

X REALITY – Projet Manga 4A – IA & Data



ASSOUMA Tata Joseph
HOUESSOU Boris
ANAGO Yoann

DIA Issa
DE SEVIN Alexandre
ZEIDAN Carla

Table des matières

Introduction.....	2
Présentation du projet.....	3
Outils mis en œuvre	3
Développements effectués dans ces outils.....	3
Dialogue des personnages	3
Création des personnages et animation.....	4
AR et Tracking d'images	4
Audio.....	4
Vidéo.....	4
Scripts	5
Image utilisée pour le tracking	6

Introduction

Dans le cadre de notre Projet X Reality de quatrième année à l'ESIEA, nous avons pour but de découvrir l'environnement Unity et effectuer un projet pour découvrir l'univers complexe de la réalité virtuelle.

Ce projet doit répondre à un ensemble de critères et d'exigences tel qu'il soit bien évidemment fonctionnel ou qu'il existe une interaction utilisateur.

Pour cela nous avons formé une équipe de six personnes composées de ASSOUMA Tata Joseph, DIA Issa, HOUESSOU Boris, DE SEVIN Alexandre, ANAGO Yoann et ZEIDAN Carla.

Cette équipe a été créée en tenant compte des spécialités IA et affinités de chacun afin d'avoir une meilleure ambiance de travail tout en étant productif et de maximiser nos chances de réussite.

Présentation du projet

Notre projet Manga consiste à créer une application permettant de découvrir l'univers des mangas.

Le choix des mangas en particulier peut s'expliquer par les goûts personnels des membres de l'équipe. Nous nous sommes inspirés du concept que Beyonce a mis en place lors de la sortie de sa nouvelle collection Ivy Park. Nous avons pour objectif de découvrir le monde de la réalité virtuelle avec un sujet qui nous passionne.

Ce projet doit permettre à l'utilisateur de découvrir certains mangas. En effet lorsqu'on clique sur le personnage souhaité, nous avons accès aux informations sur le manga, une vidéo le tout sur des fonds musical du manga en question. Nos personnages dansent sur la musique pour un aspect plus amusant et original.

Outils mis en œuvre

Nous avons utilisé l'univers de développement Unity recommandé et vu en cours avec notre professeur Monsieur Le Renard. De plus ayant fait des recherches sur les différents univers de développement Unity nous a paru le plus simple d'utilisation contenu des nombreux tutos présents sur internet et les assets qui sont très utile évitant une surcharge de script. Nous avions pour ambition de pouvoir faire par la suite de faire notre projet sur mac pour qu'il puisse fonctionner sur un Iphone mais malheureusement par manque de temps cela n'a pas été possible. Pour les personnages et les animations nous les avons trouvés sur le site Mixamo.

Développements effectués dans ces outils

Durant le projet, nous avons expérimentés plusieurs outils de développement et plusieurs idées qui se sont pour certains été un vrai échec même si nous avons tous de même appris énormément de chose. Dans cette partie, nous allons présenter les outils qui ont été un succès.

Dialogue des personnages

Pour les dialogues des personnages nous avons utilisé un asset présent sous Unity. Pour l'utiliser nous avons besoin d'écrire un script qu'on a nommé « *PersonnageConversation* » pour récupérer le dialogue, créer un objet vide qu'on appelait par exemple « *Conversation Naruto* » puis on lui attribuait le script déjà présent « *NPC Conversation* » et enfin d'un « *Canvas* » pour l'affichage du dialogue. Il nous restait plus qu'à ajouté le tout dans le personnage Naruto et lui ajouté un component « *box collider 2D* » et le dialogue s'affiche en cliquant vers le bas du personnage.

Création des personnages et animation

Nous avons créé 3 personnages à savoir : Naruto, Michelle et Kachujin.

Le personnage Naruto est un modèle que nous avons récupéré sur le site « <https://free3d.com/3d-models/naruto> ». Pour pouvoir l'animer nous l'avons uploadé sur le site Mixamo qui nous permettait d'avoir des animations de characters 3D.

Les personnages Michelle et Kachujin sont des personnages du site Mixamo.

Pour pouvoir les animé sur Unity, nous avons utilisé le component « *Animator* » pour pouvoir intégrer les mouvements que nous voulions.

AR et Tracking d'images

Pour pouvoir faire la partie Tracking d'image pour notre projet, nous avons utilisé la librairie Vuforia. Cette librairie est facile à prendre à main, surtout pour des débutants que nous somme.

Pour pouvoir intégrer nos personnages dans l'espace nous avons utilisé « Ar Camera », la camera dédié pour faire du AR avec Vuforia et pour le tracking d'image nous avons utilisé « Image Target ».

Nous avons créé une base de données sur le site Vuforia afin de pouvoir intégrer les images que la camera pourra détecter pour afficher nos personnages.

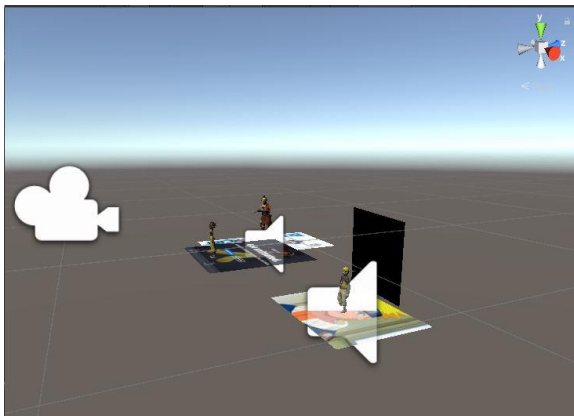
Audio

Ensuite pour intégrer la musique à nos personnages Naruto et Kachujin, nous avons créé un script « *ImageTargetPlayAudio* » qui, permet de lancer la musique lorsqu'il détecte l'image de chaque « *Image Target* » de nos personnages.

Nous avons ensuite ajouté notre musique au component « *AudioSource* » de l'image Target de nos personnages.

Vidéo

Pour intégrer la vidéo dans l'environnement, nous avons utilisé le component « *Video Player* » de Unity que nous avons ajouté à un « *GameObject* » préalablement créer pour contenir notre vidéo sur l' « *Image Target* ».



Affichage finale de
notre scène

Scripts

```

C# PersonnageConversation.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using DialogueEditor;
5
6  public class PersonnageConversation : MonoBehaviour
7  {
8      public NPCConversation myConversation;
9
10     private void OnMouseOver()
11     {
12         if (Input.GetMouseButton(0))
13         {
14             ConversationManager.Instance.StartConversation(myConversation);
15         }
16     }
17 }
18

```

Script pour les dialogues des personnages

```

C# ImageTargetPlayAudio.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using Vuforia;
5  using UnityEngine.Playables;
6
7  public class ImageTargetPlayAudio : MonoBehaviour, ITrackableEventHandler
8  {
9      private TrackableBehaviour mTrackableBehaviour;
10     public AudioSource AudioObject;
11     // Start is called before the first frame update
12     void Start()
13     {
14         GetComponent();
15         mTrackableBehaviour = GetComponent<TrackableBehaviour>();
16         if (mTrackableBehaviour)
17         {
18             mTrackableBehaviour.RegisterTrackableEventHandler(this);
19         }
20     }
21
22     public void OnTrackableStateChanged(TrackableBehaviour.Status previousStatus, TrackableBehaviour.Status newStatus)
23     {
24         if(newStatus == TrackableBehaviour.Status.DETECTED ||
25             newStatus == TrackableBehaviour.Status.TRACKED ||
26             newStatus == TrackableBehaviour.Status.EXTENDED_TRACKED)
27         {
28             AudioObject.Play();
29         }
30         else
31         {
32             AudioObject.Stop();
33         }
34     }
35 }

```

Script pour lancer un audio lorsqu'on détecte une image

Image utilisée pour le tracking

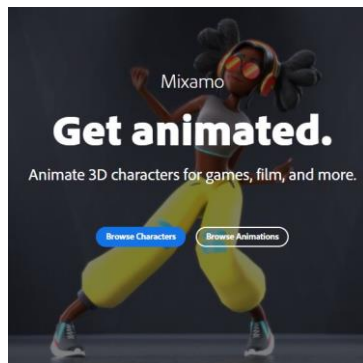


Image utilisée pour faire
apparaître le personnage
Michelle

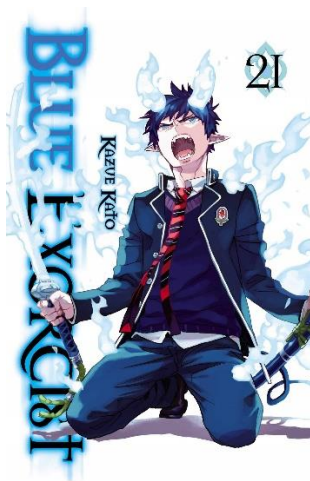


Image utilisée pour faire
apparaître le personnage
Kachujin



Image utilisée pour faire
apparaître le personnage
Naruto