

Master of Science HES-SO in Engineering

Orientation : Technologies de l'information et de la communication
(TIC) ou bien orientation Technologies industrielles (TIN)

Énigmes digitales en AR pour la formation en milieu hospitalier

Fait par

Quentin FORESTIER

Sous la direction de

Prof. Yassin Aziz REKIK

Dans le laboratoire/institut de/ à l'école HEPIA

Stéphane MALANDAIN, Expert principal

Lieu, HES-SO//Master, 2023

Accepté par la HES-SO//Master (Suisse, Lausanne) sur proposition de

Prof. Yassin Aziz REKIK, conseiller du projet d'approfondissement
Stéphane MALANDAIN, Expert principal

Lausanne, le 2 juin 2023

Prof. Yassin Aziz REKIK
Conseiller

Table des matières

Table des matières	iii
Abréviations	v
Résumé	vi
1. Introduction	1
1.1. Cahier des charges	1
1.1.1. Évolution du projet en travail de Master	1
1.2. Planification du projet	1
1.2.1. Écriture du scénario	1
1.2.2. Mécanisme des énigmes	1
1.2.3. Création du monde virtuel	1
1.2.4. Association des énigmes avec le monde virtuel	1
1.2.5. Indices et événements	2
1.3. Organisation du projet	2
2. Analyse de l'existant	3
2.1. Application de modération	3
2.2. Application HoloLens 2	3
2.3. Points importants	3
2.3.1. Surcouche Serveur TCP	3
2.3.2. Emplacement du scénario, indices et perturbateurs	4
2.3.3. Ancrage du monde virtuel	4
2.4. Limitations du projet actuel	5
2.4.1. Mono scénario	5
2.4.2. Manque de scénario fil rouge entre les énigmes	5
2.4.3. Surcouche Host TCP	5
2.4.4. Inscriptions des méthodes RPC	5
3. Description du nouveau scénario	6
3.1. Contexte	6
3.2. Participants	6
3.3. Étapes et énigmes	6
3.3.1. Étape hémorragie	6
3.3.2. Étape Message codé	7
3.3.3. Étape opération	7
3.4. Indices	9
3.5. Perturbateurs	10
4. Technologies	11
4.1. Technologies utilisées dans le projet existant	11
4.2. Technologies rajoutées	11

4.2.1.	Microsoft QR	11
4.2.2.	Vuforia	11
5.	Développement du projet	13
5.1.	Compréhension de l'existant	13
5.2.	Nouvelles mécaniques de jeu	13
5.2.1.	Affiche d'objet sur QR Code	13
5.2.2.	Synchronisation accrue entre les HoloLens, donc plus de collaboration	13
5.3.	Conception du monde virtuel	14
5.4.	Conception des énigmes du scénario	14
5.4.1.	Différentes phases d'une énigme	14
5.4.2.	Prise en compte des éléments réels	16
5.5.	Communication réseau	16
5.5.1.	Utilisation de HostNetworkManager	16
5.5.2.	Synchronisation des HoloLens	17
5.6.	Utilisation des HoloLens 2	17
5.6.1.	Déploiement	17
6.	Problèmes rencontrés	19
6.1.	Communication réseau avec les HoloLens	19
6.2.	Vidéo problématique	20
6.3.	Crash dû à un GameObject	20
6.4.	Détection de QR Code	21
6.5.	Détection d'images	21
6.6.	Difficulté de tester	21
7.	Conclusion	22
7.1.	Rappel de l'objectif	22
7.2.	Résultat obtenu	22
7.3.	Conclusion personnelle	22
7.4.	Améliorations et perspectives	22
8.	Références	23
9.	Annexes	24

Abréviations

RPC	Remote Procedure Call
TCP	Transfer Control Protocol
MRTK	Mixed Reality Tool Kit
UWPO	Universal Windows Platform
IL2CPP	Intermediate Language To C++
JSON	JavaScript Object Notation

Résumé

Le projet HOST vise à créer une simulation d'accouchement d'urgence à bord d'un avion, où les participants doivent résoudre des énigmes en réalité augmentée (AR) pour mener à bien l'accouchement. Il a été développé avec Unity et comprend une application pour les HoloLens ainsi qu'une application de modération.

L'objectif de ce projet d'approfondissement est d'étendre les fonctionnalités d'HOST en ajoutant notamment un nouveau scénario et de nouvelles énigmes. Étant donné que le projet doit être réalisé dans un délai court, il est nécessaire de réutiliser les mécanismes déjà en place afin d'accélérer le développement.

De plus, il est demandé de comprendre et de se familiariser avec tous les aspects du projet, tant au niveau de l'application HoloLens que de l'application de modération. L'objectif final est d'acquérir une connaissance globale du projet afin de pouvoir poursuivre ce travail en tant que projet de Master.

Cela nécessite donc de comprendre les principaux mécanismes mis en place pour la réalité augmentée, de comprendre le fonctionnement de la communication réseau et d'analyser la structure globale du projet.

Key Words: HoloLens 2, Énigme, Simulation, Formation, Réalité augmentée, Jeux, Unity, Médical

1. Introduction

1.1. Cahier des charges

Ce projet vise la conception et le développement d'une bibliothèque d'énigmes digitales en réalité augmentée qui seront intégrés à une solution de formation existante (HOST). La solution HOST permet de simuler une situation d'accouchement en urgence dans un avion, et exige de la part des participants de résoudre des énigmes AR pour réaliser l'accouchement. Les énigmes sont souvent combinées : avec des indices digitaux et d'autres réels. Ceci afin d'encourager le travail en équipe. L'objectif est de développer de nouvelles énigmes et de les intégrer à la solution existante.

Techniquement, la solution existante est une application Unity fonctionnant sur un PC pour le client monitoring et supervision et sur HoloLens 2 pour la partie participante. Les énigmes développées doivent respecter ces choix technologiques.

1.1.1. Évolution du projet en travail de Master

Pendant le projet, il a été envisagé de poursuivre avec un travail de Master, ce qui a entraîné une légère modification des objectifs. L'objectif principal est désormais d'explorer tous les mécanismes du projet actuel afin de les comprendre et de les rendre plus génériques. Ceci permettrait notamment de créer une "boîte à énigmes" et de simplifier la création de nouveaux scénarios, car l'approche actuelle est très limitée à cet égard.

1.2. Planification du projet

1.2.1. Écriture du scénario

Le scénario a été développé à partir d'un scénario préexistant, qui avait été conçu pour une expérience unique et était entièrement basé sur des éléments réels. Certaines énigmes ont dû être adaptées, voire modifiées, afin qu'elles aient du sens dans un contexte de réalité augmentée.

1.2.2. Mécanisme des énigmes

Certaines énigmes nécessitent une interaction avec les utilisateurs. Ainsi, une étape a été entreprise pour explorer des solutions visant à créer des énigmes ergonomiques en lien avec le scénario. Il s'agissait d'une phase de "bac à sable" permettant d'évaluer rapidement la viabilité des énigmes et de déterminer s'il était nécessaire de les modifier.

1.2.3. Création du monde virtuel

Cette phase a contribué à conférer une dimension immersive à la simulation. Étant donné que l'objectif de la réalité augmentée est d'immerger le joueur dans le jeu, il était nécessaire de rendre ce monde à la fois attrayant et motivant. Cela impliquait la mise en place des décors ainsi que la création d'une ambiance globale en accord avec le scénario.

1.2.4. Association des énigmes avec le monde virtuel

Étant donné que les différentes énigmes ont été créées indépendamment du contexte, il a été nécessaire de les intégrer dans les différents décors. L'objectif était de rendre les mécanismes aussi naturels que possible, de manière à ce que le joueur puisse les remarquer sans compromettre l'immersion.

1.2.5. Indices et événements

La dernière étape consistait à créer des indices et des événements perturbateurs. Les indices devaient être progressifs, c'est-à-dire que le premier indice d'une énigme devait fournir une piste, tandis que le dernier devait presque révéler la réponse. C'était un aspect crucial, car il était difficile d'évaluer à quel point un indice pouvait être évident ou non.

1.3. Organisation du projet

Le code du projet Unity est disponible sur GitHub :

<https://github.com/Host-Project/HostProject>

Pour un souci de simplification, il a été choisi d'utiliser un mono repository. Cependant, il serait intéressant de créer un dossier GitHub distinct par application, d'autant plus que les kanbans GitHub permettent désormais de suivre plusieurs dossiers.

Le suivi du projet est effectué à l'aide d'un kanban et d'issues GitHub. Un Milestone est disponible pour le projet d'approfondissement, ainsi que pour le travail de Master. Cela permet déjà de réfléchir à des idées d'améliorations spécifiques au travail de Master, tout en les séparant de ce qui doit être réalisé pour le projet d'approfondissement.

Le dossier GitHub est accessible via une organisation. Cela signifie que si de nouveaux collaborateurs doivent travailler sur le projet, il suffit de les ajouter à l'organisation avec les droits appropriés.

2. Analyse de l'existant

2.1. Application de modération

L'application Windows développée pour la modération du jeu offre plusieurs fonctionnalités importantes. Tout d'abord, elle permet d'envoyer des messages personnalisés aux participants, ce qui permet d'interagir avec eux et de leur fournir des informations ou des instructions spécifiques. Elle offre également la possibilité d'introduire des perturbations dans le jeu pour rendre l'expérience plus stimulante et immersive.

L'application permet également de fournir des indices aux participants au besoin, afin de les aider à progresser dans les énigmes ou à surmonter des obstacles. Les modérateurs peuvent envoyer ces indices de manière ciblée aux joueurs qui en ont besoin.

Une fonctionnalité clé de l'application est la visualisation du flux vidéo en direct provenant des HoloLens portés par les participants. Cela permet aux modérateurs de voir ce que les joueurs sont en train de faire, de suivre leur progression et de comprendre leur point de vue. Ces vidéos sont également enregistrées, accompagnées de commentaires, ce qui permet de les consulter ultérieurement lors du débriefing après la session.

Pour faciliter le débriefing, l'application génère un fichier PDF qui associe les paires d'images capturées à partir des vidéos avec les commentaires correspondants. Cela permet une analyse plus détaillée de l'expérience de jeu et facilite la discussion et l'évaluation post-session.

Enfin, l'application agit en tant que serveur central auquel les HoloLens se connectent. Elle gère les sessions de jeu et les différentes étapes, assurant ainsi la synchronisation entre les participants et le bon déroulement de l'expérience.

2.2. Application HoloLens 2

L'application HoloLens 2 est la plateforme sur laquelle le scénario du jeu se déroule. Elle se connecte au serveur, c'est-à-dire à l'application de modération, pour échanger des informations et assurer la synchronisation avec les autres participants.

L'application HoloLens 2 est responsable de l'affichage du Game World, c'est-à-dire l'environnement virtuel dans lequel les joueurs évoluent. Elle permet aux participants de voir et d'interagir avec les différents objets et éléments du jeu.

L'application HoloLens 2 gère également les interactions physiques avec les objets virtuels. Elle utilise les capteurs et les fonctionnalités des HoloLens 2 pour détecter les mouvements et les gestes des joueurs, leur permettant d'effectuer des actions telles que saisir, déplacer ou manipuler des objets virtuels.

Pendant le déroulement du jeu, l'application HoloLens 2 communique avec le serveur pour informer de la progression des énigmes. Lorsque les participants résolvent une énigme, l'application envoie un signal au serveur pour indiquer que l'étape est terminée, ce qui peut déclencher la disponibilité de nouvelles informations ou énigmes.

En résumé, l'application HoloLens 2 est l'interface immersive sur laquelle se déroule le scénario du jeu. Elle assure la communication avec le serveur de modération, affiche le Game World, gère les interactions avec les objets virtuels et informe le serveur de la progression des énigmes.

2.3. Points importants

2.3.1. Surcouche Serveur TCP

Le projet Host fournit des classes permettant la gestion des connexions TCP et la mise en œuvre de la communication par procédure distante (RPC) entre les clients et les serveurs.

En utilisant ces classes, les utilisateurs peuvent créer des connexions TCP entre les différents clients et le serveur, ce qui permet d'établir un réseau de communication. Ils ont également la possibilité d'inscrire leurs propres méthodes en RPC, c'est-à-dire des méthodes qu'ils souhaitent rendre accessibles aux autres participants du réseau.

L'inscription de vos propres méthodes en RPC signifie que vous déclarez ces méthodes de manière à ce qu'elles puissent être appelées à distance par d'autres clients ou serveurs connectés au réseau. Cela facilite la communication et la collaboration entre les participants, car ils peuvent invoquer les méthodes des autres participants pour échanger des informations, partager des données ou coordonner des actions.

En résumé, les classes de gestion de connexion TCP du projet Host vous permettent de créer des connexions entre les clients et les serveurs, et d'établir des appels de méthodes à distance grâce au mécanisme RPC. Cela offre une communication efficace et une collaboration fluide entre les différents participants du réseau.

2.3.2. Emplacement du scénario, indices et perturbateurs

Le scénario de base, ainsi que les événements perturbateurs, sont stockés dans une base de données. Les indices, quant à eux, sont également présents dans la base de données, mais ils ne sont pas utilisés directement. La véritable implémentation des indices se trouve dans un script où ils sont codés en dur. Ce script est accessible depuis l'interface de modération, permettant aux modérateurs d'accéder aux indices de manière pratique et rapide.

```
Enigmes = new List<string>();
Enigmes.Add("Garrot");
Enigmes.Add("Message crypté");
Enigmes.Add("Monitoring");
Enigmes.Add("Seringues");
Enigmes.Add("Calcul mental - Tableau périodique élément");

Helps = new List<List<HelpCommand>>();

var garrot = new List<HelpCommand>();

garrot.Add(new HelpCommand() { CommandName = "SendText", Text = "Code non sécurisé", TextParameter = "Le code n'est pas sécurisé." });
garrot.Add(new HelpCommand() { CommandName = "SendText", Text = "Numéro de la salle", TextParameter = "Le numéro de la salle a son importance" });
garrot.Add(new HelpCommand() { CommandName = "SendText", Text = "Garrot sur le patient", TextParameter = "Amener les bandages au patient !" });

Helps.Add(garrot);

var cryptedMessage = new List<HelpCommand>();
cryptedMessage.Add(new HelpCommand() { CommandName = "SendText", Text = "Toquer à la porte", TextParameter = "Quelqu'un a toqué à la porte. Regardez dans les alentours." });
cryptedMessage.Add(new HelpCommand() { CommandName = "SendImage", Text = "Image tablette reçue", ActionIndex = 54 });
cryptedMessage.Add(new HelpCommand() { CommandName = "SendEvent", Text = "Flèche tablette", ActionIndex = 55 });
cryptedMessage.Add(new HelpCommand() { CommandName = "SendImage", Text = "Exemple de déchiffrement", ActionIndex = 56 });
cryptedMessage.Add(new HelpCommand() { CommandName = "SendEvent", Text = "Lumière sur toutes les clés", ActionIndex = 57 });

Helps.Add(cryptedMessage);
```

Figure 1 : HelpSelection.cs - Création des indices côté modération

```
Scenario sce1 = new Scenario("Scenario Operation");
sce1.AddVirtualEvent(new VirtualEvent(scenario: sce1.id, "1", recipient: "All", name: "Arrêt des lumières temporaires"));
sce1.AddVirtualEvent(new VirtualEvent(scenario: sce1.id, "2", recipient: "All", name: "Alarme bombardement"));
```

Figure 2 : DBManager.cs - Création du scénario et des événements perturbateurs

2.3.3. Ancrage du monde virtuel

Un GameObject englobe l'ensemble du monde virtuel et permet son ancrage dans le monde réel. Cet ancrage peut être personnalisé au début de chaque partie, permettant aux participants d'adapter l'environnement virtuel en fonction de leur environnement réel.

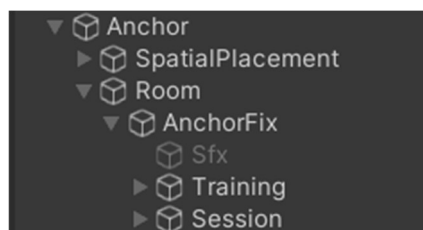


Figure 3 : GameObject principal sur lequel l'ancrage est basé

2.4. Limitations du projet actuel

2.4.1. Mono scénario

Le projet Host est initialement conçu pour un seul scénario spécifique. Bien que des éléments tels que la base de données puissent suggérer une adaptation pour plusieurs scénarios, il est évident que des compromis ont été faits probablement dans le but de respecter un deadline précédent.

2.4.2. Manque de scénario fil rouge entre les énigmes

Il est important de souligner qu'il n'existe pas vraiment de concept de scénario dans le sens traditionnel au sein de l'application. Les énigmes sont indépendantes les unes des autres et ne suivent pas une progression linéaire. Au lieu de cela, toutes les énigmes sont disponibles simultanément, sans qu'il y ait de fil rouge ou de séquence prédéfinie à suivre. Les participants ont la liberté de résoudre les énigmes dans l'ordre qui leur convient, sans contraintes de chronologie ou de dépendance entre les énigmes.

2.4.3. Surcouche Host TCP

La surcouche disponible dans l'application présente cependant certaines limitations. Dans la version actuelle, les synchronisations entre les participants sont assez rares, permettant alors que le buffer soit vide avant la prochaine écriture dans celui-ci. Cependant, si des communications plus fréquentes étaient mises en place, cela pourrait poser un problème, car le buffer risquerait de ne pas être vidé avant la prochaine communication.

2.4.4. Inscriptions des méthodes RPC

Il est nécessaire d'inscrire les méthodes RPC du serveur sur le client et vice versa, même si certaines de ces méthodes ne sont pas utilisées. Cela signifie que les mêmes inscriptions de méthodes doivent être présentes à la fois côté client et côté serveur, ce qui peut conduire à une liste considérable de méthodes inscrites, mais non utilisées.

3. Description du nouveau scénario

3.1. Contexte

Dans un hôpital situé dans une zone touchée par une catastrophe naturelle, les équipements sont rudimentaires, mais apparemment suffisants pour les tâches qui doivent être accomplies. Aucun médecin ou chirurgien n'est disponible, les participants doivent se débrouiller par eux-mêmes.

Une équipe d'urgence vient d'amener un blessé sur la table d'opération. Les participants se retrouvent "coincés" dans la salle d'opération avec le patient, et ils doivent se débrouiller et tout gérer par eux-mêmes. Tous les outils et équipements nécessaires se trouvent dans la pièce, ils en sont conscients. Certains éléments seront également virtuels. Le patient est réel et présente visiblement des traumatismes multiples. Il est déjà traité avec des perfusions, des sondes et une intubation.

Les participants disposent de casques de réalité augmentée HoloLens 2, qui fourniront des éléments virtuels pour renforcer l'immersion. Un maître du jeu est présent pour observer les actions des participants, prendre des notes et les aider si nécessaire. Il communiquera avec eux via des indices textuels, transmis dans le casque HoloLens pour activer un projecteur qui affichera le message.

3.2. Participants

Le scénario a été spécifiquement conçu pour un minimum de deux participants équipés de HoloLens. Idéalement, il serait également souhaitable d'avoir au moins un participant sans HoloLens, favorisant ainsi la collaboration.

Le choix du nombre de participants équipés de HoloLens s'explique par plusieurs raisons. Tout d'abord, il est essentiel qu'au moins deux joueurs aient accès aux informations fournies par la réalité augmentée, car cela leur permettra de communiquer efficacement sur les éléments virtuels présents. Ils seront également en mesure d'expliquer aux participants sans casque ce qu'ils voient et de les guider.

De plus, pendant l'étape de l'opération, un joueur portant un casque HoloLens sera "bloqué" à la surveillance du patient. Par conséquent, l'un de ses collègues devra prendre en charge la tâche de transmettre les informations nécessaires pour mener à bien le jeu.

En résumé, la présence d'au moins deux participants avec HoloLens facilite la communication et l'interaction avec les éléments virtuels, tandis qu'un participant sans casque peut contribuer à la collaboration et à la coordination des actions au sein du groupe.

3.3. Étapes et énigmes

Chaque étape du jeu comprend une ou plusieurs énigmes. La résolution de toutes les énigmes de l'étape précédente déclenche l'activation de l'étape suivante. En d'autres termes, les participants doivent résoudre avec succès toutes les énigmes d'une étape donnée afin de progresser vers l'étape suivante du jeu.

3.3.1. Étape hémorragie

L'objectif de cette étape est de localiser un garrot pour stopper l'hémorragie du patient. Les participants doivent trouver un garrot et l'appliquer immédiatement pour arrêter le saignement.

Énigme du garrot

Dans le but de familiariser les participants avec les interactions du HoloLens et la communication entre joueurs, cette énigme reste simple. Les joueurs devront ouvrir une boîte de premiers secours qui contient un garrot. Cependant, la boîte est fermée par un cadenas. Les personnes responsables de la mise en place du cadenas n'ont pas été très créatives et ont utilisé le même code que le numéro de la salle.

Les participants qui n'ont pas de casque HoloLens auront plus de facilité à repérer le numéro de la salle, car ils ne seront pas distraits par des objets virtuels. Cela leur donnera un avantage pour résoudre cette énigme.



Figure 4 : Visual de l'énigme du garrot

3.3.2. Étape Message codé

Énigme de décodage

Dans le monde virtuel, les participants entendront un coup frappé à la porte, suivi du message codé glissé sous celle-ci. Ils devront trouver un moyen de déchiffrer le message. Ce message révélera que le patient est allergique à la céphalosporine, une information essentielle à prendre en compte pour assurer son traitement adéquat.

Pour déchiffrer le message, les participants devront retrouver des morceaux de la clé de cryptage dispersés dans la salle, à la fois dans le monde réel et virtuel. Les parties de la clé virtuelle seront générées de manière aléatoire, ce qui signifie que le message chiffré changera à chaque simulation, mais conservera toujours la même signification.

Cela ajoutera un élément de recherche et de découverte aux interactions des participants, les obligeant à explorer à la fois l'environnement réel et virtuel pour rassembler les morceaux de la clé et déchiffrer le message.

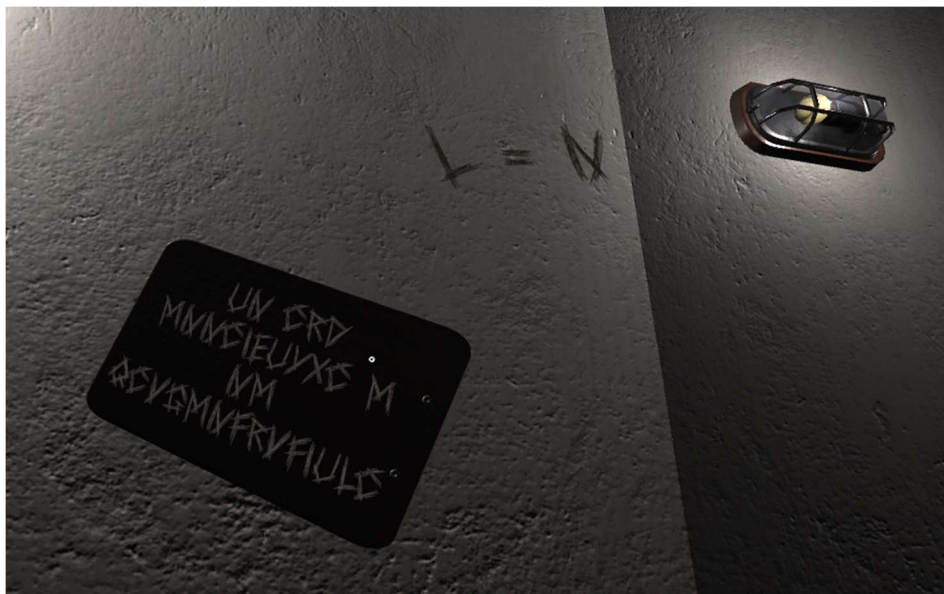


Figure 5 : Message crypté avec une partie de la clé de déchiffrement

3.3.3. Étape opération

L'opération comprendra plusieurs énigmes à résoudre, et l'ordre dans lequel les énigmes sont résolues n'a pas d'importance. Les participants auront la liberté de choisir l'énigme sur laquelle ils souhaitent se concentrer en premier, et pourront progresser à leur propre rythme. Cela permettra une approche flexible et adaptée à la

dynamique de l'équipe, où chacun pourra contribuer à résoudre les énigmes dans l'ordre qui leur semble le plus approprié.

Énigme du Monitoring

Dans cette partie du jeu, un participant devra constamment surveiller l'état de santé du patient en examinant le monitoring. Cependant, l'appareil de monitoring est en mauvais état et nécessite un recalibrage toutes les 10 secondes. Sans ce recalibrage régulier, l'appareil risque de transmettre de fausses informations pouvant entraîner la mort du patient.

L'interface de monitoring sera réalisée virtuellement, et des indices seront affichés à proximité de celui-ci. Ainsi, le participant pourra participer à la réflexion sur les autres énigmes tout en surveillant l'état du patient. Ces indices n'auront pas besoin d'être transmis par le maître du jeu.

Pour progresser dans cette énigme, il suffira d'appuyer sur le bouton correspondant à la lumière clignotante. Il y aura cinq boutons, chacun associé à une lumière, qui s'activeront de manière aléatoire. Seul un bouton pourra être actif à un moment donné. Plusieurs interactions correctes devront être effectuées afin que les indices nécessaires à la résolution des autres énigmes s'affichent.



Figure 6 : Énigme du Monitoring

Énigme de l'aspirateur

Pour procéder à l'opération du patient, un aspirateur est nécessaire, mais il est bloqué dans une armoire protégée par un code. L'objectif de cette énigme est de favoriser la collaboration entre les participants.

Pour résoudre cette énigme, une personne devra se déplacer sur une grille au sol comportant des chiffres. La personne qui est chargée de surveiller le patient recevra des informations sur la résolution de l'énigme grâce à des flèches. Ces flèches indiqueront le chemin à suivre sur la grille, permettant ainsi de déterminer le code nécessaire pour déverrouiller l'armoire et obtenir l'aspirateur.

Cela nécessitera une communication efficace entre les participants, car ceux qui se trouvent sur la grille devront transmettre les informations fournies par les flèches à celui qui est au monitoring. En travaillant ensemble, ils pourront déchiffrer le code et obtenir l'aspirateur.



Figure 7 : Flèches directrices

Énigme de la pince à clamper

Pour arrêter le saignement de l'artère malgré le garrot, il est nécessaire de clamper l'artère en utilisant une pince à clamper. Cette pince se trouve dans une boîte qui est fermée à l'aide d'un code.

Dans le monde réel, sur la boîte, il y aura un calcul. Cependant, les chiffres du calcul seront remplacés par des éléments du tableau périodique des éléments. Le résultat de ce calcul sera le code permettant d'ouvrir le cadenas de la boîte.

Dans le monde virtuel, sur le monitoring, le tableau périodique des éléments apparaîtra si la série de boutons réussie est suffisamment élevée. Cela fournira des indices nécessaires pour résoudre le calcul et obtenir le code.

Il faudra donc combiner les informations trouvées dans le monde réel et virtuel pour déterminer le code du cadenas et obtenir la pince à clamper, permettant ainsi de traiter l'artère qui saigne.

Énigme de la suture

Pour procéder à la suture, il est nécessaire d'amener sur la table d'opération le fil, le porte-aiguille et les 2 pincettes. Cependant, il faut d'abord réussir à les trouver et à les atteindre.

Des dessins de labyrinthes seront dissimulés virtuellement dans la pièce. Chaque résolution de labyrinthe représentera un chiffre spécifique qui permettra d'ouvrir une boîte contenant les outils nécessaires à la suture.

Il faudra donc faire preuve d'observation, de réflexion et de collaboration pour trouver les dessins de labyrinthes, les résoudre et récupérer les outils nécessaires à la suture.

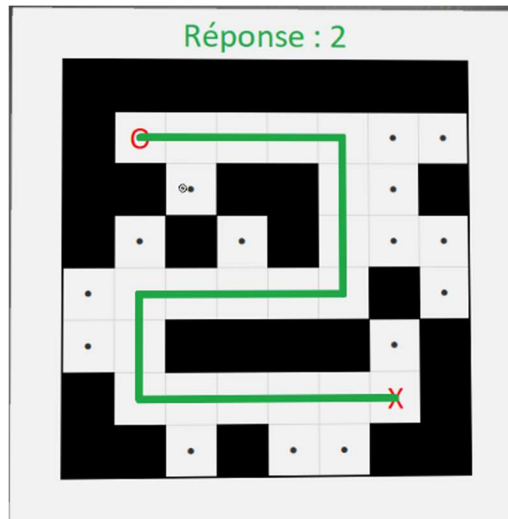


Figure 8 : Exemple de labyrinthe résolu

Antibiotique

La plaie est fortement souillée et nécessite l'injection d'un antibiotique. Pour cela, il faut trouver et administrer de la vancomycine, car le patient est allergique à la céphalosporine. Cependant, avant de pouvoir accéder aux antibiotiques, il faudra réussir à ouvrir l'armoire qui les contient.

Le code du cadenas de l'armoire sera découvert en rassemblant des seringues qui comportent des lettres d'un côté et des chiffres de l'autre.

En assemblant toutes les lettres des seringues, on obtiendra le mot "opération". De l'autre côté, on obtiendra un code qui permettra l'ouverture de l'armoire.



Figure 9 : Seringue virtuelle guidant les participants

3.4. Indices

Chaque étape du jeu est accompagnée d'indices pour aider les participants dans leur résolution des énigmes. Ces indices sont conçus de manière progressive, offrant initialement des indications générales pour donner une idée de ce qu'il faut chercher ou accomplir. Cependant, si les participants rencontrent des difficultés persistantes,

le maître du jeu pourra envoyer des indices plus spécifiques, indiquant les éléments sur lesquels ils devraient se concentrer.

Dans le cas où les participants se retrouveraient vraiment bloqués et ne parviendraient pas à résoudre l'énigme, le maître du jeu aura la possibilité d'envoyer un indice détaillant la méthode de résolution de l'énigme. Cela permettra de maintenir une progression fluide dans le jeu, en veillant à ce que les participants puissent avancer tout en conservant un niveau de défi approprié.

Ainsi, les indices sont conçus pour guider les participants tout au long du jeu, en offrant des indications adaptées à leur niveau de difficulté et en assurant une expérience de jeu stimulante et engageante.

3.5. Perturbateurs

Deux éléments perturbateurs ont été intégrés dans ce scénario. Le premier consiste en un son d'alarme inattendu, conçu pour surprendre les participants et les déconcentrer. Cette perturbation vise à ajouter une dose d'incertitude et à mettre à l'épreuve leur capacité à se concentrer malgré les distractions.

Le deuxième élément perturbateur est plus contraignant, car il entraîne l'extinction des lumières dans le monde virtuel. Cette situation rend la progression de l'étape plus difficile, car les participants sont confrontés à un environnement obscur. Cela met leur capacité d'adaptation à l'épreuve et nécessite une réflexion supplémentaire pour trouver des solutions malgré les contraintes visuelles.

Ces perturbateurs ont été incorporés dans le scénario dans le but de créer des défis supplémentaires et de susciter une réaction rapide et stratégique de la part des participants. Ils ajoutent une dimension d'imprévisibilité et d'urgence, renforçant ainsi l'immersion et l'excitation du jeu.

4. Technologies

4.1. Technologies utilisées dans le projet existant

Le projet Host a été développé avec la version d'Unity 2020.3, qui est une version "Long Term Support" et bénéficiera de mises à jour jusqu'au milieu de l'année 2023.

Une partie importante du projet concerne les HoloLens 2, qui sont des casques de réalité augmentée utilisant la plateforme de développement "Universal Windows Platform" (UWP). Ces casques permettent d'ajouter des éléments virtuels dans le monde réel, offrant ainsi une expérience de réalité augmentée.

Pour faciliter l'utilisation des interactions des HoloLens dans un projet Unity, Microsoft propose les paquets MRTK (Mixed Reality Toolkit). Ces paquets fournissent de nombreux scripts et éléments préfabriqués pour faciliter l'intégration des fonctionnalités des HoloLens 2. On peut y trouver des boutons, des interfaces utilisateur, des objets déplaçables, et bien d'autres exemples d'utilisation des HoloLens 2, offrant ainsi un large éventail de possibilités pour le développement du projet.

4.2. Technologies rajoutées

Dans le cadre de l'énigme visant à faire apparaître un objet sur une image, il a été nécessaire d'ajouter une librairie de détection d'images. Cette librairie permet de reconnaître et de localiser des motifs spécifiques sur des images, qu'il s'agisse d'images ordinaires ou de codes QR.

Grâce à cette librairie de détection d'images, l'application est capable d'analyser en temps réel les images capturées par les HoloLens ou les codes QR présents dans l'environnement virtuel. Cela permet de déclencher des actions ou des événements en fonction de la détection d'une image spécifique, ou d'utiliser les informations extraites de l'image pour résoudre une énigme.

4.2.1. Microsoft QR

Lors de mes recherches, j'ai exploré la librairie Microsoft dédiée à la détection des codes QR. Cette librairie offrait des fonctionnalités intéressantes, mais elle présentait une limite importante : les codes QR devaient avoir une taille comprise entre 5 et 10 cm pour être détectés correctement. Cette contrainte de taille pouvait être un inconvénient, notamment si les codes QR devaient être intégrés dans des supports plus petits ou discrets.

Des projets existants ont été examinés, qui étaient conçus spécifiquement pour les HoloLens et visant à la détection de codes QR. Malheureusement, lors des tests, aucun de ces projets n'a fonctionné, même ceux de Microsoft.

Face à ces limitations, il a fallu explorer d'autres options pour répondre aux besoins de détection d'images et de codes QR dans le projet.

4.2.2. Vuforia

Face aux limitations rencontrées avec la librairie Microsoft pour la détection des codes QR, la décision de me tourner vers Vuforia a été prise. C'est une librairie plus polyvalente offrant des fonctionnalités de détection d'images, de modèles, d'objets, et bien d'autres.

Cependant, un nouvel obstacle est survenu : la version 10 de Vuforia n'était pas compatible avec Unity 2020.3, la version utilisée dans le projet. Curieusement, la documentation de la version 9 n'était plus disponible sur le site de Vuforia, mais j'ai pu trouver les packages d'installation correspondants.

Il a donc été choisi d'utiliser la version 9 de Vuforia et de prendre le temps par soi-même de comprendre son fonctionnement et ses possibilités. Malgré l'absence de documentation officielle, il a été possible de se familiariser avec les fonctionnalités essentielles et les intégrer dans le projet.

Cette adaptation à une nouvelle librairie a nécessité des efforts supplémentaires pour comprendre son utilisation et s'assurer de sa compatibilité avec Unity 2020.3, mais cela m'a permis de résoudre les problèmes liés à la détection d'images dans le projet.

5. Développement du projet

5.1. Compréhension de l'existant

Le principal défi de ce projet résidait dans la compréhension du code existant. Étant donné qu'il avait été développé par plusieurs personnes, il y avait très peu de documentation disponible, ce qui rendait la tâche complexe. J'ai donc dû procéder à des vérifications minutieuses des scripts les plus importants afin de comprendre les étapes clés du fonctionnement.

En outre, j'ai dû effectuer des recherches pour localiser les scripts, les GameObjects et les composants nécessaires. J'ai également constaté qu'une des interfaces utilisateur était entièrement créée dans un script, ce qui était plutôt inhabituel pour moi.

Enfin, j'ai dû me familiariser avec les communications réseau, plus précisément la surcouche mise en place pour le projet Host. J'ai consacré une partie du projet à la décomposition et aux tests de l'existant, afin de comprendre les mécanismes et les éventuelles limitations. Cette étape m'a permis d'avoir une vision plus claire des possibilités offertes et des éléments que je pourrais réutiliser pour accélérer le développement de mon propre scénario.

Cependant, il est important de mentionner que j'ai également dû investir du temps dans la compréhension du code après avoir commencé le développement, car certaines parties se sont révélées plus importantes que prévu.

5.2. Nouvelles mécaniques de jeu

5.2.1. Affiche d'objet sur QR Code

Initialement, l'idée était de faire apparaître différents garrots sur des images. Cependant, lors de la mise en œuvre du mécanisme, j'ai réalisé que le temps imparti ne serait pas suffisant en raison de problèmes techniques et d'une technologie peu adaptée au projet existant. Malheureusement, cette interaction a dû être abandonnée. Toutefois, je reste convaincu que cela aurait apporté une nouvelle dimension aux énigmes du projet, et je considère que c'est un élément incontournable à explorer dans le cadre du travail de Master.

Il est toutefois important de noter qu'afficher un objet fonctionnait avec Vuforia, mais qu'il était compliqué de le faire suivre l'image pour deux raisons.

La première était que l'objet n'apparaissait pas toujours au bon endroit, et se retrouvait parfois sous le sol, ou dans les airs. La deuxième était plus un problème de performance. L'objet se déplaçait de manière saccadée, ce qui ne rendait les choses que très peu immersives, et très peu pratique.

La mise en place était plutôt simple, il suffisait d'ajouter un script sur la caméra, et d'utiliser une de leur prefab, contenant l'objet à afficher, et ayant l'image en paramètre.

5.2.2. Synchronisation accrue entre les HoloLens, donc plus de collaboration

Lors de mon essai du projet existant, j'ai remarqué un manque de cohérence entre les HoloLens. Aucune énigme n'est collaborative dans le monde virtuel et seuls certains événements "importants" sont synchronisés. Dans cette optique, l'objectif est d'ajouter cet aspect de collaboration dans le nouveau scénario, afin de créer de nouvelles interactions et de repousser encore davantage les limites du projet. Cette amélioration permettra aux participants de travailler ensemble de manière plus étroite et de bénéficier d'une expérience immersive et collaborative plus enrichissante dans le monde virtuel.

J'ai donc créé une énigme pouvant être réalisée par plusieurs participants en simultanée, dû au fait que les informations transmises sont les mêmes.

Il serait cependant très intéressant d'accentuer encore cette synchronisation, par exemple au déplacement des objets, car cela permettrait d'avoir le Game World exactement pareil dans tous les HoloLens. Cela amènerait

éventuellement les participants à se donner les objets virtuels, ou bien simplement qu'ils puissent mieux comprendre ce que sont en train de faire leurs collègues.

5.3. Conception du monde virtuel

Le Game World a été construit à partir d'assets disponibles sur l'Assets Store. Étant donné que le scénario se déroule dans un hôpital désaffecté, l'objectif était de créer une ambiance lourde et oppressante. Ainsi, on peut y trouver des lits renversés pour faire de la place pour nos patients, des accessoires anciens, des murs sales et une luminosité réduite.

Le lit central du patient a été conçu de manière à servir de point d'ancrage lors de la phase de placement du monde virtuel. Il suffit donc de positionner ce lit virtuel au même emplacement que le lit réel, permettant ainsi une synchronisation entre le monde virtuel et le monde réel.



Figure 10 : Monde virtuel du nouveau scénario

5.4. Conception des énigmes du scénario

Dans le scénario Avion, les énigmes ne sont pas directement liées les unes aux autres et ne nécessitent généralement pas d'interactions multiples. Pour résoudre ce problème, un système a été mis en place. Chaque énigme dispose d'un script qui agit comme un "contrôleur" pour cette énigme. Ce contrôleur contient généralement la solution de l'énigme.

Les contrôleurs d'énigmes peuvent se synchroniser entre eux, du moins sur certains événements, afin de maintenir la cohérence des états des HoloLens. Cela permet de garantir que les participants voient les mêmes informations et que les actions effectuées par un participant affectent également les autres participants de manière appropriée.

5.4.1. Différentes phases d'une énigme

Chaque énigme possède plusieurs phases. Ces phases sont les suivantes.

Début de scénario

Au début du scénario, certains éléments graphiques sont mis en place et apparaissent dès que le scénario est lancé. Un exemple concret est l'énigme du message crypté, où des indications sont affichées à différents endroits de la pièce. Il est tout à fait normal que ces informations soient affichées dès le début, même si l'énigme en elle-même n'a pas encore commencé.

Cette approche permet de fournir aux participants des indices visuels dès le départ, créant ainsi une atmosphère immersive et les incitant à observer attentivement leur environnement. Cela peut également les aider à se familiariser avec les éléments qui seront nécessaires pour résoudre les énigmes ultérieures.

Début de l'énigme

Dans le cas de l'énigme du message crypté, le début de l'énigme est déclenché lorsque l'on toque à la porte virtuelle. C'est à ce moment précis que l'énigme entre en action et que les participants peuvent commencer à interagir avec les éléments liés à cette énigme en particulier.

Le fait de déclencher l'énigme au moment opportun, tel que toquer à la porte virtuelle, permet de créer une progression narrative et d'introduire des éléments de jeu de manière cohérente. Cela ajoute un aspect interactif et dynamique à l'expérience, en invitant les participants à agir pour déclencher et résoudre chaque énigme.

Ainsi, chaque énigme possède son propre déclencheur spécifique qui marque le début de son activation, offrant aux participants une expérience progressive et guidée à travers le scénario.

Fin de l'énigme

Effectivement, la fin de l'énigme survient lorsque les joueurs ont réussi à effectuer l'action requise, dans ce cas précis, le rendu correct de la tablette des allergies pour l'énigme du message crypté. Une fois que les joueurs ont accompli cette tâche avec succès, cela marque la résolution de l'énigme et sa conclusion.

Il est important de souligner que la fin de chaque énigme peut varier en fonction de la nature de l'énigme elle-même. Dans certains cas, il peut s'agir de trouver un objet spécifique, de résoudre une équation, de déverrouiller une porte, ou d'accomplir toute autre action spécifiée par l'énigme. La réussite de cette action est ce qui détermine la fin de l'énigme et permet aux joueurs de passer à la suite.

Cette approche de définir la fin de l'énigme en fonction d'une action précise offre une structure claire et permet de mesurer la progression des joueurs tout au long du scénario.

```
1 référence
public void SetCryptedMessage(string message, string pairs)...

1 référence
public void GiveMessage()...

1 référence
public void CheckAnswer()
{
    if (direction != Direction.None) return;

    messageObject.GetComponent<NearInteractionGrabbable>().enabled = false;
    messageObject.transform.localPosition = inPosition.localPosition;
    messageObject.transform.localRotation = rotation;
    direction = Direction.Out;
    if(!CephalosporineCheck.activeInHierarchy || VancomycineCheck.activeInHierarchy || AmoxicilineCheck.activeInHierarchy)
        Invoke("GiveMessage", 5);
    else
    {
        rpc.TriggerCryptedMessageDone();
    }
}

1 référence
```

Figure 11 : CryptedMessageRiddle.cs - Exemple de mise en place d'une énigme

Sur l'image ci-dessus, on peut voir qu'une fois que la réponse est trouvée, on transmet l'information via RPC afin de donner l'information à tous les autres participants.

Toutes les phases sont synchronisées entre les HoloLens et l'application de monitoring. Cependant, il est important de noter que les liens entre le début et la fin de chaque phase ont été codés manuellement dans le projet actuel.

L'idée d'avoir un manager de scénario qui pourrait automatiser cette gestion serait certainement bénéfique. Ce manager de scénario pourrait permettre d'assigner les énigmes dans une liste ordonnée, où chaque énigme serait déclenchée automatiquement lorsque la précédente se termine avec succès.

Cela simplifierait la tâche du développement et de la gestion du scénario, en évitant d'avoir à coder manuellement chaque lien entre les phases. Le manager de scénario pourrait prendre en charge cette logique de progression et de synchronisation, assurant ainsi une transition fluide d'une énigme à l'autre.

Une telle approche permettrait également de rendre le scénario plus flexible, car il serait possible d'ajouter, supprimer ou réorganiser les énigmes dans la liste sans avoir à modifier directement le code de chaque énigme.

En résumé, l'implémentation d'un manager de scénario automatisé serait une amélioration intéressante pour simplifier le développement et la gestion des phases et énigmes, en assurant une synchronisation efficace entre les HoloLens et en facilitant l'évolution du scénario.

5.4.2. Prise en compte des éléments réels

Effectivement, la conception d'énigmes qui combinent des éléments réels et virtuels peut être complexe et nécessite une attention particulière pour éviter les redondances ou les incohérences entre les deux aspects. Dans le cas de l'énigme du message crypté, certaines parties de la résolution se déroulent dans le monde réel, tandis que d'autres se déroulent dans le monde virtuel.

L'un des défis consiste à créer des éléments virtuels aléatoires tout en tenant compte des éléments réels pour éviter les répétitions ou les incohérences. Cela demande un développement supplémentaire pour intégrer les contraintes du monde réel dans la génération aléatoire des éléments virtuels. Par exemple, il peut être nécessaire de prendre en compte les informations ou les objets physiques disponibles dans l'environnement réel lors de la création des énigmes virtuelles.

Il est important de ne pas négliger les éléments réels lors du développement des parties virtuelles des énigmes. Cela peut inclure la correspondance entre les objets physiques et les objets virtuels, l'interaction entre les deux mondes, et la cohérence globale entre les éléments réels et virtuels de l'énigme.

5.5. Communication réseau

5.5.1. Utilisation de HostNetworkManager

L'utilisation du HostNetworkManager, une surcouche du serveur TPC qui facilite l'utilisation des appels de procédure à distance (RPC), est simple et intuitive. Il suffit d'enregistrer les méthodes souhaitées en tant que RPC et de les appeler via le gestionnaire. Ainsi, les appels sont automatiquement transmis à la cible avec les paramètres appropriés, sans nécessiter la mise en place d'autres mécanismes complexes. Cette fonctionnalité simplifie considérablement l'utilisation du réseau et m'a permis de progresser rapidement dans l'élaboration du scénario.


```

1 référence
private void InitRpc()
{
    HostNetworkManager.RegisterGameObject(HostNetworkId, this);
    HostNetworkManager.RegisterRPC(HostNetworkId, 0, "TriggerMessageEvent");
    HostNetworkManager.RegisterRPC(HostNetworkId, 1, "TriggerVirtualEvent");
    HostNetworkManager.RegisterRPC(HostNetworkId, 2, "TriggerHelpEvent");
    HostNetworkManager.RegisterRPC(HostNetworkId, 3, "StopSimulation");
    HostNetworkManager.RegisterRPC(HostNetworkId, 4, "StartStreaming");
    HostNetworkManager.RegisterRPC(HostNetworkId, 5, "SeatsFreeNotification");
    HostNetworkManager.RegisterRPC(HostNetworkId, 6, "SetSwitchState");
    HostNetworkManager.RegisterRPC(HostNetworkId, 7, "BreakerPanelOpen");
    HostNetworkManager.RegisterRPC(HostNetworkId, 8, "CryptedMessage");
    HostNetworkManager.RegisterRPC(HostNetworkId, 9, "MonitoringPressedButton");
    HostNetworkManager.RegisterRPC(HostNetworkId, 10, "MonitoringActiveButton");
    HostNetworkManager.RegisterRPC(HostNetworkId, 11, "MonitoringFeedback");
    HostNetworkManager.RegisterRPC(HostNetworkId, 12, "GarrotDone");
    HostNetworkManager.RegisterRPC(HostNetworkId, 13, "CryptedMessageDone");
    _rpcInitDone = true;
}

```

Figure 12 : HelpRPC.cs - Inscriptions des méthodes en RPC

```

1 référence
public void TriggerCryptedMessageDone()
{
    HostNetwork.RPC(HostNetworkId, "CryptedMessageDone", HostNetworkTarget.All);
}

0 références
public void CryptedMessageDone()
{
    cryptedMessage.Done();
}

```

Figure 13 : HelpRPC.cs – Exemple de méthodes émettrices / réceptrices

5.5.2. Synchronisation des HoloLens

Le networking est principalement utilisé pour synchroniser l'état du jeu entre les HoloLens. J'ai choisi de l'implémenter en utilisant une architecture client/serveur, bien qu'il soit également possible d'établir des communications directes entre les appareils. Cette approche de client/serveur permet de centraliser la gestion de la synchronisation et de garantir un état cohérent du jeu sur tous les appareils.

5.6. Utilisation des HoloLens 2

La prise en main des HoloLens n'a pas été une tâche facile, car je n'avais jamais utilisé ces appareils auparavant. J'ai dû consacrer du temps à me familiariser avec les interactions possibles et à comprendre les fonctionnalités offertes par le casque. Il regorge d'un large éventail de fonctionnalités qui peuvent être exploitées si l'on en connaît l'existence. Une partie importante de mon début de projet a été d'explorer ces fonctionnalités, de les tester et de réfléchir à leur pertinence pour l'ajout dans mon projet.

5.6.1. Déploiement

Le déploiement sur les HoloLens nécessite de suivre un processus spécifique. Bien que Microsoft fournisse des instructions pour le déploiement à partir d'Unity, toutes les étapes ne sont malheureusement pas spécifiées. La partie concernant les certificats n'est pas mentionnée, ce qui peut entraîner des difficultés.

Pour déployer sur les HoloLens, il est essentiel de signer l'application avec un certificat. Vous avez la possibilité de créer un certificat dans Unity ou d'utiliser un certificat existant sur votre ordinateur.

Le pipeline classique de déploiement comprend donc plusieurs étapes.

Paramétrage sur Unity

Dans Unity, la configuration implique la création du certificat, la définition du nom du package, de la version et d'autres informations relatives au projet. Elle permet également de spécifier la plateforme cible, le type de build et d'autres paramètres.

Compilation sur Unity

La compilation doit tout d'abord se faire sur Unity pour créer un projet IL2CPP.

Paramétrage sur Visual Studio

Une fois que le projet a été compilé avec Unity, cela génère un projet Visual Studio qui doit être ouvert et configuré davantage. Il est nécessaire de spécifier à nouveau un certificat, qui n'a pas besoin d'être le même que celui utilisé précédemment.

Compilation et déploiement sur Visual Studio

Une fois que tous les paramètres ont été remplis, on peut passer à la compilation et au déploiement. On spécifie la méthode de déploiement, que ce soit via une connexion USB directe ou en utilisant la méthode Remote. L'application est ensuite déployée sur les HoloLens.

Déploiement depuis Unity

La possibilité de déployer directement depuis Unity, en contournant Visual Studio, existe, mais je n'ai pas trouvé cette méthode fiable. Bien que pratique, elle a échoué à plusieurs reprises sans aucune erreur explicite pour expliquer le problème. En revanche, le déploiement en utilisant le même build avec Visual Studio a fonctionné correctement.

Il est donc assez déconcertant de constater que l'application se déploie dans un certain contexte, mais pas dans un autre, sans aucune information sur la cause du dysfonctionnement.

6. Problèmes rencontrés

6.1. Communication réseau avec les HoloLens

L'utilisation du réseau avec la surcouche HostNetworkManager m'a facilité la tâche jusqu'à ce que je dépasse ses limites. Ces limites n'étaient pas les mêmes lorsque j'utilisais la simulation sur Unity, car j'étais en train de tester l'application de modération et l'application HoloLens sur le même ordinateur, évitant ainsi les délais liés au réseau.

Le scénario de base ne nécessitait que peu d'échanges sur le réseau, et le buffer avait été créé pour contenir un seul objet JSON. Cependant, dans mon cas, il arrivait que plusieurs objets se succèdent dans le buffer. Cela a posé un problème de format, car les objets JSON multiples dans le buffer n'étaient pas encapsulés dans un tableau, mais simplement collés les uns à la suite des autres, comme suit :

```
{
  "MessageType": 0,
  "NetworkTarget": 0,
  "SourceIP": "0.0.0.0",
  "TargetIP": "",
  "Data": "eyJPY3RjZCI6MSwiUnBjSWQiOjExLCJQYXlsb2FkIjpbZmFsc2VdfQ=="
}
{
  "MessageType": 0,
  "NetworkTarget": 0,
  "SourceIP": "0.0.0.0",
  "TargetIP": "",
  "Data": "eyJPY3RjZCI6MSwiUnBjSWQiOjEwLCJQYXlsb2FkIjpbNF19"
}
```

Figure 14 : Pseudo JSON problématique

Le désérialiseur JSON n'était donc pas en mesure de lire ce message. J'ai donc rapidement adapté le code en transformant le message en un tableau d'objets JSON, ce qui donne le résultat suivant :

```
[
  {
    "MessageType": 0,
    "NetworkTarget": 0,
    "SourceIP": "0.0.0.0",
    "TargetIP": "",
    "Data": "eyJPY3RjZCI6MSwiUnBjSWQiOjExLCJQYXlsb2FkIjpbZmFsc2VdfQ=="
  },
  {
    "MessageType": 0,
    "NetworkTarget": 0,
    "SourceIP": "0.0.0.0",
    "TargetIP": "",
    "Data": "eyJPY3RjZCI6MSwiUnBjSWQiOjEwLCJQYXlsb2FkIjpbNF19"
  }
]
```

Figure 15 : JSON sous forme de tableau d'objets

De cette manière, le sérialiseur pouvait traiter les objets un par un.

```
// Workaround to fix problem when multiple message in buffer...
try
{
    HostNetworkMessage message = JsonConvert.DeserializeObject<HostNetworkMessage>(data);
    MessageReceived?.Invoke(this, new HostNetworkMessageEvent(message));
}
catch (JsonReaderException e)
{
    string[] messages = data.Split(new string[] { "}" }, StringSplitOptions.None);
    foreach (string message in messages)
    {
        string tmp = message;
        if (!message.StartsWith("{"))
        {
            tmp = "{" + message;
        }

        if (!message.EndsWith("}") )
        {
            tmp += "}";
        }

        HostNetworkMessage msg = JsonConvert.DeserializeObject<HostNetworkMessage>(tmp);
        MessageReceived?.Invoke(this, new HostNetworkMessageEvent(msg));
    }
}
```

Figure 16 : HostTcpClient.cs - Résolution du problème du buffer

6.2. Vidéo problématique

Actuellement, le problème lié à l'arrêt de l'application HoloLens lors du lancement de la vidéo de cardiogramme n'a pas été résolu. Malheureusement, aucun log n'est disponible pour identifier la cause exacte de ce problème. Par conséquent, la raison derrière cet arrêt inattendu n'est pas connue à l'heure actuelle.

6.3. Crash dû à un GameObject

Au début du projet, lors du déploiement de l'application HoloLens pour tester le pipeline, j'ai rencontré un problème où un GameObject présent sur la scène provoquait un crash du programme. Il est important de souligner qu'aucun problème n'était observable dans la simulation Unity. Initialement, je me suis remis en question, me demandant si j'avais commis une erreur. Étant donné que les logs ne fournissaient pas d'informations explicites, j'ai passé du temps à effectuer divers tests, cherchant des solutions. Malheureusement, j'ai épuisé toutes mes idées et j'ai finalement décidé de désactiver et de réactiver chaque GameObject de la scène.

Après avoir réalisé cette manipulation, j'ai réussi à isoler l'objet problématique. Ne voyant aucune utilité à cet objet, je l'ai simplement désactivé. Il est important de noter que le processus de déploiement successif de l'application prend du temps, et c'est l'un des problèmes qui m'a fait perdre le plus de temps dans ce contexte.

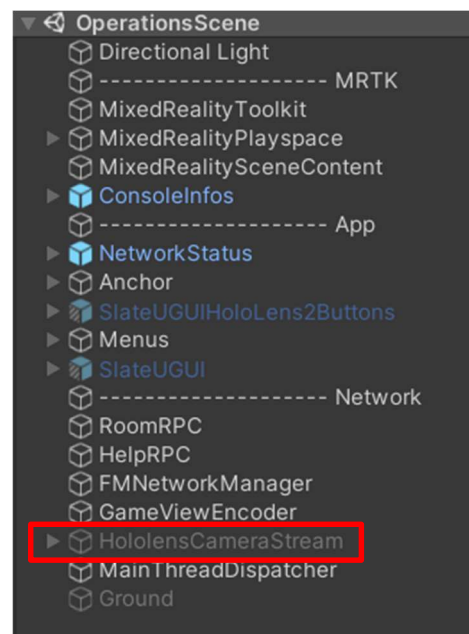


Figure 17 : GameObject problématique

6.4. Détection de QR Code

Malgré les tutoriels promettant une détection de QR Code facile, j'ai personnellement rencontré des difficultés et n'ai jamais réussi à la faire fonctionner. J'ai investi beaucoup de temps dans ce problème, en testant les projets HoloLens de détection de QR Code fournis par Microsoft, mais rien n'a abouti.

Face à ces échecs répétés, j'ai finalement pris la décision d'abandonner cette mécanique et de me tourner vers la détection d'images comme alternative. J'ai trouvé cette approche mieux adaptée à mes besoins, ce qui m'a permis de continuer à progresser dans mon projet.

6.5. Détection d'images

Vuforia est en effet une librairie bien connue et j'ai également eu l'occasion de travailler avec elle par le passé, la trouvant très simple à utiliser. J'ai même testé avec succès un exemple d'utilisation disponible sur l'Asset Store.

Dans le cadre de mon projet actuel, j'ai entrepris d'intégrer Vuforia et j'ai rencontré un premier problème de compatibilité entre les versions. Cependant, j'ai pu le résoudre en utilisant une version antérieure de Vuforia. J'ai alors commencé à développer une énigme basée sur la détection d'images. Initialement, tout semblait fonctionner correctement, avec l'objet se positionnant au-dessus de l'image comme prévu.

Cependant, lorsque j'ai essayé de mettre en place la détection de plusieurs images, j'ai rencontré des difficultés. Aucun objet ne s'affichait plus du tout. J'ai alors réduit le nombre d'images à une seule, en modifiant légèrement l'énigme. Malheureusement, l'objet ne se positionnait plus correctement sur l'image, mais dans le sol ou à un emplacement indésirable. J'ai passé beaucoup de temps à essayer de corriger le positionnement de l'objet, mais je n'ai jamais réussi à le faire correspondre à l'image comme je le souhaitais.

De plus, l'objet ne s'affichait pas systématiquement à chaque déploiement, ce qui a rendu la situation encore plus frustrante. Après avoir consacré beaucoup de temps et d'efforts à tenter de résoudre ces problèmes, j'ai finalement pris la décision d'abandonner cette fonctionnalité basée sur la détection d'images, bien que cela me tenait à cœur. J'ai alors réfléchi à une nouvelle énigme pour la remplacer, chose faite, mais malheureusement, j'ai perdu un temps considérable à tâtonner et à chercher des solutions pour une fonctionnalité qui n'a jamais donné de résultats fiables.

Ces difficultés ont été une expérience frustrante, mais elles m'ont également appris à être plus sélectif dans les fonctionnalités que je choisis de développer et à évaluer plus soigneusement les contraintes et les défis potentiels avant de m'engager pleinement dans un développement.

6.6. Difficulté de tester

La réalité augmentée nécessite que le joueur se déplace et que l'environnement réel soit approprié. Des contraintes telles qu'une pièce trop petite ou une table mal positionnée peuvent entraîner des difficultés lors du jeu.

Étant donné que j'habite dans un studio, il est évident que la pièce était trop petite. Pour tester ma simulation, j'ai dû effectuer plusieurs essais en déplaçant à chaque fois l'ancrage afin d'accéder à l'énigme que je souhaitais tester.

Bien que ce problème n'ait pas été grave en soi, il a néanmoins entraîné une légère perte de temps. J'ai dû procéder à des ajustements fréquents pour trouver la meilleure configuration permettant de jouer dans les limites de mon espace restreint.

7. Conclusion

7.1. Rappel de l'objectif

Le but était de créer un scénario inédit pour la simulation et de concevoir des énigmes supplémentaires pour enrichir les options du projet Host. Il était également nécessaire de comprendre et de maîtriser les divers mécanismes en vigueur afin de les exploiter de la meilleure façon possible.

7.2. Résultat obtenu

Il y a maintenant un scénario qui vient avec son propre ensemble d'énigmes. Ces énigmes utilisent des mécanismes existants, mais elles ajoutent des petites fonctionnalités ici et là pour rendre le scénario unique. Un monde virtuel a également été conçu, permettant aux joueurs de s'immerger dans un environnement lugubre et oppressant. Le modérateur du jeu dispose également d'un ensemble d'indices et d'événements perturbateurs pour influencer la progression des participants.

7.3. Conclusion personnelle

Je suis extrêmement satisfait d'avoir eu l'opportunité de travailler sur ce projet. En tant qu'amateur de résolution d'énigmes, le fait de les créer moi-même a été un défi auquel je ne m'attendais pas. Travailler avec une technologie que j'apprécie particulièrement, les HoloLens, a été une chance inestimable. J'ai adoré concevoir le scénario et donner vie à celui-ci en l'illustrant dans le monde virtuel.

J'ai été confronté à un projet d'envergure, que j'ai réussi à maîtriser dans une large mesure, en un laps de temps relativement court. J'ai également su surmonter la plupart des problèmes auxquels j'ai été confronté, même si certains se sont avérés plus tenaces que prévu.

Dans l'ensemble, je suis fier de mon travail. Je suis convaincu qu'il apporte une réelle valeur ajoutée au projet, et qu'il me prépare également pour le travail de Master que j'aurai à réaliser sur ce projet.

7.4. Améliorations et perspectives

Le projet Host offre une multitude de possibilités. Bien que le concept de fil rouge ait été introduit dans ce projet, il pourrait être étendu à grande échelle, simplifiant ainsi le développement de nouveaux scénarios.

Une généralisation des mécanismes des énigmes permettrait également la création de scénarios originaux tout en réduisant les coûts de développement. Cela pourrait être réalisé grâce à une boîte à outils d'énigmes facile à mettre en place. Il serait intéressant d'ajouter de nouveaux types d'énigmes, en particulier en utilisant la détection d'images.

De plus, une assistance à la création du monde virtuel serait bénéfique, par exemple en spécifiant les dimensions réelles, afin d'éviter que des objets virtuels se retrouvent inaccessibles à certains endroits.

8. Références

- [1] 1. **Microsoft**. Build and deploy to the HoloLens. *Microsoft Learn*. [En ligne] 09 09 2022. <https://learn.microsoft.com/en-us/windows/mixed-reality/develop/unity/build-and-deploy-to-hololens>.
- [2] 2. **PTC, Inc**. Image Targets. *Vuforia Library*. [En ligne] <https://library.vuforia.com/objects/image-targets>.
- [3] 3. **Microsoft**. QR code tracking overview. *Microsoft Learn*. [En ligne] 10 04 2022. <https://learn.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/qr-code-tracking-overview>.
- [4] 4. **van Schaik, Joost**. Positioning QR codes in space with HoloLens 2 - building a 'poor man's Vuforia'. *DotNetByExample - The Next Generation*. [En ligne] 3 02 2021. [Citation : 02 04 2023.] <https://localjoost.github.io/Positioning-QR-codes-in-space-with-HoloLens-2-building-a-'poor-man's-Vuforia'/>.

9. Annexes

Liste des annexes

- HoloLens avec l'application Host-Operation
- Code source du projet sur GitHub
- Exécutable de l'application de monitoring