**Projet EVAH - README**

Printemps 2021

Client : eManRisk

Product Owner : KUBICKI Sébastien

Étudiants travaillant sur ce projet (Scrum Team) : BRANDILY Hugo

ELEONORE Solène

IQUEL Fergal

HERBRETEAU Guillaume

**Présentation**

Ce projet a été commencé en 2021 au semestre de printemps dans le cadre du module de PRI. Il a été réalisé sous Unity (version 2019.4.20f1).

L'objectif du projet est de concevoir un environnement virtuel représentant un laboratoire de supermarché servant à la conception de produits alimentaires afin de former de nouvelles recrues.

Pour télécharger le projet, veuillez cloner le répertoire git suivant : <https://git.enib.fr/g6herbre/pri-evah.git>

"MainScene.unity" est la scène principale, celle qui représente le laboratoire, elle se trouve dans le répertoire *Assets/Scenes/*.

Cette scène est composée d'une salle principale avec une table, un four, un réfrigérateur, un lavabo, un distributeur de savon, un distributeur de serviettes, un burger qui sert d'aliment "test"

Il y a aussi un couloir qui mène vers une autre pièce représentant la chambre froide.

**Avertissement** :

Travailler à plusieurs via Git sur un projet Unity peut entraîner de nombreux conflits.

La meilleure chose à faire est de :

-avoir une scène personnelle par développeur

-créer et partager des objets, assets, ...etc en les passant en prefabs

-n'avoir qu'une seule personne à la fois qui modifie la scène principale

Le projet utilise "SteamVR" qui offre de nombreux objets, scripts, attributs et méthodes facilitant l'interaction avec l'environnement virtuel. Si besoin, vous pouvez le trouver gratuitement dans l'Asset Store Unity puis l'importer.

Voici un lien qui permet de facilement prendre en main "SteamVR" : <https://valvesoftware.github.io/steamvr_unity_plugin/articles/Interaction-System.html>

**Organisation**

Ce projet a été réalisé en suivant les méthodes agiles, plus précisément la méthode Scrum.

Il a été divisé en 3 sprints de 4 semaines (avec 1 jour de travail sur le projet par semaine).

Les tableurs de sprints contiennent :

- les tâches à réaliser (Backlog)

-les avancées effectuées chaque jours (Kanban)

-le bilan des tâches réalisées pendant le sprint (Review)

-le bilan des points positifs/négatifs du sprint (Retrospective)

**Scripts**

Les Scripts du projet sont présents dans le répertoire *Assets/Scripts*.

Voici la liste des scripts servant au projet présentés de la manière suivante :

<**Nom**>.cs (<**types des variables publiques**>) : <**Description**>.

areaTrigger.cs : script détectant la présence du joueur autour de l’objet

Clock.cs : Placé sur l’horloge

Dirty.cs :placé sur …

DisplayTimeUnderWater.cs (Canva) : placé sur la "Collide Box" sous le robinet. Permet d'afficher un slider devant la caméra de l'utilisateur qui se remplit en 30s, il représente le temps que pass l'utilisateur à se laver les mains.

DropSoap.cs (GameObject) : placé sur la "Collide Box" sous le distributeur de savon. Permet de distribuer une dose de savon représentée par un système de particules.

EchelleMovement.cs : placé sur …

Food.cs : placé sur …

FPSController.cs : placé sur …

Fridge.cs : placé à l'intérieur du réfrigérateur et à l'intérieur de la chambre froide. Placer un objet avec le tag "Food" fera diminuer l'attribut "temperature" de l'objet.

Oven.cs : placé à l'intérieur du four. Placer un objet avec le tag "Food" fera augmenter l'attribut "temperature" de l'objet.

PickNapkin.cs : placé sur le carré blanc représentant une serviette. Permet à l'utilisateur de prendre une serviette, l'objet deviendra alors sensible à la gravité et sera dupliqué à l'emplacement initial.

ScreenOnWrist.cs : placé sur …

Sink.cs :placé sur le système de particules représentant l'eau qui coule du lavabo. Si l'utilisateur approche assez prêt de se dernier, l'eau coulera automatiquement.

Sonde.cs :placé sur …

Test.cs : //

TriggerDoorController.cs : placé sur les "Box Collider" sur les poignées (porte, réfrigérateur, four). Ouvre automatiquement la porte lorsqu'une main de l'utilisateur passe dans le "Box Collider".

**Prefabs**

Des objets créés peuvent être convertis en “Prefabs” sur Unity. Cela permet d’archiver cet objet et de le réutiliser à maintes reprises. Modifier une instance de cet objet modifiera le Prefab.

Ils sont tous disponibles dans *Assets/Prefabs*.

Utilisation de "Universal Render Pipeline" (URP).

L'un des objectifs est d'intégrer à l'environnement une tablette permettant à l'utilisateur de se servir du logiciel conçu par eManRisk.

Via ce logiciel, un utilisateur peut assurer un suivi des produits du laboratoire.

Nous n'avons pas eu le temps d'intégrer cela au projet pour le moment mais nous avons mené différentes réflexions et recherches pour mettre cette fonctionnalité en place.

Une des solutions qui a été imaginée est de permettre une communication entre l'environnement Unity et le logiciel. Cela se ferait grâce à des requêtes http envoyées suites à des actions de l'utilisateur, une capture d'écran serait ensuite prise et affichée sur la tablette virtuelle.

Cette même tablette serait couverte de capteurs pour détecter les zones touchées par l'utilisateur. Cela est possible en utilisant la bibliothèque "PuppeteerSharp".

Voici un lien vers la documentation : <http://www.puppeteersharp.com/api/index.html>

La seconde option, qui paraît plus simple mais qui pourrait donner un rendu moins "réaliste", serait de simplifier les fonctions proposées par le logiciel et de les afficher devant la caméra de l'utilisateur via un Canva avec lequel il peut interagir.

L'utilisateur aurait à effectuer une action avec la tablette et le Canva s'afficherait.

Pour en savoir plus sur les l'utilisation de l'environnement vous pouvez consulter le manuel utilisateur (Manuel Utilisateur.pdf).