05/01/2019

ÉCOLE NATIONALE SUPÉRIEUR
D'ÉLECTROTECHNIQUE, D'ÉLECTRONIQUE, D'INFORMATIQUE
D'HYDRAULIQUE ET DES TÉLÉCOMMUNICATIONS
INSTITUT NATIONAL POLYTECHNIQUE

INSTITUT TOULOUSAIN D'OSTÉOPATHIE

Documentation Technique : Osteo - Android

Projet Long 2019 :
Détection de pathologie cervicale par Réalité virtuelle

Membres : Txomin Itoiz, Clément Brunie, Thibaud Ishacian, Lucien Haurat, Clément Guillaumin, Edmond Boulet-Gilly
Chef de projet : Edmond Boulet-gilly

Version 1.0

Introduction

Ce papier sert de Documentation Technique pour accompagner le développement l'application Android Osteo développé lors du Projet Long de 2019 par les membres du projet désigné au dessus. L'application permet de paramétrer un environnement de test en réalité virtuel disposant d'un cible qui se déplace de droite à gauche et d'enregistrer les mouvements de tête de l'utilisateur. Les mouvements de tête sont ensuite enregistrées localement et peuvent être envoyer par mail, l'idée étant que ces données soient exploités par un autre logiciel pour détecter les pathologies cervicales.

Le test dans l'environnement VR constitue à suivre du regard une cible qui se déplace en arc de cercle à droite et à gauche de l'utilisateur à une vitesse constante et en effectuant une pause à chaque bout de l'arc de cercle.

Prérequis et Plateforme ciblée

Le développement et la compilation ont été effectué sous Unity 2018.3.2f1, comme la plateforme visée est android il est nécessaire d'avoir android studio installé pour profiter du SDK-Android et Unity devra détecter ce dernier pour fonctionner correctement.

L’application nécessite aussi des packages annexes (compris dans le code source fournis) tel que :

- *GoogleVR* package for unity
- *Textmesh Pro* (inclus la plus part des distributions unity)

Compiler en .apk

Une fois les prérequis remplis, il est possible d’ouvrir le projet à l’aide du launcher Unity en sélectionnant le dossier “*VRApp*”.

Après avoir ouvert le projet, allez dans File > Build Settings, sélectionnez Android, vous avez ensuite la possibilité de compiler un .apk ou bien de compiler le .apk et de lancer l’application sur un téléphone connecté au PC. Dans tous les cas il est possible d’installer l’application depuis le .apk à postériori.

Architecture générale

L’application est séparée en 2 scènes : une scène dédiée au menu principale “MainMenu” et une scène dédiée à l’environnement VR “App”.

Les script responsables du passage d’une scène à l’autre sont *SceneLoader* et *VREnabler* pour passer du menu principale à la VR et *LoadSceneVR* et *VRDisabler* pour le sens inverse.

De plus des arguments sont passés d’une scène à l’autre avec les conteneurs “*PlayersPrefs*”, les scripts qui sont responsables du passage d’arguments entre scènes sont situés dans /Asset/ParameterPasser.

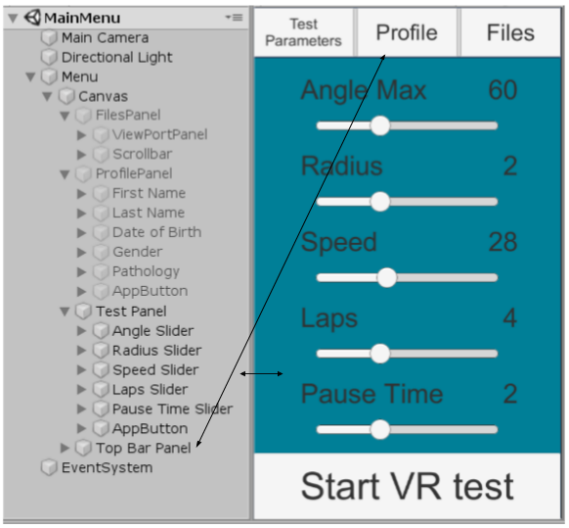
Les dossiers *MenuScript* et *VRScript* contiennent respectivement les scripts dédiés au menu et à l’environnement VR.

Le Dossier *Ressource* contient les Prefab (entité Unity préfabriquée), nécessaire au fonctionnement du menu, c’est le seul dossier qu’il est impératif de ne pas mélanger avec les autres, le bon fonctionnement de script lié au menu “*FilePanel*” repose sur son emplacement.

Menu Principale

Scène

Le menu principale permet d’accéder à 3 sous menus grâce aux 3 onglets supérieur puis de lancer le test avec le le bouton inférieur “Start VR Test”.



La scène menu principal est constitué d'une entité "Menu" parent d'un "Canvas", la barre supérieur onglet "*Top Bar Panel*" et les 3 menus "Test Panel", "*ProfilePanel*" et "*FilesPanel*" liées aux onglets supérieurs respectivement "*Test Parameters*", "Profile et "*Files*".

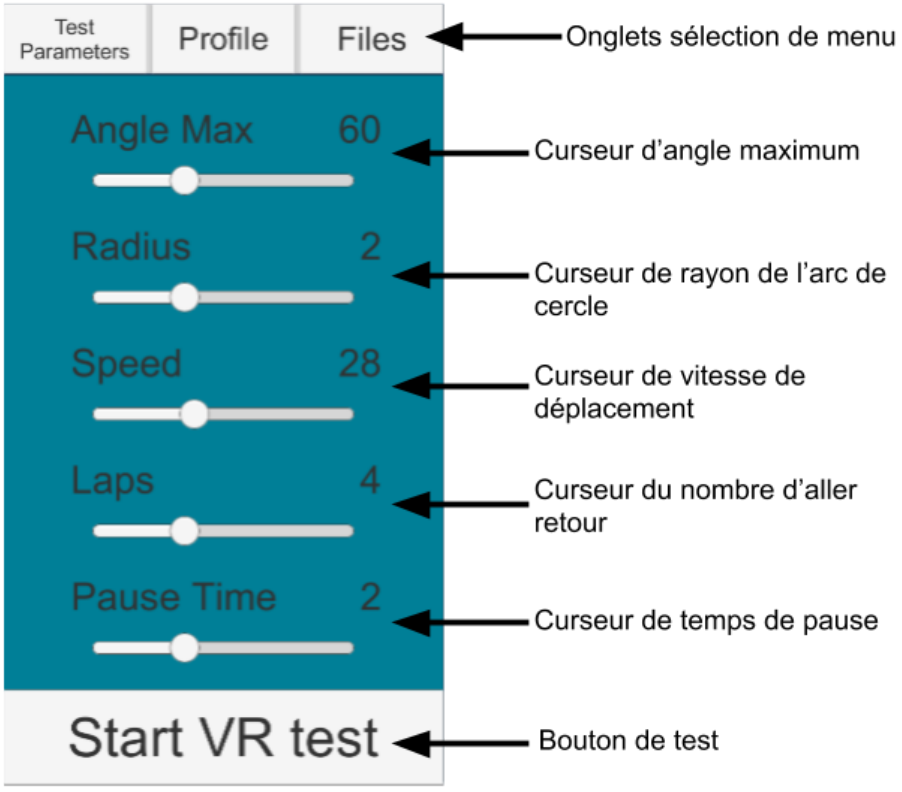
Les boutons de "*Top Bar Panel*" activent et désactivent les entités menus à l'aide d'event Unity propre à chacun des boutons.

Seul les 2 premiers sous-menus ont un "*AppButton*" qui permet d'accéder à la seconde scène.

Paramètre de test

Le premier onglet , " Test Parameters " repose sur une série de curseurs glissant qui sont chacun liés à leur script spécifique :

- Angle Max : *AngleMaxPasser* écrit la valeur donnée dans "*AngleMax*"
- Radius : *RadiusPasser* écrit la valeur donnée dans "*Radius*"
- Speed : *SpeedPasser* écrit la valeur donnée dans "*Speed*", la valeur affichée n'est pas la même que la valeur réelle du slider et c'est encore une autre valeur qui est passée à l'autre scène en raison de la conversion à la vitesse angulaire et que le slider ne supporte pas bien les décimales.
- Laps : *MaximumLapsPasser* écrit la valeur donnée dans "*MaximumLaps*"
- Pause Time : *PauseTimePasser* écrit la valeur donnée dans "*Pause Time*"



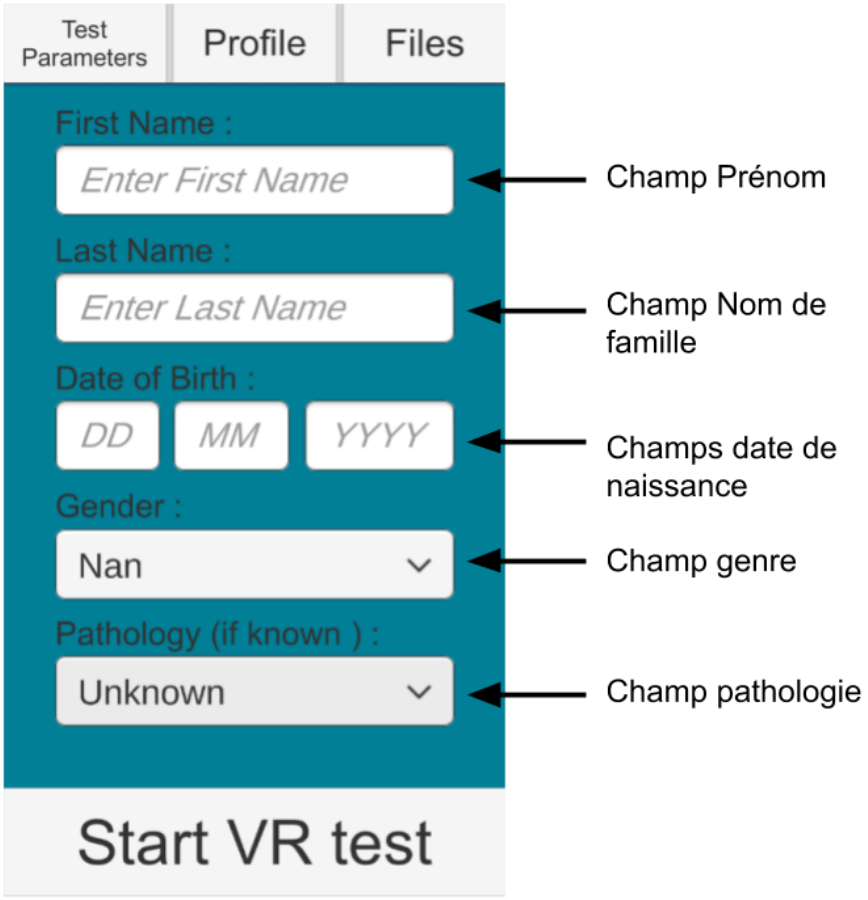
Profile

Le second onglet propose d'entrer un certain nombre d'information sur l'utilisateur, elles sont toutes passées en tant que string avec le script *StringPasser*:

- Le prénom écrit la valeur donnée dans "*FirstName*"
- Le nom écrit la valeur donnée dans "LastName"
- La date de naissance écrit la valeur donnée dans "*DayBirth*", "*MonthBirth*" et "*YearBirth*"
- Genre écrit la valeur donnée dans "*Gender*"
- Pathologie (si connue) écrit la valeur donnée dans "*Phatology*"

Du fait que le genre et la pathologie ne sont pas de champs mais des menus déroulants ils nécessitent aussi le script *DropDownHelp* pour lire la valeur sélectionnée.

Si l'utilisateur ne remplit pas ces champs, le string vide sera pris comme paramètre par la scène VR.



Gestion de Fichiers

Cette onglet donne un liste des fichiers enregistrées, les fichiers sont triés par noms est sont labélisé d'un I si l'enregistrement a été interrompu.

Ce menu est composé d'une barre de défilement et d'un espace pour faire apparaître le nom de chaque fichier et des boutons. Le script responsable de faire apparaître la liste des entités qui permettent d'interagir avec les fichiers est *FileListSpawner*.

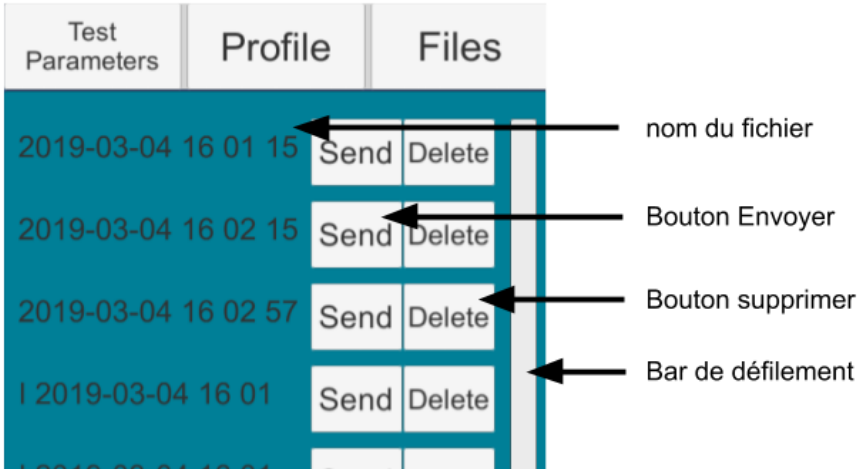
La barre de défilement permet d'accéder à la liste complète si il y a trop de données pour que tout rentre sur un seul écran.

Les boutons font appellent au script *DataTool*:

Le bouton "*Send*" permet de créer un email dont le corps du texte comporte les données, pour cela l'application fait appel à la fonction "OpenURL("mailto..")".

Le bouton "*Delete*" supprime les données du téléphone, et donc de la liste.

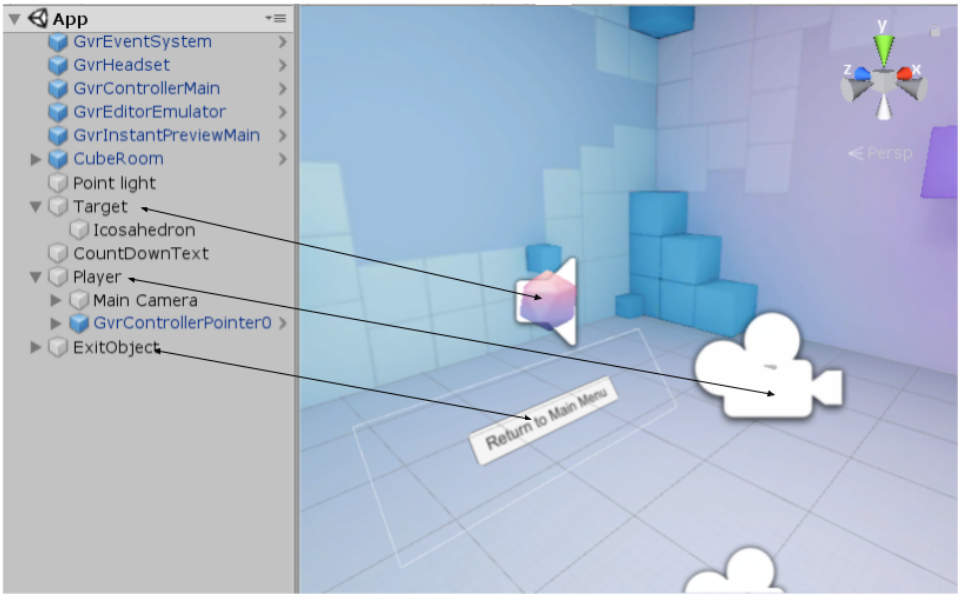
Les entités qui apparaissent dans la liste sont tirées des Prefabs, il est possible de trouver cette dernière est localisée à l'emplacement "*Ressource/DataItem*".



Environnement VR

Scène

Après avoir appuyer sur bouton Start VR Test, la deuxième scène se charge. Il est possible de la charger directement en double cliquant sur cette dernière dans l'explorateur de l'éditeur. Dans les 2 cas, tant que vous êtes dans l'éditeur, l'environnement VR sera chargé comme une seule fenêtre et non pas 2 rendus cote à cote pour chaque oeil. Cette scène est simplement une copie de la scène fournie en exemple par *GoogleVR* à laquelle des scripts et des entités ont été ajoutés.



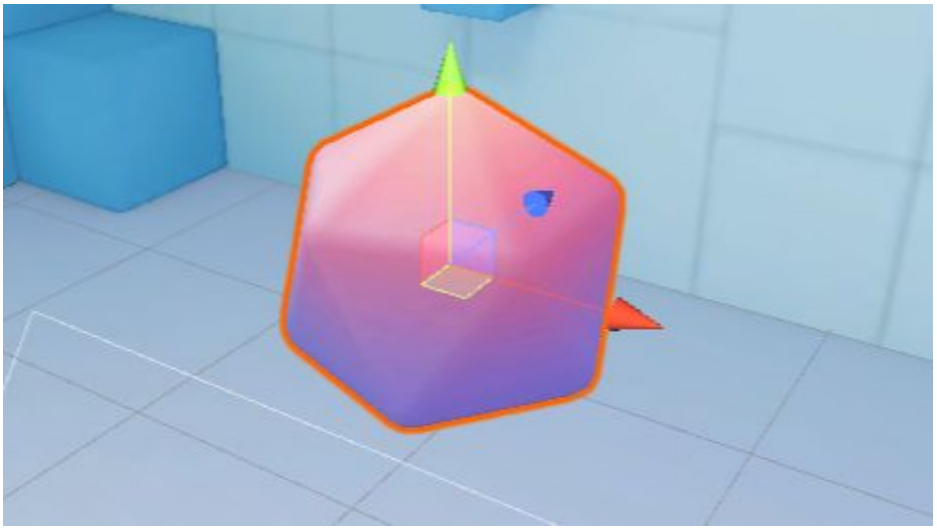
Target

Entité représentant la cible que l'utilisateur va suivre du regard.

Dispose du script *MoveTarget* responsable du mouvement de la cible, du calcul de sa trajectoire et de ses temps de pauses.

Ce script nécessite au script *MoveCapture* pour notifier de l'initiation,l'interruption, l'annulation ou de la complétion de du test.

L'entité Icosahedron représente la cible dans l'environnement 3D. A travers le système d'évent Unity, l'entité change de couleur quand l'utilisateur la regarde et notifie les scripts *CountDown* et *MoveTarget*.



CountDownText

Entité affichant le compte à rebour pour lancer le test.

Dispose du script CountDown, responsable pour l'affichage du décompte et initier le départ de la cible.

Nécessite un accès à *MoveTarget* pour notifier l'entité de la fin du décompte et lancer le mouvement.

Player

Entité lié à la perspective du joueur.

Dispose du script *MoveCapture*, responsable pour la capture du mouvement de tête de l'utilisateur. Le format d'enregistrement des données a été conçues pour être homogène à l'application windows développé lors du projet long 2017 par l'ENSEEIHT.

Les données sont écrites dans le répertoire persistant de l’application, spécifique à chaque plateforme, sous forme de *.txt* .

ExitObject

Entité permettant de revenir au menu principale depuis l’environnement VR.

Dispose du script *SceneLoader* et *LoadScene* VR, responsable pour initier le décompte, l’annuler si l'utilisateur arrête de regarder et changer de scène si il est complété.

Notes

Il est possible que l’utilisateur entre des lettres à la place de chiffre pour sa date d’anniversaire.

Il est possible de regrouper les scripts qui passent les arguments de paramètre de test et de profile pour éviter la duplication de script très similaire.

Un aller retour entre la scène VR et le menu efficace les données enregistrée dans le menu profile.

L’envoie des données se fait par corps de mail et non par pièce jointe.

Dans l’environnement VR il y a 4 scripts dont la méthode *void Update()* est appelée à chaque image, pour réduire l’impact en performance il serait pertinent d’activer et de désactiver les scripts dynamiquement.