins nous avons réussi notre

objectif qui était de n'utiliser aucun asset déjà existant et de créer notre univers par nous-mêmes.

Une fois les tiles obtenues, il nous a fallu nous pencher sur l'aspect qui donnerait envie au joueur de s'impliquer dans cette aventure : les différents niveaux. En effet, un niveau est en grande partie ce qui va retenir l'intérêt du joueur, et lui permettre de s'immerger dans l'univers proposé. Si les niveaux sont fades, ou se ressemblent, l'utilisateur sera rapidement blasé par ce qu'il a devant lui et décidera sans doute qu'il a mieux à faire. C'est cette multitude de mondes

que l'on retrouve par exemple dans Mario, ou encore dans Zelda, qui sont deux classiques des jeux en deux dimensions.

Nous avons donc décidé, pour le premier niveau, qui est avant tout un tutoriel, de créer une carte relativement plate, afin de permettre aux joueurs de se familiariser avec l'environnement et les contrôles.

Pour les niveaux suivants, nous avons décidé de jouer sur la verticalité permise par l'utilisation de la deuxième dimension. Contrairement à Mario, où les tuyaux permettent de rejoindre des niveaux, la verticalité sera exploitée par chaque niveau séparément.

La question de la création de notre univers nous a amené à utiliser un logiciel en ligne nous permettant de créer nos blocs de pixels : pixilart. Grâce à celui-ci, nous avons pu réaliser notre ambition qui était de créer notre propre univers avec les éléments que nous voulions.



Après cela, nous avons donc créé nos cartes comme expliqué précédemment. Ensuite, pour que le joueur puisse profiter d'une immersion pleine et entière, il était nécessaire de créer une ambiance, une atmosphère permettant à l'utilisateur de vivre l'aventure comme un des personnages. Cette ambiance est mise en place par la musique qui permet de transmettre des émotions.

Pour que les musiques placent l'ambiance que nous souhaitions, nous avons décidé de les créer nous-mêmes, dans la même optique que la création de nos propres tiles. Pour ce faire, nous avons utilisé FL Studio 12, un logiciel de création de musique très complet et facile d'accès.



Ainsi, comme les utilisateurs ont pour mission de sauver le royaume de Zores alors qu'un monstre vient de le priver de sa richesse, nous avons axé les sons sur une ambiance angoissante afin que l'utilisateur ressente la situation du royaume de Zores. Créer la musique a été quelque peu compliqué, car il fallait mobiliser nos connaissances musicales, qui se résumaient, pour tous les membres de notre groupe, à écouter de la musique et non à en jouer ou encore moins de la créer.

Ces trois logiciels, Unity, FL Studio 12 et Pixilart nous ont ainsi permis de développer notre projet, avec certaines difficultés certes, mais avec une satisfaction profonde de savoir que cela n'existe que dans notre projet.

Nous nous sommes donc répartis le travail selon les intérêts de chacun. Ceux qui connaissaient déjà le domaine, ou qui en savaient déjà plus que les autres ont été nommés chefs de pôle. Ainsi, cela permet une meilleure coordination et organisation afin de centraliser les différents pôles.

Répartition des différents pôles entre les membres de PiCaSaAr :

	Camille	Arthur	Pierre	Samy
Environnement	✓		✓	✓
Gameplay		✓	✓	
Interface	✓			✓
Son		✓	✓	
Réseau	✓		✓	✓
IA	✓	✓		
Site Web		✓		✓
Test	✓	✓	✓	✓

Cependant, il était impossible de pouvoir réaliser ce projet sans un planning réaliste et surtout déterminé. En effet, sans une organisation stricte, il était certain que nous ne réaliserais jamais les objectifs fixés. Nous avons donc établi un planning pour être sûr d'être toujours dans les temps.

Planning répartit sur les différentes soutenances pour le groupe PiCaSaAr

	Soutenance 1	Soutenance 2	Soutenance 3
Environnement	30%	90%	100%
Gameplay	60%	90%	100%
Interface	40%	70%	100%
Son	0%	45%	100%
Réseau	50%	80%	100%
IA	50%	90%	100%
Site Web	10%	70%	100%
Test	/	/	/

## B/: Mise en place du projet :

### a/: Gameplay :

Comme tous les platformers, le but est de venir à bout de plusieurs niveaux en terrassant les monstres qui les infestent, avant de défier le Boss final ce qui, lors de la victoire contre celui-ci, clos le jeu. Chaque niveau sera différent du précédent par le nombre et les types de monstre présents, mais aussi par sa construction. De plus, chacun sera créé de manière logique par rapport à la région où se situe le joueur.

Une fois tous les monstres d'un niveau éliminés, le joueur obtient une clé permettant de débloquer le niveau suivant et de manière plus rare, un objet générant un dialogue, révélant à chaque fois un peu plus l'histoire de l'univers et des personnages. Ainsi le joueur peut à la fois jouer pour l'histoire qu'il suit ou bien pour le plaisir de finir un niveau compliqué.

Ce type de gameplay correspond à notre vision des caractéristiques d'un jeu auquel nous voudrions jouer. A la fois motivant par la difficulté de compléter un niveau mais aussi la richesse d'une histoire et des personnages, trop souvent négligés dans les jeux.

Pour ce qui est du Gameplay, comme écrit au-dessus, les joueurs doivent finir les niveaux en battant les ennemis afin de sauver le Royaume de Zorestone Realm. Ils doivent donc parcourir tous les niveaux de la gauche vers la droite comme de nombreux platformer. Le joueur est accompagné par des dialogues et une révélation progressive de l'histoire tout au long de la progression du jeu.

Nous avons ainsi implémenté les déplacements des joueurs, voici le script qui permet au joueur de se déplacer vers la droite ou vers la gauche ainsi que de sauter.

```

65    void MovePlayer(float _horizontalMovement)
66    {
67        Vector3 targetVelocity = new Vector2(_horizontalMovement,rb.velocity.y);
68        rb.velocity = Vector3.SmoothDamp(rb.velocity, targetVelocity, ref velocity, .05f);
69        if(isJumping)
70        {
71            rb.velocity = (new Vector2(0f ,jumpForce)); // Jump
72            isJumping = false;
73        }
74    }

```

Pour que le personnage puisse sauter il faut qu'il touche le sol, c'est ce que la ligne ci-dessous vérifie.

47

```
isGrounded = Mathf.Abs(rb.velocity.y) < 0.001f;
```

Ou bien, si le personnage est capable de faire un double saut et qu'il n'a fait qu'un seul saut alors il peut sauter également.

Pour finir le personnage change de côté lorsque le joueur appuie sur la flèche gauche ou droite. C'est ce que le script ci-dessous permet.

```
76 void Flip(float fleche)
77 {
78     if (fleche > 0.0)
79         spriteRenderer.flipX = false;
80     else if (fleche < 0.0)
81         spriteRenderer.flipX = true;
82 }
```

Nous avons aussi implémenté le script suivant accordant au joueur une barre de vie en haut à gauche. Celle-ci est dynamique et sera automatiquement modifiée à chaque fois que le personnage sera attaqué.

```
1  using UnityEngine;
2
3  public class PlayerHealth : MonoBehaviour
4  {
5      public int maxHealth = 100;
6      public int currentHealth;
7      public HealthBar healthBar;
8      void Start()
9      {
10         currentHealth = maxHealth;
11         healthBar.SetMaxHealth(maxHealth);
12     }
13     void Update()
14     {
15         if (Input.GetKeyDown(KeyCode.H))
16         {
17             TakeDamage(20);
18         }
19     }
20
21     void TakeDamage (int damage)
22     {
23         currentHealth -= damage;
24         healthBar.SetHealth(currentHealth);
25     }
26 }
```

De plus, puisque le joueur peut incarner plusieurs personnages différents, il était nécessaire de les différencier. Nous avons donc choisi pour cela d'impacter les déplacements des différents personnages. Ainsi , à l'aide de différentes variables, nous avons pu modifier leur vitesse de déplacement, la hauteur de leur saut, ou encore leur nombre de sauts :

Archer : Vitesse => 1.25  
Hauteur de saut => 1.25  
Nombre de sauts => 1

Assassin : Vitesse => 1.5  
Hauteur de saut => 1.5  
Nombre de sauts => 2

Guerrière : Vitesse => 1  
Hauteur de saut => 1  
Nombre de sauts => 1

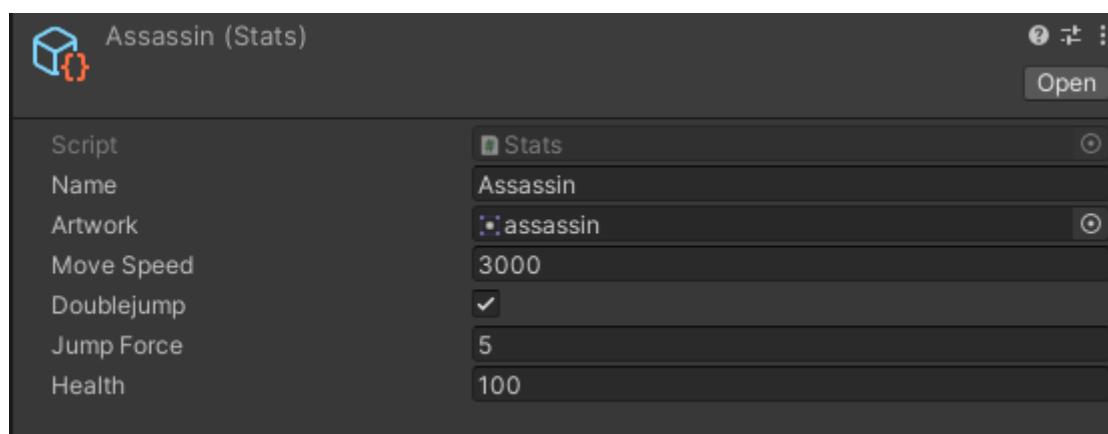
Mage : Vitesse => 1  
Hauteur de saut => 1  
Nombre de sauts => 1

Par ailleurs, pour accorder toutes ces valeurs aux différents personnages, nous avons utilisé des scriptableObjects. Ils permettent le stockage de grosses masses de données plus facilement en un seul fichier de petite taille, et d'y avoir accès rapidement. Ainsi la classe Stats fut créée et contient toutes les données essentielles d'un personnage.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  [CreateAssetMenu]
6  public class Stats : ScriptableObject
7  {
8      public string name;
9      public Sprite artwork;
10     public int moveSpeed;
11     public bool doublejump;
12     public int jumpForce;
13     public int health;
14 }
```

Nous avons ci-dessous un exemple d'Objet de type Stats contenant les stats de notre assassin:



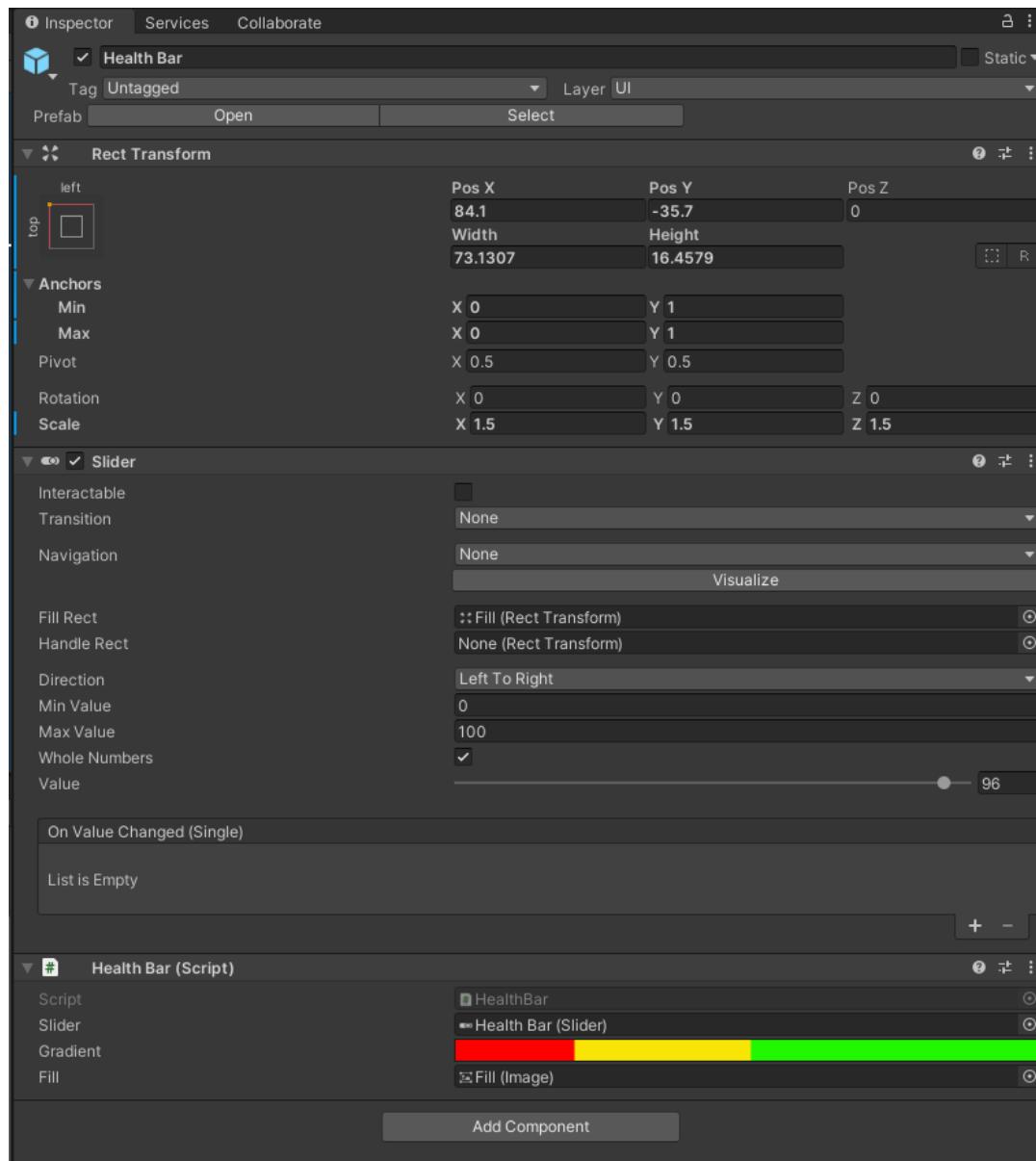
Pour ce qui est du Gameplay, comme écrit au-dessus, les joueurs doivent finir les niveaux en battant les ennemis afin de sauver le Royaume de Zorestone Realm. Ils doivent donc parcourir tous les niveaux de la gauche vers la droite comme de nombreux platformer. Le joueur est accompagné par des dialogues et une révélation progressive de l'histoire tout au long de la progression du jeu.

Pour permettre au jeu d'être complexe, afin d'accorder au joueur un réel défi, dès que la barre de vie du joueur tombe à zéro, celui-ci doit reprendre le

niveau depuis le départ, sans ses coéquipiers encore en vie plus loin. Il faut donc choisir en connaissance de cause s'il est préférable d'éviter un ennemi ou bien de l'abattre pour permettre un retour rapide si un joueur venait à tomber.

La barre de vie, l'élément le plus important pour les joueurs a été implémentée pour que les utilisateurs ne se perdent pas dans les chiffres.





Comme nous pouvons le voir sur cet Inspecteur de Unity, la barre de vie est appliquée aux joueurs en fonction du protagoniste qu'ils choisissent. Ensuite, celle-ci devient propre au joueur en question, ce qui permet d'éviter que lorsqu'un joueur subit des dégâts, ceux-ci soient répercutés injustement sur un autre équipiers.

```

 6  Script Unity (2 références de ressources) | 1 référence | Changed by samy.yacef@epita.fr on jeudi 21 avril 2022
 7  public class HealthBar : MonoBehaviour
 8  {
 9
10     public Slider slider;
11     public Gradient gradient;
12     public Image fill;
13
14     1 référence | Changed by samy.yacef@epita.fr on jeudi 21 avril 2022
15     public void SetMaxHealth(int health)
16     {
17         slider.MaxValue = health;
18         slider.value = health;
19
20         fill.color = gradient.Evaluate(1f);
21     }
22
23     1 référence | Changed by samy.yacef@epita.fr on jeudi 21 avril 2022
24     public void SetHealth(int health)
25     {
26         slider.value = health;
27
28         fill.color = gradient.Evaluate(slider.normalizedValue);
29     }

```

Ainsi, chaque barre de vie est composée de deux variables, qui permet, à l'initialisation, de définir la vie maximum, différente en fonction du protagoniste choisi, et la vie actuelle.

### b/: Multijoueur :

Une fois que nous avions pris connaissance de nos outils et répartit le travail selon les forces et faiblesses de chacun, nous avons pu commencer à réellement nous intéresser au cœur du projet, qui était donc la programmation.

Pour commencer, nous avons implémenté les déplacements des personnages car c'était le plus important. En effet, même si le jeu était terminé sans les déplacements, il est impossible d'y jouer. En plus des déplacements, nous avons dû prendre en compte l'aspect multijoueur du projet.

Après de nombreuses réflexions et de nombreux débats sur les différentes possibilités à notre disposition, nous avons finalement décidé d'utiliser Mirror pour le domaine du multijoueur. Ainsi notre concept pouvait être réalisé facilement car nous souhaitions un jeu de coopération et la contrainte de réseau nous paraissait comme un avantage.

Le mode multijoueur sera un mode à quatre joueurs ou moins, de tel sorte que chaque joueur contrôle un protagoniste. Si le nombre de joueurs est strictement inférieur à quatre, les joueurs pourront changer de protagoniste si celui voulu n'est

pas déjà contrôlé par un autre joueur. Cela permettra à chacun de jouer le rôle qu'il souhaite. Au début de l'aventure chaque joueur doit choisir son personnage, et lorsque le jeu commence, les changements de protagoniste deviennent possibles, si la situation et le nombre de joueurs le permettent.

Cependant, ce jeu serait trop simple si la difficulté restait la même pour un joueur et pour quatre. Nous pensons donc qu'il serait bon d'augmenter les caractéristiques des monstres en fonction du nombre de joueurs présents dans la partie.

Après nous être mis d'accord sur ce que devait être notre jeu, nous avons commencé à travailler sur ce domaine. En effet, nous avons conclu que Zorestone Realm devait être un jeu où les joueurs devront coopérer. Le réseau local nous a donc paru être une bonne idée, tout comme une bonne référence à Diablo. De plus, si nous le souhaitons, nous pourrons toujours implémenter un serveur dédié pour héberger les parties dans le futur.

Nous nous sommes alors heurtés à un problème de taille, qui était le contrôle du personnage. En effet, lorsque nous lancions le jeu après avoir implémenté Mirror en suivant la documentation fournie avec le module, le "host" du serveur contrôlait les personnages de tous les joueurs.

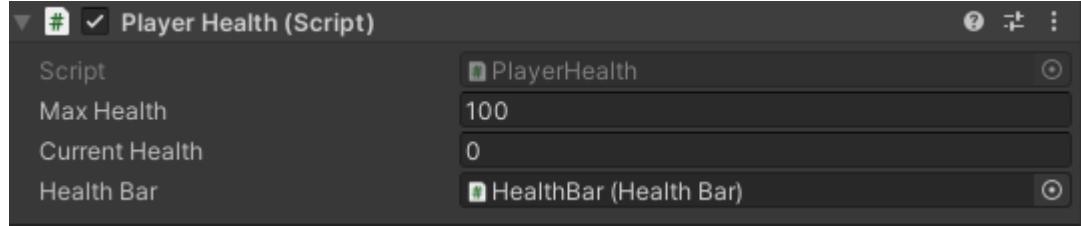
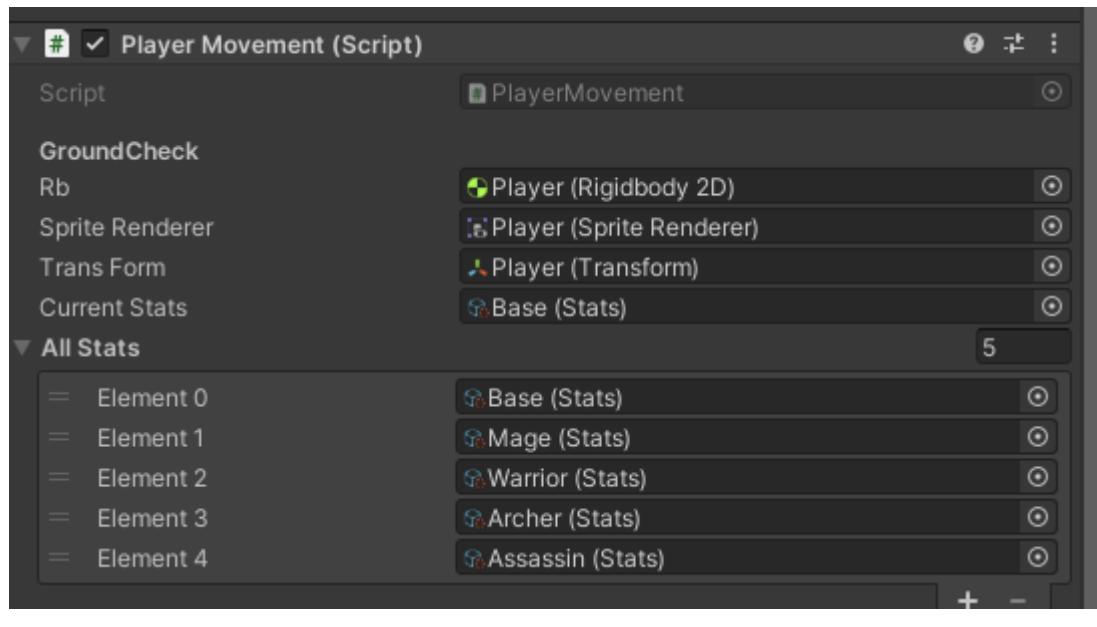
Il était donc ici nécessaire de rajouter un script de PlayerSetUp, qui héritait de la classe NetworkBehaviour, classe intégrée à Mirror. Ce script permettait de désactiver les composants que le joueur en question ne devait pas contrôler.

```

1  using UnityEngine;
2  using Mirror;
3
4  public class PlayerSetup : NetworkBehaviour
5  {
6
7      public Behaviour[] componentsToDisable;
8      private void Start()
9      {
10         if(!isLocalPlayer)
11         {
12             for (int i = 0; i < componentsToDisable.Length; i++)
13             {
14                 componentsToDisable[i].enabled = false;
15             }
16         }
17     }
18 }
19

```

Les composants à désactiver pour l'instant étant les suivants:



Lors des dialogues, nous avons décidé que cela ne mettrait pas le jeu en pause, pour ne pas forcer les joueurs à supporter la vitesse de lecture des autres joueurs. De plus, si un joueur, pendant un combat, essaie de se sauver en ouvrant le menu comme dans Skyrim, il aura le plaisir de voir le combat continuer sans lui, tout en continuant à subir des dégâts comme dans Genshin Impact.

Afin de permettre au joueur de jouer ensemble, nous avons pensé, après avoir synchronisé les déplacements des joueurs, qu'il était plus agréable que chaque joueur dépende de son point de vue. Ainsi, nous avons implémenté un système permettant que chaque joueur ait sa propre caméra suivant son personnage et pas un autre.

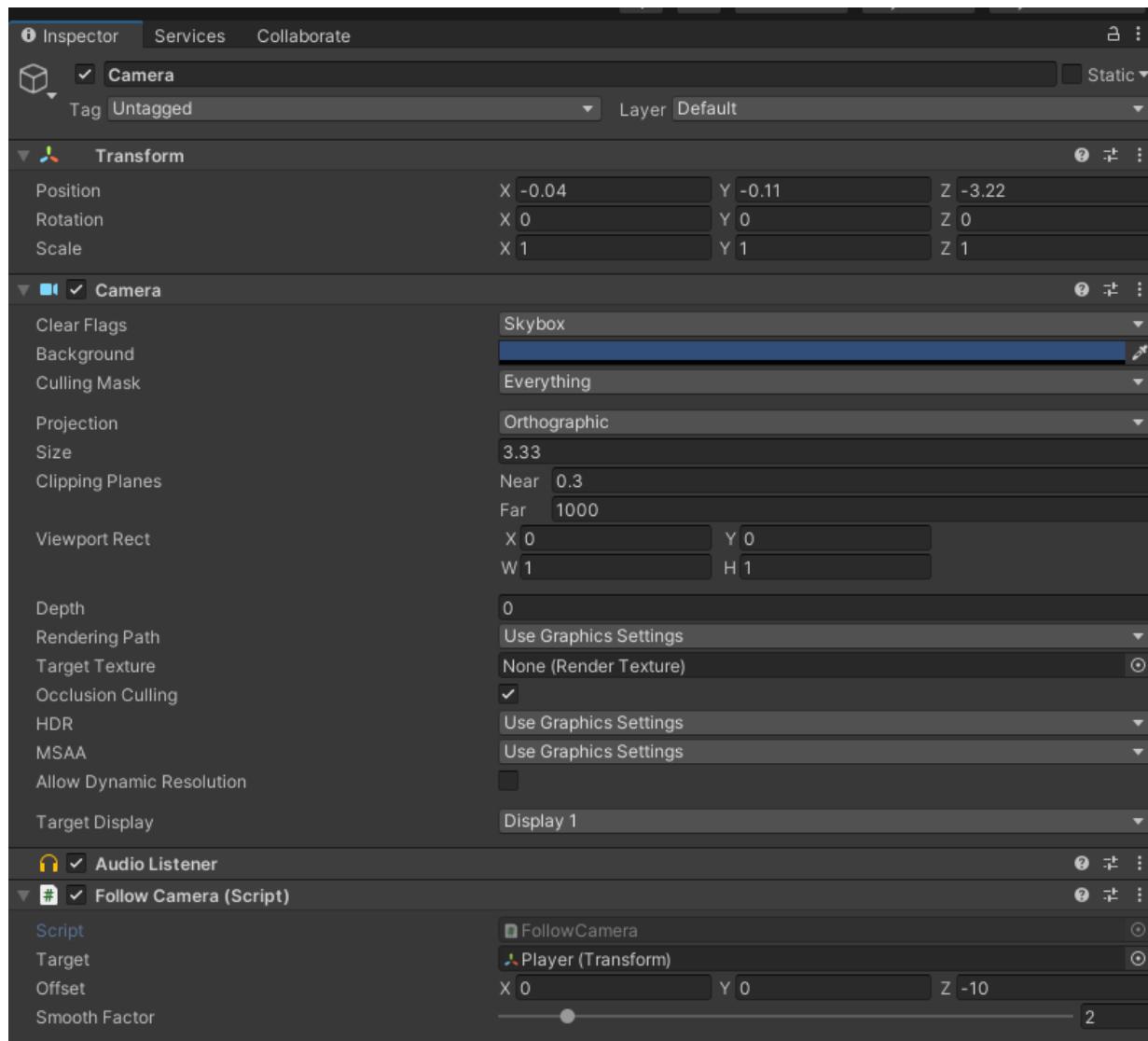
```
public class FollowCamera : MonoBehaviour
{
    public Transform target;
    public Vector3 offset;
    [Range(1,10)]
    public float smoothFactor;

    © Message Unity | 0 références | Changed by samy.yacef@epita.fr on mercredi 20 avril 2022
    private void FixedUpdate()
    {
        Follow();
    }
    1 référence | Changed by samy.yacef@epita.fr on mercredi 20 avril 2022
    void Follow()
    {
        // Definir des valuers Maximums pour x,y,z
        Vector3 targetPosition= target.position + offset;

        Vector3 smoothPosition = Vector3.Lerp(transform.position, targetPosition, smoothFactor * Time.fixedDeltaTime);
        transform.position = smoothPosition;
    }
}
```

Dès qu'un joueur initialise son personnage, lorsqu'il est téléporté dans l'aventure, une caméra est attachée à son personnage, ce qui lui permet ainsi d'avoir son point de vue centré sur son personnage, et ainsi lui permettre de mieux prendre en compte l'univers dans lequel il évolue.

Ici, l'Inspector de Unity, qui nous a permis de créer des "object" "camera" afin de pouvoir les attacher au "player". De fait, grâce à cela la caméra suit sans problème et de manière fluide le personnage contrôlé par le joueur, permettant de simplifier la prise en main du jeu et de l'immersion.



### c/: Intelligence artificielle :

L'IA sera représentée par les monstres que le ou les joueurs rencontreront au cours de leur aventure. Il y aura plusieurs types de monstres, de tailles différentes, avec des caractéristiques de combat variant entre chaque. Il y aura aussi des personnages non-joueurs avec lesquels le joueur pourra interagir. A la fin de certains niveaux, des monstres, plus puissants que ceux précédemment rencontrés, affrontent le joueur, ce qui permet de conserver l'attention du joueur tout au long de l'histoire.

S'il est possible et si le temps le permet, les monstres pourront se déplacer dans les niveaux où ils seront placés. Leur rôle sera de ralentir la progression du joueur, et de les attaquer dès qu'ils seront à portée. Les montres de haut niveau

(Boss de niveau) effectueront des enchaînements d'attaques qui dépendront de leurs points de vie, des points de vie du joueur et de la position de celui-ci par rapport aux boss.

Les personnages non-joueurs seront là pour permettre de donner des indications aux joueurs, ou simplement développer l'histoire. Ainsi, ils ne seront pas utiles lors des combats, mais éviteront la désorientation du joueur en lui indiquant où se situe l'objectif à atteindre.

Pour ce qui concerne l'IA, nous avons pu implémenter le déplacement de base des ennemis, qui sont les mêmes que les déplacements des personnages non joueurs.

Le jeu étant aussi basé sur le combat avant d'être simplement une histoire interactive, il était nécessaire d'implémenter l'élément le plus important du gameplay, celui qui mettra le plus en difficultés les joueurs : Les ennemis.

Dans le premier rapport de soutenance, nous vous avions présenté les ennemis classiques des joueurs : les gobelins.



un goblin, ennemi basique.

Nous avons pu ainsi les implémenter dans notre univers afin que le joueur puisse les affronter. Ainsi, le joueur perd de la vie si jamais il venait à heurter cet ennemi.



Comme on peut le voir, ici le joueur incarne le mage, et se retrouve en face à face avec un gobelin, celui-ci se déplaçant entre deux points préalablement placés sur le framework.

De plus, le projectile tiré par le mage, ici la balle rouge, est défini par le script suivant.

```

5  public class Bullet : MonoBehaviour
6  {
7      public float speed = 16f;
8      public Rigidbody2D rb;
9      public SpriteRenderer spriteRenderer;
10     public int damage;
11 
12     void Start()
13     {
14         rb.velocity = transform.right * speed;
15     }
16 
17     void OnTriggerEnter2D(Collider2D hited)
18     {
19         EnemyHealth enemy = hited.GetComponent<EnemyHealth>();
20         if (enemy != null)
21             enemy.TakeDamage(damage);
22         Destroy(gameObject);
23     }

```

Celui-ci se comporte grâce au script ci-dessous (1), tandis qu'il est instancié par le script venant ensuite (2).

(1)

```

5  public class ProjectilBehaviour : MonoBehaviour
6  {
7      public Transform firePoint;
8      public GameObject Bullet;
9      public Stats[] allStats;
10 
11     void Update()
12     {
13         if(Input.GetButtonDown("Fire1"))
14             Shoot();
15     }
16 
17     void Shoot()
18     {
19         int r = PlayerPrefs.GetInt("CharacterIndex2");
20         Bullet.GetComponent<Bullet>().spriteRenderer.sprite = allStats[r].Bullet;
21         Bullet.GetComponent<Bullet>().damage = allStats[r].attack;
22         if(allStats[r].longRangeAttck)
23             Instantiate(Bullet, firePoint.position, firePoint.rotation);
24     }

```

(2)

```
6  public class EnemyHealth : MonoBehaviour
7  {
8      public int maxHealth;
9      public int currentHealth;
10     public HealthBar healthBar;
11     public GameObject ObjectToDestroy;
12
13
14     private void Start()
15     {
16         currentHealth = maxHealth;
17         healthBar.SetMaxHealth(maxHealth);
18     }
19
20     private void OnTriggerEnter2D(Collider2D collision)
21     {
22         if (collision.CompareTag("Projectile"))
23         {
24             TakeDamage(20);
25         }
26     }
27
28     private void Update()
29     {
30         if (Input.GetKeyDown(KeyCode.P))
31             TakeDamage(20);
32     }
33
34     public void TakeDamage(int damage)
35     {
36         if(currentHealth<damage)
37         {
38             currentHealth -= damage;
39             Destroy(ObjectToDestroy);
40         }
41     }
42 }
```

De plus, il y a d'autres ennemis implémentés dans le jeu, comme par exemple l'ogre ou encore le boss de fin. Cela permet une certaine diversité dans l'aventure.



#### d/: Site web :

Dans un monde ultra connecté, un projet, et ce quelque soit son envergure, doit être représenté sur internet et sur les réseaux sociaux. Nous avons décidé pour notre jeu qu'il sera accompagné d'un site internet qui nous permettra de le promouvoir ainsi que de créer un lien avec les joueurs à travers des nouvelles régulières de l'avancement du projet, ainsi qu'une présentation individuelle des membres du groupe.

Il sera aussi possible de télécharger la dernière version du jeu sur ce même site et d'accéder à un serveur discord qui sera créé pour, une fois encore, garder un lien avec la communauté et répondre aux éventuelles questions des joueurs.

Le site étant l'élément de communication principal pour les nouveaux joueurs, nous avons décidé de commencer à l'implémenter sans pour autant le prioriser car cela nous paraissait aberrant de présenter notre projet si celui-ci n'était pas à un stade que nous pouvions qualifier d'avancé, voire même jouable.

A présent que nous avons un projet jouable et présentable, nous nous sommes penchés plus en détails sur la manière dont nous pouvions capter l'attention de potentiels joueurs. Pour plus de confort, il a été décidé de limiter un maximum la place des différents menus. Ainsi le joueur découvrant le site pourra avoir toutes les informations qu'il souhaite, tandis que le joueur le connaissant pourra immédiatement se rendre sur la page du téléchargement.

#### e/: Effets sonores :

Pour une meilleure immersion dans le jeu, nous voulons adapter la musique à chaque situation dans laquelle le joueur pourra se trouver. Ainsi une musique calme de paix sera utilisée pour l'ambiance d'un village tandis qu'une musique frénétique ou épique sera utilisée pour combattre un boss. La présence de son pour chaque action est aussi importante pour nous, car sans cela le joueur se retrouve séparé du jeu et l'immersion est brisée.

Pour ce faire, nous pensons utiliser FL Studio et Audacity en fonction des compétences de chacun. Ces logiciels permettant un contrôle simple sur l'entièreté du processus de création de musique, il sera utile de pouvoir avoir accès à de

nombreuses fonctions intégrées et à de nombreux instruments gratuitement afin d'obtenir le son voulu.

Comme expliqué dans le cahier des charges, nous souhaitons que l'ambiance sonore se prête au jeu. Cependant, avant que les joueurs puissent s'immerger dans le jeu, il faut que le jeu soit jouable. C'est pourquoi nous avons jugé que la création de l'ambiance musicale et sonore n'était pas la priorité lors de cette première soutenance.

Pour que les joueurs aient vraiment l'impression d'évoluer dans le même univers, malgré les changements de niveaux, nous avons pensé à un thème musical récurrent, que nous pourrons modifier, en le rendant plus calme ou plus épique en fonction de la situation dans laquelle est plongé le joueur.

Ce concept a été utilisé dans Star Wars, avec le thème de la force qui se décline en plusieurs versions, chacune d'entre elles suscitant une émotion particulière chez le spectateur.

### III/: Expériences personnelles :

#### A/: Camille Michon :

Étant passionnée par les nouveautés technologiques, l'ensemble des informations que j'ai pu collecter concernant l'EPITA et sa notoriété excellente m'ont permis de renforcer ma motivation pour y rentrer.

Ma personnalité est à la fois cartésienne, analytique comme synthétique, mais également sensible aux arts. C'est pourquoi, étant une passionnée de jeux vidéos, je me suis toujours interrogée sur le travail accompli derrière un jeu: les graphismes, les scénarios, la musique.

Lorsque j'ai pris connaissance du projet du second semestre; l'idée de la conception d'un jeu vidéo était, pour moi, la meilleure idée. Pouvoir toucher à tous les secteurs tout en restant dans un même domaine me permettrait d'acquérir de nouvelles compétences tout en appréciant mon travail.

Ce projet S2 m'est extrêmement gratifiant. J'ai eu la chance de voir ce projet se concrétiser . Nous avions aussi élaboré une grande organisation . Nous travaillons toujours par groupe de 3, nous nous entraidons, nous nous informons, en résumé, nous sommes une vraie équipe. Nous avons toujours quelque chose à faire, les tâches ne sont pas forcément distribuées par rapport à nos compétences mais dans l'optique de nous faire évoluer.

Pour bien disperser le travail, Pierre, notre chef de projet, a décidé de nous diviser en plusieurs pôles. Nous aurons tous un rôle dans chacun d'eux. Ayant une âme légèrement artistique, j'ai été promue cheffe du pôle graphisme, et travaille dans l'équipe du multijoueur tenue par Samy , mais aussi de l'IA et du site web tenue par Arthur. Nous avons tous un rôle important à jouer, mais nous pouvons aussi bien nous adapter au cas où l'un de nous viendrait à quitter le groupe.

Avoir une véritable ambiance de travail permet d'avancer et d'améliorer ses capacités (que cela soit en programmation ou même dans le vivre ensemble). En effet, pour débuter nous avons tous commencé par des vidéos explicatives, qui nous montraient comment marche Unity,et comment régler des problèmes liés à

cette plateforme. Il faut savoir que pour Unity, je n'avais en bagage que les cours de NTS et peu d'entraînement à mon actif. J'avais donc peur de ne pas pouvoir aider mon groupe comme il le fallait ou de simplement réussir.

Mais la persévérance de l'équipe m'a aussi permis d'être plus patiente, plus engagée dans ce que je fais, à mieux gérer mes émotions, ne plus abandonner mais d'essayer de comprendre comment et pourquoi le programme est dysfonctionnel . Nous avons aussi eu la chance de découvrir des logiciels tels que Fresco, Photoshop, Unity , d'apprendre à les manier, ce qui pourrait me permettre de me démarquer lors d'un entretien par exemple.

Je vis donc dans une ambiance où travailler n'est pas une contrainte. Nous essayons de dépasser nos limites. Nous nous voyons grandir , mais nous voyons aussi grandir notre jeu.

### B/: Pierre Fouché :

J'ai commencé à m'intéresser à l'informatique lorsque j'ai compris que derrière n'importe quel objet électronique se trouvait tout un système complexe. C'est pour comprendre le fonctionnement de ceux-ci que j'ai rejoint l'EPITA. J'ai été extrêmement heureux d'être accepté.

De plus, j'ai toujours été passionné de logique, c'est pourquoi l'algorithme m'a tout de suite attiré. Avec un cursus scientifique et une initiation à l'informatique dans mon lycée, j'ai pu implémenter quelques programmes en physique et en maths. De manière personnelle, j'ai créé quelques jeux mais à travers ce projet, j'espère pouvoir enfin réaliser quelque chose dont je puisse être fier.

Le fait que ce projet soit libre nous permet d'imposer notre vision d'un bon jeu et c'est pour moi une motivation importante et une source d'inspiration constante. J'attends de ce projet un enrichissement et une affirmation dans mon projet professionnel après l'EPITA, mais aussi une expérience de travail en équipe qui sera valorisante pour tous dans notre groupe.

Ce projet S2 m'est extrêmement gratifiant. Je vois de plus en plus ce projet se concrétiser . Notre bonne organisation permet aussi d'évacuer un stress. Nous travaillons toujours par groupe de 3 minimum, nous nous entraînons, nous nous informons, en résumé, nous sommes une vraie équipe. Nous avons toujours

quelque chose à faire, les tâches ne sont pas forcément distribuées par rapport à nos compétences mais dans l'optique de nous faire évoluer.

Pour bien disperser le travail, Pierre, notre chef de projet, a décidé de nous diviser en plusieurs pôles. Nous aurons tous un rôle dans chacun d'eux. Ayant une âme légèrement artistique, j'ai été promue cheffe du pôle graphisme, et travaille dans l'équipe du multijoueur tenue par Samy , mais aussi de l'IA et du site web tenue par Arthur. Nous avons tous un rôle important à jouer, mais nous pouvons aussi bien nous adapter au cas où l'un de nous viendrait à quitter le groupe.

Avoir une véritable ambiance de travail permet d'avancer et d'améliorer ses capacités (que cela soit en programmation ou même dans le vivre ensemble). En effet, pour débuter nous avons tous commencé par des vidéos explicatives, qui nous montraient comment marche Unity,et comment régler des problèmes liés à cette plateforme. Il faut savoir que pour Unity,je n'avais en bagage que les cours de NTS et peu d'entraînement à mon actif. J'avais donc peur de ne pas pouvoir aider mon groupe comme il le fallait ou de simplement réussir.

Mais la persévérance de l'équipe m'a aussi permis d'être plus patiente, plus engagée dans ce que je fais, à mieux gérer mes émotions, ne plus abandonner mais d'essayer de comprendre comment et pourquoi le programme est dysfonctionnel . Nous avons aussi eu la chance de découvrir des logiciels tels que Fresco, Photoshop,Unity , d'apprendre à les manier, ce qui pourrait me permettre de me démarquer lors d'un entretien par exemple.

Je vis donc dans une ambiance où travailler n'est pas une contrainte. Nous essayons de dépasser nos limites. Nous nous voyons grandir , mais nous voyons aussi grandir notre jeu.

### C/: Arthur Le Flem :

Au cours de mon parcours scolaire j'ai pu développer des premières compétences informatiques : langages (python, c#, html, css), interfaces utilisateur ou encore technologie web. Et je n'ai qu'une grande motivation, travailler sur ces sujets. C'est pourquoi le choix d'EPITA était logique.

Ce n'est pas la première fois que je réalise un tel projet de groupe, puisqu'en terminale mon groupe et moi avons mis au point un puissance 4 dans le cadre de la

spécialité numérique et science informatique en classe de première. Nous étions 3 et nous avons développé le jeu en python.

Cela m'a permis de perfectionner mes compétences en python, interface utilisateur et expérience utilisateur puisque le jeu est fait sur une fenêtre à part, ainsi qu'en intelligence artificielle car nous avons doté le jeu d'un robot que l'on peut défier. Le jeu se joue à deux joueurs ou alors à un joueur contre le robot.

Ce projet me permettra de renforcer mes compétences ou d'en acquérir de nouvelles. De plus, je peux apporter mon savoir faire à mes camarades.

C'est la première fois que je crée un jeu vidéo aussi complet. J'appréhendais la répartition des tâches pour ce genre de projet mais finalement nous nous sommes bien organisés et j'ai pu apprendre beaucoup de choses grâce à ce projet. J'ai pu compter sur la disponibilité de mes camarades pour leur poser des questions lorsque j'utilisais le travail de l'un d'eux. Par exemple, j'ai créé la barre de vie du joueur et pour que cela s'organise parfaitement avec le multijoueur j'ai pu poser mes questions à Samy pour que je puisse la paramétriser correctement.

Dès le début nous avons réparti le travail en cinq grandes catégories, pour ma part je me suis chargé de l'intelligence artificielle du jeu ainsi que de la partie site web du projet. En ce qui concerne la partie IA je me suis occupé de créer des ennemis qui pour l'instant se déplacent sur une zone prédéfinie. D'autre part pour la partie site web, j'ai fait de nombreuses recherches pour commencer à implémenter un site qui permette de présenter notre jeu et notre équipe puis également pour que la version finale du jeu soit téléchargeable depuis celui-ci.

Chacun a travaillé en collaboration sur la catégorie prévue d'un camarade pour l'aider dans son travail mais également pour que chacun découvre presque toutes les tâches associées à la création de notre projet. Ainsi j'ai pu laisser mon âme d'artiste parler pour créer des tiles en pixel art. Mais j'ai également corrigé des bugs notamment pour le saut du personnage.

#### D/: Samy Yacef :

Étant étudiant étranger, j'ai rejoint l'EPITA, après un cursus scolaire dans un établissement français à l'étranger. J'ai pu rencontrer de nombreuses personnes d'origine diverses et variées, et cela m'a permis d'évoluer et d'en apprendre plus sur moi-même.

Après cette introspection, j'ai commencé une recherche concernant ma carrière future et après le stage de 3ème dans une entreprise de matériel informatique concurrente à celle de ma mère, j'ai pu découvrir le métier de technicien informatique dans leur service après vente. J'ai ressenti une certaine attirance pour le domaine de l'IT.

De mère ingénierie électronique et cheffe d'entreprise en Informatique, j'ai été, et ce depuis le plus jeune âge, baigné dans ce domaine. J'ai, d'ailleurs, toujours été un fan de Minecraft, jeu dans lequel, le joueur a une liberté totale de création grâce au système de redstone qui permet d'implémenter une version très basique de circuit électrique. J'espère ainsi à travers ce projet, évoluer encore une fois, et déterminer une possible attirance pour le domaine des jeux vidéo.

J'attendais avec impatience le début du projet de S2 car je savais que cela allait être notre premier projet concret dont on pourrait voir les avancements en temps réel. La plupart des tâches à réaliser, notamment les TPs. que nous avons fait ont été assez théoriques et centrés sur la maîtrise d'une ou plusieurs composantes de la programmation informatique tel que le système de classe en POO.

De plus, ce projet m'a permis de mettre en pratique toutes les connaissances que j'ai acquises au cours de cette première année universitaire, mettant en lumières les progrès que j'ai fait et ceux malgré certaines difficultés au début.

C'est pour cette raison que je me suis porté volontaire en tant que chef du pôle Réseau, domaine dont je ne connaissais rien et que j'étais très impatient de découvrir. J'ai eu certes de nombreuses difficultés au départ mais des recherches approfondies quant à l'asset Mirror, asset qui permet l'implémentation d'un multijoueur sur unity m'ont permis de venir à bout de la plupart des problèmes rencontrés.

De plus, l'ambiance conviviale et chaleureuse au sein de notre groupe m'a poussé à donner le meilleur de mes capacités dans ce projet. Un esprit d'entraide était aussi très présent, cela nous a permis d'avancer ensemble et d'échanger sur les problèmes que chacun rencontre et réfléchir ensemble à une solution si nécessaire. Ce travail d'équipe est d'autant plus nécessaire au niveau du multijoueur où le fonctionnement des objets est très différent de celui d'un projet solitaire.

#### IV/: Conclusion :

Le but de ce projet est avant tout de se familiariser avec la Programmation Orientée Objet avec le C#, mais aussi de découvrir les différents avantages et inconvénients d'un travail de groupe. Ainsi, à titre individuel, nous souhaitons pouvoir développer nos capacités, acquérir des compétences et être satisfait de ce que nous faisons, tandis qu'à titre collectif, nous cherchons à développer notre capacité à travailler en équipe, et nos compétences orales de présentation et de mise en valeur d'un projet qui nous tient à cœur.

