

# RAPPORT DE STAGE

Nicolas KOHLER

IIM-3ème Année-Jeux Vidéo

2023-2024



**Game Programmer**

22 avril – 18 juin

Tuteur de stage : Kevin Roussel



## REMERCIEMENTS

Je tiens tout d'abord à remercier nos Lead Kevin Roussel, Clémence Gueidan et Elisa Péron pour leur suivi tout au long de ce stage et qui grâce à qui j'ai pu apprendre beaucoup.

Je souhaite également remercier les membres de mon équipe lors de ce stage au sein de l'agence Level Up! Agency. Leur collaboration et leur esprit d'équipe ont grandement contribué au succès de notre projet commun :

- Lucas Guichard
- Nicolas Coulliais
- Eliass Godson
- Melvin Litière
- Morgane Hyun-Perez
- Enzo de Corte
- Flora Denis
- Noé Henriot

Enfin, je tiens à remercier l'ensemble de l'équipe pédagogique de l'IIM et notamment Thomas Nicolet, Jérôme AUDO et Diana Paniah.

# SOMMAIRE

## Table des matières

REMERCIEMENTS .....	2
SOMMAIRE .....	3
INTRODUCTION .....	4
PARTIE 1 : L'ENTREPRISE .....	5
A. L'agence Level up ! .....	5
B. Les clients .....	5
B.1. Les différents clients .....	5
B.2. Notre client .....	6
C. L'équipe .....	6
PARTIE 2 : LES MISSIONS .....	7
A. Présentation de la mission .....	7
A.1. Concept .....	7
A.2. Gameplay .....	7
B. Recherche et développement et préproduction .....	8
B.1. Familiarisation avec la VR .....	8
B.2. Gestion de la Motion Sickness .....	8
B.3. Tests de Performances et Limitations .....	9
C. Production .....	9
C.1. La réalité virtuelle .....	9
C.2. Le multijoueur en ligne .....	13
C.3. Les énigmes .....	16
PARTIE 3 : RELFEXION SUR LE PROJET PERSONNELS ET PROFESSIONNEL .....	20
CONCLUSION .....	21
GLOSSAIRE .....	22
ANNEXES .....	23

## INTRODUCTION

En 2023, j'ai commencé mes études à l'IIM en troisième année de Bachelor. Avant cela, j'avais suivi un parcours plus traditionnel en tant que développeur web/d'applications. Pendant que suivait mes cours de Licence professionnelle en tant que Concepteur développeur d'applications, j'ai réalisé que ce n'était pas la carrière que je souhaitais poursuivre.

Après réflexion, j'ai décidé de me consacrer à ce qui me passionnait vraiment : le développement de jeux vidéo. J'ai appris par moi-même à utiliser Unity, un moteur de jeu, et j'ai approfondi mes connaissances techniques dans ce domaine. Mon entrée à l'IIM a été décisive, car elle m'a permis de solidifier les compétences que j'avais acquises en autodidacte, mais elle m'a surtout offert l'opportunité de travailler au sein d'équipes de développement de jeux vidéo, renforçant ainsi mon expérience pratique et mes capacités de collaboration.

Pendant mon stage chez Level Up! Agency, j'ai été immergé dans un environnement passionnant où l'on exploite les jeux vidéo pour répondre aux besoins de clients variés. Notre principal projet était de créer une expérience en Réalité Virtuelle pour le CNES, mettant en valeur les missions d'Ariane 5. J'ai contribué activement au développement de cette expérience multijoueur en ligne, en travaillant étroitement avec une équipe dynamique.

Dans un premier temps, je présenterais l'organisation Level Up! Agency dans laquelle j'effectue mon stage ainsi que le projet que j'ai effectué avec mon équipe. Ensuite je décrirai du développement et de l'avancement du projet ainsi que ses problématiques. Pour finir, je conclurai partageant mes réflexions sur cette expérience et en évoquant les compétences et connaissances que j'ai acquises.

## PARTIE 1 : L'ENTREPRISE

### A. L'agence Level up !



Contrairement à de nombreuses agences de communication qui utilisent des tactiques conventionnelles telles que les réseaux sociaux et les campagnes à grande échelle, Level Up! Communication se distingue en intégrant le jeu vidéo comme principal vecteur de communication. Cette approche ludique et mémorable permet à l'agence d'atteindre de nouvelles cibles pour ses clients. Level Up! conçoit des jeux vidéo et des expériences numériques sur mesure, ce qui constitue le cœur de son activité. L'agence travaille avec une grande diversité de clients, y compris des lieux culturels et patrimoniaux, des organismes publics et des entreprises du secteur privé, utilisant les technologies les plus récentes pour transmettre des messages de manière innovante.

Chaque année, divers clients sollicitent Level Up! pour des campagnes de communication variées. Par exemple, certains lieux souhaitent être mis en valeur, et rien de mieux qu'une expérience de jeu pour y parvenir. Outre les jeux vidéo, l'agence propose également des jeux de société, des jeux live immersifs et des jeux hybrides, permettant ainsi de toucher un public plus large et de s'adapter aux besoins d'un grand nombre d'entreprises. Chaque projet est personnalisé pour répondre aux besoins spécifiques du client, qu'il s'agisse de campagnes de communication, de sensibilisation ou d'engagement du public.

Le fonctionnement de Level Up! est particulier. Une fois la liste de clients établie, l'agence forme des équipes de stagiaires en Game Design, Game Art et Game Programming, toutes supervisées par des "Leads". Ces équipes ont pour mission de trouver, présenter et développer divers concepts après validation par les clients. Chaque équipe devra ainsi présenter différents concepts pour ensuite développer celui choisi par le client, garantissant que les projets répondent précisément aux attentes et aux objectifs des clients

### B. Les clients

#### B.1. Les différents clients

Lors de mon stage chez Level Up!, l'agence a collaboré avec quatre clients ayant fait un appel d'offres : le Centre National d'Etudes Spatiales (CNES), le Château de Versailles, l'Institut Pasteur et le Grand Palais Immersif. Chaque équipe, composée de 7 à 9 étudiants, travaillait sur des projets aux objectifs spécifiques et diversifiés, permettant d'explorer différents domaines de la conception d'expériences numériques. Par exemple, une équipe était chargée de concevoir une expérience en réalité augmentée pour prolonger une exposition du Grand Palais Immersif, tandis qu'une autre équipe travaillait sur un projet pour l'Institut Pasteur visant à sensibiliser un public plus

jeune à la recherche médicale. Ces projets mettaient en lumière les capacités de l'agence à répondre à des demandes spécifiques et innovantes de divers secteurs.

## B.2. Notre client

Dans le cas de mon équipe, notre client était le CNES. L'appel d'offres portait sur la création d'une expérience en Réalité Virtuelle (VR) destinée à rendre hommage aux principales missions du lanceur maintenant à la retraite, Ariane 5. L'objectif était de capitaliser sur le succès de leur premier jeu "Orbital Dance" et de préparer le terrain pour le premier lancement d'Ariane 6, prévu entre mi-juin et mi-juillet 2024.



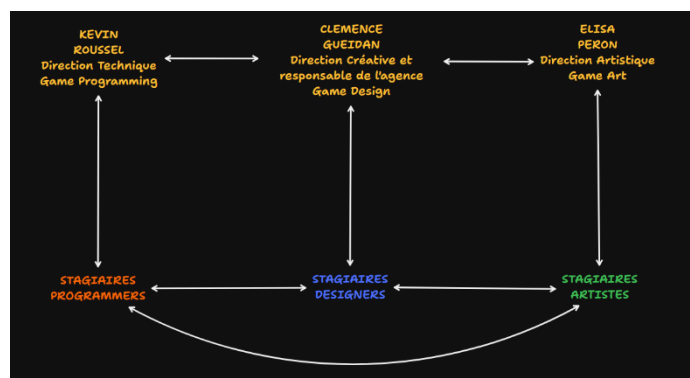
Cette expérience VR devait non seulement éduquer le public, mais aussi les impliquer en leur offrant une immersion dans les missions historiques d'Ariane 5, marquant ainsi une transition vers les futures missions d'Ariane 6.

## C. L'équipe

Mon équipe était constituée de 9 personnes, réparties selon trois spécialités principales :

- Quatre Game Artists (Morgane Huynh-Perez, Enzo de Corte, Melvin Guichard et Flora Denis)
- Trois Game Designers (Eliass Godson, Nicolas Coulliais et Noé Henriot)
- Deux Game Programmers (Lucas Guichard et moi-même).

Chaque corps de métier était supervisé par un Lead : Éliisa Péron pour les artistes, Clémence Gueidan pour les designers et Kevin Roussel pour les programmeurs. Leur rôle était de s'assurer que nous respections les consignes de notre client tout en nous enseignant de nouvelles techniques et méthodologies pour s'assurer du bon déroulement du stage



## PARTIE 2 : LES MISSIONS

### A. Présentation de la mission

#### A.1. Concept

Au début de la production de notre projet de jeu vidéo pour le CNES, il était crucial de définir un concept solide. Après avoir exploré diverses idées et effectué des recherches approfondies sur la réalité virtuelle (VR), nous avons décidé de créer une expérience coopérative multijoueur asymétrique.

Inspirés par des jeux comme "Keep Talking and Nobody Explodes", où deux joueurs doivent collaborer pour désamorcer une bombe en utilisant des rôles distincts, nous avons voulu pousser ce concept encore plus loin. Notre objectif était de développer un jeu où chaque joueur aurait des tâches et des expériences de jeu complètement différentes, reflétant ainsi les différents rôles dans une mission spatiale.

Après plusieurs jours d'idéation et de tests, nous avons conclu que le gameplay en multijoueur était non seulement le plus amusant, mais aussi le plus représentatif de la coopération nécessaire dans les missions spatiales réelles. Nous avons donc décidé de centrer notre jeu sur une coopération asymétrique, où un joueur se trouve dans le lanceur tandis que l'autre se situe dans la base de contrôle, chacun dépendant de l'autre pour mener à bien la mission.

#### A.2. Gameplay

Lors de la conceptualisation, nous avons été inspirés par le jeu "Mission ISS" où les déplacements des joueurs se faisaient en apesanteur grâce à des barres de maintien. Cette mécanique, qui permet de simuler l'absence de gravité en s'agrippant et en se propulsant, nous a semblé essentielle pour offrir une expérience immersive et réaliste au joueur dans le lanceur Ariane 5.

Par contre, pour le joueur dans la base de contrôle, nous avons pris inspiration du jeu "Keep Talking and Nobody Explodes", où l'un manipule une bombe avec les instructions fournies par l'autre joueur, statique avec un manuel. Nous avons voulu intégrer cette dynamique de communication et de coopération dans notre jeu, en adaptant le gameplay aux spécificités de chaque rôle.

#### *Joueur dans le lanceur*

Pour le joueur situé dans le lanceur Ariane 5, nous avons adopté une mécanique de déplacement inspirée de "Mission ISS". En utilisant des barres de maintien disposées dans tout l'intérieur du lanceur, le joueur peut s'agripper et se propulser en avant, simulant ainsi les déplacements en apesanteur. L'impression d'être dans l'espace est primordiale pour l'expérience utilisateur, et ce système de mouvement dynamique contribue grandement à cette immersion.

Toutefois, nous sommes conscients des défis posés par ces déplacements complexes. Pour équilibrer le gameplay, les énigmes assignées à ce joueur ont été conçues pour être plus simples et moins précises. Au lieu de manipuler des objets spécifiques, le joueur dispose de commandes plus générales, réduisant ainsi la nécessité de précision. Cependant, pour compenser cette simplicité, les énigmes requièrent une grande dose de dynamisme et d'agilité, rendant le gameplay à la fois engageant et accessible.

### *Joueur dans la base de contrôle*

Le joueur dans la base de contrôle, également appelé Mission Control, doit gérer un environnement riche en technologie, inspiré des vraies bases de contrôle que nous avons étudiées. Le joueur dans la base de contrôle est entouré de nombreux boutons, écrans et autres dispositifs informatiques, simulant un véritable centre de commande et doit transmettre ces informations efficacement au joueur dans le lanceur.

Le joueur doit manipuler différentes interfaces et contrôles pour résoudre les énigmes et assurer le bon déroulement de la mission. Contrairement au joueur dans le lanceur, celui-ci reste statique mais doit faire preuve de coordination et de communication efficaces. Cette asymétrie dans le gameplay offre une expérience de coopération unique où chaque joueur a des responsabilités et des défis distincts, renforçant la collaboration et l'immersion dans le jeu.

## **B. Recherche et développement et préproduction**

### **B.1. Familiarisation avec la VR**

Pour la R&D, c'était la première fois que nous devions travailler avec des casques de réalité virtuelle. Nous avons commencé par tester les Oculus Quest 3 que nous avions à disposition. Tous les membres de l'équipe ont eu l'occasion de s'habituer à la VR en essayant diverses applications et jeux. Cela nous a permis de nous familiariser avec les spécificités et les possibilités offertes par la réalité virtuelle. Nous avons passé plusieurs jours à explorer différentes expériences VR pour comprendre comment elles fonctionnent et quelles sensations elles procurent aux utilisateurs.

### **B.2. Gestion de la Motion Sickness**

Une question cruciale que nous avons abordée était celle de la "motion sickness" (nausée due aux mouvements en VR). La motion sickness, ou cinétose, est un problème fréquent en réalité virtuelle, causé par des signaux contradictoires entre les yeux et l'oreille interne, responsables de l'équilibre. Nous avons commencé par des



recherches approfondies sur les causes de la motion sickness, notamment la vitesse des mouvements et la nécessité d'un temps d'adaptation pour les nouveaux utilisateurs.

Nous avons testé le jeu "Mission ISS" du Meta Quest Store, qui utilise des déplacements en zéro gravité via des barres de maintien pour se propulser. Après plusieurs sessions de tests, nous avons observé que, bien que certains membres de l'équipe aient ressenti un inconfort initial, la plupart se sont rapidement adaptés. Cette mécanique de déplacement, où les joueurs contrôlent activement leurs mouvements, s'est avérée efficace pour réduire la motion sickness par rapport à d'autres méthodes de déplacement.

Sur la base de ces observations, nous avons décidé d'intégrer une mécanique similaire dans notre jeu pour le joueur dans le lanceur Ariane 5.

En conclusion, tout au cours développement nous avons gardé en mémoire cette problématique pour minimiser les risques d'inconfort pour les joueurs qui ne serait pas habituer à la réalité virtuelle afin de garantir une immersion agréable pour tous les joueurs

### B.3. Tests de Performances et Limitations

Pour évaluer les performances du casque Oculus Quest 3, nos Leads nous ont informés des défis posés par les textures alpha (transparences) en VR. Nous avons donc effectué un stress test pour comprendre les limites du casque. Avec l'aide d'un game artist, nous avons instancié de nombreux objets, ajouté des particules avec de l'alpha, et utilisé plusieurs caméras pour évaluer la charge.

Les résultats ont montré que l'Oculus Quest 3 est très performant et plus avancé que les générations précédentes. Ces tests nous ont permis de définir un scope réaliste pour les game designers, en identifiant clairement les capacités et les limites techniques du matériel.

## C. Production

### C.1. La réalité virtuelle

La réalité virtuelle (VR) est une technologie immersive qui permet aux utilisateurs de se plonger dans un environnement virtuel tridimensionnel créé par ordinateur. En utilisant des casques VR, les utilisateurs peuvent interagir avec ces environnements comme s'ils y étaient réellement, grâce à des capteurs de mouvement et des contrôleurs. Cette technologie a évolué rapidement au cours des dernières années, offrant des expériences de plus en plus réalistes et engageantes. Pour notre projet, la VR

a été une composante essentielle pour simuler l'univers spatial et les missions de l'Ariane 5.

L'un des aspects les plus fascinants de la VR est sa capacité à transporter les utilisateurs dans des mondes complètement différents, qu'il s'agisse de paysages fantastiques, de simulations de formation professionnelle, ou, dans notre cas, d'un lanceur spatial. Les casques VR modernes, comme l'Oculus Quest 3 que nous avons utilisé, intègrent des écrans haute résolution et des systèmes de suivi de mouvement avancés, offrant une immersion visuelle et sensorielle presque totale. Cette immersion permet non seulement de créer des expériences de jeu captivantes, mais aussi de répondre à des objectifs éducatifs et de formation en simulant des environnements complexes de manière sûre et contrôlée.



*Oculus Quest 3*

Après avoir exploré les possibilités offertes par la réalité virtuelle, nous devons comprendre les spécificités du casque et ses performances pour notre projet. En discutant avec notre Lead de programmation, nous avons décidé d'utiliser Unity comme moteur de jeu, car c'était celui avec lequel nous étions le plus à l'aise. Il nous a ensuite orientés vers un package appelé XR Interaction Toolkit, qui inclut de nombreuses interactions prêtes à l'emploi, nous permettant ainsi de gagner du temps dans notre production future. Ce choix a été motivé par notre besoin de rapidité et d'efficacité dans le développement de notre prototype.

Après avoir installé le package dans notre projet Unity, j'ai ouvert une scène de test incluse dans le package pour me familiariser avec les différents éléments déjà présents dans celui-ci. Cette scène préconstruite nous a permis de visualiser comment les interactions VR étaient implémentées, notamment les contrôles de déplacement et d'interaction avec les objets virtuels. J'ai pu observer comment les scripts étaient organisés et comment les différents composants étaient configurés pour assurer une expérience utilisateur fluide et intuitive. Cette première étape m'a aidé à comprendre la

structure de base que nous pouvions utiliser comme point de départ pour notre propre développement.

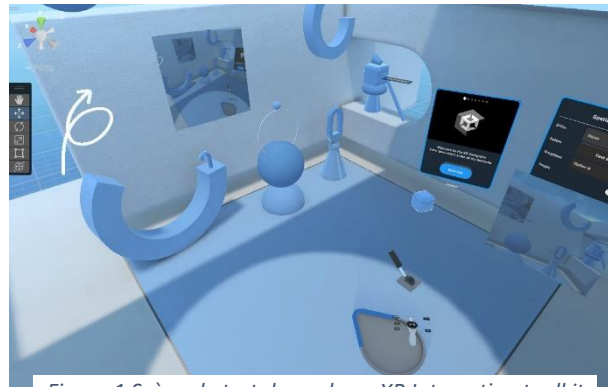


Figure 1 Scène de test du package XR Interaction toolkit

### *Le mouvement du joueur dans la base de contrôle*

Pour les mouvements du joueur dans la base de contrôle, nous avons adopté une approche différente comparée à celle du joueur dans le lanceur Ariane 5. Contrairement à l'environnement dynamique et spatial du lanceur, où les déplacements en zéro gravité sont cruciaux, la base de contrôle (Mission Control) nécessitait une interaction plus statique.

Nous avons opté pour une approche centrée sur l'utilisation du joystick et de la téléportation, des fonctionnalités déjà incluses dans le package que nous avons intégré à notre projet Unity. Ces choix de contrôles visaient à fournir une expérience intuitive et fluide que les joueurs déjà habitués à VR connaissent. Ce sont les deux modes de déplacement les plus utilisés dans les jeux et applications qui fonctionnent avec la VR.

Le joystick utilisé pour contrôler les déplacements du joueur dans la base de contrôle a été choisi pour sa familiarité et son intuitivité, rappelant les joysticks que l'on retrouve sur les manettes de consoles de jeu. Cette décision a été motivée par le désir de rendre l'expérience de jeu aussi accessible et naturelle que possible pour les utilisateurs habitués à ce type de contrôle. Cependant, malgré sa familiarité et son intuitivité, l'utilisation du joystick peut encore entraîner un léger inconfort lié à la motion sickness chez certains utilisateurs, en particulier lors de mouvements prolongés ou rapides dans l'environnement virtuel.

En complément au joystick, la téléportation a été intégrée pour offrir une méthode de déplacement instantané vers des emplacements prédéfinis dans la base de contrôle, elle offre une expérience différente et moins intuitive que le joystick, mais présente l'avantage notable de réduire considérablement les risques de motion sickness. Cette fonctionnalité s'est révélée essentielle pour se déplacer rapidement sans nécessiter de mouvements physiques continus. En se téléportant instantanément d'un point à un autre de la base de contrôle, les utilisateurs évitent ces mêmes mouvements continus qui peuvent parfois provoquer une gêne ou un malaise en réalité virtuelle pour certains.

Le joueur a donc le choix d'utiliser soit les mouvements continus du joystick ou soit la téléportation en fonction de sa préférence et de ses besoins dans la base de contrôle. Cette flexibilité permet aux utilisateurs d'adapter leur méthode de déplacement en fonction de la situation à laquelle ils font face dans le jeu.

### *Le mouvement du joueur dans le lanceur Ariane 5*

Pour le joueur évoluant dans le lanceur d'Ariane 5, nous avons conçu un système de déplacement qui reflète l'environnement complexe et dynamique de l'espace. Inspiré par des simulations de gravité zéro comme celles observées dans des jeux tels que "Mission ISS", nous avons intégré des mécanismes permettant au joueur de se déplacer en utilisant des barres de maintien comme points d'ancrage. Ces éléments offrent au joueur une sensation d'apesanteur et une liberté de mouvement immersive.

Nous souhaitions initialement retirer la possibilité de se déplacer à l'aide du joystick et de se téléporter dans le lanceur afin de contraindre le joueur à utiliser les barres de maintien et ainsi rendre l'expérience plus immersive. Cependant, nous avons rencontré un problème : le joueur risquait de rester bloqué au milieu du lanceur lorsque les barres de maintien étaient hors de portée.

Pour résoudre cette problématique, nous avons décidé de maintenir l'utilisation du joystick pour les déplacements horizontaux. Cette adaptation permet au joueur de se déplacer de manière fluide à travers l'environnement du lanceur, en évitant ainsi les situations où les points d'ancrage nécessaires pourraient être inaccessibles. Cela signifie que même en utilisant le joystick pour les déplacements horizontaux, le joueur doit toujours utiliser les barres de maintien pour se déplacer verticalement dans le lanceur. Ainsi, le joueur peut principalement utiliser les barres de maintien et n'utiliser les mouvements au joystick que lorsqu'il n'a pas pu se projeter dans la trajectoire souhaitée.

Cette approche garantit une expérience immersive où les actions du joueur reflètent les contraintes réelles de la navigation en gravité zéro, tout en offrant une navigation plus libre et intuitive dans l'ensemble de l'environnement virtuel du lanceur d'Ariane 5.

### *Problèmes rencontrés lors de l'implémentation :*

Le package VR que nous utilisons inclut un mur d'escalade permettant au joueur de s'accrocher pour se déplacer. Après plusieurs tests, nous avons jugé que cette fonctionnalité était idéale pour simuler les déplacements en apesanteur dans le lanceur spatial.

Pour simuler correctement les déplacements en apesanteur dans le lanceur, il était nécessaire de désactiver la gravité sur le Rigidbody du joueur et des objets. Bien que cela fonctionnait pour les objets, les déplacements du joueur posaient problème. En effet, le script de déplacement sur le mur d'escalade utilisait directement la position du Transform du joueur, ce qui ne prenait pas en compte la vélocité nécessaire pour simuler l'apesanteur avec le Rigidbody. Étant donné que le script intégré au package était en lecture seule, nous avons dû suivre les conseils de notre Lead programmer et créer un nouveau script. Ce nouveau script conservait le contenu de l'ancien tout en adaptant les mouvements pour utiliser le Rigidbody du joueur.

Ainsi, le nouveau script ressemble à l'ancien mais intègre désormais le Rigidbody du joueur pour gérer correctement la physique du moteur et simuler l'apesanteur dans le lanceur.

## C.2. Le multijoueur en ligne

Lors de notre réunion sur la priorisation et l'ordre de développement des fonctionnalités du jeu, nous avons décidé de donner la priorité à l'implémentation du multijoueur en ligne plutôt qu'aux énigmes. Cette décision s'explique par le fait que la conception et la structure du code diffèrent considérablement entre un mode solo et un mode multijoueur en ligne. Plutôt que de développer d'abord les énigmes en solo et de modifier ensuite le code pour le multijoueur, il était plus efficace de commencer directement à concevoir et développer les fonctionnalités avec le multijoueur à l'esprit.

Le multijoueur en ligne est crucial dans les jeux vidéo modernes, permettant aux joueurs de se connecter et de jouer ensemble en temps réel à travers le monde. Contrairement aux modes solo ou multijoueur local, il nécessite une infrastructure réseau sophistiquée pour synchroniser les actions des joueurs, minimisant la latence pour garantir une expérience fluide. Le système repose sur des serveurs gérant la logique du jeu et les états des joueurs en temps réel, tandis que les clients envoient et reçoivent des données sur les actions des joueurs. Cette complexité dépasse celle du multijoueur local, où la synchronisation est directe, et impose des défis comme la gestion de la latence, la compensation des décalages, et la prévention des tricheries, souvent résolus par des techniques d'interpolation et de prédiction des mouvements.

Dans notre projet, en corrélation avec notre Lead, nous avons décidé d'utiliser Unity Cloud avec les services Relay et Lobby pour gérer la connectivité et les interactions entre les joueurs, ce qui le rend évidemment très facile à implémenter et très stable au sein de Unity en lui-même.

L'estimation initiale de deux jours pour l'implémentation réseau a été rapidement dépassée en raison de la difficulté à faire communiquer les jeux entre eux via le réseau fourni. Après de nombreuses tentatives infructueuses et des discussions approfondies

avec le service informatique, il s'est avéré que le problème provenait du réseau lui-même, et non de notre code ou de notre implémentation. Cette découverte a retardé notre progression d'une semaine. Finalement, nous avons opté pour une solution temporaire en utilisant le partage de connexion cellulaire d'un téléphone d'un membre de l'équipe. Bien que cette approche soit loin d'être idéale et présente des problèmes de stabilité, elle nous a permis de tester notre implémentation réseau dans des conditions extrêmement défavorables. Cela nous a rassurés, sachant que la majorité des joueurs disposeront de connexions bien meilleures que la nôtre.

Je pense que nous nous sommes un peu trop précipités lorsque nous avons fait l'implémentation réseau et lancé les tests de communication. En effet, il aurait été plus judicieux de commencer par faire un test de ping, qui était inclus dans les exemples du package Relay, entre deux instances. Cela nous aurait permis de détecter le problème bien plus tôt et de comprendre que celui-ci ne provenait pas de notre implémentation du code, mais du réseau lui-même.

## UI et lobby



Figure 2 - UI du menu

Les joueurs arrivent dans un lobby (*figure 2*) avec une UI flottante permettant de créer ou de rejoindre un lobby. Le joueur qui crée un lobby (*figure 3*) reçoit un code de la part du service Unity Lobby qu'il doit partager à l'autre joueur pour qu'il le rejoigne. Le joueur qui doit rejoindre le lobby (*figure 4*), appuie sur le bouton *Join a game* qui affiche tous les lobbys disponibles et doit choisir le code qu'il lui a été communiqué. Les joueurs sont ensuite regroupés dans le même lobby et peuvent voir les mouvements de l'autre joueur.

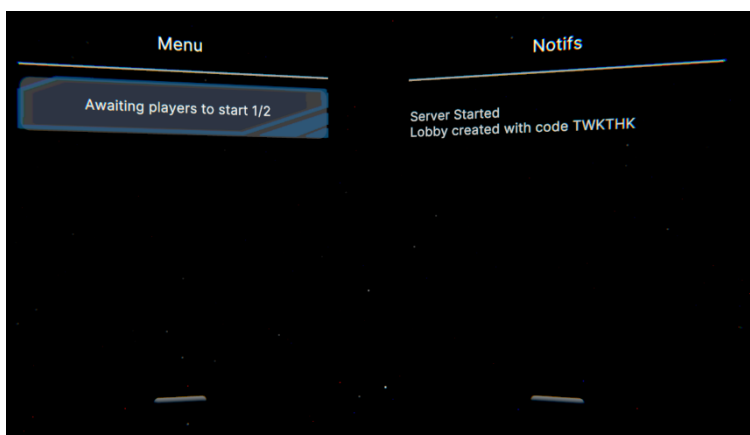


Figure 3 - UI Créer un lobby

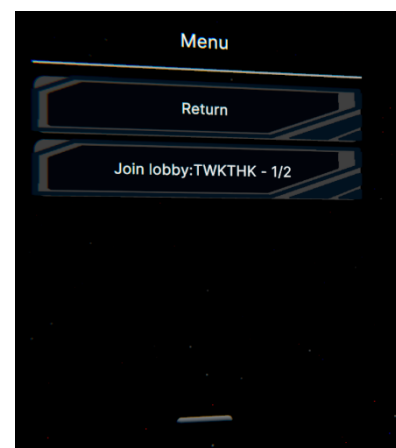


Figure 4 - UI rejoindre un lobby

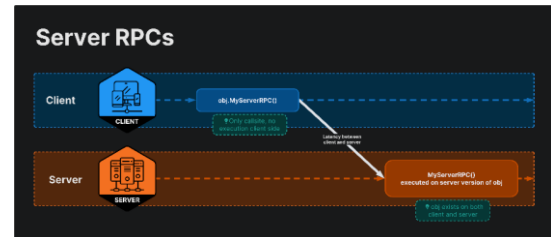
## RPCs

Les RPCs (Remote Procedure Calls) que l'on utilise avec Unity Relay sont essentiels pour synchroniser les actions entre les joueurs. Les RPC permettent à une instance du jeu (un client ou un serveur) d'exécuter des fonctions sur une autre instance, assurant que les deux joueurs vivent le même état de jeu.

### Server RPCs

Les Server RPCs sont invoqués par le client mais exécutés sur le serveur. Ce processus implique :

1. Le client envoie une requête au serveur en utilisant un appel RPC.
2. Le serveur reçoit cette requête et exécute la fonction correspondante.
3. Le serveur met ensuite à jour l'état du jeu et peut potentiellement renvoyer des informations au client.

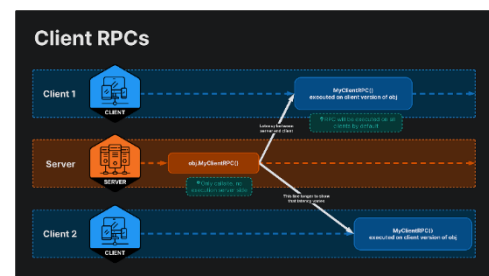


Cette méthode est utilisée pour assurer un contrôle autoritaire par le serveur, réduisant ainsi les risques de triche et maintenant la cohérence dans la logique du jeu.

### Client RPCs

Client RPCs sont invoqués par le serveur et exécutés sur les clients. Cela inclut :

1. Le serveur envoie une commande aux clients via un RPC.
2. Chaque client exécute la fonction, mettant à jour son état de jeu local.



Cette approche garantit que tous les clients ont une vue synchronisée de l'état du jeu.

## NetworkVariables

Les NetworkVariables sont des outils essentiels pour la synchronisation des données entre les clients et le serveur dans un environnement de jeu multijoueur. Elles permettent de stocker et de synchroniser des valeurs de manière efficace et automatique à travers le réseau, assurant que tous les participants voient et interagissent avec les mêmes données en temps réel. Ces variables peuvent être de différents types, tels que des entiers, des flottants ou des chaînes de caractères, et sont conçues pour gérer les changements d'état fréquents, comme les positions des joueurs ou les scores. L'utilisation des NetworkVariables simplifie considérablement le processus de développement multijoueur en éliminant le besoin de coder

manuellement les routines de synchronisation, réduisant ainsi les erreurs potentielles et le temps de développement.

Dans notre projet, l'utilisation des NetworkVariables a été particulièrement avantageuse puisque les joueurs ne se voyaient pas en jeu. Cela simplifiait la tâche de synchronisation, car nous n'avions pas besoin de synchroniser chaque objet en temps réel. Au lieu de cela, il suffisait de transmettre uniquement l'information sur l'état d'activation des objets, réduisant ainsi la complexité et les ressources nécessaires pour maintenir la cohérence du jeu.

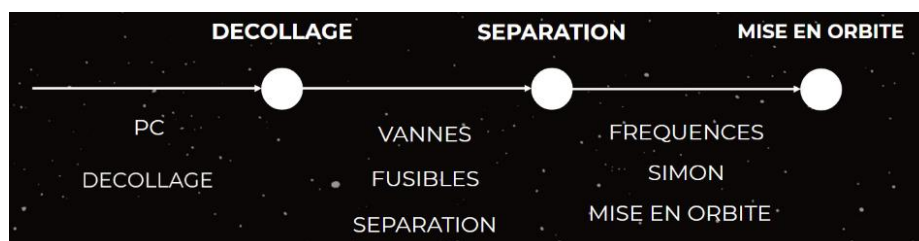
## Analyse

Le fait de prioriser le multijoueur nous a donné un avantage significatif en permettant une optimisation du développement et une intégration fluide du multijoueur en ligne tout au long du processus de création du jeu. Cependant, nous avons également constaté un inconvénient majeur : en repoussant le développement des énigmes à une étape ultérieure, les Game Designers ont reçu ces éléments plus tardivement. Cela a posé des problèmes lors des phases de test et de validation, nécessitant des ajustements et des améliorations de certaines énigmes vers la fin de la production.

### C.3. Les énigmes

Pour ce qui en est de la boucle de jeu principale, nous savions que nous voulions baser cela sur des énigmes que les joueurs devront réaliser en coopérant. Nous avons essayé d'équilibrer les énigmes de sorte que chaque joueur doive communiquer des informations pour débloquer l'autre joueur et en recevoir de façon égale.

Nous avons fait 8 énigmes séparés en 3 parties. Personnellement j'ai développé 5 énigmes qui sont les suivantes : décollage, séparation, fréquences, simon says et la mise en orbite.





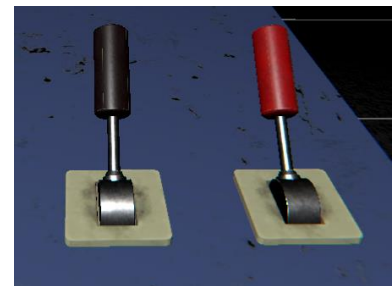
## Décollage

Pour faire décoller le lanceur une alerte va signaler aux joueurs que le joueur dans le lanceur va devoir s'attacher. Une fois attaché, un compte à rebours va se lancer, lorsqu'il atteint 0, un gros bouton rouge va se débloquent permettant au joueur dans la base de contrôle de démarrer le décollage du lanceur.



## Séparation et mise en orbite

Pour la phase de séparation et mise en orbite, ces deux énigmes utilisent la même logique qui est : les deux joueurs auront des leviers qu'ils devront tirer en même temps. Le levier noir est celui pour la séparation et le rouge celui de la mise en orbite. Les joueurs peuvent attraper ces leviers et le baisser mais s'ils les relâchent, ces leviers reviendront à leur position initiale. Les joueurs doivent donc maintenir les leviers vers le bas en même temps pour réussir l'énigme.



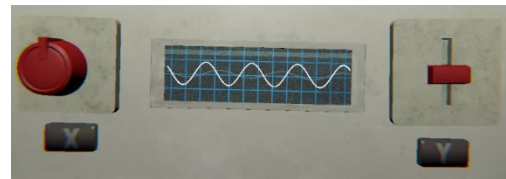
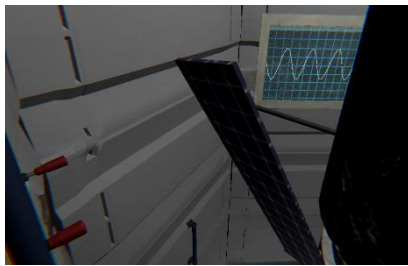
Les leviers étaient intégrés dans le package de la VR et j'ai utilisé les Events disponible pour récupérer l'état du levier dans un NetworkVariable qui contient un boolean. Donc à chaque fois qu'un levier a été activé ou désactivé, le server vérifie l'états des leviers. Si les deux NetworkVariables sont true, alors les deux joueurs ont baissé les deux leviers en même temps et validé l'énigme.

## Fréquences

Les deux joueurs vont trouver des bandes de fréquences ainsi que des outils liés à ces bandes. Ces outils vont changer les valeurs des différentes fréquences mais la subtilité est qu'ils vont contrôler la bande qu'ils ne voient pas. Ils vont donc devoir coopérer pour trouver la bonne valeur l'un de l'autre. Lorsque la fréquence approche de la valeur souhaitée, la courbe devient verte.

Les fréquences utilisent 2 scripts **Frequency** qui gère chaque fréquence et sont validé par un script appelé **Fréquences** qui vérifie que les valeurs sont les bonnes pour finir l'énigme. Le script **Frequency** utilise des NetworkVariables avec des float pour les

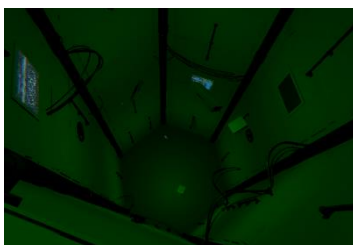
valeurs de la courbe et un autre NetworkVariable en boolean pour vérifier si la fréquence est à la valeur souhaitée.



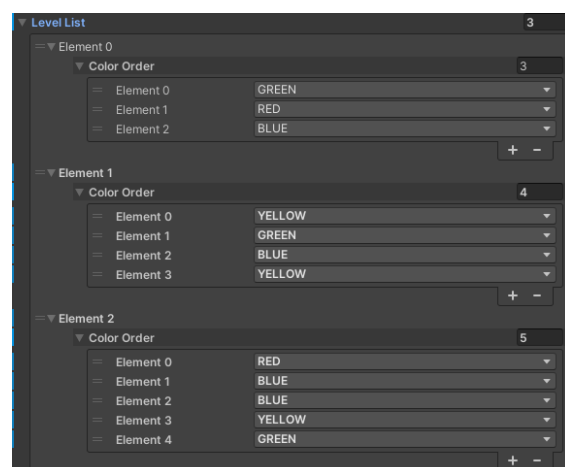
*Simon*

Une alarme de différentes couleurs va s'afficher dans le lanceur. Pour couper cette alarme d'urgence, le joueur dans la base de contrôle va devoir appuyer sur des boutons de couleurs. Pour connaître l'ordre d'exécution, les joueurs vont devoir communiquer puisqu'il sera donné grâce aux couleurs des lumières du lanceur. Lorsque les joueurs entrent les bonnes séries, ils passent à la prochaine étape.

Lorsque le joueur dans la base de contrôle appuie sur les boutons de couleur, cela enregistre l'ordre localement et lorsque le nombre nécessaire à la séquence est atteint, cela envoie l'ordre des boutons appuyé au serveur pour qu'il les valide.

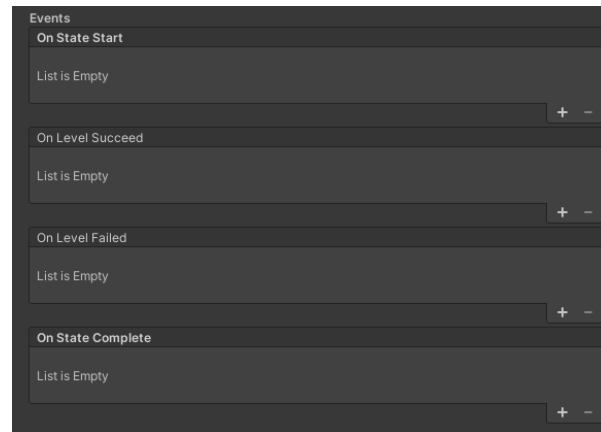


Nous devons ajouter l'ordre des couleurs directement dans l'éditeur d'Unity et cela permettait aux Game Designer de pouvoir changer l'ordre des couleurs ou ajouter une séquence lorsqu'ils voulaient.



## Events

Pour chaque énigme, j'ai ajouté des Events OnStateStart et OnStateComplete, ainsi que des Events spécifiques à chaque énigme. Cela permet à nos Game Designers d'afficher, d'activer ou de faire des changements à des moments clés des énigmes. Comme expliqué dans la partie sur les RPCs en multijoueur ci-dessus, ces Events doivent être déclenchés dans un Client RPC pour qu'ils s'activent pour tous les clients.



*Events de l'énigme Simon*

## PARTIE 3 : RELFEXION SUR LE PROJET PERSONNELS ET PROFESSIONNEL

### Compétences et bénéfices techniques

Ce stage a été une véritable immersion dans des technologies innovantes que je n'aurais pas pu explorer autrement, notamment la réalité virtuelle. Travailler avec des équipements spécifiques tels que les casques de réalité virtuelle a été une opportunité unique, nécessitant une approche technique et une adaptation que je n'avais pas rencontrées auparavant. Cette expérience m'a permis de comprendre les défis et les exigences techniques associés à la création de contenu en réalité virtuelle, enrichissant ainsi considérablement mes compétences techniques.

De plus, j'ai pu approfondir mes connaissances en matière de développement de jeux en multijoueur en ligne, une compétence que j'avais déjà amorcée lors d'un projet personnel antérieur. Travailler sur un projet professionnel a amplifié ma compréhension de la complexité de la gestion des interactions réseau et des synchronisations entre les joueurs. La conception et la structure du code pour le multijoueur sont notablement différentes de celles du mode solo ou du multijoueur local, ce qui m'a poussé à adopter une approche plus rigoureuse et méthodique dans ma programmation. Ces expériences m'ont non seulement permis d'élargir mes compétences techniques, mais aussi d'appréhender de manière concrète les exigences et les dynamiques de travail dans le développement de jeux vidéo professionnels.

### Apports professionnels

Ce stage a été particulièrement enrichissant sur le plan professionnel, m'offrant l'opportunité d'acquérir et de renforcer des compétences essentielles. Il m'a permis de développer une plus grande autonomie ainsi que travailler sur un projet destiné à un client, ce qui représentait une expérience différente de mes précédents projets académiques et personnels.

## CONCLUSION

En résumé, ce stage chez Level Up! Agency a été une expérience transformative à plusieurs égards pour moi. Sur le plan technique, j'ai eu l'occasion d'explorer des technologies avancées telles que la réalité virtuelle et le multijoueur en ligne. La découverte de la réalité virtuelle m'a permis de comprendre ses défis uniques et d'adapter mes compétences de développement pour répondre à ses exigences spécifiques.

D'un point de vue professionnel, ce stage m'a offert une véritable immersion dans la réalité du travail pour un client réel, ce qui a renforcé ma capacité à travailler de manière autonome et à gérer les attentes des clients tout en maintenant des standards élevés de qualité. Cette expérience m'a également permis d'apprécier l'importance de la communication et de la collaboration au sein d'une équipe multidisciplinaire.

Enfin, sur le plan personnel, ce stage a été enrichissant en termes de développement personnel. J'ai appris à naviguer dans un environnement professionnel dynamique, à surmonter des défis techniques et à cultiver ma passion pour le développement de jeux vidéo. Ces apprentissages sont précieux pour mon avenir professionnel, me préparant à relever de nouveaux défis et à saisir de nouvelles opportunités dans l'industrie du jeu vidéo.

## GLOSSAIRE

CNES (Centre National d'Études Spatiales): Agence spatiale française responsable des programmes spatiaux civils et de la recherche en astronautique.

Keep Talking and Nobody Explodes: Jeu vidéo de coopération où un joueur désamorce une bombe pendant que l'autre fournit des instructions à partir d'un manuel.

Alpha: En infographie, cela désigne la transparence d'une texture ou d'une couleur.

Stress Test: Test permettant de mesurer la capacité d'un système à fonctionner sous une charge maximale pour identifier ses limites et points de défaillance.

Scope: Ensemble des fonctionnalités et des contenus prévus dans le projet, défini pour guider le développement et éviter les dérives.

Instancier: En programmation, créer une instance (un exemplaire) d'un objet défini par une classe.

GameObjects: Éléments fondamentaux dans Unity, représentant tous les objets présents dans une scène de jeu.

## ANNEXES

Technologies, Unity. « Unity Cloud ».

<https://unity.com/fr/products/unity-cloud>

Unity RPCs :

<https://docs-multiplayer.unity3d.com/netcode/current/advanced-topics/message-system/rpc/>

Unity NetworkVariables :

<https://docs-multiplayer.unity3d.com/netcode/current/basics/networkvariable>

XR Interaction Toolkit :

<https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@3.0/manual/index.html>