

UNIVERSITÉ NATIONALE DU VIETNAM, HANOI (UNVH)



Option : Système intelligent et multimédia

Promotion : 22

Niveau : master II

Module : Réalité Virtuelle et Augmentée

Professeures : Dr. Thi-Lan Le et Dr. Thanh-Hai Tran

Rapport-Projet final

Création d'une tour Virtuelle pour le Musée d'Ethnographie du Vietnam

Présenté par :

- *Bendy LATORTUE*
- *Mongetro GOINT*
- *Myderson SEMEURAND*
- *Rooldy ALPHONSE*

Février 2019

SOMMAIRE

I-	INTRODUCTION.....	2
II-	Description du projet.....	2
III-	Outils utilisés.....	3
IV-	Présentation conceptuelle du projet.....	3
V-	Implémentation et expérimentation.....	4
VI-	Conclusion.....	18
VII-	BIBLIOGRAPHIE / WEBOGRAPHIE.....	19

I- INTRODUCTION

Un musée est une institution permanente sans but lucratif au service de la société et de son développement ouvert au public, qui acquiert, conserve, étudie, expose et transmet le patrimoine matériel et immatériel de l'humanité et de son environnement à des fins d'études, d'éducation et de délectation. (ICOM, 2007).

Les musées servent de promouvoir la présentation des collections et aussi de veiller à l'utilisation la plus rationnelle des richesses publiques, culturelles et humaines.

D'autres parts, la visite virtuelle qui est une simulation d'un lieu existant, généralement composée d'une séquence de vidéos ou d'images fixes vient en aide à une meilleure découverte des musées, notamment en influençant sur le télé tourisme.

Dans ce contexte, nous avons choisi de créer, dans le cadre du cours de Réalité Virtuelle et Augmentée en master en informatique à l'Institut Francophone International (IFI), une tour virtuelle pour le musée d'Ethnographie du Vietnam.

II- Description du projet

Le projet consiste à mettre en pratique les différentes notions vues au cours de Réalité Virtuelle et Augmentée. Il s'agit de la création d'une tour virtuelle à l'aide des images de 360 % pour le Musée d'Ethnographie du Vietnam. A l'aide de ce tour virtuel, un utilisateur doit pouvoir visualiser, à travers des scènes créées dans Unity, différents espaces du musée d'Ethnographie du Vietnam présentés en format 3D. L'application doit non seulement être capable de tourner dans Unity, mais aussi sur n'importe quel appareil (smart phone, tablette...) avec une version minimum d'Android 4.4 "Kit Kat". Outre, sur l'appareil en question on doit aussi préalablement installer "google cardboard" qui servira de visionneuse pour examiner nos scènes.

III- Présentation conceptuelle du projet

Notre scène sera constituée de cinq (5) objets (sphères) à travers lesquels l'utilisateur pourra transiter pour visualiser les différents espaces du Musée d'Ethnographie du Vietnam.

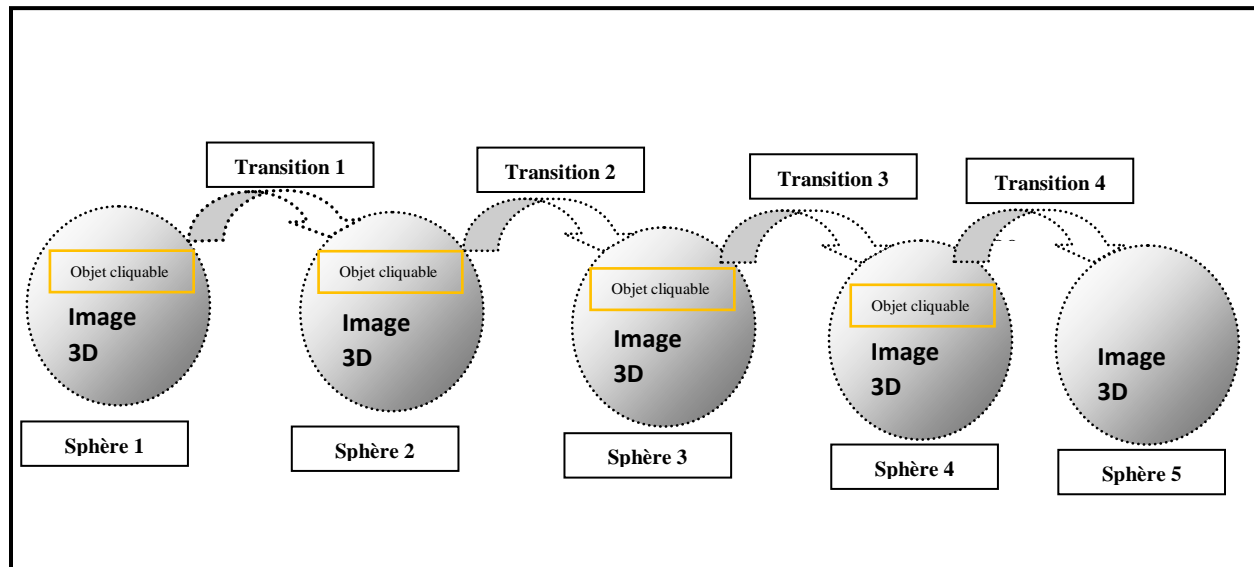


Fig1: Vue conceptuelle de notre environnement virtuel

IV- Outils Utilisés

Dans le cadre de ce projet, nous avons utilisé différents outils :

- **Unity 2017.4** : Pour le développement et la simulation de notre application de réalité virtuelle ;
- **Android studio V 3.3** : Pour paramétrer notre application afin qu'elle puisse tourner sur des appareils avec systèmes Android ;
- **GVR SDK for Unity v1.190.1** : SDK Google VR pour Unity contenant un ensemble de composants afin de créer des applications VR mobiles pour Android ou iOS ;
- **360 Tour Assets**: Un packadge contenant des composants permettant de créer des tours virtuelles.

VI- Implémentation et expérimentation

Pour la phase d'implémentation, nous avons d'abord téléchargé et installé *Android Studio*.

Ensuite, Après avoir créé notre projet dans Unity, nous avons téléchargé *GVR SDK for Unity v1.190.1* et *360 Tour Assets*, puis importé leurs composants dans Assets dans Unity.

De plus, nous avons classé les différents autres composants que nous avons utilisés dans leur dossier respectif :

- *Scripts* (contenant le script « *SphereChanger* » téléchargé via 360 Tour Assets) ;
- *Shaders* (contenant le composant « *Insideout* » téléchargé via 360 Tour Assets)
- *Textures* (contenant les différentes images 360% pour les différentes scènes) ;
- *Sprite* (contenant les images « *cliquable* » pour le passage d'une sphère à un autre) ;
- *Sound* (contenant les sons à utilisés pour notre scène).

D'autre part, nous avons supprimé le composant « *Directional Light* » dans Unity, puisqu'il n'est pas nécessaire dans notre contexte. Ensuite, nous avons procédé à la réinitialisation de la caméra.

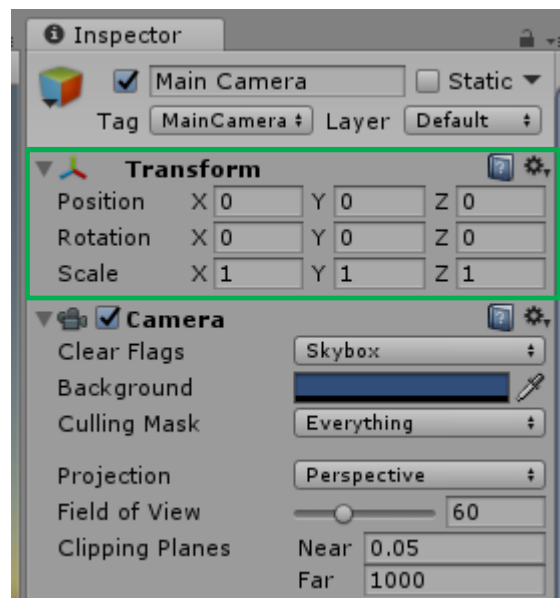


Fig2 : Image de la caméra réinitialisée

Dans une autre phase, nous avons procédé au paramétrage d'Android :

- a) « Switch platform » pour prendre en compte la plateforme Android ;
- b) « Play settings », pour les différentes configurations Android (voir Fig3, Fig4, Fig5), notamment les parties encadrées en vert.

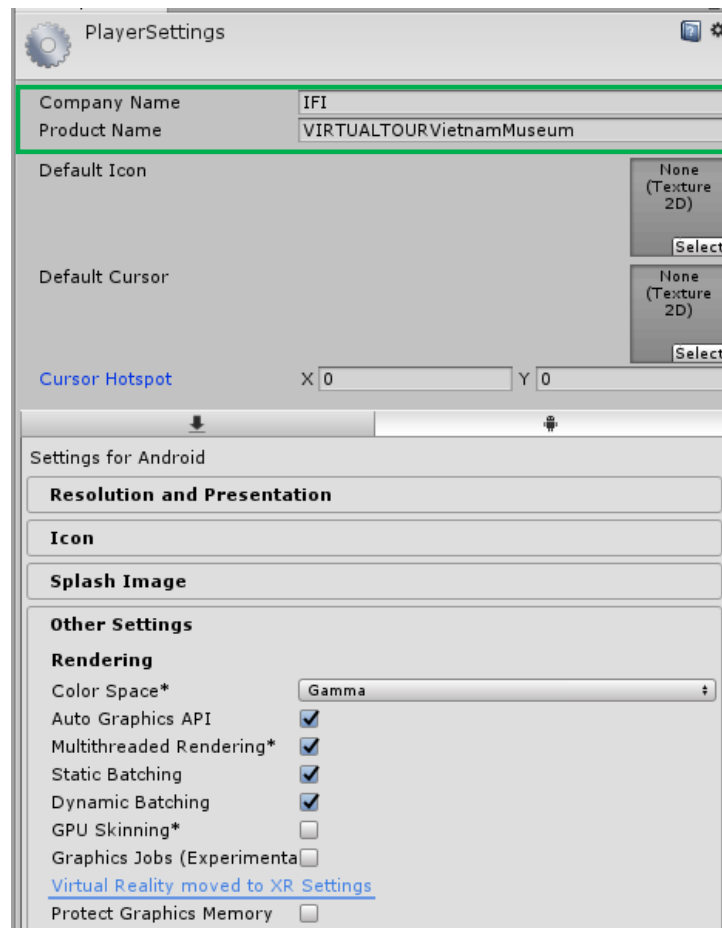


Fig3 : Illustration du paramétrage Android

Identification	
Package Name	ifi.edu.vn
Version*	1.0
Bundle Version Code	1
Minimum API Level	Android 4.4 'Kit Kat' (API level 19)
Target API Level	Automatic (highest installed)

Configuration	
Scripting Runtime Version*	Stable (.NET 3.5 Equivalent)
Scripting Backend	Mono
Api Compatibility Level*	.NET 2.0 Subset
Mute Other Audio Sources*	<input type="checkbox"/>
Disable HW Statistics*	<input type="checkbox"/>
Device Filter	FAT (ARMv7+X86)
Install Location	Prefer External
Internet Access	Auto
Write Permission	Internal
Filter Touches When Obscured	<input type="checkbox"/>
Low Accuracy Location	<input type="checkbox"/>
Android TV Compatibility	<input checked="" type="checkbox"/>
Android Game	<input checked="" type="checkbox"/>
Android Gamepad Support	Works with D-pad
Scripting Define Symbols*	
Active Input Handling*	Input Manager

Optimization	
Prebake Collision Meshes*	<input type="checkbox"/>
Keep Loaded Shaders Alive*	<input type="checkbox"/>
► Preloaded Assets*	
Stripping Level*	Disabled
Enable Internal Profiler* (Dev)	<input type="checkbox"/>

Fig4 : Illustration du paramétrage Android

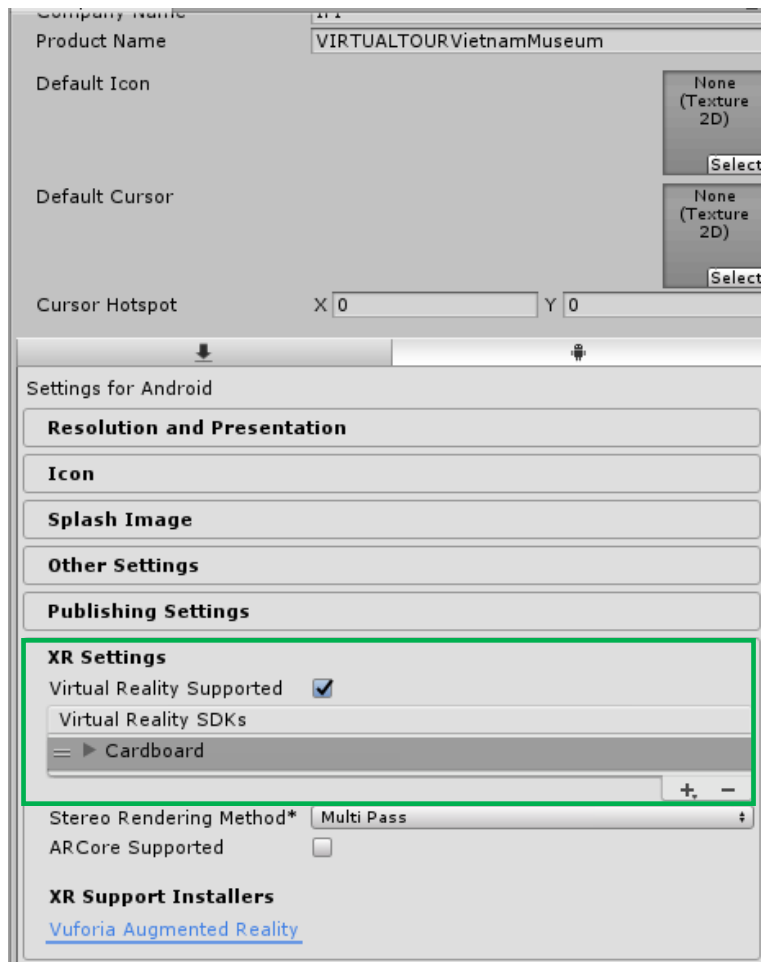


Fig5 : Illustration du paramétrage Android

Après les différentes configurations, nous avons passé à la Phase vraiment fondamentale qui est la mise en place de notre environnement virtuelle et la création de nos scènes en utilisant les composants importés préalablement dans *assets*.

- 1- *GvrEditorEmulator* (Utilisé comme émulateur pour simuler nos scènes) ;
 - 2- *GvrEventSystem* (Pour prendre en compte des évènements interactifs entre les objets) ;
 - 3- *GvrReticlePointer* (importé sur la caméra pour signaler à celle-là les actions des objets cliquables.
 - 4- *GvrPointerPhysicsRaycaster* (importé sur le composant *GvrEventSystem*)
- Pour une meilleure manipulation de la caméra, nous avons créé un objet (Tripod) et placé celle-là sur celui-ci.
 - Sur la caméra, nous avons créé un objet 3D(*Quad*), le renommé *Fader*. Cela explique que cet objet 3D placé en face de la caméra fera apparaître en 3D tout autres objets se trouvant à sa suite, en face de la caméra.

- Dans un dossier (Materials) créé dans assets, nous avons créé un nouveau matériel (Fader), changé son propriété « Shader » en « Sprite/default », puis passer la valeur de la couleur de 255 à 0 afin qu'il puisse donner une allure transparente.
 - Ce matériel a été importé sur l'objet *Fader* que nous avons créé.
- Pour pouvoir mettre en scène notre image 360° importée dans « Textures » au part avant, nous avons créé un objet 3D(Sphère), le renommer « Sphère 1 » puis enlever ses collisionneurs.
- Nous avons créé un nouveau matériel (Insideout), changé son propriété « Shader » en « Insideout ».
 - Ce matériel a été importé sur l'objet *Sphère 1* que nous avons créé afin de permettre de tourner à l'envers dans la scène, l'image qui sera placée dans la sphère.
 - Ensuite, nous avons importé une image 360° de notre répertoire « Textures » au sphère 1, et l'image ci-dessous (voir Fig6) montre une vue de la scène.

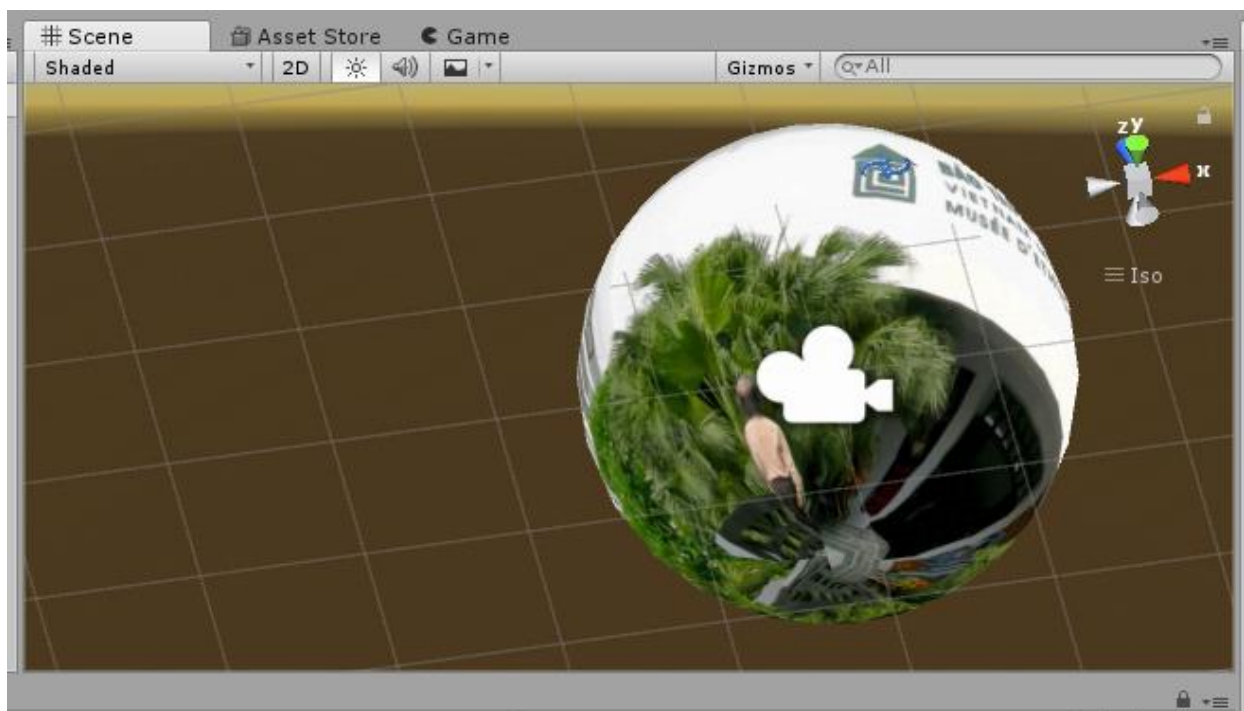


Fig6 : Illustration de la scène avec notre premier objet 3D

Ensuite, quand on démarre la scène, en tenant enfoncé la touche ALT du clavier et en bougeant la souris, on peut tourner l'image et constater une vue 3D des différentes parties de celle-ci (voir Fig7).

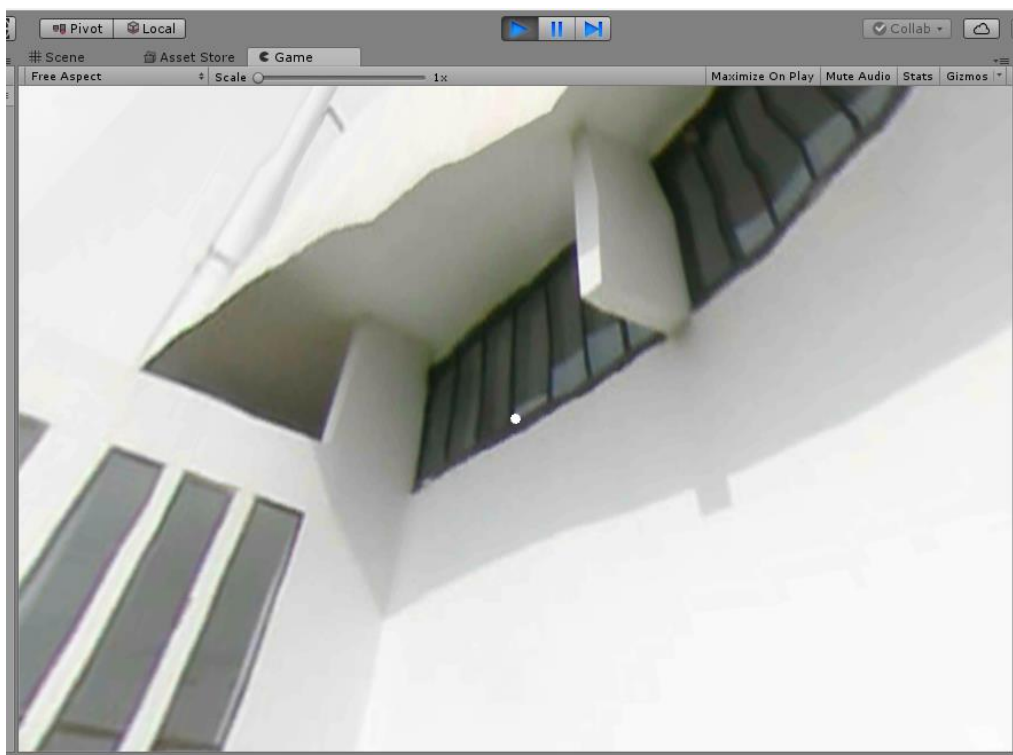


Fig7 : Illustration de la scène avec notre premier objet 3D en mode « play ».

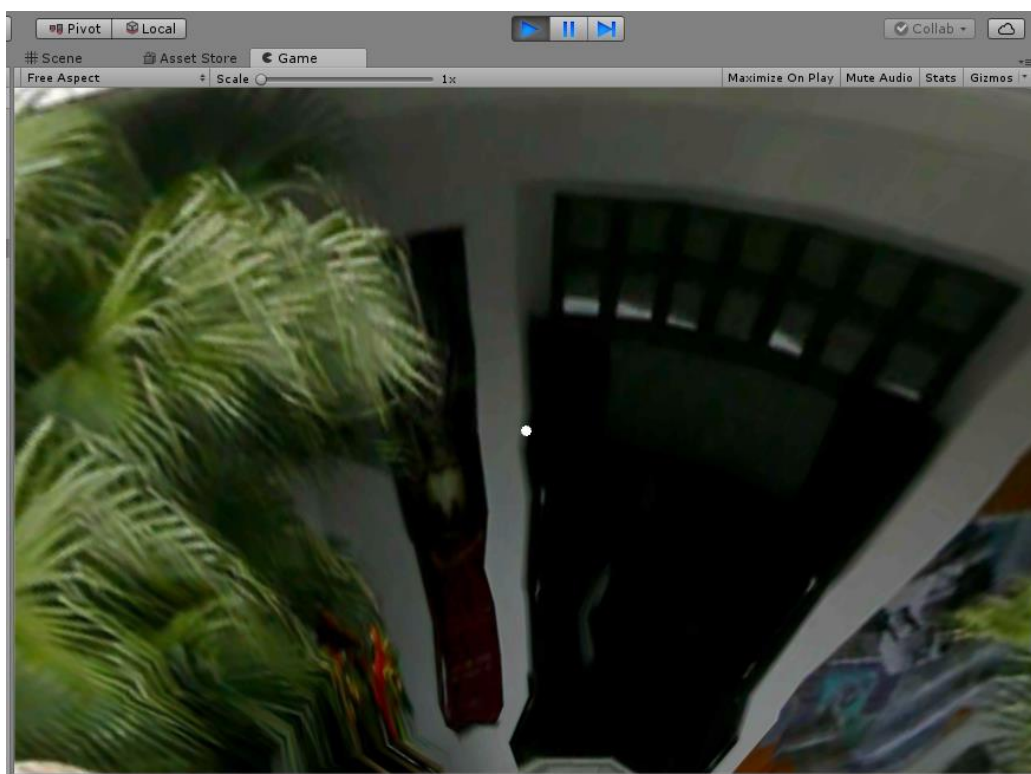


Fig8 : Illustration de la scène avec notre premier objet 3D en mode « play ».

➤ **Création d'une image cliquable pour transiter entre sphères 1 et sphère 2.**

Pour pouvoir transiter d'un objet à un autre, nous avons créé une image cliquable, qui, lorsque le curseur passera dessus, montrera un petit cercle et qui permettra de faire la transition entre objet, en cliquant.

Pour ce faire, nous avons configuré une des images (arrow1) que nous avons placée dans notre répertoire « Sprites », donc nous avons changé sa propriété « Texture Type » en « *Sprite 2D and UI* ». Ensuite, nous avons importé notre image cliquable dans la scène, l'arranger dans notre sphère, puis nous ajoutons le composant « sphere collider » pour montrer l'espace cliquable quand le curseur passera dessus. Puis, nous avons importé le script « *SphereChanger* » sur le l'objet « *tripod* » contenant la caméra, ajouté un composant « Event Tigger » afin que l'objet puis être interractif. Outre, nous avons ajouté « PointerDown » comme nouveau « Event Trigger » pour pour que l'objet puis répondre quand on clic avec la souris. En effet, pour pouvoir accéder au script « *SphereChanger* », nous avons ajouté l'objet « Tripod ». En plus, nous avons modifié la propriété « *Function* » à « *SphereChanger/ChangeSphere* » pour le changement d'une sphère à un autre.

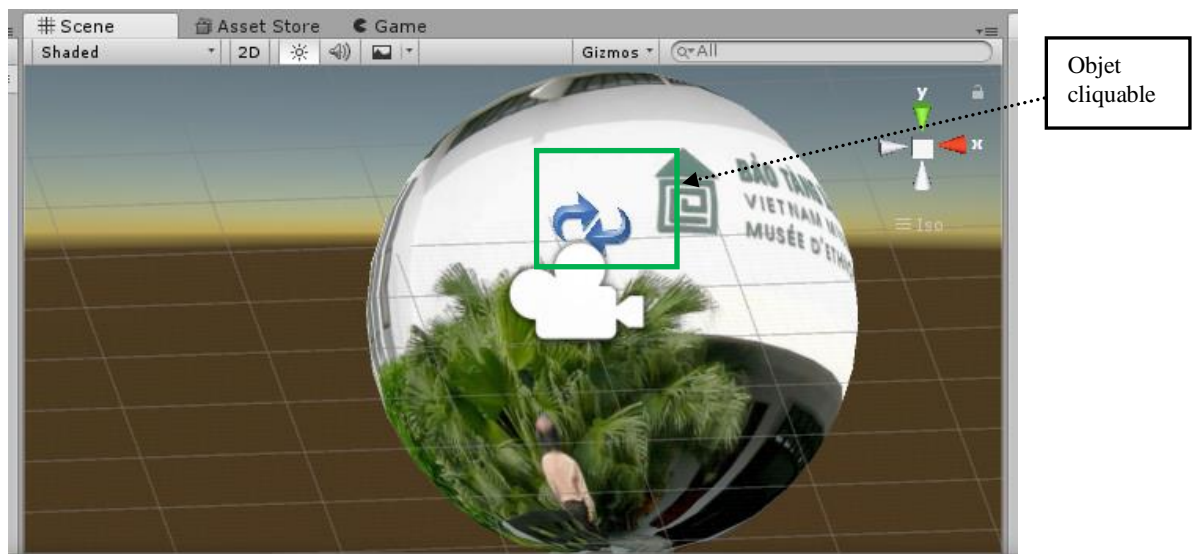


Fig9 : Illustration de la scène avec notre objet cliquable

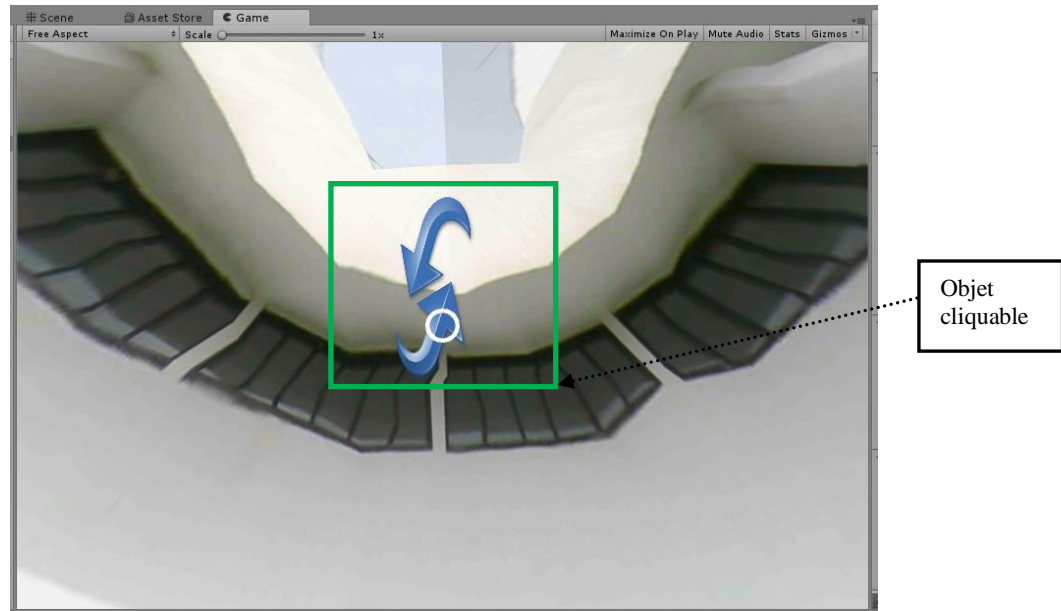


Fig10 : Illustration de la scène avec notre objet cliquable montrant la possibilité de changement en mode « play »

Enfin nous avons créé une nouvelle sphère, et répété la même procédure qui a été appliquée pour la première sphère. Outre, dans notre objet cliquable, nous avons importé cette sphère dans « *None Transform* » afin qu'elle soit prise en compte par l'objet cliquable. Puis, la même procédure a été répétée pour la création d'un objet cliquable sur les autres sphères pour passer aux sphères suivantes.

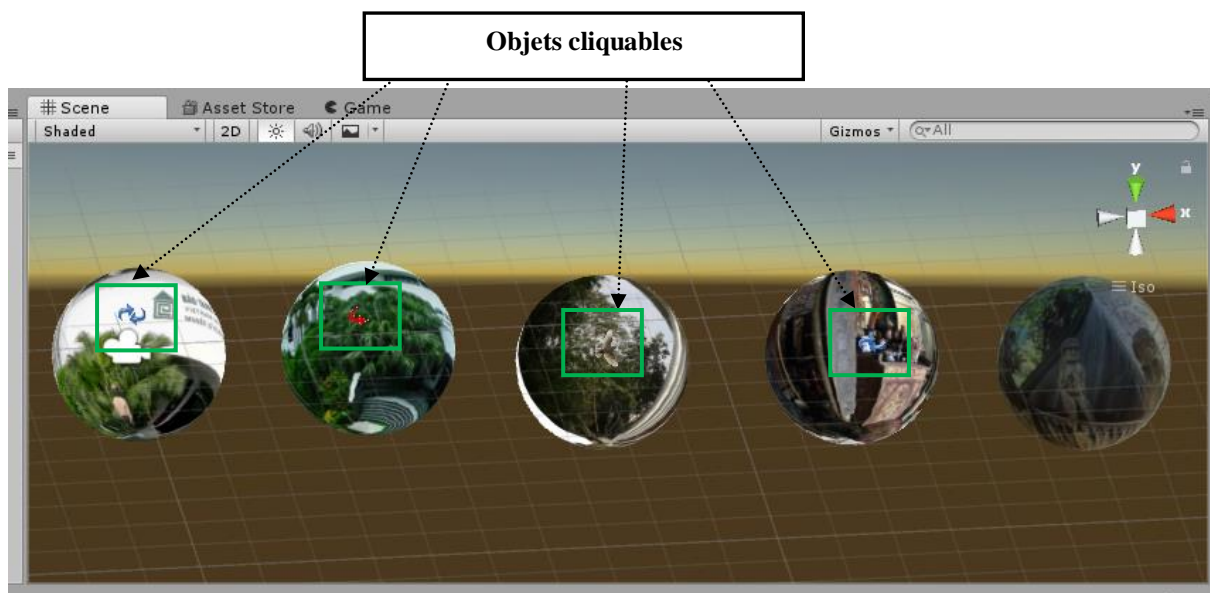


Fig11 : Illustration de la scène avec nos cinq (5) sphères

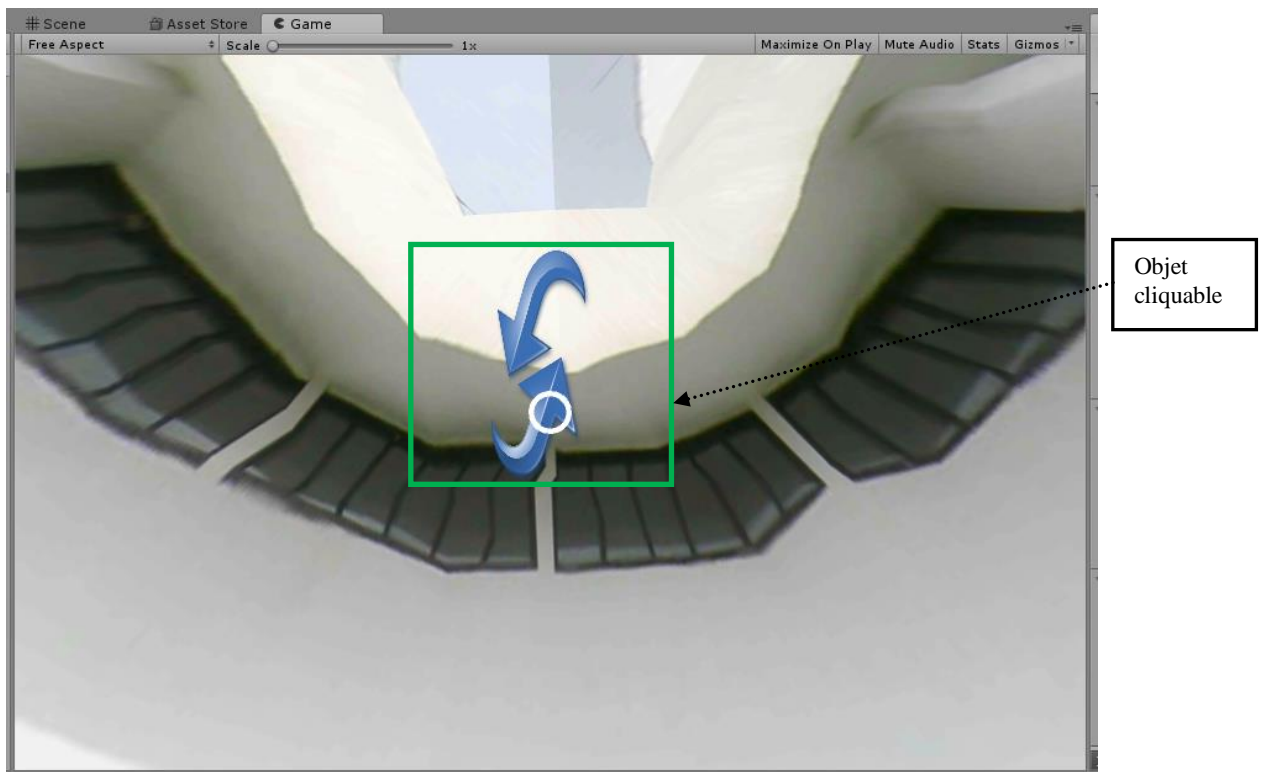


Fig12 : Illustration de la scène avec notre sphère 1 en mode « play »

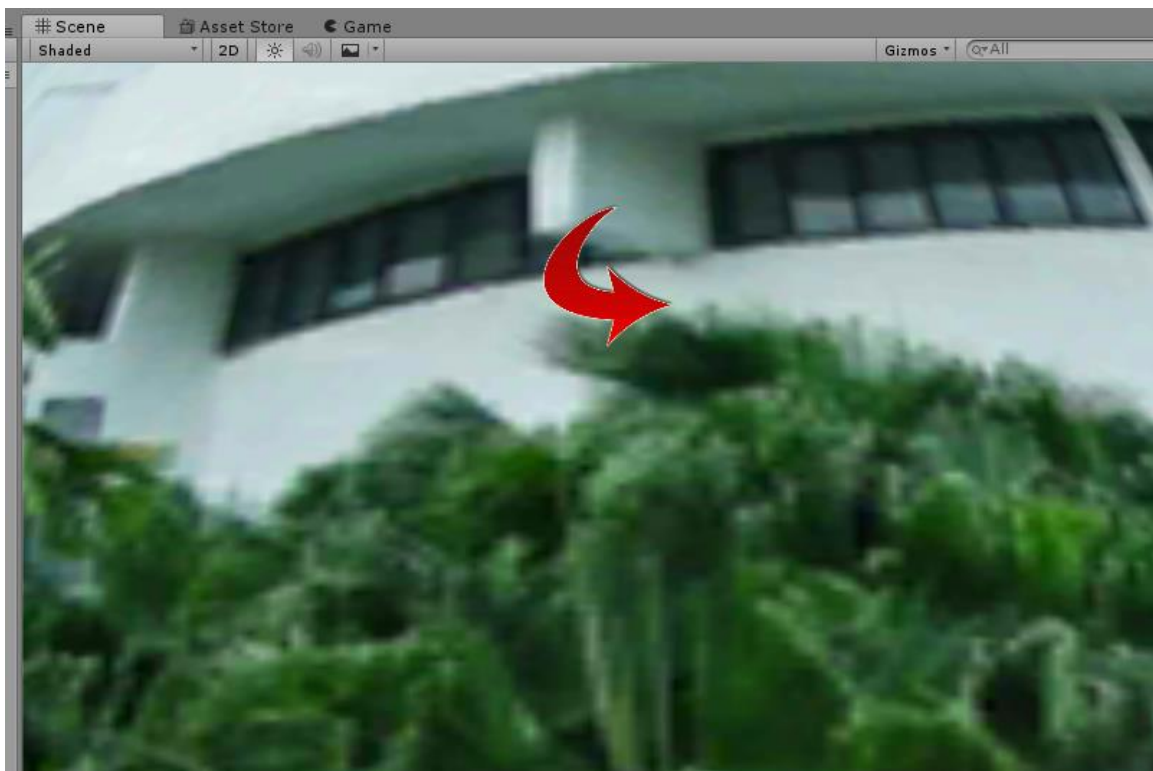


Fig13 : Illustration de la scène avec le passage de sphère 1 à la sphère 2.

Comme surplus dans notre projet, nous avons ajouté un son en arrière-plan qui sera joué tout au long de la simulation de la scène.

Pour ce faire, nous avons placé notre fichier audio dans un répertoire *son*, créé préalablement dans « assets » du projet. Ensuite, sur « Main caméra » nous avons choisi d'ajouter le composant « sound », puis ajusté les paramètres comme le montre la *figure 14*.

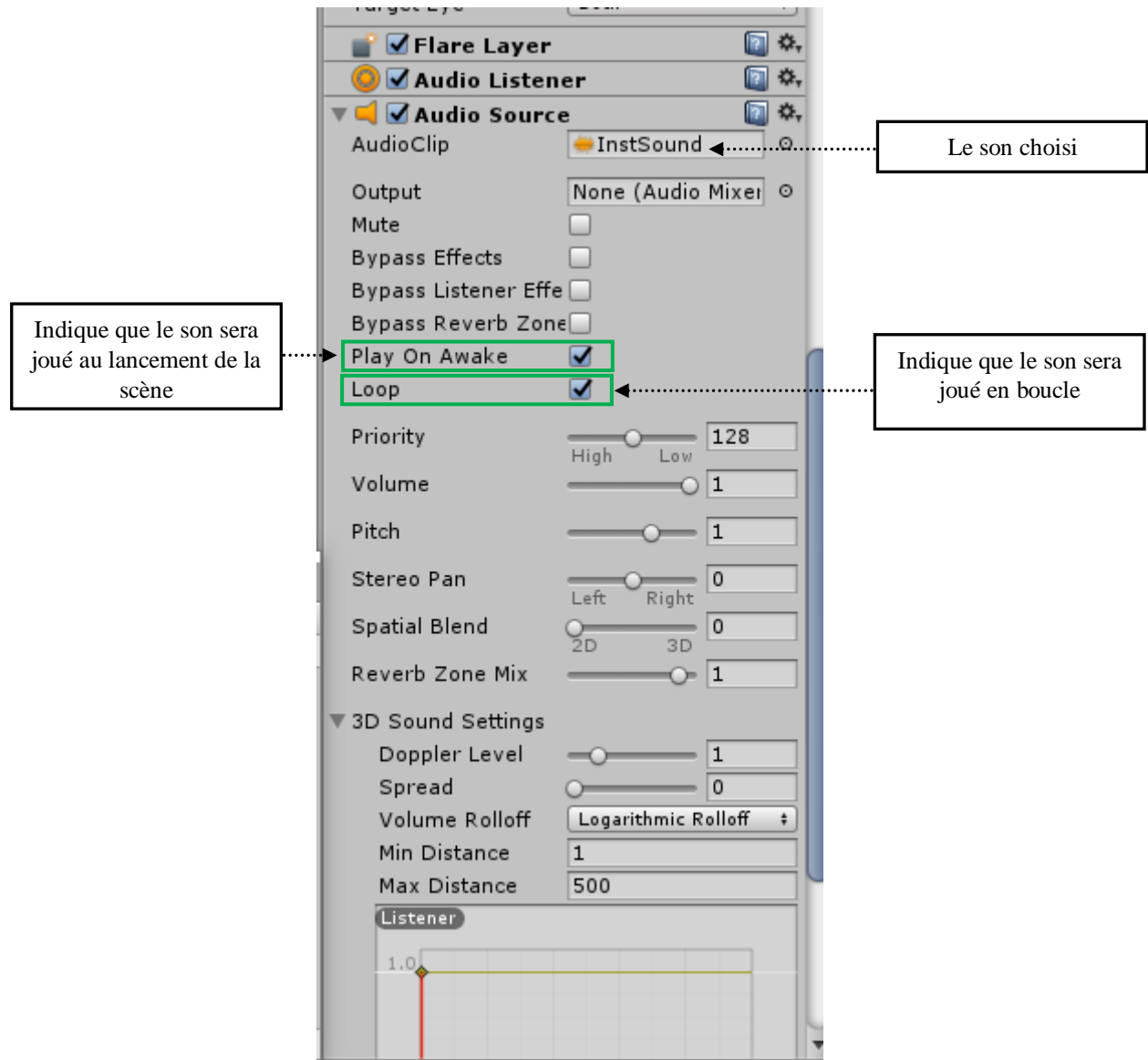


Fig14 : Illustration du paramétrage du son ajouté à la scène.

Après avoir finalisé le projet, nous avons converti en un fichier exécutable « apk » en l'installant sur un téléphone Android. Après avoir mis notre téléphone en mode développeur, activé l'option de débogage par défaut, nous télécharge et installé sur la machine, les pilotes USB correspondant au téléphone afin qu'elle puisse le détecter. Pour la conversion de notre projet en fichier exécutable « apk », nous avons connecté le téléphone avec la machine via un câble USB, puis procédé par « *File* → *Build and Run* ». Les images ci-dessous (voir *figure 15*, *figure 16* et *figure 17*) montrent, en partie, le processus de la conversion.

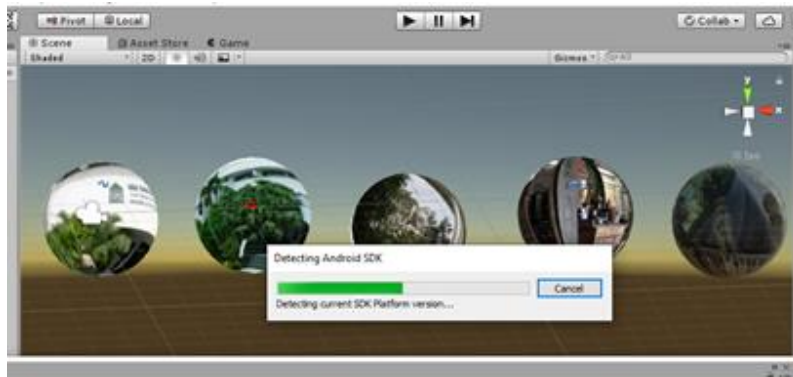


Fig15 : Illustration de la conversion du projet en fichier apk sur Android.



Fig16 : Illustration de la conversion du projet en fichier apk sur Android.

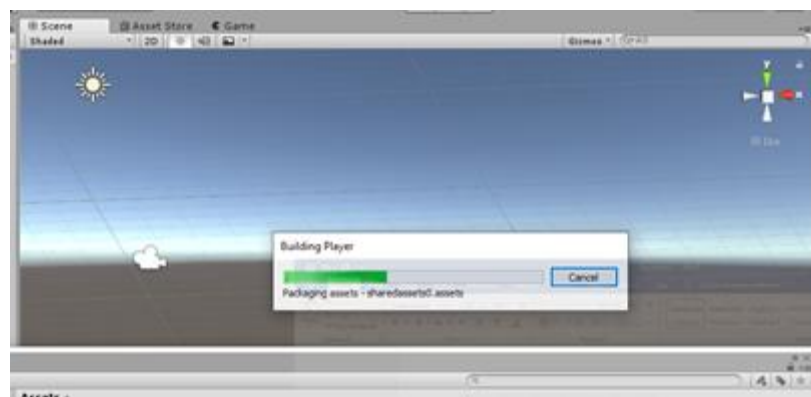


Fig17 : Illustration de la conversion du projet en fichier apk sur Android.

Enfin, l'application a été installée sur notre téléphone Android. L'image ci-dessous (voir fig 18) montre l'interface de notre téléphone avec notre application (***VIRTUALTOURVietnamMuseum***) installée.

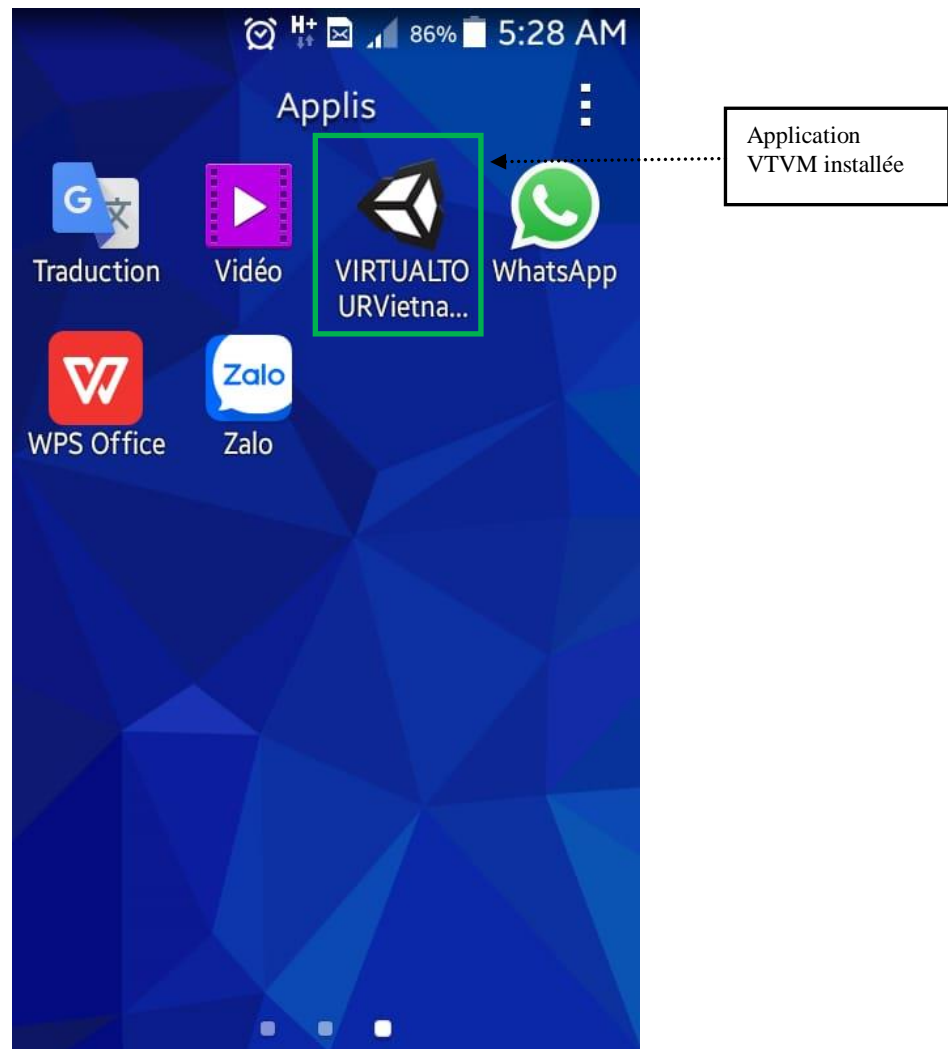


Fig18 : Interface du téléphone Android avec l'application *VIRTUALTOURVietnamMuseum* installé

Pour faire fonctionner l'application sur le téléphone, nous avons installé, via Google Play Store, un « Cardboard ». En effet, les images ci-dessous (voir figure 19, figure 20 et figure 21) montrent quelques vues du fonctionnement de notre application sur le téléphone.



Fig19 : fonctionnement de notre application sur un téléphone



Fig20 : fonctionnement de notre application sur un téléphone



Fig21 : fonctionnement de notre application sur un téléphone

V- Conclusion

Ce rapport résume l'implémentation d'une tour virtuel pour le musée d'ethnographie du Vietnam, réalisé dans le cadre du cours de Réalité Virtuelle et Augmenté au master II en informatique à l'Institut Francophone International (IFI)/Université Nationale du Vietnam, Hanoï (UNVH).

Pour la mis en place de ce projet, les différents outils : *Unity 2017.4* ; *Android studio V 3.3* ; *GVR SDK for Unity v1.190.1* ; *360 Tour Assets* et aussi quelques images du musée ethnographique du Vietnam prises sur le site <https://www.vanupied.com/hanoi/musees-hanoi/musee-d-ethnolohie-d-hanoi.html> nous ont servis de support.

Somme toute, à travers ce projet nous avons pu réaliser une tour virtuelle avec différentes scènes permettant de visiter, notamment avec une vue 3D, 5 espaces du musée ethnographique du Vietnam. Grace aux objets cliquables, on peut transiter dans les différents espaces dans notre environnement virtuel.

D'une part, nous avons notre projet qui fonctionne avec le « Cardboard » intégré dans Unity, d'autre part, nous avons converti notre application en exécutable apk et l'installé sur un téléphone Android, et cela fonctionne parfaitement après avoir installé un « cardboard » sur ce dernier.

VI- BIBLIOGRAPHIE / WEBOGRAPHIE

[1] - Google Virtual Reality : <https://developers.google.com/vr/develop/unity/download> , [consulté le 02 Février 2019]

[2]- 360 Tour Assets:
https://www.pmiwdc.org/sites/default/files/presentations/201703/PMIW_LocalCommunity_Tyson_presentation_2017-02.pdf , [consulté le 02 Février 2019]

[3]- Florent Lothon, « La méthode Agile, mieux gérer ses projets et son entreprise », <https://www.unow.fr/blog/le-coin-des-experts/methode-agile-mieux-gerer-projets-entreprise/>, [consulté le 02 Février 2019]

[4]- Cardbord Viewer, <https://vr.google.com/cardboard/get-cardboard/>, [consulté le 24 Février 2019]