



COARV - Projet Unity

Développement d'un Puzzle Game sous Unity en Réalité Virtuelle

Phase II

Par les élèves suivant :

BRUNOT Matthieu, HUA Alexandre, LESDALONS Christopher, QU Michel,
OPPMANN Thomas

Option Réalité Virtuelle

Lien du Git du projet :

<https://github.com/MichelQu/TimeTravel>

Sommaire

1. Introduction

2. Présentation de l'escape game

2.1. Le scénario

2.2. La scène

2.3. Les scripts

2.4. Mécaniques du jeu

2.5. Fonctionnalités non-implémentées

3. Travail effectué

3.1. Mise à niveau du projet

3.2. Commentaire de code et documentation

3.3. Ajout d'une nouvelle scène

3.4. Répartition du travail

4. Ce que nous aurions aimé faire

5. Conclusion

1. Introduction

Dans un premier temps, nous nous sommes familiarisés avec le projet pour pouvoir ensuite y apporter nos modifications. Mais il a d'abord fallu mettre à niveau la version d'Unity, et résoudre les problèmes qui en ont découlé.

Nous allons ensuite implémenter certaines de ces améliorations, et ajouter les parties qui n'ont pas pu être faites par le groupe de l'année dernière, mais nous n'avons pas eu le temps car le coronavirus nous a empêché de tester avec le Vive.

2. Présentation de l'escape game

2.1. Le scénario

On présente ici une version courte qui met l'accent sur les interactions du jeu, pour la version longue, se référer au rapport G1 - Time Travel 2019.pdf section 2 "Présentation de l'application"

Le joueur est d'abord dans la temporalité années '80.

- Son soi futur l'invite à venir voir les dégâts de la pollution: il joue la cassette avec le walkman pour se téléporter dans le futur.
- Il joue une musique western sur le synthétiseur, qui fait apparaître une cassette. On utilise cette cassette pour se téléporter dans la temporalité Western.
- Le joueur détruit un mur fissuré avec de la dynamite. Il trouve la clé du coffre-fort. Il retourne dans le futur pour trouver la cassette années '80.
- Il place la cassette non-rembobinée dans le tiroir inter-dimensionnel, et retourne dans le western. Il la rembobine avec un crayon, place la clé dans le tiroir et se téléporte dans les années '80.
- Il récupère la clé, ouvre le coffre, prend la graine et la plante.
- Il peut aller dans le futur voir des arbres.

2.2. La scène

Le jeu se déroule dans 3 époques différentes: Western, années '80 et années 2050 futuriste.

Dans la scène, on retrouve ces trois environnements qui sont représentés par des préfab: *Western*, *80s_large* et *2050*. *[CameraRig]* et *[SteamVR]* proviennent de SteamVR et représentent le joueur. Enfin on a le *walkman* qui sera téléporté avec le joueur.

2.3. Les scripts

Dans cette section, nous décrivons rapidement les rôles des scripts, et les améliorations possibles que nous n'avons pas eu le temps d'implémenter.

Les trois époques

Les différentes époques sont modélisées et gérées par la classe *ZoneManager*. Il est attaché au *walkman*. On y définit aussi la classe *Zone*.

Pour faciliter le développement, on pourrait séparer ces trois époque dans des scènes différentes. Mais le choix d'avoir les trois lieux dans la même scène permet de téléporter facilement par translation. À creuser.

Scripts joueur

On utilise la téléportation pour se déplacer. Elle est implémentée dans *LaserPointer*. On utilise un cube aplati pour le laser.

Une amélioration possibles serait d'utiliser un arc au lieu d'une ligne droite pour sélectionner l'endroit où se téléporter car cela permet d'avoir plus de précision lorsqu'on essaye de se déplacer loin. Si on est en ligne droite on a un petit angle de travail. Avec un arc, on peut compenser. cf. Le déplacement dans les démos SteamVR.

Le grab est Implémenté dans *ControllerGrabObject*, on attache l'objet au contrôleur sur l'appui d'un bouton. On utilise une boîte de collision devant le contrôleur qui est "trigger" lorsqu'il entre en collision avec un rigidbody. On attache alors l'objet à notre contrôleur.

2.4. Mécaniques du jeu

Walkman et cassettes

Lorsqu'une cassette est jouée dans le walkman, la musique est lancée et le joueur est téléporté. Pour le moment, c'est la cassette qui joue la musique et qui téléporte le joueur. On a commencé à déplacer ce code dans la classe walkman. Il est envisageable de stocker la musique sur la cassette pour la jouer après.

Éventuellement, on pourrait implémenter un fadeout lorsqu'on utilise une cassette, et un bruit de cassette abîmé pour les mauvaises.

Synthétiseur

Il faut jouer une mélodie sur le synthétiseur durant un temps imparti pour faire apparaître une cassette. L'implémentation de cette interaction n'est pas complète car le joueur n'est pas obligé de poser les mains sur le clavier.

Destruction du mur avec de la dynamite

Le script Dynamite prend en charge l'animation et la réapparition de la dynamite. Il déclenche aussi la fracturation du mur. La mèche commence à brûler lorsqu'on attrape la dynamite. La fracturation du mur est implémentée dans Fracturable. Il va ajouter une force sur les morceaux, et détruire le tout dès qu'il touche le sol.

Tiroir inter-dimensionnel

Le script tiroir expose une fonction qui permet de récupérer ses enfants pour les déplacer lorsqu'on change d'époque. Il semblerait qu'il manque une méthode pour placer des objets dans le tiroir.

Le script 80s/Drawer_constraint restreint la position du tiroir selon l'axe X jusqu'à une position d'arrêt. Il est uniquement appliqué sur le tiroir des années '80. Ce script peut être généralisé pour appliquer les contraintes sur les autres tiroirs.

2.5. Fonctionnalités non-implémentées

Ces interactions étaient prévues par le groupe précédent, mais on été abandonnées par manque de temps.

- Rembobinage de la cassette années '80 avec le crayon
- Ouverture du coffre avec la clé trouvée dans le passé
- Planter la graine et observer l'effet dans le futur
- Dialogue d'introduction par le détective du futur

3. Travail effectué

3.1. Mise à niveau du projet

Cette partie a consisté à mettre à niveau le projet pour qu'il fonctionne avec une version plus récente d'unity (la version 2019.3.0) pour ce faire, les étapes suivantes ont été suivi :

- Supprimer les anciens asset steamVR pour pouvoir importer la dernière version de steamVR,
- Corriger toutes les parties du code déjà écrite qui ne fonctionnait pas avec les nouvelles versions d'unity et de steamVR (correspond en partie de l'appelle au Inputs des contrôleurs et à l'utilisation de certaines bibliothèques),
- Modifier le code existant pour qu'il soit adapté aux nouveaux appels d'Inputs de steamVR,
- Associer les actions aux différents inputs des contrôleurs,
- Recréer un nouveau projet en dupliquant l'ancien pour corriger un bug dans lequel les contrôleurs n'apparaissait pas dans le jeu.

Malheureusement suite à l'annonce du confinement, nous n'avons pas pu finir de déboguer les différentes interaction coder par nos prédécesseurs, celle ci étant :

- Un souci au niveau du point de téléportation mal calibré,
- Une correction pour rendre plus naturel l'utilisation du tiroir temporel (la liaison appliquée manquait de restrictions permettant des faire coulisser le tiroir sans qu'il soit agripper par l'utilisateur et ne limitant pas la distance de coulissage).

3.2. Commentaire de code et documentation

Le projet que nous avons reçu possédait un git défaillant. En effet, il avait 2 git : le premier dont le dernier commit ne fonctionnait pas et un autre contenant le code qui marche mais ne possédant pas d'historique. Pour réparer cela, nous avons dû :

- Récupérer l'historique du git cassé,
- Récupérer le commit avec l'application qui marchait,
- Fusionner le tout dans un 3e lien git car nous n'avions pas les accès sur les 2 git fournis par le groupe précédent.

Le lien du git fonctionnel est fourni en début du rapport du projet. De plus, en respect avec les pratiques d'un projet nous avons :

- Ajouter des informations dans les onglets Issues et Project du git pour permettre aux prochains de voir l'avancement du projet.

Dans un second temps, certains scripts qui étaient à disposition possédait peu voire aucun commentaire utile à la compréhension globale ou détaillé des scripts. Pour l'utilité du projet,

nous avons donc décidé de commenter et de créer de la documentation pour ces scripts. Nous avons ajoutés les points suivants :

- Commentaire globale et détaillé sur scripts suivant : Cassette, Dynamite, Walkman, ZoneManager, Tiroir, Fracturable pour permettre une meilleure transmission et compréhension du projet. Les autres scripts essentiellement sur la gestion des controllers et de ses fonctionnalités, possédaient déjà du commentaire en anglais.
- Ajout de quelques lignes de code en rapport avec les musiques sur le script Walkman et des indications pour les futurs groupes.
- Écriture d'un fichier de documentation à partir des commentaires du code faits pour aider les prochains à savoir les points à régler, à ajouter et aussi rappeler l'utilité des scripts. Cette documentation rappelle également les grandes lignes du projet.

Nous avons également fait une réorganisation des dossiers dans le dossier Assets pour mieux s'y retrouver dans les fichiers et pour éviter d'avoir une surcharge de fichier à l'écran qui nous perdrait.

3.3. Ajout d'une nouvelle scène

- Ajout d'une nouvelle scène comme l'avaient voulu les créateurs de l'application, à savoir une scène de fin avec de la verdure et des arbres, le tout en restant dans l'esprit de faire les objets à la main, sous Blender.
- Ne pouvant hélas pas tester le code écrit, j'ai pensé préférable de ne pas modifier les codes de téléportation.



3.4. Répartition du travail

	Commentaire et documentation	Ajout de la scène	Réparation du code
Christopher			<input checked="" type="checkbox"/>
Michel	<input checked="" type="checkbox"/>		
Matthieu		<input checked="" type="checkbox"/>	
Thomas			<input checked="" type="checkbox"/>
Alexandre	<input checked="" type="checkbox"/>		

4. Ce que nous aurions aimé faire

L'objectif que l'on s'était finalement fixé mais que nous n'avons pas pu réaliser était de mettre en place totalement la nouvelle scène dans le jeu, à savoir permettre la téléportation vers cette scène une fois le jeu fini.

Nous aurions également voulu compléter les codes que le précédent groupe nous avaient fourni. Cependant, n'ayant pas à disposition des casques VR, nous n'avons pas pu avancer sur les scripts que nous avons à disposition car nous ne pouvions pas tester ces scripts avec les erreurs apparaissant dû au manque du casque VR.

5. Conclusion

Malgré les circonstances, nous avons essayé d'améliorer au mieux le projet que l'on nous avons reçu avec les moyens que nous avons. Avec le matériel à disposition, nous avons mis à jour la compatibilité du code avec une nouvelle version de Unity et nous avons ajouté une scène dans la logique du projet. Puis, nous nous sommes concentrés sur la transmissibilité du projet en ajoutant un maximum de documentation et de commentaire sur les scripts et sur le projet que nous avons reçu.