



UNIVERSITÉ DE TECHNOLOGIE DE BELFORT-MONTBÉLIARD

Rapport de projet VI50

Réalisé par :

DAYET Félix

FRITSHY Christophe

MBYAS SAROUKOU Abdel

Table des matières

I. Synopsis

II. Les utilisateurs cibles

III. Les interactions

IV. Le moi virtuel

V. L'univers graphique et sonore

VI. Formalisation du problème et méthode proposée

VI.1 Le robot

VI.1 Les blocs et leur arrivée

VI.2 Le pistolet à laser

VII. Les difficultés que nous avons rencontrées

VIII. Les améliorations possibles et perspectives d'ouverture

IX. Conclusion

Introduction

Dans le cadre de l'UV VI50, nous sommes amenés à effectuer un programme de réalité virtuelle afin d'appliquer nos connaissances et d'élargir notre expérience dans ce domaine. Nous avons choisi de réaliser un jeu vidéo, dont nous expliquerons le principe par la suite.

Dans notre rapport, nous expliquerons tout d'abord le principe et synopsis du jeu créé, puis nous ferons une explication de notre spécification et analyse du jeu selon son univers graphique et ergonomique, par la suite nous parlerons de la méthode proposée pour créer ce projet, et les difficultés rencontrées durant la conception du projet, pour enfin finir sur les améliorations possibles et les perspectives d'ouverture possibles du sujet traité.

I. Synopsis

Le programme créé est un jeu vidéo en réalité virtuelle utilisant le matériel oculus, il a été créé dans le but de divertir le joueur et de lui faire passer un bon moment.

Ce jeu est destiné à tous les publics, et ne limite pas son utilisation grâce à un sujet intéressant permettant à toute personne d'accrocher au fonctionnement et à l'intrigue.

Principe du jeu :

Le joueur débute le jeu dans un désert s'étendant à perte de vue autour de lui. Il voit tout d'abord un écran blanc qui laisse place à l'apparition du désert. Il n'a donc aucune idée d'où il se trouve. Il peut observer à côté de lui d'un bunker, et trouve un pistolet, ayant la capacité de tirer des lasers, situé à sa ceinture. Il aperçoit alors un objet volant s'approcher de lui, il s'agit d'un robot, celui-ci lui effectue alors une introduction du jeu en lui expliquant le fonctionnement, et lâche un bouton après avoir terminé sa tâche. Le joueur se saisit du bouton et l'actionne, le jeu se lance.

Le joueur peut alors apercevoir des blocs avancer sur lui selon un couloir. Son but va être de les esquiver en effectuant des mouvements de gauche à droite ou inversement. Il peut également utiliser le pistolet qui est à sa disposition afin de détruire certains des projectiles lui arrivant dessus. Si le joueur échoue l'esquive des blocs, il perd le jeu.

II. Les utilisateurs cibles

On vise avec ce jeu un ensemble d'utilisateur possédant un matériel de réalité virtuelle à disposition et ayant les capacités d'installer leur matériel et de lancer le jeu.

Une fois le jeu lancé, une grande nécessité d'expérience dans le domaine de la réalité virtuelle n'est pas nécessaire, si ce n'est que pour l'utilisation basique des manettes.

Cela représente donc une majorité d'utilisateurs "jeunes", en n'omettant pas les autres utilisateurs. Il fallait donc trouver un principe de jeu qui pourrait attirer n'importe quel personne, c'est pourquoi nous avons choisi de créer un jeu d'esquive dans un désert, qui est susceptible d'intéresser une grande majorité des utilisateurs cibles.

Pour trouver un sujet correspondant à tous, il nous fallait donc trouver une intrigue divertissante avec un décor agréable à observer. C'est pourquoi nous avons opté pour un désert. De plus il fallait chercher un fonctionnement de jeu qui ne nécessiterait pas des aptitudes physique hors du commun ou une connaissance très importante des manettes ou du casque de VR.

III. Les interactions

Le tableau des interactions que nous appliquons dans notre projet est celui-ci :

I ² fonctionnelles	PCV	Param PCV	Schème métaphore s réelles	Interfac e idéales	Schèmes et métaphore s magiques	interfaces dégradée s	Variantes
Esquiver obstacle	se déplacer	2DDL (y, z)	Marcher	Cave + lunette 3D	Déplacer le casque	Casque VR	Accroupi, debout
Casser obstacle avec pistolet	prendre objet	sélection de l'objet	Préhension	gant de donnée s	Appuyer bouton	Manette	
	déplacer objet	3DDL (x,y,z)	Porter, lancer	gant de donnée s	Appuyer bouton	Manette	
Evaluer environnement	Observer	Champ visuel humain(220°)	Regarder autour de soi	Cave + lunette 3D	Déplacer le casque	Casque VR	Présence/absenc e d'obstacle
	Se déplacer	3DDL (x, y Rz)	Marcher	Cave + lunette 3D	Déplacer le casque	Casque VR	
Attrapper	se	2DDL (y, z)	Marcher	Cave +	Déplacer le	Casque	

objet	déplacer			lunette 3D	casque	VR	
	Attrapper	-	serrer poing	gant de données	Appuyer bouton	Manette	
	déplacer main	3DDL(x,y,z)	déplacer main	gant de données	Appuyer bouton	Manette	

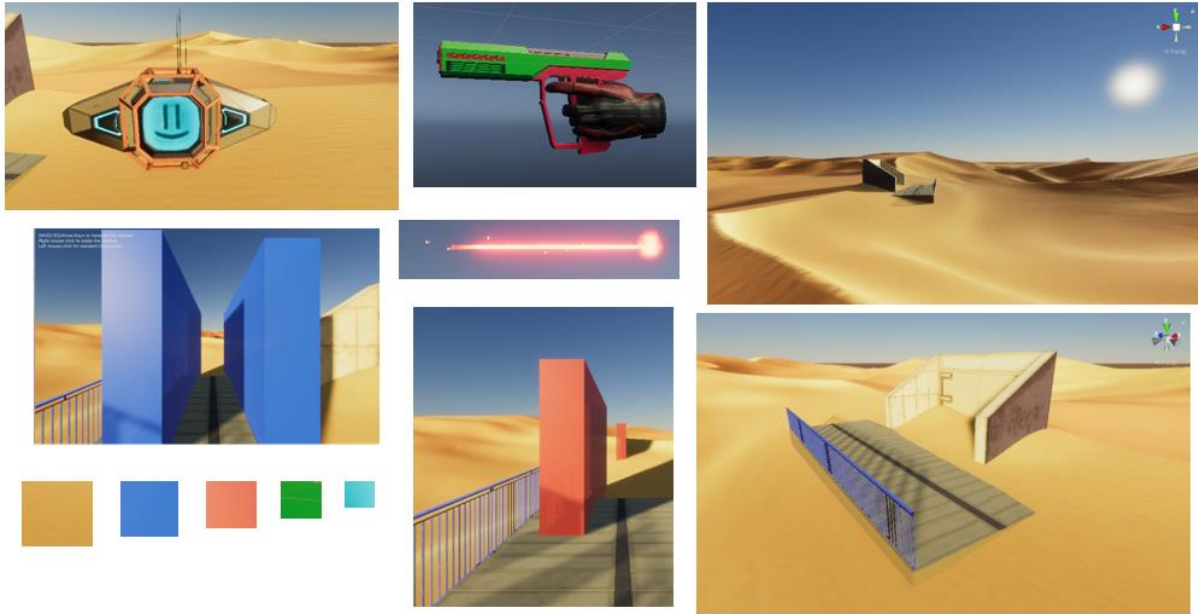
IV. Le moi virtuel

Élément de l'avatar	Moyen de contrôle	Représentation
Tête	Casque	Aucune
Main gauche	Manette gauche	Main 3D virtuelle
Main droite	Manette droite	Main 3D virtuelle

Nous avons opté pour l'utilisation de l'oculus rift s, ou les autres casque oculus en général, pour l'utilisation du projet. Cela nous semblait le choix le plus adapté pour offrir la meilleure expérience possible à l'utilisateur.

V. L'univers graphique et sonore

Il nous semble bon de préciser que dans le projet, tous les assets visuels ont été créés par notre équipe, en utilisant Blender pour désigner les mesh et les textures. L'univers graphique a donc été totalement conçu par notre l'équipe.



L'environnement principal dans lequel se déroule le jeu est un désert. Le joueur s'y trouve pendant le jeu et plus précisément à côté d'un bunker.

Apparaissent également dans le jeu différents objets, tels que le robot, les obstacles (blocs rouges et bleus) ainsi que le pistolet tirant des lasers rouges. Au loin, le joueur pourra également apercevoir un triangle d'où arrivent les blocs à esquiver.

Nous avons également créé un univers sonore dans notre jeu, sur une multitude d'actions et d'objets.

Tout d'abord, concernant l'environnement, une ambiance sonore entoure le joueur. Tout d'abord le joueur entend un bruit de vent ambiant, rendant l'environnement plus crédible. Puis au lancement des cubes apparaît une pyramide effectuant un effet sonore, ainsi qu'une musique qui se lance lors du jeu.

Concernant les objets, on peut citer le robot qui utilise une bande son enregistrée pour effectuer l'introduction du jeu. Le pistolet laser émet également un son de tir lors de l'appui sur la détente. Également lors de l'approche des cubes du joueur, un bruit de frottement apparaît, rendant l'illusion plus crédible.

VI. Formalisation du problème et méthode proposée

Afin de réaliser le jeu, il a fallu créer un ensemble d'objets pouvant interagir entre eux, et plaçant le joueur au sein de l'environnement. Par exemple, il nous était nécessaire d'effectuer une introduction au jeu pour le joueur, pour cela nous avons créé un robot.

De plus, il fallait créer des blocs et les envoyer vers le joueur, ainsi que créer un pistolet qui serait à la disposition du joueur pour détruire certains de ces blocs.

VI.1 Le robot

Le robot a pour but principal d'introduire le joueur à la scène et de lui expliquer rapidement ce qu'il doit faire lors de l'apparition des obstacles. De plus celui-ci permet d'ajouter un peu de "vie" au jeu pour qu'il y ait plus qu'une simple scène d'action.

Celui-ci doit pouvoir se déplacer autour de la map et jouer des pistes audio et des animations.

Il est d'abord loin du joueur, puis va s'approcher de lui selon une certaine trajectoire donnée. Il va ensuite effectuer différentes animations tout en lui donnant les règles du jeu.

VI.1 Les blocs et leur arrivée

Pour l'arrivée des blocs nous avons créé un canon virtuel. Il est doté d'une liste préétablie de préfab à instancier puis à lancer sur notre joueur. Les paramètres pouvant être choisis pour le lancement sont le type de préfab, la vitesse, la cible, le delay avant le suivant. La fonction de lancement est une coroutine qui est lancée.

Les cibles sont des objets empty disposés derrière le joueur. Elles sont au nombre de 6, 3 disposé au ras du sol et 3 autres à hauteur du joueur.

Les projectiles sont des des prefabs ayant à minima le component Projectile. Leurs propriétés de bases sont la cible, la vitesse. Ces propriétés sont assignables au lancement par le Canon.

Juste avant les cibles se trouve un layer de suppression de projectiles. Il se charge de désactiver les blocs.

Tous les blocs désactivés soit par le laser soit par le layer de suppression sont retournés dans un pool d'objets. Ce pool se charge de conserver les références des objets pour que ceux-ci puissent être réutilisés à l'infini et ainsi optimiser la consommation de mémoire.

Les scripts utilisés et leurs fonctions:

- Projectile: Script définissant les propriétés d'un projectile: cible, vitesse, type
- Canon: Script définissant le premier niveau de jeu grâce à une queue de projectile à envoyer vers le joueur. Ce script se charge entre autre de démarrer et arrêter le lancement des blocs grâce à une coroutine.

- Uninstall: Script permettant de détecter les collisions avec un projectile et de les désactiver
- ObjectPoolManager: Script permettant de stocker au long du jeu les références d'objets pour ne pas les perdre et ainsi les réutilisés.

VI.2 Le pistolet à laser

Pour effectuer le pistolet à laser il a fallu tout d'abord créer un modèle de pistolet sous blender puis l'exporter dans unity. Il fallait ensuite créer le laser.

La méthode que nous avons choisie pour effectuer ceci a été d'utiliser un line renderer. Nous avons ensuite créé un shader graph dans le but d'introduire un matériau qui constituerait le laser, afin de lui donner l'aspect le plus réel possible, en lui créant différents masques et introduisant du bruit le rendant plus convaincant.

Pour le tir du laser il fallait également créer un flash pour la bouche du pistolet, nous avons utilisé pour ceci un système de particule. Le matériau utilisé est un matériau similaire à celui du laser, néanmoins différent, notamment par l'utilisation de masques différents.

Une fois l'ensemble terminé, nous avons conçu deux différents types de collision pour le laser :

- Le premier type est la collision avec les objets destructibles de la scène, dans ce cas, on va détruire les objets lors de la collision avec le laser et faire appel à un système de particules pour le rendre plus réel.
- Le second type de collision est celui avec les objets non destructibles de la scène, comme par exemple le désert, le bunker etc... Dans ce cas on instancie le prefab d'une marque de brûlure à l'endroit de l'impact.

Pour ces collisions on utilise un collider directement placé sur le prefab du laser qui activera le système de particule ou instanciera les marques de brûlure selon le tag ou layer de l'objet rencontré. Pour éviter tout lag dus à la destruction des laser, nous avons choisi de les désactiver et non de les détruire.

En effet, chaque laser, une fois la collision effectué, est désactivé. S'il n'interagit avec aucun élément durant une certaine période il se désactive également. Afin d'éviter la création intempestive de nouveaux laser, nous avons utilisé un tableau qui prend en entrée chacun des lasers présents dans notre jeu, pour toute nouvelle création de laser on va vérifier si le tableau comprend pas déjà un laser inactif. Ainsi, dans le cas où certains lasers ne sont plus utiles (inactif), on les réutilise dans le jeu.

On effectue le même travail pour les marques de brûlures sur les objets, elles sont désactivées au bout d'un certain temps, et réutilisées si nécessaire.

Pour l'implémentation au sein du jeu, nous avons décidé de placer le pistolet à côté du joueur lorsque le jeu débute afin de le laisser découvrir son environnement et les éléments à sa disposition. Egalement, nous avons réglé les mains du joueur pour qu'elles s'adaptent au pistolet de manière à rendre la prise en main du pistolet plus réelle.

Le pistolet fera également du bruit lors du tir du laser afin de rendre l'illusion de tir plus authentique.

Les scripts utilisés et leurs fonctions:

LaserSpawnScript : C'est le script qui permet l'instanciation des lasers lorsque l'on tire avec le pistolet. Il place les lasers créés dans un tableau et réutilise ceux qui sont inactif.

LaserImpactScript : Il permet la détection de collision des lasers avec les objets du décor. Il teste le tag des objets avec lesquels il engage une détection afin de savoir si il a besoin de détruire l'objet ou d'instancier une marque de brûlure à l'endroit du laser.

LaserFirepointScript : Utilisé pour la désactivation du flash de la bouche de pistolet. Il est placé directement sur le prefab du flash.

ScorchPrefabScript : C'est le script qui va permettre de faire disparaître les marques de brûlure au bout d'un certain temps. Elles sont désactivées puis réutilisées si nécessaire.

LaserRayScript : Comme ScorchPrefabScript, il permet de faire disparaître les lasers au bout d'un certain temps.

VII. Les difficultés que nous avons rencontrées

Concernant le pistolet, nous avons pu observer que les marques de brûlure n'apparaissent pas forcément, notamment sur le désert. Le fait que l'impact soit sur un mesh collider ayant une surface creuse par endroits, cela provoque que les marques de brûlure n'apparaissent pas forcément.

Dans notre projet, nous utilisons la pipeline hd afin d'essayer rendre l'expérience visuelle la meilleure possible. Nous avons remarqué cependant un problème de performance au cours du jeu, et il semblerait que cela provienne directement d'un bug Unity.

VIII. Les améliorations possibles et perspectives d'ouverture

Une des améliorations que l'on pourrait apporter à notre projet est de mettre en place de nouveaux niveaux, apportant d'autres difficultés à notre jeu, et utilisant d'autres styles graphiques (par exemple des environnements différents comme une jungle, une ville...).

De plus, il serait possible de mettre en place de nouvelles interactions avec les obstacles, comme par exemple les briser avec le poing ou avec la crosse du pistolet, ou encore les escalader.

Une ouverture de notre sujet serait l'implémentation d'un algorithme permettant de créer automatiquement l'arrivée des objets au cours du niveau. Cela serait utile, notamment pour ne pas avoir à écrire un script prévoyant l'arrivée des blocs.

IX. Conclusion

Le projet est une application concrète des notions vues en cours, nous avons ainsi solidifié nos connaissances en Unity et Blender. De plus, le fait d'avoir un sujet libre a permis de travailler sur un sujet qui nous intéressait, ce qui est une source de motivation car il était plaisant de travailler dessus. Ce fut également pour nous un moyen de mieux comprendre les principes de développement en matière de réalité virtuelle et de nous rendre plus familier avec le matériel nécessaire pour son application. Nous avons bien pu définir nos tâches afin d'avancer au mieux, et ainsi tirer de ce projet une bonne expérience.