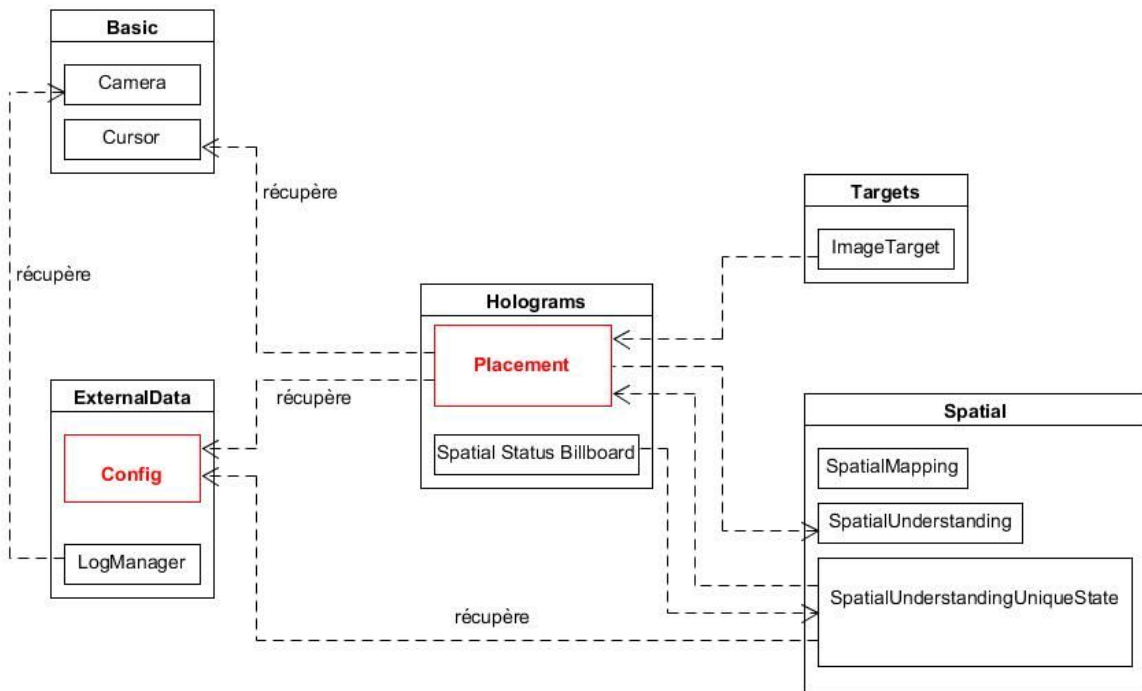


Description des différents scripts



DESCRIPTION DE LA SCÈNE

I. Basics

1) Control

- Control est un prefab permettant de gérer les entrées, que ce soit les entrées sur la scène de Unity Editor (clics, clavier) mais aussi dans l'HoloLens (gestures). Il est indispensable au fonctionnement de l'application
-

2) BiCamera

- BiCamera est un prefab contenant les fonctions de deux cameras : Vuforia et HoloToolkit. Les scripts Vuforia (Vuforia Behaviour et Default Initialization Error Handler) permettent la reconnaissance des Target et contient la configuration des éléments Vuforia. Le script Manual Gaze Control permet de contrôler la caméra avec le clavier et la souris afin de bouger dans la scène de Unity Editor

3) Cursor

- Cursor est un prefab contenant le curseur visible dans l'application. Le script WorldCursor permet de placer le curseur et de récupérer les objets avec lesquels il entre en contact

II. Spatial

1) Spatial Mapping

- Spatial Mapping est un prefab de base de HoloToolkit permettant de définir les limites du monde réel afin de permettre aux holograms de communiquer avec

2) Spatial Understanding

- Spatial Understanding est aussi une prefab de base de Mixed Reality ToolKit permettant de comprendre l'environnement réel en se basant sur la Mesh de Spatial Mapping. En effet cet outil permet de reconnaître des murs, le sol et les autres surfaces. Dans notre cas on utilise le Spatial Mapping avec le script ObjectPlacer afin de trouver le meilleur placement possible d'objets.

3) Spatial Understanding Unique State

- Ce GameObject était créé afin de délimiter le scan et d'avoir un minimum de scan permettant le bon fonctionnement de l'application. De plus il permet d'afficher le mesh du Spatial Mapping durant l'application et d'activer ou non les Taps. Il déclenche la fin du scan et appelle la création des objets (du ObjectPlacer).

III. Holograms

1) Spatial Status Billboard

- Affichage en temps réel du statut de Spatial Understanding Unique State lors de la phase de scan.

2) Placement

Placement est composé de plusieurs scripts:

- Object Placer
Ce script permet à l'aide de Spatial Understanding de trouver les emplacement les plus cohérents pour les objets afin d'être en harmonie avec l'environnement. C'est ici qu'on déclare toutes les contraintes nécessaires afin de placer les objets (distance entre objets, objets proche d'un point ou au contraire objet loin d'un point). Pour le mode debug il y a une possibilité d'afficher les boîtes de debug afin de connaître les positions données par Spatial Understanding. Les positions trouvées par Object Placer sont ensuite envoyée à Object Collection Manager à fin d'être créé.
- Object Collection Manager
Object Collection Manager gère l'ensemble des objets. En effet, en récupérant les positions des objets, Object Collection Manager associé aux divers objets instancié par la suite des prefabs correspondants. Les objets créés devient des enfant d'Object Collection Manager
- Gaze Interaction Manager
 - Ce script permet d'interagir avec les différents objets présents avec le regard. En effet, il envoie le message à ses enfants et ce message est géré par eux-même.
- World Anchor Manager
 - Ce script permet la création de World Anchor et le stockage de ces derniers, permettant de garder en mémoire la position des objets pour les futures utilisations de l'application
- Spatial Mapping Renderer
 - Ce script permet de faire de l'occlusion, c'est-à-dire à cacher les objets virtuels si ceux-ci se trouvent derrière des objets réels

IV. Targets

- Ce prefab contient les ImageTarget, prefabs Vuforia permettant de détecter certaines images prédéfinies et de lancer des scripts à la détection de ces derniers. Les ImageTarget ont en fils les éléments qu'elle affiche si on regarde la cible

V. ExternalData

1) Config

- Config est une classe permettant la lecture du fichier XML myConfig.xml. On peut ainsi récupérer les éléments de cet xml en fonction du nom des valeurs, avec les valeurs sous forme de double, int, string et bool.

2) LogManager

- LogManager permet 2 choses : écrire dans un fichier texte la position de l'HoloLens dans le temps, afin de pouvoir étudier ces valeurs en les récupérant ultérieurement et de récupérer la distance parcourue par l'utilisateur durant l'utilisation de l'application sous la forme d'un fichier texte.

-Ressources Externes :

Meshs:

Attribution (pour coffre) mrsCrash

<https://sketchfab.com/models/92a8d45d2d5c43b8b0b5bf7d97816ada>

(pour clé) Pedrovalente

<https://sketchfab.com/models/422b4acc7db04fa3b6b7e80bcb2840f8>

Attribution pour le bus Yo ri

<https://sketchfab.com/models/f895bb31d5d244f6828608f438189993>

Licence pr les meshs

<https://creativecommons.org/licenses/by/4.0/>

Sons :

coins falling

<https://www.youtube.com/watch?v=miZpDS2HWmA>

keyCollected

<http://soundbible.com/2068-Woosh.html>

chestOpening

<https://freesound.org/people/spookymodem/sounds/202092/>

psst hey

<https://freesound.org/people/SkyRaeVoicing/sounds/363585/>

Simatic autobus

<https://www.innovation-pedagogique.fr/article1816.html>