# Introduction

## Cahier des charges

Ce projet vise la conception et le développement d'une bibliothèque d'énigmes digitales en réalité augmentée qui seront intégrées 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 des 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.

### Évolution du projet en travail de Master

Au cours du projet, il a été discuté de continuer le projet. De ce fait, les objectifs pour ce travail ont donc été légèrement modifié. Le but étant maintenant d’explorer tous les mécanismes du projet existant, afin de les comprendre et pouvoir les rendre plus générique, notamment pour la création d’une « boite à énigme » et de la simplification de création de nouveau scénario, l’existant étant très peu permissif sur ce point.

## Planification

### Écriture du scénario

Le scénario a été élaboré à partir d’un scénario fourni. Ce dernier avait été fait pour une expérience unique et était purement avec des éléments réels. Il a fallu adapter, voir modifier certaines énigmes afin qu’elles aient un sens dans un univers de réalité augmentée.

### Mécanisme des énigmes

Certaines énigmes ont un besoin d’interactions avec les utilisateurs. Cette étape donc consisté à explorer des solutions pour créer des énigmes ergonomiques à partir du scénario. Le moment « bac à sable » qui a permis de voir rapidement si oui ou non, des énigmes étaient viables ou si elles nécessitaient une modification.

### Création du Game World

Cette partie a permis de donner un aspect immersif à la simulation. L’idée de la réalité augmentée étant de plonger le joueur dans le jeu, il fallait donc rendre ce monde attractif et motivant. Les décors ont donc été mis en place, ainsi que l’ambiance globale du scénario.

### Association des énigmes avec le Game World

Les différentes énigmes ayant été créent sans contexte, il a fallu les incruster dans les différents décors. Il a fallu rendre naturel au maximum les différents mécanismes afin que le joueur puisse le remarquer, mais sans casser le côté immersif.

### Indices et événements

Cette dernière étape a été de créer des indices et des événements perturbateurs. Ces indices devaient être croissant, c’est-à-dire que le premier indice d’une énigme devait donner une piste, alors que le dernier devait presque donner la réponse. Un point important car il est difficile de juger à quel point un indice peut être évident ou non.

## Organisation

Le code du projet Unity est disponible sur GitHub :

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

Pour des raisons de simplification, j’ai opté pour un mono repository. Cependant, je pense qu’il serait intéressant de créer un dossier Github pour chaque application, d’autant plus que dorénavant, les kanbans GitHub permettent un suivi de plusieurs dossiers.

Le suivi du projet est fait avec un kanban et des issues GitHub. Un Milestone pour le projet d’approfondissement est disponible, et également pour le travail de Master. De ce fait, il est déjà possible de réfléchir à des idées d’améliorations pour le travail de Master, tout en les séparant de ce qui doit être réalisé pour le projet d’approfondissement.

Le dossier GitHub est disponible au travers d’une organisation. Ce qui veut dire que si de nouveau collaborateur doivent travailler sur le projet, il suffit de les ajouter dans l’organisation avec les droits adéquats.

# Analyse de l’existant

## Host

### Points fondamentaux

Basé sur Serveur TCP avec une surcouche « HostNetwork » -> UDP est généralement utilisé pour les jeux, changé ça pour TM ?

### Limitations

Vraiment fait pour 1 seul scénario

Pas vraiment de scénario a proprement parlé, toutes les énigmes faisables à tout moment, donc pas de trame. Donc il a fallu callé un manager de scénario. -> grosse amélioration pour TM

Le système de communication client-serveur est minimaliste, et a du mal avec de l’envoie de donnée fréquente (typiquement pour synchronisé la position d’un objet)

Les méthodes RPC doivent être inscrites sur les 2 app, même si une des 2 les utilisent pas car ne fait que l’invoker chez l’autre

# Scénario

## Contexte

Dans un hôpital situé en zone de catastrophe naturelle. Les équipements sont rudimentaires, mais a priori suffisants pour ce qu'ils ont à faire. Il n'y a pas de médecins / chirurgiens disponibles, les participants doivent se débrouiller par eux-mêmes.

Des gens viennent d'emmener sur la table un blessé, qui a déjà été pris en charge par une équipe d'urgence. Les participants se retrouvent "coincés" dans le bloc, avec un patient, et n'ont d'autre possibilité que de se débrouiller et de tout faire par eux-mêmes. Tout ce dont ils auront besoin est dans la pièce, ils le savent. Une partie des éléments sera également virtuelle. Le patient est réel et est visiblement polytraumatisé. Il est "équipé" : perfusé, sondé, intubé.

Les participants ayant des casques de réalité augmentée HoloLens 2, des éléments virtuels seront présents afin d’appuyer le côté immersif.

Un maitre du jeu pour voir ce que font les participants, prendre des notes ou éventuellement les aider si cela s’avère nécessaire. Il le fera à partir d’indice textuel, transmissible dans le casque HoloLens afin d’activer un projecteur qui affichera le message.

L’intégralité du scénario est disponible en annexe.

## Énigmes

Ici faut mettre des infos sur comment ont été faite les énigmes, et également comment modifier les codes ou autre pour matcher la réalité -> Ce serait d’ailleurs un super point à simplifier pour le TM

## Indices

## Perturbateurs

# Choix des technologies

## Détection d’images ou QR codes

Premiere tentative avec la lib QRCode de Microsoft -> infructueux, même les exemples de Microsoft n’ont pas fonctionné. Limitation net également avec la taille très peu pratique des QR Code.

Du coup Vuforia, qui permet de faire ce qu’on veut, même si limiter à al version 9 car le projet est sous unity 2020

# Développement du projet

## Nouvelles mécaniques de jeu

### Affiche d’objet sur QR Code

Dans un premier temps, il a été question de faire apparaitre différent garrot avec des QR Code. Cependant, lors de la réalisation du mécanisme, je me suis rendu compte que le temps à disposition ne permettrait pas de réaliser cette tâche.

Cette interaction a donc été malheureusement abandonnée, mais je reste persuadé que cela apporterait une nouvelle dimension aux énigmes du projet, et que dans le cadre du travail de Master, ce serait un élément incontournable.

### Synchronisation accrue entre les HoloLens, donc plus de collaboration

Lors de mon essaie du projet existant, j’ai remarqué un manque de cohérence entre les HoloLens. Aucune énigme n’est collaborative dans le monde virtuel, et seule certains événements « importants » sont synchronisés.

Le but étant donc de rajouter cet aspect dans le nouveau scénario, créant ainsi de nouvelles interactions et repoussant encore les limites du projet.

## Game world

Le monde a été crée avec les assets « … ». Le but étant d’être un hôpital désaffecté, n y retrouve de vieux objets tel que :

Projecteur

Seringue

…

## Énigmes

Les énigmes du scénario Avion n’ayant pas de réel lien entre elles, ni ne nécessite plusieurs interactions, il a fallu instaurer ce système. Chaque énigme possède un script, qui est le « contrôleur » de l’énigme. Le choix de solution s’y trouve pour la plupart. Les contrôleurs peuvent se synchroniser entre eux, en tous cas sur certains événements, afin de laisser les états des HoloLens cohérent au maximum.

### Détection d’images

Utilisation de Vuforia pour détecter une image et placer un objet sur celle-ci.

### Prise en compte des éléments réels

Création du message crypté. Une partie de la clé de chiffrage doit être un élément réel, et donc non aléatoire, besoin de penser un algorithme qui ne modifie pas certaines lettres, cf code CryptedMessage pour mieux expliquer

## Networking

### Utilisation de HostNetworkManager

### Synchronisation des HoloLens

Mode client/serveur ou l’app de monitoring est le serveur et les hololens sont les clients -> on pourrait peut être passé à peer to peer, mais c’était plus simple pour le temps à dispo.

## HoloLens

### Différence avec la simulation Unity

# Plus lent d’envoyer/lire buffer -> problème

### Déploiement

Ne pas oublier le certificat

Possible de build and run via Unity avec le device portal manager (mais nécessite de s’être connecté une fois via browser pour créer une sorte de user)

# Problèmes rencontrés

## Networking avec les HoloLens

Message Json successif sous force "{…}{…}" n’est pas un format reconnu, problème juste à cause du buffer qui était pas vide avant d’écrire. Ça marchait sur simulation unity car probablement instantané et donc buffer était vide à chaque fois.

Exemple de message qui posait problème : {"MessageType":0,"NetworkTarget":0,"SourceIP":"0.0.0.0","TargetIP":"","Data":"eyJPYmplY3RJZCI6MSwiUnBjSWQiOjExLCJQYXlsb2FkIjpbZmFsc2VdfQ=="}{"MessageType":0,"NetworkTarget":0,"SourceIP":"0.0.0.0","TargetIP":"","Data":"eyJPYmplY3RJZCI6MSwiUnBjSWQiOjEwLCJQYXlsb2FkIjpbNF19"}

# Conclusion

# Annexe

Code sur Github

Dossier de l’exécutable de l’app de monitoring

HoloLens avec l’application