

Escape Game - Laboratoire de Chimie

Aubane Espérance, Léa Prémont, Hugo Allemand, Thomas Zoude, Maël Lagadeuc, Axel Goris



Table des matières

Introduction	3
Organisation	4
Etat des lieux	5
Travail réalisé	6
Intégration VR	6
Déplacement Leap Motion	6
Interaction Leap Motion	7
UI lié à la main	8
Scénario	9
Intro	10
Effet sonores - musique de fond	12
Cinématique de défaite	12
Menu d'accueil	13
Améliorations futures	15
Bilan	16



<https://github.com/Axelgoris99/Chemistry-Lab/tree/master>

1. Introduction

Dans le cadre du cours de COARV, nous avons repris un projet de jeu d'escape game solo, "Escape Labo", jouable en réalité virtuelle avec l'utilisation d'un Leap Motion.

Le projet fut initié il y a deux ans par des élèves de l'option RV et repris l'année dernière par un autre groupe pour le continuer et le faire évoluer.

Le scénario du jeu est le suivant : dans les années 1850, un virus dangereux a envahi la Terre. Nous allons devoir tout faire pour sauver le monde et échapper à cette mortelle épidémie. Dans ce but, nous nous introduisons discrètement dans le laboratoire d'un savant fou, qui pourrait avoir trouvé le remède mais le garde jalousement pour lui. Heureusement, il a laissé des notes de ses recherches dans son laboratoire. Cependant, nous ne devons pas traîner longtemps dans ce laboratoire car le scientifique est aussi méfiant que fou et fait régulièrement des rondes pour s'assurer que ses secrets restent bien gardés. Et il ne risque pas de se montrer compréhensif s'il vous trouve dans son repaire sans son autorisation...

Pour nous en sortir, nous devons décrypter les notes du scientifique afin de résoudre les énigmes. Ces dernières sont des équations chimiques à résoudre en obtenant les réactions désirées. Nous pourrons ainsi avancer et reconstituer la formule complète du remède. Pour ce faire, nous disposons d'un établi avec différents réactifs contenus dans des fioles et un contenant pour y verser la solution finale. Bon courage pour cette mission...



<https://github.com/Axelgoris99/Chemistry-Lab/tree/master>

2. Organisation

Notre organisation était assez simple :

Nous avons défini un certain nombre de fonctionnalités sur Github Projet, les avons attribuées et chaque personne avait au moins une tâche à accomplir. Une fois celle-ci effectuée, il était possible de continuer avec une autre idée que nous avions.

The screenshot shows a GitHub repository interface with several open pull requests and issues. The issues are listed on the left, and pull requests are listed on the right. Each item includes a title, a number, the author, and a small profile picture.

Issue/Pull Request	Description	Author	Icon
Scénario simple	#3 opened by Llleea	Llleea	Green plus icon
Rajouter des énigmes	#9 opened by Llleea	Llleea	Green plus icon
Menu d'accueil	#16 opened by Axelgoris99	Axelgoris99	Purple puzzle icon
Interaction saisir objet	#2 opened by hugoalle	hugoalle	Purple puzzle icon
Menu pour relancer	#10 opened by Llleea	Llleea	Yellow gear icon
Faire une intro simple	#8 opened by Llleea	Llleea	Green plus icon
Faire une intro simple	#8 opened by Llleea	Llleea	Green plus icon
Gérer la défaite !	#7 opened by Llleea	Llleea	Purple puzzle icon
Effets sonores - musique de fond	#6 opened by Llleea	Llleea	Avatar icon
Déplacement Leap Motion	#11 opened by Llleea	Llleea	Avatar icon
Montée en version	#4 opened by Llleea	Llleea	Avatar icon
Implémentation VR	#1 opened by Axelgoris99	Axelgoris99	Avatar icon

Ainsi, chaque personne a pu réaliser quelque chose. Pour chaque fonctionnalité, nous avons créé une branche différente, chacun travaillait dans son coin puis il mergeait développement sur sa branche, vérifiait que tout fonctionnait et remergeait de sa branche vers développement. Cela nous a permis d'éviter des problèmes lorsque nous avions fork depuis Développement il y a longtemps. A chaque version fonctionnelle avec une véritable fonctionnalité implémentée, nous faisions un merge vers master qui ne contient que des versions fonctionnelles ainsi qu'une release qui est la dernière version à ce jour.



<https://github.com/Axelgoris99/Chemistry-Lab/tree/master>

3. Etat des lieux

Le projet tel qu'on l'a repris n'était pas entièrement fonctionnel. Le groupe précédent ayant eu à travailler dessus pendant le confinement, ils n'ont pas testé le fonctionnement en réalité virtuelle, ni avec le Leap Motion.

La version de Unity utilisée était la 2019.3.0f6.

Le dépôt Git était constitué de plusieurs branches correspondant à plusieurs fonctionnalités que le groupe avait tenté de développer, mais aucune de ces branches n'ont été fusionnées. Certaines comportaient toujours des erreurs.

Au niveau du scénario, le décor et les assets étaient présents. Le déroulement du jeu n'était pas entièrement fixé. Il s'agissait d'une épreuve en temps limité (100s), le temps restant s'affichant sur un tableau à la vue du joueur dans la scène. Un autre tableau comportait une équation de chimie, dont certains éléments manquaient. Le joueur disposait de 3 fioles colorées et d'un grand récipient, mais à ce stade du projet aucune interaction attachée à ces objets ne pouvait faire avancer le scénario. Les fioles n'étaient pas manipulables car l'interaction au Leap Motion n'était pas fonctionnelle.

Une fois l'état des lieux du projet dressé, nous avons dû prendre des décisions quant à la démarche à suivre pour poursuivre le développement de l'escape game. Tout d'abord, notre objectif (fixé avec les encadrants du projet) était d'avoir une première scène entièrement fonctionnelle et jouable, le but étant de pouvoir jouer juste en lançant un exécutable.

On a décidé de repartir de la branche master et d'abandonner les branches non fonctionnelles pour développer nos propres fonctionnalités.

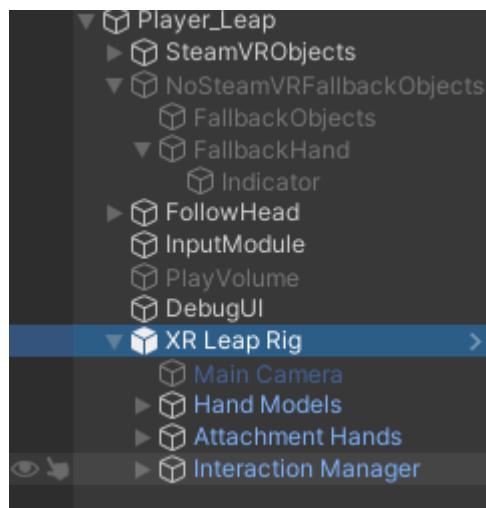


<https://github.com/Axelgoris99/Chemistry-Lab/tree/master>

4. Travail réalisé

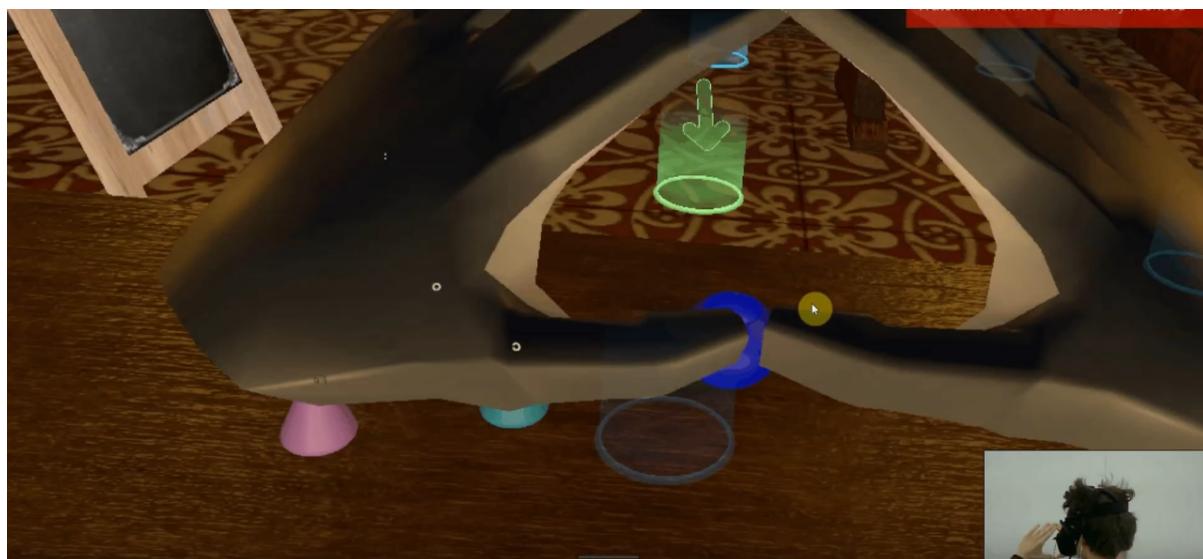
a. Intégration VR

Pour cette partie, nous avons juste eu à mettre à jour la pipeline graphique et les différents composants de Steam VR afin de les rendre compatible avec la version de Unity (2019.4.20f) que nous utilisions. Toutefois, pour le Leap Motion, il a également fallu créer un nouveau préfab prenant en compte la VR et le leap motion. Le Leap Motion est donc un enfant du système qui permet de déplacer la caméra en VR afin de ne pas avoir de problème de main n'apparaissant pas à l'écran mais ailleurs dans la scène.



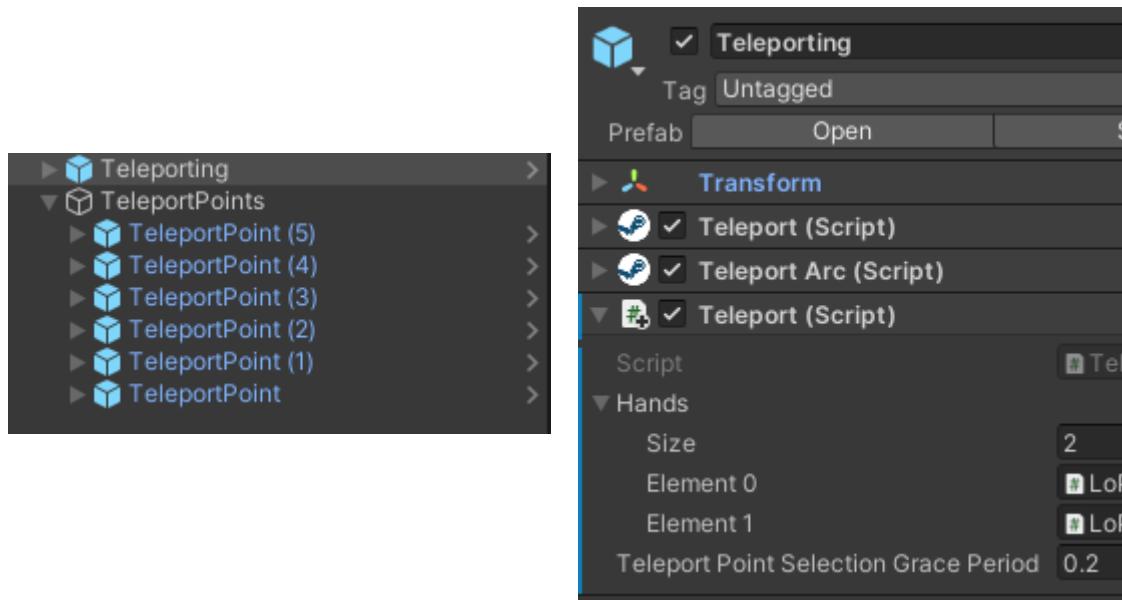
b. Déplacement Leap Motion

Pour cette partie, nous avons regardé comment divers jeux qui utilisent la reconnaissance des mains géraient le déplacement. Nous avons décidé de reprendre un système similaire au jeu Elixir VR, où il faut faire un mouvement de triangle en utilisant le pouce et l'index des deux mains pour sélectionner un des points de téléportation qui sont alors visibles, puis fermer ce triangle pour s'y téléporter.



<https://github.com/Axelgoris99/Chemistry-Lab/tree/master>

Plutôt que de tout refaire et comme SteamVR proposait déjà des prefabs gérant la téléportation, nous avons décidé de nous baser sur ce système. Puisque SteamVR ne gère que les contrôles avec des manettes et pas avec le Leap Motion, nous n'avons repris que les prefabs de TeleportPoint de SteamVR tel quel. Nous avons complété le prefab Teleporting de SteamVR par un script Teleport qui s'interface avec les TeleportPoint de SteamVR et qui se charge de la détection des divers mouvements de mains.



Afin de faciliter les choses pour la suite, nous avons également créé une classe Gesture qui facilite la création de nouveaux mouvements. Il n'y a qu'à redéfinir sa méthode Active qui détermine si le mouvement est actif ou non et on peut savoir si le mouvement vient juste de commencer, est en cours ou vient de s'arrêter.

c. Interaction Leap Motion

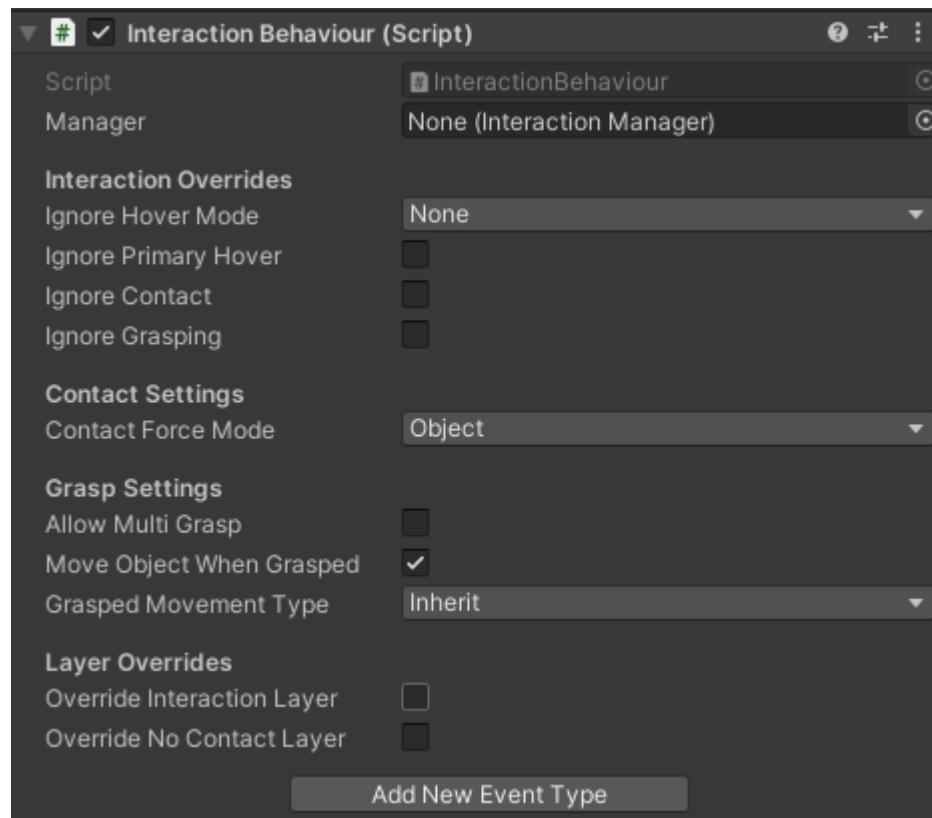
Pour cette partie, nous avons regardé ce que proposait déjà les exemples Leap et en avons déduit que plutôt que d'utiliser une méthode développée en Inter, qui marche certes mais pas forcément de manière fiable, nous allions utiliser les méthodes déjà existantes. Nous avons donc utilisé le SDK du Leap Motion et une fois le Leap mis dans la scène, tous les objets avec lesquels on peut interagir doivent intégrer un "Interaction Behaviour" puisque cela permet de savoir si l'on peut interagir avec eux via le Leap Motion. Il est donc tout simplement possible de venir grab un objet en se rapprochant et en serrant la main autour de la fiole par exemple. Attention toutefois, si jamais la main disparaît, l'objet gardera sa position et au moment où la main réapparaît vu que l'objet est censé se déplacer avec la main, il est fort possible que la force de déplacement appliquée à l'objet soit trop violente et l'éjecte au loin.

Il est tout à fait possible de prendre un objet, de le tourner puis de le jeter par dessus son épaule par exemple. Dans ce cas, au moment où on ouvre la main, l'objet garde son inertie et ceci pour toutes les interactions que nous pourrions désigner ainsi cela permet d'avoir une sensation de réalisme / de véritable physique derrière.

Voici un exemple de prefab avec un interaction behaviour.



<https://github.com/Axelgoris99/Chemistry-Lab/tree/master>



Les mains ressemblent à cela :

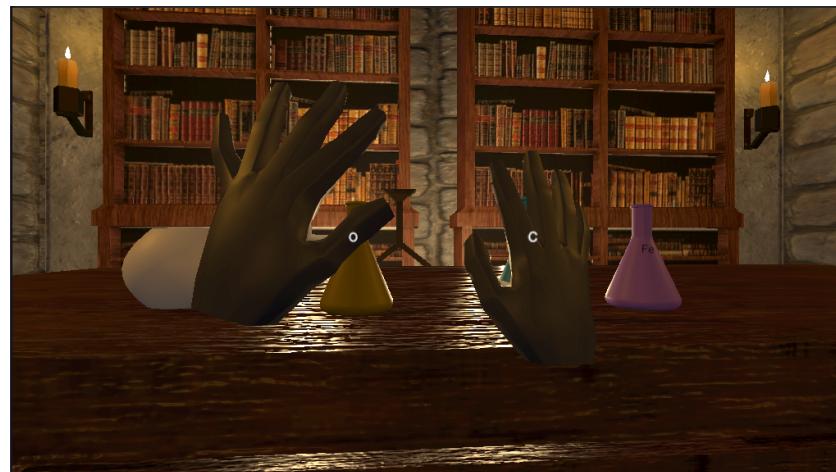


d. UI lié à la main

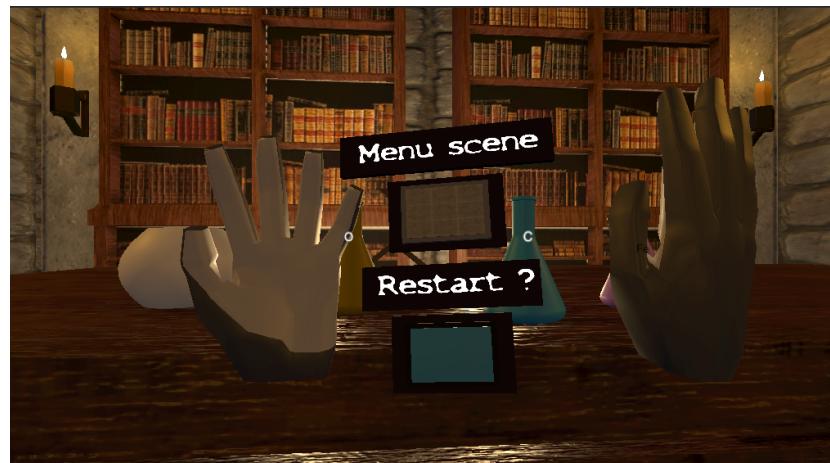
Pour cette partie, nous avons regardé ce qui pouvait déjà exister afin d'être plus efficace et d'avoir un rendu probablement plus propre. Nous nous sommes alors tournés vers une interface utilisateur qui apparaît lorsque l'on retourne la paume de sa main gauche vers soi. Ce n'est pas un mouvement courant donc le menu apparaît bien seulement quand demandé. Lorsque l'on retourne la main, il disparaît.



<https://github.com/Axelgoris99/Chemistry-Lab/tree/master>



Nous avons modifié l'interface déjà proposée pour qu'elle ne laisse apparaître que deux boutons.



Le premier permet de recommencer la scène (“Restart ?”) et donc remet le compteur à 100 mais on perd également toute l'avancée que l'on aurait pu avoir sur l'énigme (puisque le script recommence la scène). Le deuxième bouton (“Menu Scene”) permet de revenir à la scène “Menu” pour recommencer le jeu du début. Les transitions entre les scènes se font avec un fondu pour éviter un changement trop brutal et peu agréable visuellement.

Pour chacun des boutons, nous avons utilisé un script déjà présent dans le préfab qui s'appelle “Interaction button” qui nous permet de déclencher des scripts attachés à d'autres Game Objects. Ici, nous avons utilisé ceux attachés au LevelLoader (dépendant du bouton pressé) qui est parent d'autres Game Objects grâce auxquels une animation permettant d'avoir un fondu a été réalisé.

e. Scénario

Nous avons décidé de retenir l'idée des fioles à verser dans le grand bol. Nous avons modifié le tableau sur lequel était affiché l'équation.

On peut à présent manipuler les fioles à l'aide du Leap Motion; il fallait aussi simuler le fait de verser du liquide avec les fioles. Pour cela, un script “verserLiquide” est



<https://github.com/Axelgoris99/Chemistry-Lab/tree/master>

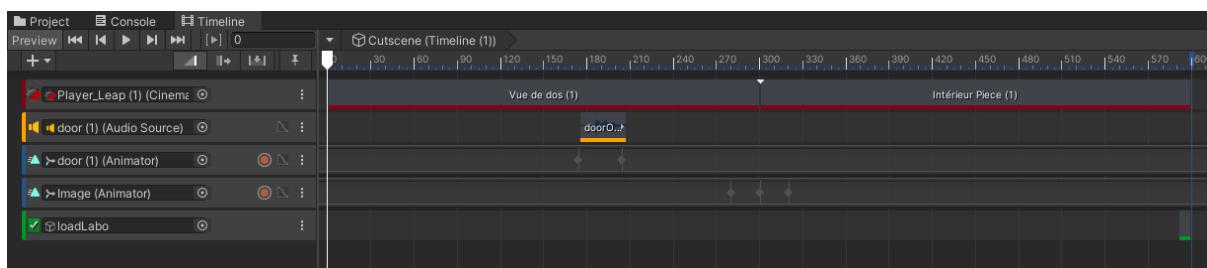
attaché aux fioles : lorsqu'une fiole est inclinée de plus de 90° par rapport à la verticale, des petites sphères colorées en tombent, comme une pluie de petites gouttes.

D'autre part, il fallait détecter lorsque l'on verse du liquide dans le grand bol, afin de pouvoir détecter le succès de l'épreuve. On donc un script "compterGoutte" attaché au contenant qui a pour rôle de détruire les sphères qui touche le contenant tout en incrémentant des compteurs. Ainsi, on peut connaître la quantité de liquide de chaque fiole versé dans le grand bol. Si les gouttes tombent sur le contenant, et non pas dedans, elles sont quand même comptées pour laisser une marge d'erreur à l'utilisateur. En effet, la manipulation précise des fioles au leap motion reste difficile.

Un script "equation1" attaché à la table permet d'instancier les fioles et le contenant sur la table puis vérifie à chaque frame si les compteurs de gouttes valident la condition pour que l'énigme soit résolue. Pour l'instant, cette condition est fixée à 50 gouttes minimum de la flasque bleu et rose et moins de 5 gouttes de la jaune. Ce script modifie aussi le texte sur le tableau pour afficher l'indice.

f. Intro

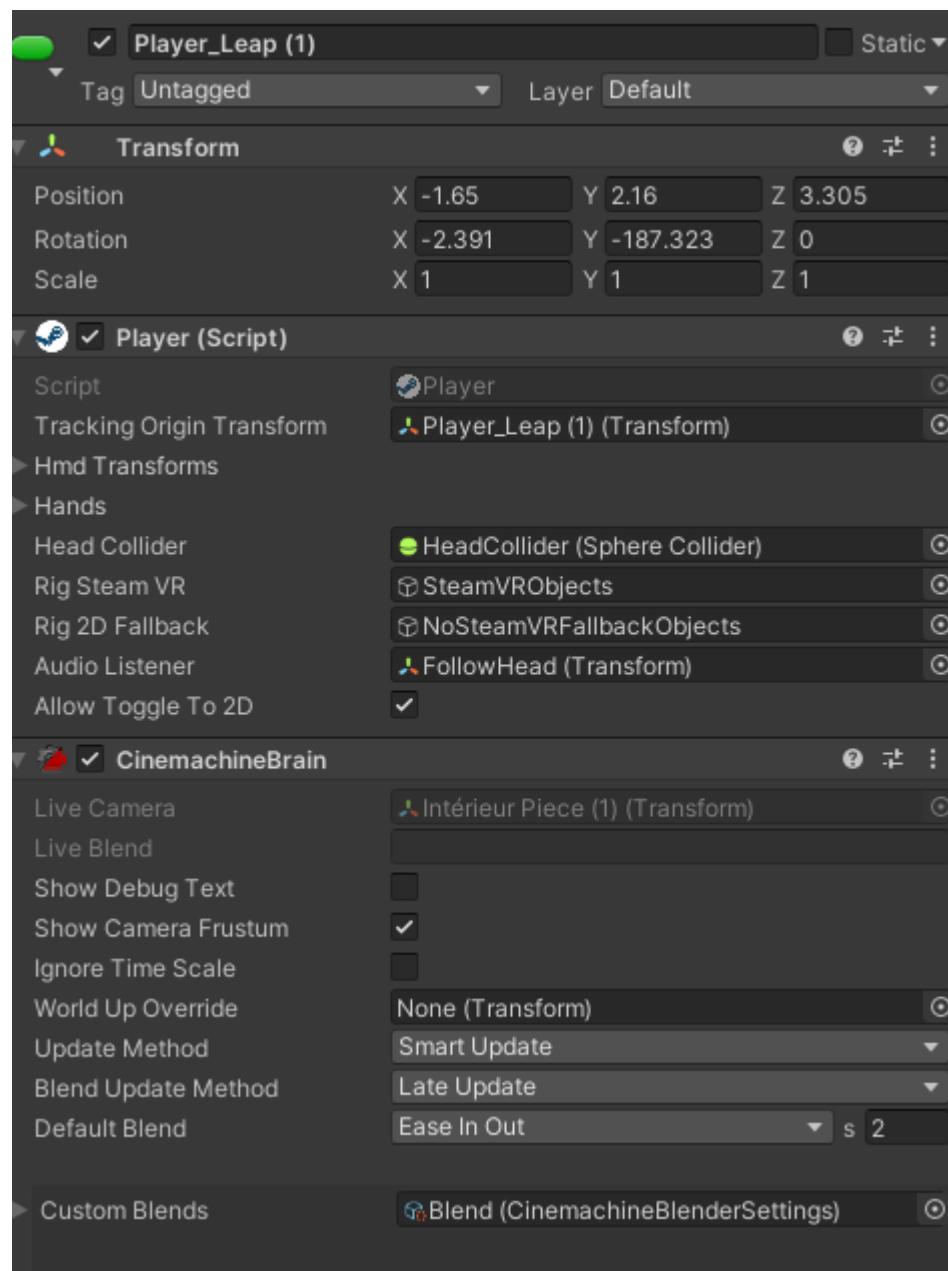
Pour la scène d'intro, nous avons utilisé Cinemachine pour réaliser une véritable cinématique ainsi que la timeline afin de gérer les différentes animations ou encore l'ouverture de la porte. On aurait également pu attacher un script à la porte qui s'ouvre mais nous voulions également apprendre à utiliser la Timeline.



Pour cela, nous avons du faire attention à la VR lors de l'utilisation de caméra virtuelle (qui permettent de se déplacer facilement dans la scène et avoir ce côté cinématique) en attachant donc le composant "Cinemachine Brain" qui permet de gérer les différentes caméra au prefab que nous avons créé pour gérer la VR ainsi que le leap.



<https://github.com/Axelgoris99/Chemistry-Lab/tree/master>



Une fois cela fait, nous avons créé deux caméras virtuelles : Une qui regarde vers la salle où nous serons enfermés et une dans la salle où nous sommes enfermés.

Une animation track est ajoutée à un canva et une image dont la transparence (via un canvas group) est modifiée afin d'avoir un effet de fondu entre les deux plans lors du changement de caméra.

Nous pourrions désactiver les mains mais cela ne change pas grand chose d'avoir ses mains pendant les cinématiques puisque les objets avec lesquels on peut interagir ne sont pas initialisés.



<https://github.com/Axelgoris99/Chemistry-Lab/tree/master>

g. Effet sonores - musique de fond

Dans la scène principale, il n'y avait aucun son. Pour améliorer l'immersion, nous avons décidé de rajouter des effets sonores et musiques de fond.

Tout d'abord, en musique de fond nous avons essayé d'utiliser une bande son déjà présente dans les assets du projet (sélectionnée par le groupe précédent). Nous sommes rapidement revenus sur ce choix compte tenu du caractère trop horrifique de la bande son (le but n'étant pas de choquer les joueurs avec des pleurs d'enfants et autres bruits étranges). Nous avons opté pour une musique moins "intense".

Pour les sons, il y a un fichier crédit dans le dossier Audio. Si un jour, le jeu devait être publié, pensez à les mettre en avant.

Pour accentuer le fait que l'épreuve est à durée limitée, nous avons ajouté le tic-tac d'une horloge au niveau du tableau indiquant le temps restant. Ce tic-tac s'arrête au moment où le compte à rebours tombe à 0.

Enfin, pour ajouter des sons au niveau des interactions et augmenter le réalisme, nous avons attaché un script aux préfabs de verrerie qui déclenche un petit bruit de verre qui s'entrechoque lorsque 2 objets en verre entrent en contact. Pour avoir cet effet, il faut attacher à l'objet le script *GlassEffect* et lui attribuer le layer Glass.

h. Cinématique de défaite

Comme pour la cinématique d'intro, celle-ci a été réalisée en utilisant Cinemachine. De la même manière, nous avons utilisé des caméras virtuelles mais ce coup-ci, nous avons utilisé des fonctionnalités propres à Cinemachine telles que "Look At" ou encore "Follow" ce qui permet de suivre un objet du regard avec un caméra dans un coin par exemple ou bien d'avoir une caméra derrière l'épaule d'un personnage que l'on anime pour qu'il marche. Attention toutefois à bien créer une timeline et pas juste à copier-coller l'objet puisque sinon ce dernier se réfère au même objet "cutscene" qui est créé dans un dossier de notre choix. Sinon, les modifications sur une scène impactent également la deuxième, ce qui n'a rien de désirable.

Pour garder ce côté un peu fou, terrifiant, nous avons décidé d'utiliser le modèle de "Lu" de la cinématique "Adam" sortie en 2016.

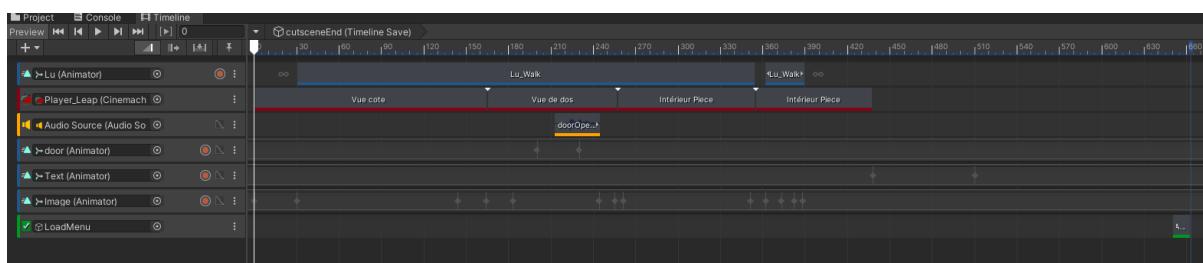


<https://github.com/Axelgoris99/Chemistry-Lab/tree/master>



Dans un premier temps, nous nous sommes dit qu'une vision de la scène d'intro mais depuis le coin, comme si l'on observait en spectateur était intéressant. Puis nous avons décidé d'avoir la caméra qui suit Lu de dos ce qui donne ce côté terrifiant de voir la mort approcher. Finalement, on reprend notre place de joueur et l'on voit Lu s'approcher de nous avant d'avoir un fondu noir, une phrase disant que nous ne sommes pas échappés qui apparaît encore une fois en gérant la transparence et une fois cela fait, on retourne au menu.

Pour essayer d'avoir un véritable côté terrifiant, nous avons décidé de diminuer grandement la luminosité par rapport à ce qu'avaient décidé nos prédécesseurs et nous avons retiré la majorité des sources de lumière dans cette scène ainsi que dans la scène d'intro. De plus, nous avons choisi une musique un peu dérangeante pour cette scène de fin.



i. Menu d'accueil

Le menu n'était pas forcément un objectif à l'origine mais nous nous sommes vite aperçu que cela permettait d'avoir un lieu à l'ambiance neutre dans lequel le joueur pourrait se préparer, passer le casque à l'un de ses camarades ou tout simplement revenir après chaque partie avant d'entamer la suivante jusqu'à finalement réussir à s'échapper. Nous avons tout simplement créé deux morceaux de marbre qui sortent du sol et il y a deux énormes boutons : un pour quitter l'application et un pour lancer la lecture du jeu.



<https://github.com/Axelgoris99/Chemistry-Lab/tree/master>



La téléportation n'a pas été implémentée pour ne pas perdre le joueur dès le début du jeu.



<https://github.com/Axelgoris99/Chemistry-Lab/tree/master>

5. Améliorations futures

Nous avons concentré nos efforts sur la scène principale du projet, le premier niveau.

Maintenant que celui-ci marche, il pourrait être intéressant de réfléchir à la suite de l'escape game, une fois la première épreuve achevée. Peut-être complexifier les prochaines énigmes pour ajouter de la difficulté au fur et à mesure du jeu, et finalement, faire une scène de victoire.

Le déplacement peut être amélioré car il fonctionne bien mais il est effectif même si seulement l'une des mains fait le geste requis par moment, il y a l'air d'y avoir un léger défaut. Il serait également nécessaire d'avoir une explication de comment le déplacement marche pour ne pas avoir de problème à le comprendre.

Il pourrait être bien de réaliser une scène d'introduction un peu plus poussée et donnant un peu plus de contexte.

On pourrait implémenter plus d'éléments dans le décor pour que le joueur puisse découvrir toute la vérité derrière ce lieu par lui-même en se contenant de porter ses yeux partout.

Il faudra probablement changer la version de SteamVR utilisée voire les méthodes utilisées puisque de nouveaux plugins sont sortis et la VR utilisée actuellement ne sera peut-être plus supportée par Unity pendant très longtemps.

Il serait probablement bon de donner un avatar au personnage même si ce dernier n'est qu'accessoire mais cela permettrait lors de la scène de défaite de voir son propre corps lorsque Lu avance vers sa cible. Cela permettrait également de voir des pieds lorsque l'on regarde vers le bas !

Changer la pipeline graphique n'est pas nécessaire mais cela pourrait grandement améliorer le rendu final. A priori, deux ans auparavant, il utilisait le pipeline hdp mais ce dernier s'est avéré incompatible avec un module l'année dernière et nous ne nous sommes pas spécialement intéressés à la partie graphique puisqu'il n'y avait aucune fonctionnalité et aucune possibilité de jeu lorsque nous avons récupéré la scène, nous n'avons pas travaillé de ce côté là mais cela pourrait être intéressant.

Afin de ne plus avoir de problèmes de positionnement, il peut-être intéressant de rajouter une interaction de type grab the air pour se repositionner après s'être téléporté.



<https://github.com/Axelgoris99/Chemistry-Lab/tree/master>

6. Bilan

En conclusion, nous avons récupéré un projet avec plusieurs branches non fonctionnelles en raison de l'impossibilité pour le groupe précédent de tester ses modifications avec le casque ainsi que le Leap Motion. L'abandon de ces branches nous a permis de repartir avec les bases du projet et de faire beaucoup de choses par nous même (tout en nous inspirant de ce qui pouvait déjà exister ailleurs pour gagner en temps et efficacité).

Une bonne partie de notre travail a consisté en implémenter les fonctions essentielles (déplacement, interaction, cinématique et menu d'introduction...) et qu'elles soient toutes fonctionnelles. Ainsi, le prochain groupe qui reprendra le projet pourra s'appuyer sur une base utilisable et poursuivre voire modifier le scénario.

Les objectifs de cette année (i.e. avoir une base de gameplay fonctionnelle et ajouter quelques fonctionnalités et des cinématiques) ont donc bien été respectés et atteints, et le test final s'est avéré probant. Finalement, ce projet s'est trouvé être intéressant, et nous a permis de développer de nouvelles compétences en nous formant sur de nouveaux sujets.

A priori, les fonctionnalités permettant le jeu ont été mises en place, il ne reste plus qu'à y ajouter une bonne histoire ! Bonne chance :)

Pour rappel du lien Github :

<https://github.com/Axelgoris99/Chemistry-Lab/tree/master>



<https://github.com/Axelgoris99/Chemistry-Lab/tree/master>