



Haute Ecole de la Province de Liège



DÉPARTEMENT SCIENCES SOCIALES ET COMMUNICATION

Avenue Montesquieu, 6 - 4101 Seraing (Jemeppe sur Meuse)

Multiverse Project, jeu vidéo en 3D à la première personne

Quentin Delvaux

Travail de fin d'études présenté en vue de l'obtention du grade de
Bachelier en écriture multimédia

Année académique : 2019 - 2020

Siège social :

Avenue Montesquieu, 6
4101 Jemeppe (Seraing)
Belgique

www.hepl.be

Remerciements

Je tiens avant tout à remercier M. François Stasiuk, superviseur et premier testeur de mon TFE pour ses précieux conseils autant pour la partie écrite que pratique. Je tiens également à le remercier pour m'avoir fait découvrir les différentes facettes de la création 3D.

Merci aux quelques testeurs qui ont pris le temps d'essayer mon jeu jusqu'au bout, de m'avoir soumis leurs avis et suggestions.

Je tiens également à remercier d'une manière générale toutes les personnes qui, d'une façon ou d'une autre, ont permis à ce projet d'évoluer dans la bonne voie.

Enfin, j'aimerais remercier mes parents pour m'avoir encouragé à aller jusqu'au bout de ce projet et pour avoir relu et supervisé avec attention l'écrit de ce TFE.

Table des matières

1) Introduction.....	1
2) Aspect du jeu.....	2
a. Pourquoi un jeu en 3D à la première personne?.....	2
b. Exemples d'autres jeux.....	2
c. Le public cible.....	3
d. La création d'un objet 3D.....	3
e. Utilisation du Low Poly.....	5
f. Travail sur l'optimisation.....	6
g. Configuration minimale.....	7
3) Narration.....	8
a. Résumé du scénario.....	8
b. Multiples chapitres.....	8
c. Narration par l'exploration.....	8
4) L'immersion.....	9
a. L'ambiance visuelle.....	9
b. Choix des couleurs.....	10
c. L'ambiance sonore.....	10
5) L'animation.....	11

6) La programmation.....	12
a. Dans le cadre d'une animation.....	12
b. Dans le cadre de dialogues.....	13
c. Dans le cadre d'exploration.....	15
d. Dans un menu de pause.....	16
7) Contenu annexe du jeu.....	17
a. Succès et easter eggs.....	17
b. Indications cachées.....	18
c. Détails minimales.....	18
8) Fins alternatives.....	20
9) Tests utilisateurs.....	21
10) Problèmes rencontrés.....	22
11) Conclusion.....	27

1) Introduction

Dans le but d'achever cette dernière année de bachelier en Écriture Multimédia, il m'a été demandé de réaliser un travail de fin d'études interactif en rapport avec les cours qui m'ont été présentés. Étant quelqu'un de très créatif et de passionné par la 3D, j'ai voulu réunir ces deux qualités. C'est la raison pour laquelle j'ai décidé de créer un jeu vidéo.

Ayant particulièrement aimé les cours sur les logiciels 3DS Max et Unity, j'ai utilisé ceux-ci pour réaliser la majeure partie de mon travail. Je voulais avant tout que l'entièreté du jeu ait été créée par moi-même. Bien qu'au début je comptais réaliser une simple démo, j'ai décidé d'aller plus loin: travailler sur mon projet une heure par jour en commençant au mois d'août. Je voulais que mon jeu soit le plus complet possible et qu'il soit la meilleure représentation des ces trois années d'études. C'est pourquoi il comporte au moins un menu, un début, une fin et un maximum de détails.

Afin d'énumérer les différentes étapes, les choix et les sources d'inspirations dont je me suis servi, j'ai décidé de les fractionner en plusieurs phases. La première est avant tout le choix d'un jeu en 3D à la première personne sur un jeu en 2D.

2) Aspect du jeu

a. Pourquoi un jeu en 3D à la première personne?

Avant tout, un jeu à la première personne est un jeu où le joueur incarne le personnage qu'il contrôle en voyant le décor à travers les yeux de son avatar.

J'ai choisi ce mode de jeu pour la raison suivante: l'immersion. Dans un jeu où l'exploration est primordiale, c'était pour moi une évidence. Je voulais avant tout que le joueur puisse se déplacer un peu partout dans le jeu et qu'il puisse découvrir par lui-même des petits bouts d'histoire dissimulés.

b. Exemples d'autres jeux

En ce qui concerne l'ambiance pour la création de ce jeu, je me suis beaucoup inspiré de «Portal 2».



«Portal 2» (2011)

«Portal 2» est un jeu vidéo de plates-formes et de réflexion en vue subjective développé et édité par Valve. C'est un jeu où l'immersion est fantastique. C'est cette ambiance sombre et isolée que j'ai souhaité incorporer dans le mien.

En ce qui concerne l'ambiance que l'on retrouve à l'intérieur d'un bâtiment, je me suis inspiré de celle présente dans le jeu «The Stanley Parable». C'est un jeu d'exploration en vue à la première personne, sans aucun gameplay autre que le déplacement. C'est une atmosphère où l'on se sent isolé de tout et où le but est de trouver la sortie.



«The Stanley Parable» (2013)

c. Le public cible

«Multiverse Project» est un jeu qui ne contient aucune violence. Il a été conçu et adapté de sorte à ce qu'il ne soit pas trop compliqué de trouver son chemin. Ce jeu s'adresse à toute personne friande de découvertes, d'exploration et de surprises. Par conséquent, en raison de son aspect plutôt calme, il pourrait ne pas plaire à une personne souhaitant un jeu énergique et rempli d'action.

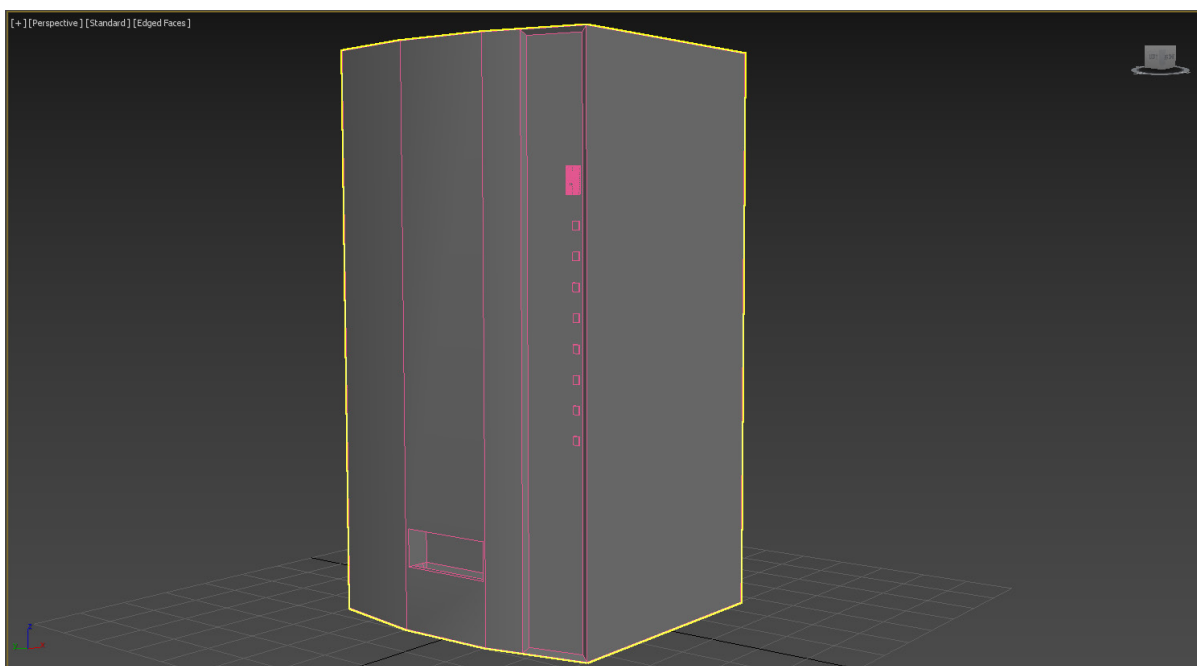
d. La création d'un objet 3D

La totalité des objets 3D ont été créés par mes soins. Quant aux textures, j'ai utilisé Adobe Photoshop pour la plupart d'entre elles.

Pour le distributeur de sodas, par exemple, j'ai créé l'objet en 3D, créé la texture et appliqué un effet de lumière une fois sur Unity pour le rendre plus vivant. Pour plus de clarté, j'ai décidé d'énumérer les différentes étapes types de la création d'un objet 3D dans mon jeu.

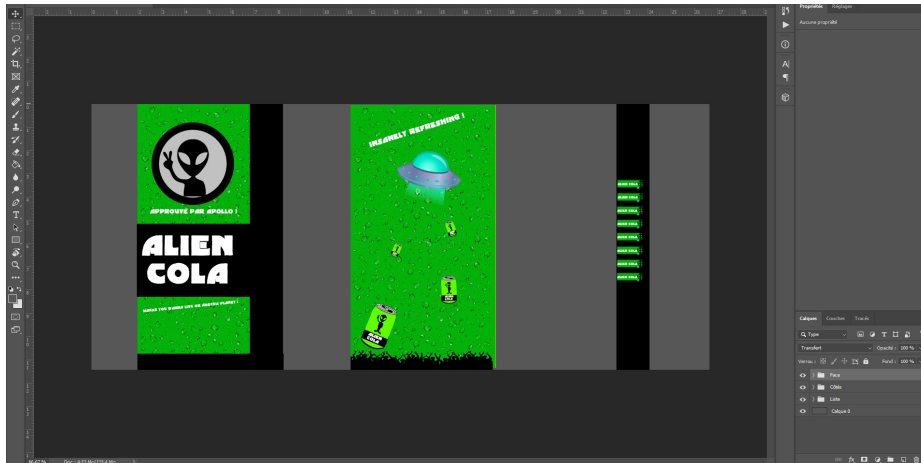
Étape 1: La modélisation sur 3DS Max

La totalité de mes objets 3D ont été modélisés sur 3DS Max. Dans le cas d'un distributeur, je suis simplement parti d'un rectangle. Je me suis servi des segments de l'objet afin de le déformer. L'avantage de travailler avec deux écrans est que j'ai pu modéliser sur un et m'inspirer d'une image sur l'autre. C'est de cette façon que la plupart des objets ont été créés.



Étape 2: Le travail des textures sur Photoshop

Pour certains objets 3D, j'ai créé la texture moi-même. Dans le cas du distributeur, il reflète une partie de l'histoire du jeu. Il devait être entièrement personnalisé. Je me suis donc appuyé sur les dimensions de mon objet 3D et les ai ancrées dans Photoshop. De ce fait, la hauteur et la largeur de ma surface de travail étaient précises.



Étape 3: L'incorporation dans le jeu en fichier FBX

Une fois la texture créée, je l'ai importée sur 3DS max et appliquée à mon objet 3D. J'ai, par la suite, exporté mon modèle 3D en fichier FBX pour l'importation sur Unity.

Néanmoins, un petit problème m'empêchait d'y ajouter un détail qui me tenait à cœur: la texture n'émettait pas de lumière.

En effet, pour ajouter une touche de réalisme, mon distributeur devait émettre un peu de lumière dans un environnement très sombre. En cherchant sur la documentation de Unity, je suis parvenu à paramétrer une lueur sur la texture.

Finalement, j'ai décidé de lui ajouter plus de détails. Parmi ceux-ci, j'ai choisi de lui ajouter une petite animation qui consiste à faire tomber une canette lorsque l'on appuie sur «e» devant le distributeur ainsi qu'un petit son de grésillement électrique quand on s'en approche.



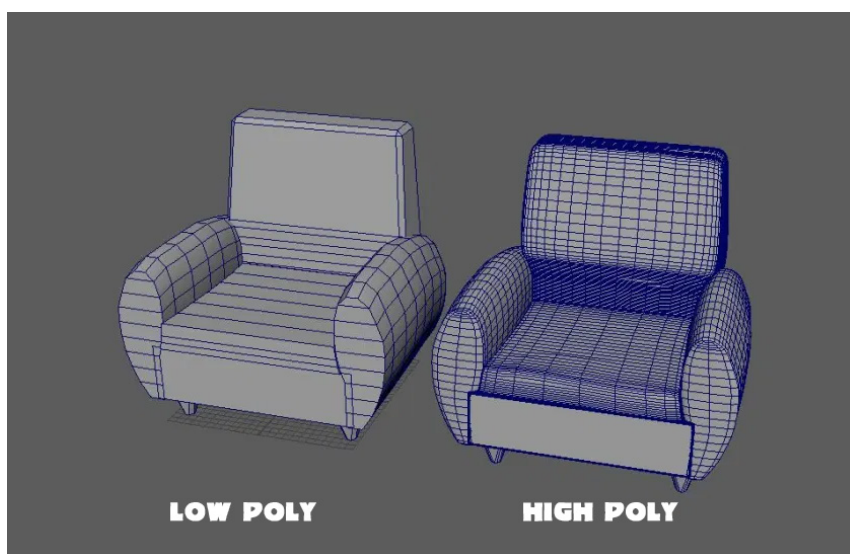
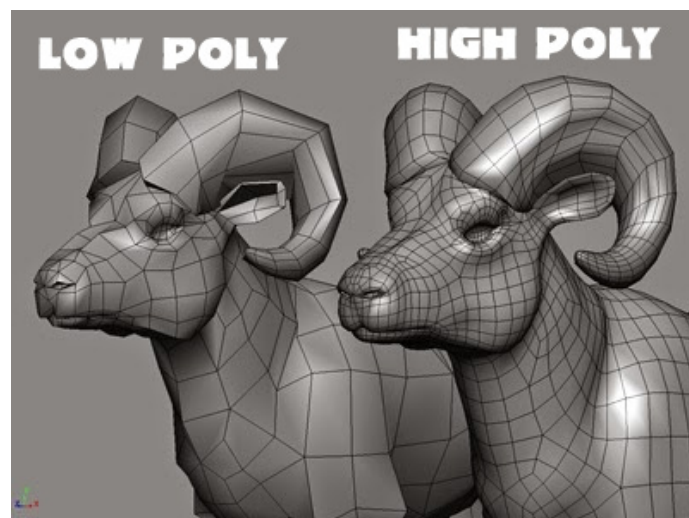
e. Utilisation du Low Poly

La technique du Low Poly (Low Polygone) est une technique de création d'objets 3D qui consiste à modéliser un objet en utilisant peu voire même très peu de polygones. J'ai décidé de créer mon jeu en utilisant cette technique pour plusieurs raisons.

La première raison est le gain de temps. En effet, un objet 3D créé en Low Poly contient beaucoup moins de détails et est donc plus rapide à créer. Sur un jeu composé de plus de 250 modèles 3D différents, il était évident qu'il me fallait utiliser cette technique.

La deuxième raison est l'optimisation. Un objet 3D qui contient moins de polygones sera, par définition, plus léger. Nous reviendrons sur l'optimisation au point suivant.

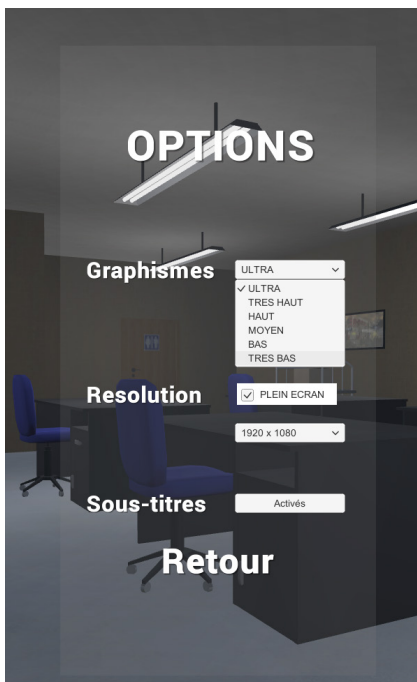
Voici quelques exemples entre un objet 3D en Low Poly et un objet 3D en high poly.



f. Travail sur l'optimisation

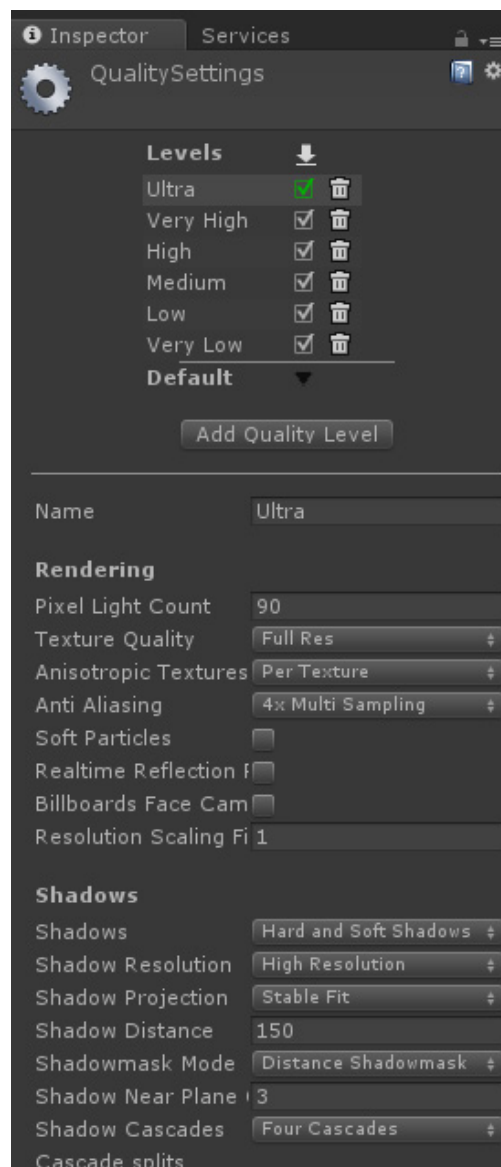
Ayant un décor plutôt riche contenant pas mal de lumières, le jeu pouvait perdre en fluidité à certains moments. Bien que je possède un ordinateur plutôt puissant, je me suis vite rendu à l'évidence que si j'avais des baisses d'images par secondes à certains moments, un joueur avec une configuration moins puissante les avaient probablement depuis le début. J'ai donc commencé à travailler sur l'optimisation de mon jeu par différents moyens.

1. Un menu en jeu permettant de baisser la qualité des graphismes.



Ce menu en jeu adapte automatiquement la qualité à la capacité graphique de l'ordinateur qu'on utilise afin d'assurer une bonne fluidité pendant l'histoire.

Il en va de même pour la résolution de l'écran qui est, elle aussi, paramétrable et qui de plus peut passer du plein écran au mode fenêtré.



J'ai défini les différents niveaux de qualité sur Unity et les ai paramétrés à l'aide de l'option «Quality Settings». Ce paramétrage comprend, pour chaque niveau de qualité, le nombre de pixels de lumières, la qualité des ombres, la qualité des textures et les reflets.

En ce qui concerne le temps d'attente entre le chargement de deux niveaux en particulier, il était de quelques secondes sur un écran noir. Suivant les conseils de mon superviseur, j'ai donc décidé, pendant ce temps d'attente (qui varie d'une personne à l'autre), d'y ajouter un écran de chargement. De ce fait, le joueur sait que le jeu charge le niveau suivant.

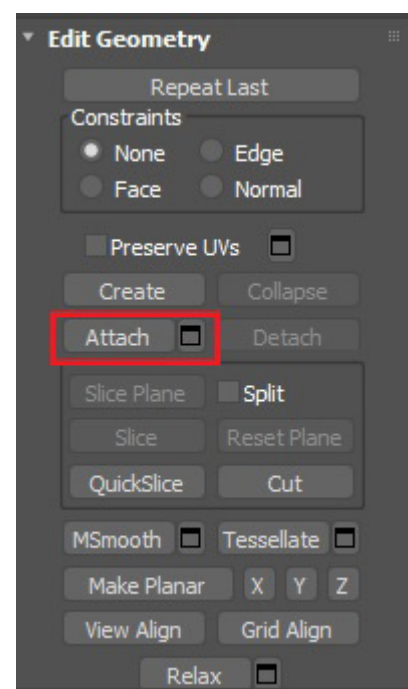
2. Un travail sur mes objets 3D

Afin d'optimiser un maximum mes créations en 3D, je me suis servi du paramètre «Attach» de 3DS Max. Ce paramètre permet, comme son nom l'indique, d'attacher plusieurs objets à la fois et de les fusionner. Cette option m'a été très utile pour optimiser la salle d'archives de mon jeu.

En effet, une étagère moyenne comprend environ une quarantaine de cartons. Si chacun de ces cartons était considéré comme un objet à part entière, le fichier serait beaucoup plus lourd et le moteur du jeu devrait charger plus de détails à la fois.



Chaque étagère n'est en fait qu'un seul objet 3D



g. Configuration minimale

Dû au fait que le joueur peut régler lui-même la qualité des graphismes et, de ce fait, avoir un jeu fluide même avec un ordinateur médiocre, il est compliqué d'établir une configuration minimale. Cependant, si une configuration minimale devait être énoncée, ce serait celle-ci:

CPU: Core i3 2.4 Ghz

RAM: 2 Go

OS: Windows 7/8/10 64-bit

VIDEO CARD: Intel HD 4000

ESPACE DISQUE: 1 Go

J'ai tout simplement énoncé la configuration minimale pour la plupart des «petits jeux» qui ne demandent pas de grande puissance. Il n'est pas exclu que le joueur puisse tout de même profiter du jeu avec une configuration en dessous de celle-ci. Il aura probablement moins d'images par secondes lors de son expérience vidéoludique.

3) Narration

a. Résumé du scénario

Nous sommes en l'an 2043 et Apollo, extraterrestre arrivé sur Terre il y a quelques années, est l'unique lien entre l'espèce humaine et une exoplanète habitable. Admis comme invité depuis son arrivée et traité comme héros pour ses pouvoirs de guérison, son utilité touche à sa fin. Une dernière chose à laquelle il peut être utile : exploiter ses pouvoirs à partir de ses cellules souches pour ses dons de guérison. Autrement dit, le tuer.

Étant l'être supposé unir deux espèces pour cohabiter, certains employés de la base où il est retenu ne le voient pas du même œil. Ils comptent bien l'aider à s'échapper et à faire aboutir le « Multiverse Project » qui consiste à envoyer des humains sur cette exoplanète et vice-versa.

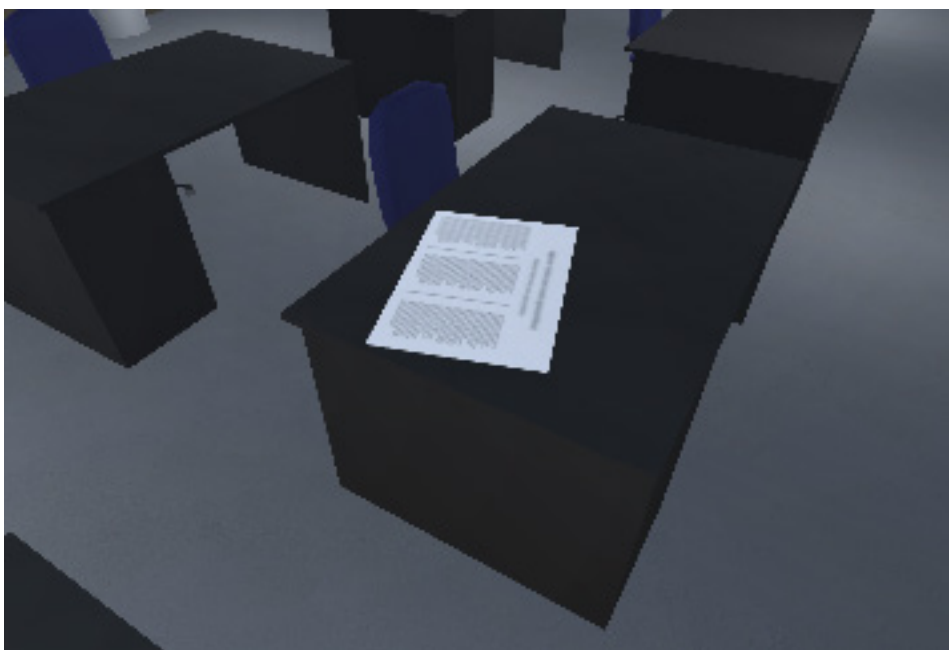
Le joueur incarne cet extraterrestre qui essaie de s'échapper.

b. Multiples chapitres

Afin d'ajouter une échelle à l'évolution de l'histoire dans le jeu, j'ai décidé d'incorporer des chapitres. Ces chapitres interviennent uniquement lors de moments clés du scénario et ne sont qu'une aide visuelle apportée au joueur.

c. Narration par l'exploration

Durant l'expérience de jeu, le joueur est soumis à de nombreux détails dans le monde qui l'entoure. Ces détails sont de plusieurs formes mais rejoignent tous le même but: raconter l'histoire de l'endroit dans lequel il se situe. Il retrouve, par exemple, des articles de journaux relatant des événements qui se sont déroulés ou encore des éléments de décors qui apportent un éclaircissement.



Exemple d'un article de journal lisible

4) L'immersion

a. L'ambiance visuelle

En ce qui concerne l'ambiance visuelle, je voulais reproduire un endroit déserté, sombre, fraîchement et précipitamment abandonné. Il devait se trouver dans un monde futuriste qui prône l'exploration spatiale et la découverte d'une nouvelle planète. C'est pourquoi tout un travail de réflexion en rapport avec l'ambiance visuelle a été réalisé. Afin d'énoncer l'ambiance dont je parle, j'ai décidé d'incorporer deux images sorties de mon jeu.



Ambiance sombre et déserte



Ambiance concentrée sur l'exploration spatiale

b. Choix des couleurs

Dans le but de troubler l'expérience du joueur, j'ai créé un grand changement d'environnement de jeu. En quelques minutes, il passe d'un monde très coloré à un monde très gris et sombre. Afin de mieux décrire ce changement, j'ai décidé de vous mettre deux palettes de couleurs: une du monde coloré et une du monde très sombre.

Palette du monde coloré



Palette du monde sombre



Nous pouvons donc constater que la transition est assez brutale. Celle-ci est tout à fait volontaire et apporte un côté étrange à l'endroit où se retrouve plongé le joueur.

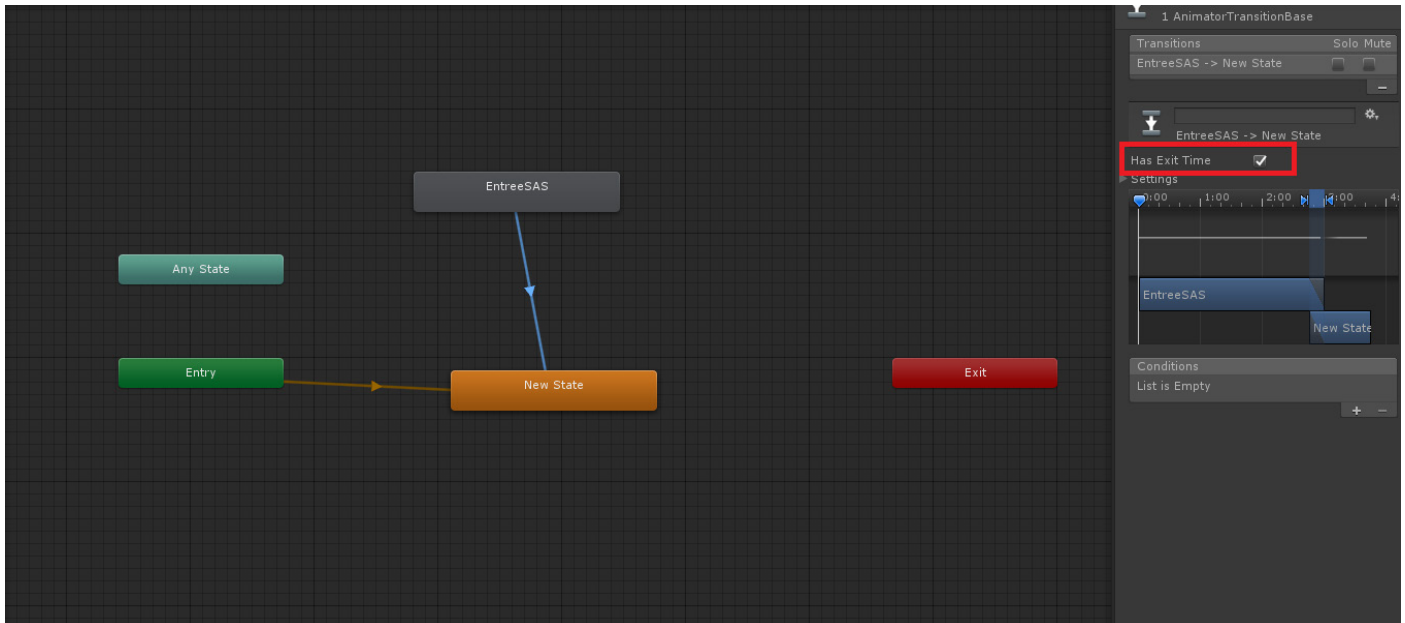
c. L'ambiance sonore

L'ambiance sonore dans le jeu est primordiale. C'est pourquoi j'ai pris la décision d'en ajouter certaines lors d'évènements spécifiques, de nouvelles découvertes ou d'actions diverses. L'ambiance sonore change constamment dans le jeu. C'est un détail qui, selon moi, ne doit pas être négligé.

Dans le cas d'un jeu vidéo qui serait destiné à la vente, je devrai, bien entendu, avoir obtenu les autorisations relatives à l'utilisation de musiques soumises aux droits d'auteurs.

5) L'animation

Une étape très importante dans la création d'un jeu vidéo est l'animation. Chaque objet avec lequel on peut interagir contient une animation scriptée. Les plus présentes sont celles associées aux portes. Prenons le cas d'une porte coulissante ci-dessous.



Une porte coulissante est un peu différente d'une porte classique. En effet, une porte classique peut s'ouvrir, rester ouverte et ne peut être interactive à nouveau. Autrement dit, il suffit de faire une animation d'ouverture. Dans le cas d'une porte coulissante, c'est légèrement différent. Elle doit, comme une porte classique, pouvoir s'ouvrir mais aussi se refermer dans le but de pouvoir être à nouveau ouverte. Pour ce faire, il suffit de cocher l'option «Has Exit Time» de l'animation. Cela m'a permis de créer un style de porte différent que j'ai utilisé notamment dans des zones de dépressurisation.

Bien entendu, afin de déclencher une animation, il fallait créer un script et l'associer à mon objet. Je m'attaque donc à la partie programmation au point suivant.

6) La programmation

Une fois mes animations mises en place, je me suis attaqué à la composition des scripts. Chaque script a été codé en langage C# (C sharp). Je les ai utilisés pour plusieurs raisons:

a. Dans le cadre d'une animation

L'exemple ci-dessous se réfère à une animation de porte. J'ai fait en sorte que si le joueur se trouvait devant la porte et appuyait sur «e», il allait jouer l'animation paramétrée au point précédent. De plus, la touche «e» disparaît de l'écran une fois qu'elle a été actionnée.

```
25 // Update is called once per frame
26 void Update () {
27
28     if (Input.GetKeyDown("e") && IsInCollider == true)
29     {
30         animPorteGarage.Play("EntreeSAS");
31         IsOpen = true;
32         animPlayed = true;
33         ToucheE.enabled = false;
34     }
35
36
37     if (IsOpen == true && IsInCollider == true)
38     {
39         ToucheE.enabled = false;
40         animPlayed = true;
41     }
42
43
44 }
```

Si le joueur rentre dans la zone de déclenchement du script, je veux que la touche «e» soit affichée sur l'interface de l'utilisateur. S'il sort de cette zone, la touche «e» disparaîtra. Cela permet de faire comprendre au joueur qu'il peut interagir avec l'objet.

```
46 void OnTriggerEnter(Collider other)
47 {
48     if (other.CompareTag("Player") && IsOpen == false)
49     {
50         ToucheE.enabled = true;
51         IsInCollider = true;
52         animPlayed = false;
53     }
54
55     if (animPlayed == true)
56     {
57         ToucheE.enabled = true;
58     }
59
60 }
61
62 void OnTriggerExit(Collider other)
63 {
64     if (other.CompareTag("Player"))
65     {
66         ToucheE.enabled = false;
67         IsInCollider = false;
68         animPlayed = true;
69         IsOpen = false;
70     }
71 }
72 }
```


b. Dans le cadre de dialogues

En vue de faciliter la compréhension du scénario, j'ai décidé d'ajouter des sous-titres (activables et désactivables) pour le joueur lors de chaque dialogue. Afin de les faire apparaître, j'ai créé une «Coroutine» sur Visual Studio.

Lors de chaque ligne de dialogue, je dois pouvoir contrôler 3 choses:

1. Le temps d'attente entre le déclenchement du dialogue et son apparition à l'écran.

Il est instantané ou parfois décalé de quelques secondes. Pour le faire apparaître, je lui demande un «WaitForSeconds» qui, comme son nom l'indique, affichera le texte après le nombre de secondes indiquées.

2. La durée du dialogue avant de passer au suivant

C'est le temps compris entre le début et la fin d'une ligne apparue à l'écran. Dans le cas de la première ligne de dialogue («Bienvenue dans la salle d'archives»), elle dure 5 secondes avant de disparaître.

3. Le temps entre le début du dialogue et sa fin.

Même principe que pour le point 2 sauf qu'après les secondes écoulées, je veux qu'il m'affiche un texte vide afin de pouvoir écrire le suivant. Sans quoi les deux textes se superposeraient. C'est la raison pour laquelle il est affiché «" »».

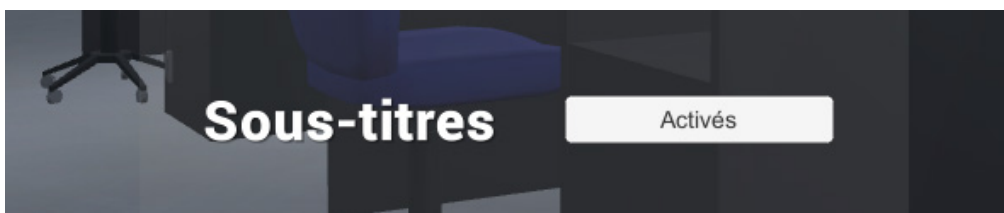
```
35 // Sous-titres
36 IEnumerator SousTitres()
37 {
38     yield return new WaitForSeconds(1);
39     textBox.GetComponent<Text>().text = "Bienvenue dans la salle d'archives ! Je pensais pas que c'était aussi grand ! C'est immense ! ";
40     yield return new WaitForSeconds(5);
41     textBox.GetComponent<Text>().text = "";
42     yield return new WaitForSeconds(0);
43     textBox.GetComponent<Text>().text = "Et encore, on est qu'au niveau Delta. Il y a encore trois niveaux au dessus du nôtre.";
44     yield return new WaitForSeconds(4);
45     textBox.GetComponent<Text>().text = "";
46     yield return new WaitForSeconds(1);
47     textBox.GetComponent<Text>().text = "Bon alors, il s'agirait de trouver la pièce essentielle à ton vaisseau, son alimentation.";
48     yield return new WaitForSeconds(5);
49     textBox.GetComponent<Text>().text = "";
50     yield return new WaitForSeconds(0);
51     textBox.GetComponent<Text>().text = "Tout est rangé par date, ça devrait pas être trop compliqué.";
52     yield return new WaitForSeconds(3);
53     textBox.GetComponent<Text>().text = "";
54 }
55
56 // Update is called once per frame
57 void Update()
58 {
59
60     if (Input.GetKeyDown("e") && IsInCollider == true && HasOpenedDoor == false)
61     {
62         anim.Play("PorteMetal");
63         IsOpen = true;
64         BruitPorte.Play();
65         MusiqueLvl5.Play();
66         Voix.Play();
67         StartCoroutine(SousTitres());
68         VoixIsPlaying = true;
69         HasOpenedDoor = true;
70     }
```

Tout ceci est valable dans le cas où les sous-titres s'affichent et restent par défaut. Étant donné que tout le monde ne veut pas forcément de sous-titres, j'ai décidé, comme dit précédemment, de les rendre activables et désactivables.

Voici, sous forme de code, la manière dont j'ai procédé. Cela fonctionne donc, comme beaucoup de mes scripts, sous forme de «if...». Ce qui signifie, dans le cas ci-dessous, que si le joueur a cliqué sur le bouton «Activés» dans le menu des sous-titres, le texte va changer et devenir «Désactivés». De plus, je lui demande de me désactiver la «textBox» qui est en fait la zone d'affichage du texte sur l'interface de l'utilisateur.

```
63 public void SwitchSubs()
64 {
65     // si j'active les sous-titres
66     if (SubDec == false)
67     {
68         textBox.SetActive(true);
69         SubDec = true;
70         buttonText.GetComponent<Text>().text = "Activés";
71         return;
72     }
73
74     // si je désactive les sous-titres
75     if (SubDec == true)
76     {
77         textBox.SetActive(false);
78         SubDec = false;
79         buttonText.GetComponent<Text>().text = "Désactivés";
80         return;
81     }
82 }
83
```

Voici le résultat en jeu. Le texte «Sous-titres» n'est qu'un simple texte ajouté sans programmation.



c. Dans le cadre d'exploration

Au fur et à mesure de l'avancement dans le jeu, le joueur est confronté à des portes verrouillées qui nécessitent une carte magnétique afin d'être ouvertes. Pour ce faire, j'ai caché à certains endroits les objets nécessaires à l'avancement dans l'histoire.

Dans le but de faire comprendre au script que l'animation de la porte ne peut se jouer que s'il a récupéré un objet essentiel, j'ai procédé comme ceci:

1. «CarteMagnBiosphere» est un autre script. Je lui demande d'aller dans ce script et d'aller chercher la variable «HasCardBio». Si elle est «true», cela signifie que le joueur a récupéré la carte. Je lui demande donc de faire disparaître «PasCarte» qui est le texte affiché lorsque l'on essaie d'ouvrir la porte sans le matériel requis.

2. S'il appuie sur «e» devant la porte et qu'il a récupéré la carte, l'animation se déclenchera et la porte s'ouvrira.

3. Inversement, s'il appuie sur «e», qu'il se trouve devant la porte et qu'il n'a pas récupéré la carte, le texte lui indiquant qu'une carte magnétique est nécessaire pour entrer s'affichera.

```
26 void Update () {
27     if(CarteMagnBiosphere.HasCardBio == true)
28     {
29         PasCarte.enabled = false;
30     }
31
32     if (CarteMagnBiosphere.HasCardBio == true)
33     {
34         if (Input.GetKeyDown("e") && IsInCollider == true && HasOpenedDoor == false)
35         {
36             anim.Play("PorteBiosphereCarte");
37             ToucheE.enabled = false;
38             IsOpen = true;
39             BipPorte.Play();
40             HasOpenedDoor = true;
41         }
42     }
43
44     if (Input.GetKeyDown("e") && IsInCollider == true && CarteMagnBiosphere.HasCardBio == false)
45     {
46         PasCarte.enabled = true;
47         ToucheE.enabled = false;
48     }
```

d. Dans un menu de pause

A n'importe quel moment, le joueur peut appuyer sur la touche «echap» de son clavier afin de mettre le jeu en pause. Dans le but de rendre cela possible, il fallait penser à plusieurs détails qui ont leur importance pour la programmation.

1. «Geler» le temps. Si le joueur était en train de sauter, par exemple, je ne voulais pas qu'il se voit retomber en ayant l'interface du menu pause active.
2. Empêcher tout mouvement de caméra. En effet, lorsque le jeu est en pause, le temps est bel et bien arrêté mais il est toujours possible de faire bouger la caméra autour de nous. «Cursor.lockState» me permet de gérer cette option. Bien entendu, j'utilise «Cursor.visible» pour permettre l'affichage de la souris en mode pause.

```
20 void Update(){
21
22
23     if (Input.GetKeyDown(KeyCode.Escape))
24     {
25         // je freeze la caméra
26         this.firstPersonController.mouseLookEnabled = false;
27         Cursor.lockState = CursorLockMode.None;
28         //je fais apparaître le curseur
29         Cursor.visible = true;
30
31         if (GameIsPaused)
32         {
33             Resume();
34         }
35
36         else
37         {
38             Pause();
39         }
40     }
41 }
42
43
44 if(GameIsPaused == false)
45 {
46     this.firstPersonController.mouseLookEnabled = true;
47 }
48
49 }
50
```

3. Couper le son. A chaque fois que le menu est activé, le son (voix, ambiance, musique...) doit s'arrêter et reprendre une fois le menu désactivé. J'utilise donc «AudioListener.pause» pour effectuer cette action.

```
51 public void Resume()
52 {
53     PauseMenuUI.SetActive(false);
54     Time.timeScale = 1f;
55     GameIsPaused = false;
56     AudioListener.pause = false;
57 }
```

7) Contenu annexe du jeu

a. Succès et easter eggs

Afin d'ajouter un petit esprit compétitif et d'amener le joueur à accroître le côté exploration, j'ai pris l'initiative d'ajouter des petits détails cachés. Ces détails sont réunis sous deux formes:

1. Les succès

Ils sont présents dans le but d'apporter un supplément au jeu pour le joueur. Ils sont, pour la plupart d'entre eux, obtenus lorsque l'on réalise des actions qui n'ont pas pour but d'aider à la progression dans l'histoire. Ils peuvent être une référence à un film / jeu vidéo ou encore une action inattendue. Lorsqu'ils sont déclenchés, ils apparaissent quelques secondes à l'écran.



Présents au nombre de huit, ils sont visibles plus en détails par la suite dans l'onglet «succès» du menu principal. A la fin du jeu, les succès non déverrouillés sont visibles mais sombres.



b. Indications cachées

Comme je l'ai déjà énoncé précédemment, «Multiverse Project» est un jeu qui n'omet pas le sens du détail. En effet, en explorant, le joueur est confronté à certains détails cachés dans le décor qui renseignent un peu plus sur l'histoire de l'endroit. Quelques indications sur un tableau, un diplôme encadré et accroché au mur etc... Cela rajoute un côté vivant en dehors des objectifs principaux et force un peu la nécessité de découverte.

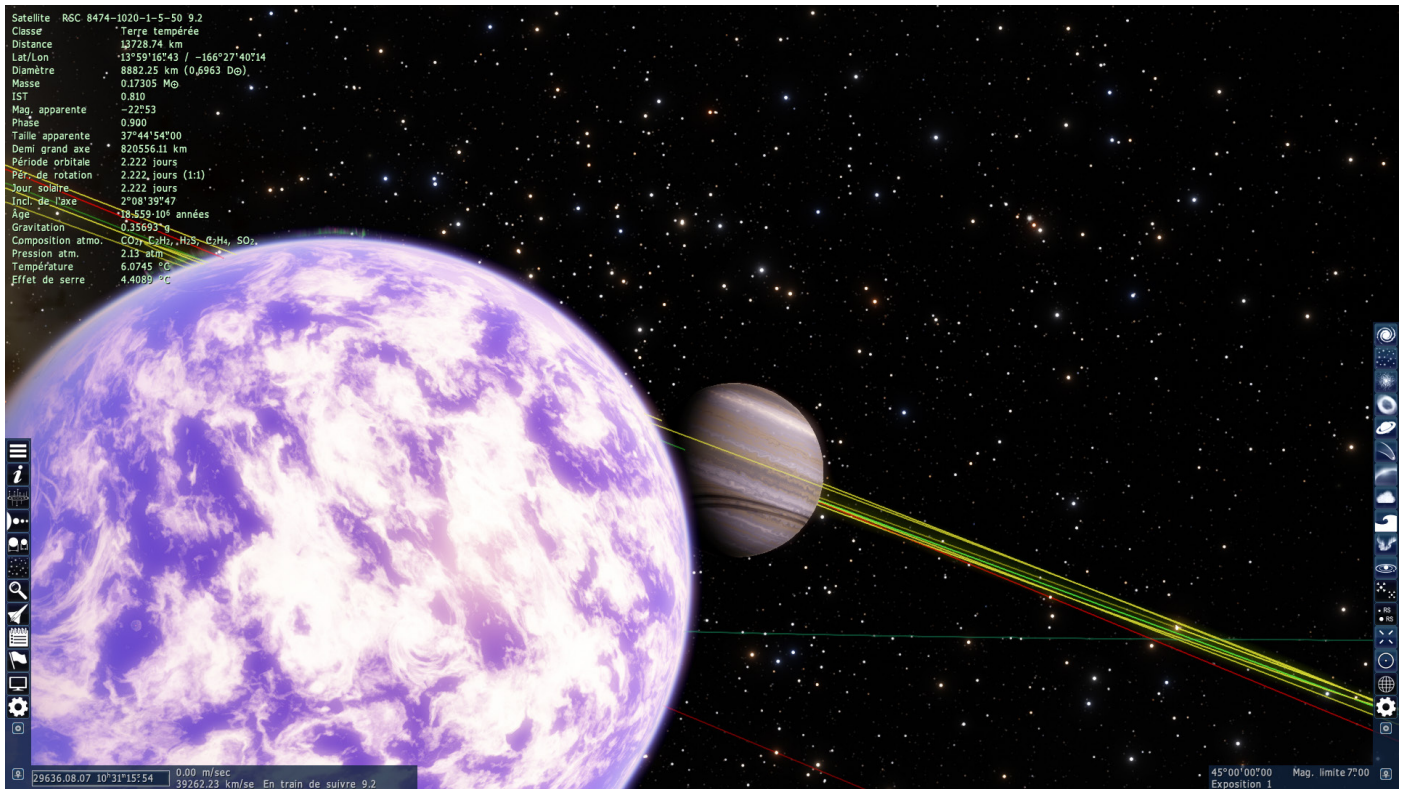


c. Détails minimes

Certains détails présents dans le jeu n'ont pas été simplement «inventés». Valley Falls (lieu du crash de l'extraterrestre) existe vraiment. C'est une petite communauté dans le comté de Lake dans l'Oregon. J'ai choisi cet endroit car il est simplement très reculé et au milieu de nulle part. Le recréer en 3D ne nécessitait donc pas beaucoup de travail. Quelques cactus, un bâtiment, un véhicule...

Le décor s'adapte aussi à l'avancée du joueur. En effet, il n'est possible d'échouer qu'une seule fois dans le jeu: lorsqu'il faut franchir une porte avant la fin du décompte. Si le joueur échoue, un écran «Game Over» apparaît. Il peut donc recommencer. Cependant, s'il est attentif, il remarquera qu'une série de flèches rouges pointant vers la porte ont été ajoutées entre-temps. Cela a été inséré dans un but humoristique.

Kepler 184F (planète d'origine d'Apollo, personnage jouable) est réellement supposée être une planète habitable. Afin de la trouver, je me suis servi du logiciel «Space Engine» (gratuit) qui est un logiciel planétarium et un moteur de jeu basé sur la simulation de l'univers. La planète s'appelle en réalité «RSC 8474-1020-1-5-50». Elle gravite également autour d'une géante de gaz (qui est représentée dans mon jeu).



Les détails de certaines textures ont été travaillés plus que d'autres. Les panneaux solaires, par exemple, reflètent la lumière du soleil. Les miroirs ont également un reflet, le ciel de la surface de Mars (dans le jeu) a été créé avec la même couleur que la vraie planète et elle contient aussi la même gravité, des petites tempêtes de sable sont visibles de temps en temps, certaines inscriptions sur les cartons de la salle d'archives relatent des évènements s'étant déroulés à une certaine époque etc...

8) Fins alternatives

L'expérience de jeu n'est pas la même pour tout le monde. En effet, au vu des choix qui vont être réalisés par le joueur, la fin ne sera pas la même. Il existe deux fins possibles.

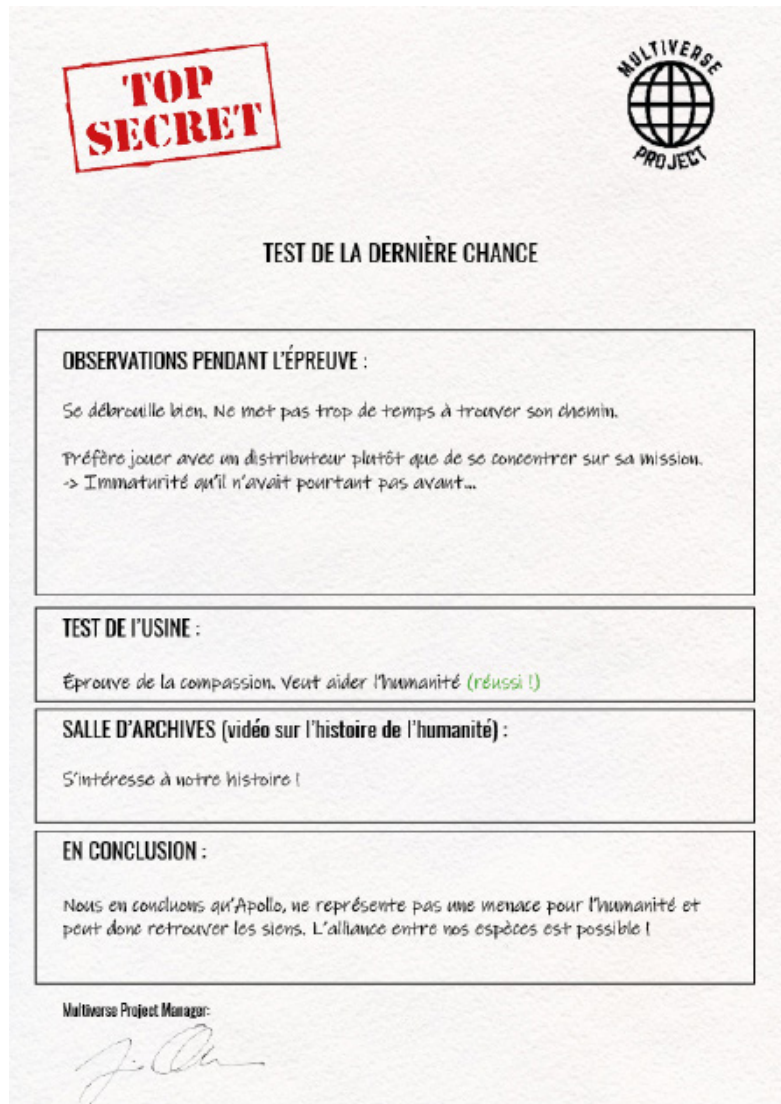
1. Une fin qui se termine bien

Au cours de l'histoire, des objectifs facultatifs sont proposés. Par exemple, appuyer sur un levier pour aider quelqu'un. Si le joueur s'est intéressé aux objectifs facultatifs et qu'il les a complétés, il déclenchera automatiquement la fin heureuse.

2. La fin qui se termine mal

A l'inverse, s'il a choisi d'ignorer ces objectifs facultatifs, il déclenchera la fin triste. Il existe donc deux vidéos de fin.

Cependant, ce n'est pas tout. Les choix réalisés sont cités à la fin du générique de fin sous une forme de rapport. Il y est écrit les différents choix que le joueur a appliqués. J'ai mis au point 8 rapports différents dont voici un exemple:



TOP SECRET

MULTIVERSE PROJECT

TEST DE LA DERNIÈRE CHANCE

OBSERVATIONS PENDANT L'ÉPREUVE :

Se débrouille bien. Ne met pas trop de temps à trouver son chemin.

Préfère jouer avec un distributeur plutôt que de se concentrer sur sa mission.
-> Immaturité qu'il n'avait pourtant pas avant...

TEST DE L'USINE :

Éprouve de la compassion. Veut aider l'humanité (réussi !)

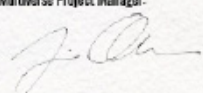
SALLE D'ARCHIVES (vidéo sur l'histoire de l'humanité) :

S'intéresse à notre histoire !

EN CONCLUSION :

Nous en concluons qu'Apollo, ne représente pas une menace pour l'humanité et peut donc retrouver les siens. L'alliance entre nos espèces est possible !

Multiverse Project Manager:



9) Tests utilisateurs

Bien qu'il soit évident pour moi de trouver les objets nécessaires à l'avancée dans le jeu, ce n'était peut-être pas le cas de tout le monde. J'ai donc pris l'initiative de faire tester mon jeu par deux types de personnes: des personnes habituées aux jeux vidéos (donc plus susceptibles de finir le jeu) et des personnes qui ne jouent jamais aux jeux vidéos. Ces tests me permettent aussi de résoudre certains problèmes qui pourraient apparaître tels que des fautes d'orthographe, des scripts qui ne se déclenchent pas normalement, des «oublis» dans la conception d'un niveau etc...

Après avoir reçu des retours de la part des testeurs, j'ai remarqué qu'il restait quelques problèmes à régler. Tout d'abord, le volume des dialogues était trop faible par rapport à l'ambiance générale du jeu. Il rendait donc obligatoire l'activation des sous-titres.

Il est possible, comme je l'ai précisé plus tôt, de devoir recommencer un niveau une seule fois au début du jeu. Cette action se produit lorsqu'on ne franchit pas une porte avant la fin du temps imparti. Il apparaît donc un écran «Game over». Le petit souci était qu'on devait à nouveau réaliser les actions nécessaires afin d'accéder à ce décompte. Cette petite modification a donc consisté à faire en sorte que le joueur puisse directement retourner à ce moment sans devoir tout recommencer.

A un certain moment au «niveau des rêves», il était compliqué de savoir où se diriger dans le brouillard. J'ai donc décidé d'ajouter des petites flèches discrètes au sol pour aiguiller le joueur. Je me suis également rendu compte que les joueurs arrivaient sans trop de problème jusqu'au bout du jeu. Les autres remarques concernaient quelques fautes d'orthographe dans des articles de journaux.



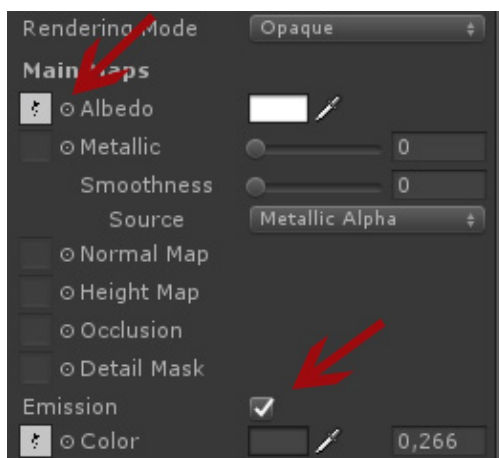
J'ai été agréablement surpris de voir que tout le monde a réussi à finir le jeu jusqu'au bout. La plupart d'entre eux m'ont partagé avec amusement les différents succès qu'ils avaient débloqués. Ce test par des utilisateurs m'a permis de peaufiner mon jeu pour la version définitive.

10) Problèmes rencontrés

Pendant le développement de mon jeu, je me suis retrouvé confronté à beaucoup de problèmes. J'ai pris le soin de les noter et d'y citer la façon dont j'ai procédé pour les résoudre. Je vais énumérer les problèmes principaux ainsi que ceux qui m'ont pris le plus de temps à résoudre.

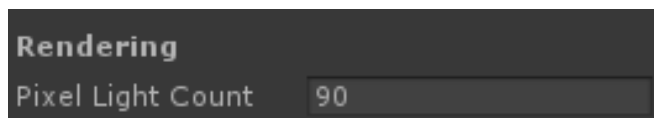
1. Comment faire en sorte que ma texture émette de la lumière ? (pour un écran par exemple)

J'ai voulu produire cet effet en priorité pour l'éclairage d'un écran d'ordinateur. Je savais produire cet effet pour une couleur mais pas pour une texture. Il suffit en réalité de cliquer sur le petit cercle à côté de «Albedo» et de sélectionner sa texture. Ensuite, dans «Emission», il suffit de la sélectionner et de choisir son intensité.



2. L'éclairage dans certains niveaux se met à clignoter quand on s'en approche

C'est un effet qui se produit lorsque beaucoup de lumières ont été ajoutées dans la scène. Pour régler ce problème, il faut augmenter le nombre de pixels de lumières «Pixel light count» à un niveau suffisant pour régler ce souci.

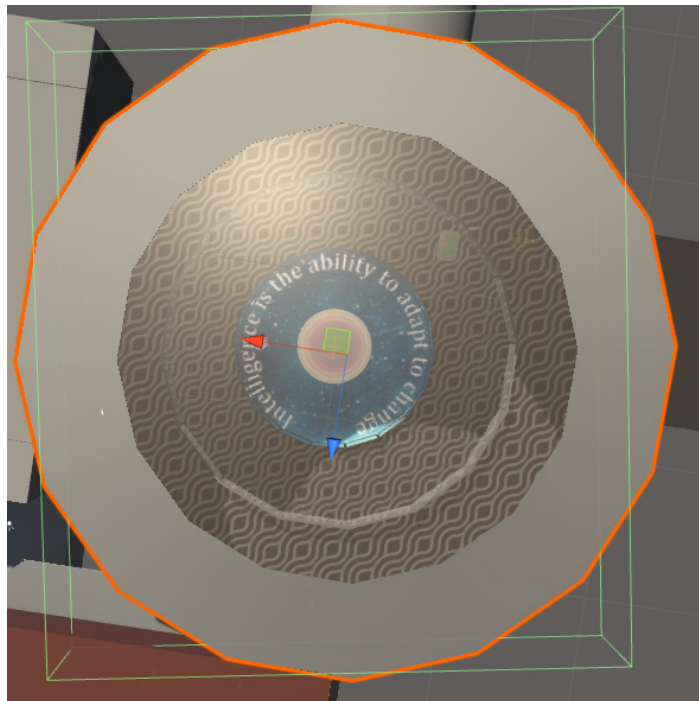


3. La rotation pendant l'ouverture d'une porte faisait n'importe quoi

La porte devait, lors de son animation, effectuer une rotation de 90° en une seconde. En la paramétrant depuis la fenêtre de transformation de l'objet, elle tournait plusieurs centaines de fois sur elle-même quand on déclenchait l'animation. Il se trouve que pour éviter ce «bug», il faut paramétrer l'animation depuis la fenêtre d'animation et non celle de transformation de l'objet.

4. Faire pivoter un «box collider»

Dans le cas d'une pièce en forme de polygône de 15 côtés, je ne pouvais pas mettre un simple «box collider» comme je l'aurais fait dans une pièce rectangulaire. En effet, si je procède de la sorte (comme ci-dessous), une partie de la pièce ne sera pas accessible car elle sera bloquée par un mur invisible. Il existe plusieurs sortes de «colliders»: le «box collider», le «capsule collider» et le «sphere collider». En aucun cas il n'existe un collider personnalisable.



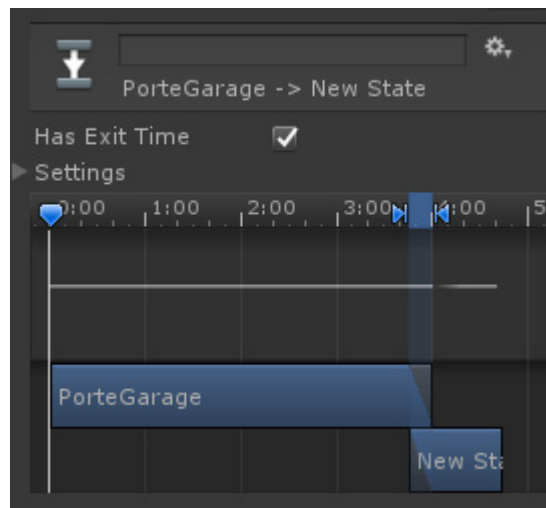
Pour régler ce problème, j'ai créé ce que l'on appelle un «empty» qui est un objet invisible. Je lui ai attaché un «box collider», ce qui a eu pour effet de pouvoir le placer où je le voulais. J'en ai donc créé 15 que j'ai placés à chaque face de la pièce. Le joueur peut donc désormais se déplacer de chaque côté sans risque de le voir bloqué par un mur invisible.

5. Le ciel de Mars change également la couleur d'éclairage au sol

Pour créer une surface de Mars réaliste, j'ai changé la couleur du ciel, ce qui a eu pour effet de changer aussi la couleur d'éclairage générale. Je me suis donc retrouvé avec un jeu très orangé. Afin de remédier à ce problème, il faut faire un petit réglage dans les options pour laisser la lumière émise par le soleil par défaut.

6. L'ouverture d'une porte ne peut être jouée qu'une seule fois

Cela était très embêtant lorsque je faisais une double porte coulissante. Le joueur devait pouvoir rentrer, se déplacer dans une pièce, pouvoir revenir et rouvrir la porte. Après pas mal de recherches, j'ai fini par découvrir qu'il existe une option dans la fenêtre «animator» de Unity. Cette option nommée «Has exit time» permet de jouer l'animation plusieurs fois.



7. Le curseur s'affiche quand on appuie sur «escape» (touche que j'avais définie avant pour «retour»)

Lorsque l'on voulait lire un article de journal ou toute autre image qui devait apparaître à l'écran, il fallait par la suite appuyer sur «escape» pour la faire disparaître. Le problème résidait dans le fait que le curseur s'affichait alors à l'écran. Etant donné que je comptais par la suite créer un menu de pause, j'ai simplement décidé de définir la touche de retour à «R». La touche «escape» affichant le menu, il était bien plus logique de voir apparaître le curseur.

8. La vidéo du menu commence par un écran blanc (mal aux yeux)

Afin d'afficher un menu un peu plus vivant, j'ai décidé d'y incorporer une vidéo à la place d'une image. Rien de sorcier, elle est composée du logo qui bouge sur lui-même. Cependant, dès l'apparition du menu, un écran blanc apparaît en premier pendant quelques secondes avant de passer à la vidéo. Pour résoudre ce petit détail, j'ai incorporé une image noire sur la totalité de l'écran. Je lui ai par la suite ajouté une animation qui contient un fondu du noir vers le transparent. La transition est désormais plus fluide et plus confortable pour les yeux.

9. Lorsque le jeu est en pause, on peut toujours faire bouger la caméra

En ce qui concerne le menu de pause, je m'étais renseigné sur internet. J'ai suivi un tutoriel très bien expliqué, ce qui m'a permis de gagner beaucoup de temps. Cependant, le tutoriel était conçu pour un jeu vidéo en 2D. L'auteur de la vidéo n'avait donc aucun problème de mouvement de caméra une fois le menu de pause activé. Afin de résoudre ce problème, j'ai ajouté quelques lignes de code dans le script de mon menu de pause.

```
if (Input.GetKeyDown(KeyCode.Escape) && GameOverUI.UiGameOverIsActive == false)
{
    // je freeze la caméra
    this.firstPersonController.mouseLookEnabled = false;
    Cursor.lockState = CursorLockMode.None;
    //je fais apparaître le curseur
    Cursor.visible = true;
}
```

Il y est indiqué que si le joueur appuie sur la touche «escape» et que l'interface du menu de pause est active, alors je veux qu'il me bloque la caméra, qu'il m'autorise à bouger le curseur et qu'il le fasse apparaître.

10. Les boutons du menu de pause ne fonctionnent pas sur les autres scènes

Le menu de pause devait bien entendu être disponible sur tous les niveaux. J'en avais donc fait un «prefab», ce qui permet de pouvoir le «copier coller» sur un autre niveau sans avoir à tout refaire. Malheureusement, lors d'un des nombreux tests réalisés, je me suis rendu compte que les boutons ne fonctionnaient pas. On pouvait afficher le menu de pause mais il était impossible de cliquer pour revenir au menu ou reprendre la partie. Le problème venait du fait qu'en important ce «prefab», il manquait un composant essentiel qui se crée automatiquement, le «Event System». Il se place automatiquement quand on crée un «canvas» (interface de l'utilisateur). J'ai donc simplement ajouté cet «Event System» et les boutons se sont mis à fonctionner à nouveau.

11. La «User Interface» de la touche «R» ne s'affichait que sur une des images

Lorsque j'ouvrais un article de journal, la touche «R» devait apparaître. Le problème, c'est que pour certains articles de journaux, elle n'apparaissait pas. Le script était pourtant presque identique. Je n'ai jamais trouvé d'explication à ce sujet. Afin de tout de même l'afficher, j'ai simplement ajouté une nouvelle «User Interface» de la touche «R».

12. Baisser la qualité des graphismes fait aussi baisser la qualité des textes lisibles

Afin d'optimiser mon jeu, j'ai mis en place une sélection de la qualité du jeu. Lors d'un test, je me suis rendu compte qu'en baissant la qualité en jeu, les textes lisibles devenaient illisibles car la qualité était trop basse. Après avoir effectué des recherches, j'ai fini par trouver la solution. Il existe, dans Unity, une option nommée «Generate Mip Map».



Cette option permet d'exclure cette texture de tout changement de qualité. De ce fait, même en qualité la plus basse, le texte restera de bonne qualité. Il suffit de décocher cette option.

11) Conclusion

Compte tenu de la grande liberté qui m'a été offerte lors de la conception de ce projet, j'ai pu créer tout ce qui me passait par la tête. Par conséquent, ce ne fut à aucun moment une «corvée» de réaliser ce projet. Bien qu'au début je voulais me contenter d'une simple démo jouable, j'ai vite trouvé cela très amusant. C'est pourquoi j'ai essayé de rendre mon jeu le plus complet possible. Grâce à cette liberté, j'ai pu créer un monde et un univers qui correspondent à ma personnalité et laisser ma créativité guider mes idées.

C'était aussi un gros challenge personnel. J'avais souvent entendu dire qu'on pouvait réaliser de beaux projets en n'y travaillant qu'une heure par jour. J'ai donc commencé en suivant ce principe: travailler une heure par jour sur mon TFE en partant du début de l'année scolaire jusqu'au mois de juin.

Bien que j'étais sceptique au début pour la partie programmation (que je maîtrise moins bien), elle ne m'a pas posé tant de problèmes grâce aux nombreux tutoriels et à la documentation disponibles sur internet.

En conclusion, cela a été une super expérience que j'aimerais réitérer dans le futur. J'espère que mon TFE pourra se démarquer des autres. Durant la conception de celui-ci, je n'ai eu aucun regret et j'ai fait du mieux que j'ai pu. J'espère que les utilisateurs de mon TFE auront autant de plaisir à y jouer que j'ai eu de plaisir à le créer.

Bibliographie

«Unity Documentation» In : <https://docs.unity3d.com/Manual/index.html> (consulté régulièrement au long du travail)

3D Bunch «Low Poly coconut tree modeling in 3ds max and free download» In: <https://www.youtube.com/watch?v=quij82eeu20> (consulté le 14 septembre 2019 et reproduit sur 3DS Max)

Info Gamer «How to Use Unity - Audio Source Game Objects» In: https://www.youtube.com/watch?v=_UI3HS_H-Xc (consulté le 17 septembre 2019)

Aubrey Millard «3DS MAX Tutorial - Model house plants - how to model a fern - Vray» In: <https://www.youtube.com/watch?v=W14l4xe6LVs> (consulté le 25 septembre 2019)

LMHPoly «Unity 2017 Tutorial - Sparks Particle Effect» In: <https://www.youtube.com/watch?v=MYU1vB-cUkfw> (consulté le 4 octobre 2019)

Design and Deploy «Unity 2018 - How to Make a Mirror - Method 1» In: <https://www.youtube.com/watch?v=pcIpGo6U2s0> (consulté le 6 octobre 2019)

MixPixVisuals «Creating a quick fire in Unity 5 - Basics of particle systems explained» In: <https://www.youtube.com/watch?v=M-Hz9Za5mCE> (consulté le 6 octobre 2019)

Arch Viz Camp «3ds max tutorial - Create basic Staircase» In: <https://www.youtube.com/watch?v=GxVwLuRnwew> (consulté le 9 octobre 2019)

Dapper Dino «How to get started with Unity - Video Player» In: <https://www.youtube.com/watch?v=O-dVGyV3rTGs> (consulté le 14 novembre 2019)

Brackeys «How to make Terrain in Unity!» In: <https://www.youtube.com/watch?v=MWQv2Bagwgk> (consulté le 19 novembre 2019)

Xenosmash Games «Creating Smoke with Shuriken Particle System in Unity» In: <https://www.youtube.com/watch?v=GebwdChivxI> (consulté le 18 décembre 2019)

CG Plugins «ATree3D v3.0 for 3ds Max | Creating plant in 3ds Max quickly» In: <https://www.youtube.com/watch?v=jS0PZVrs1tw> (consulté le 22 décembre 2019)

Brackeys «START MENU in Unity» In: https://www.youtube.com/watch?v=zc8ac_qUXQY (consulté le 14 janvier 2020)

Brackeys «PAUSE MENU in Unity» In: <https://www.youtube.com/watch?v=JivuXdrIHK0> (consulté le 27 janvier 2020)

Jimmy Vegas «Mini Unity Tutorial - How To Create Fog & Night Illusion - Beginners Tutorial» In: <https://www.youtube.com/watch?v=UgJE3TgT3o8> (consulté le 29 janvier 2020)

Brackeys «SETTINGS MENU in Unity» In: <https://www.youtube.com/watch?v=YOaYQrN1oYQ> (consulté le 9 février 2020)

«Wikipédia - Portal 2» In : https://fr.wikipedia.org/wiki/Portal_2 (consulté le 21 février 2020)

«Wikipédia - The Stanley Parable» In : https://fr.wikipedia.org/wiki/The_St Stanley_Parable (consulté le 21 février 2020)

Jimmy Vegas «HOW TO MAKE IN-GAME ACHIEVEMENTS FOR YOUR GAME IN UNITY #01 - COLLECTION BASED» In: <https://www.youtube.com/watch?v=-XuzwxkZ2Wk> (consulté le 23 février 2020)

Jimmy Vegas «HOW TO MAKE IN-GAME ACHIEVEMENTS FOR YOUR GAME IN UNITY #02 - LEVEL COMPLETE» In: <https://www.youtube.com/watch?v=k8ngpAfIeIM> (consulté le 23 février 2020)

Jimmy Vegas «HOW TO MAKE IN-GAME ACHIEVEMENTS FOR YOUR GAME IN UNITY #03 - TIME BASED» In: <https://www.youtube.com/watch?v=QesqDXGjV94> (consulté le 23 février 2020)

Jimmy Vegas «HOW TO MAKE IN-GAME ACHIEVEMENTS FOR YOUR GAME IN UNITY #04 - SECRET AREA» In: <https://www.youtube.com/watch?v=bXxXs69GR3g> (consulté le 23 février 2020)

Jimmy Vegas «HOW TO MAKE IN-GAME ACHIEVEMENTS FOR YOUR GAME IN UNITY #05 - HIDDEN OBJECT» In: <https://www.youtube.com/watch?v=wFlcinPG86s> (consulté le 23 février 2020)

Kitsune Art «The difference between Low poly and High Poly game assets» In: <http://kitsuneartutort.blogspot.com/2015/03/ladies-gentlemen-alien-high-elf-who.html> (consulté le 4 mars 2020)

Ben Loader Blog «Imagine Worlds Asset Production» In: <https://benloaderblog.wordpress.com/2017/09/26/imagine-worlds-asset-production-2/> (consulté le 4 mars 2020)

Jimmy Vegas «Mini Unity Tutorial - How To Make SUBTITLES & Turn Them On / Off C#» In: <https://www.youtube.com/watch?v=mU6Euqcm9UM> (consulté le 16 mars 2020)

«Wikipédia - SpaceEngine» In : <https://fr.wikipedia.org/wiki/SpaceEngine> (consulté le 17 mars 2020)

SonduckFilm «Create Stylized Movie Credits in After Effects | Tutorial» In: <https://www.youtube.com/watch?v=nwAFqGjKfVc> (consulté le 6 avril 2020)

Jimmy Vegas «HOW TO MAKE DUST PARTICLES IN UNITY TUTORIAL» In: <https://www.youtube.com/watch?v=tPtRCpwSgBg> (consulté le 5 mai 2020)