



Le génie pour l'industrie

GTI780 - Sujets émergents en technologie de l'information

Partie réalité virtuelle – Intégration du casque Oculus et des
contrôles Oculus Touch pour Unity3D

Session - Été 2017

Professeur – David Labbé

Chargé de laboratoire – Philippe Charbonneau

GTI780 – Utilisation de l'oculus CV1 et Oculus Touch dans Unity3D

Utilitaires nécessaires pour l'utilisation d'oculus HMD et Touch

- Oculus Utilities for Unity : [Oculus utilities](#)
- Oculus platform for Unity : [Oculus platform SDK](#)
- Oculus Avatar : [Oculus avatar SDK](#)

Importation de l'utilitaires Oculus dans Unity

- 1- Décompresser l'archive (OculusUtilities) précédemment téléchargée
- 2- Dans un projet Unity, aller dans le menu Asset-> Import Package -> Custom Package...
- 3- Sélectionner le fichier OculusUtilities.unitypackage
- 4- Dans la fenêtre d'importation, cliquer sur all et sur import
- 5- Dans Unity, aller dans le dossier OVR -> Prefabs et sélectionner le prefab OVRCameraRig
- 6- Incorporer l'objet dans votre scène
- 7- Ouvrir les options du Player unity (Project -> Player)
- 8- Cocher Virtual Reality supported
- 9- Vous pouvez maintenant cliquer sur Play et vérifier que le HMD fonctionne bien
- 10- Créé maintenant deux objets primitifs de votre choix et nommer ces objets mainGauche et mainDroite
- 11- Placer les objets afin qu'ils aient comme parents LeftHandAnchor et RightHandAnchor qui eux sont situées dans la hiérarchie de l'objet OVRCameraRig
- 12- Cliquer sur Play, les Oculus Touch devraient maintenant faire tourner les deux objets que vous avez créé.
- 13- Créer un nouveau script et associé le à l'un des deux objets précédemment créés dans la scène. Vous allez créer un script C# servant à changer les couleurs de l'objet en fonction des boutons appuyé sur l'Oculus Touch. Insérer le code suivant dans votre script

```
// Le code suivant sert à intercepter un évènement press des contrôles Touch
// en appuyant sur le bouton HandTrigger du contrôleur Touch
if (OVRInput.Get(OVRInput.Button.PrimaryHandTrigger))
    this.gameObject.GetComponent<Renderer>().material.color = Color.red;
else
    this.gameObject.GetComponent<Renderer>().material.color = Color.gray;
```

- 14- Vous pouvez tester votre code en appuyant sur Play

- 15- Remplacer le code précédent par le code suivant

```
// Le code suivant sert à intercepter un ButtonDown event sur l'indexTrigger de
// l'Oculus Touch et change la couleur de l'objet
if (OVRInput.GetDown(OVRInput.Button.PrimaryIndexTrigger))
{
    Color actualColor = this.gameObject.GetComponent<Renderer>().material.color;
    Color newColor = (actualColor == Color.gray ? Color.red : Color.grey);
    this.gameObject.GetComponent<Renderer>().material.color = newColor;
}
```

16- Vous pouvez tester votre code en appuyant sur Play

Importation de Oculus Platform dans Unity

- 1- Décompresser l'archive (OculusPlatform) précédemment téléchargée
- 2- Dans un projet Unity, aller dans le menu Asset-> Import Package -> Custom Package...
- 3- Sélectionner le fichier OculusPlatform.unitypackage
- 4- Dans la fenêtre d'importation, cliquer sur all et sur import

Importation de Oculus Avatar dans unity

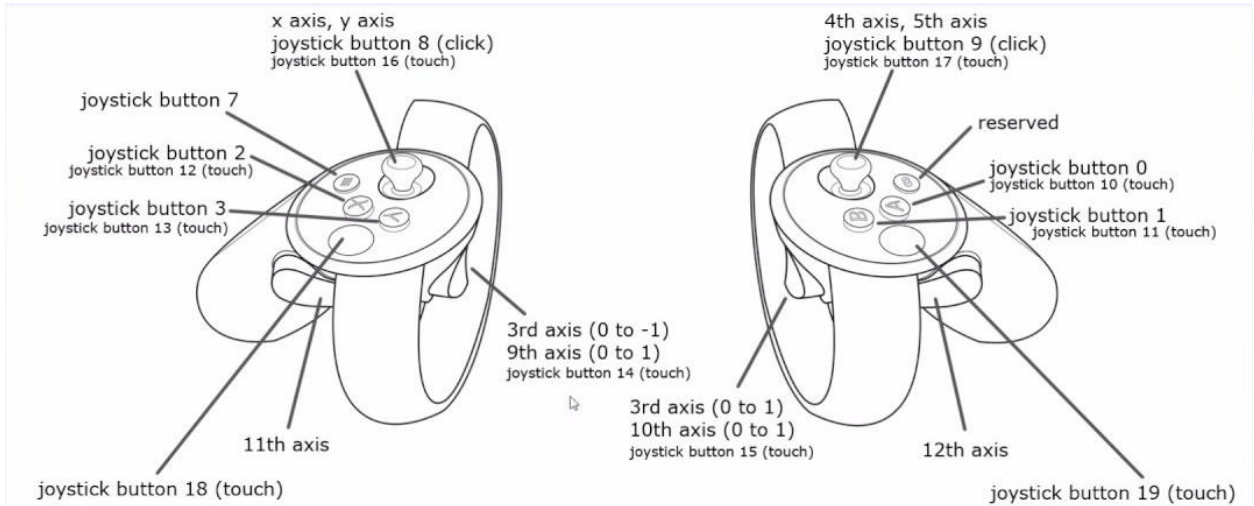
- 1- Décompresser l'archive (OVRAvatarSDK) précédemment téléchargée
- 2- Dans un projet Unity, aller dans le menu Asset-> Import Package -> Custom Package...
- 3- Sélectionner le fichier OvrAvatar.unityPackage
- 4- Dans la fenêtre d'importation, cliquer sur all et sur import
- 5- Aller dans le dossier OvrAvatar -> Samples -> Controllers
- 6- Ouvrir la scène Controllers
- 7- Dans la scène, vous avez l'objet OVRCameraRig ainsi que l'objet LocalAvatar
- 8- L'objet localAvatar comporte plusieurs options, pour une utilisation standard, assurer d'avoir décocher la case « Start With Control » et que la case « Show First Person » est coché.
- 9- Vous pouvez tester votre projet en appuyant sur Play

Utilisation des Oculus Touch pour des évènements

Suite à la réalisation de la section suivante, un projet servant à attraper des objets, les relâcher et les lancer sera exécutable.

- 1- Ajout de contrôle dans Unity. Aller dans le menu Edit -> Project Settings -> Input
- 2- Ajouter deux entrées dans la liste. Nommer les entrées LHandTrigger et RHandTrigger
- 3- Pour le LHandTrigger assurez-vous que la gravité est de 0 la densité est de 1.5 et que l'axe relié au control est la 11^{ième}

- 4- Pour le RHandTrigger, la configuration reste la même que pour le LHandTrigger, mais l'axis est la 12^{ème}.



¹Figure 1 - Oculus Touch Controllers windows mapping

- 5- Créer ensuite 1 objet (Cube, sphère, etc.) avec un rigidBody et cocher Is Kinematic
- 6- Dans l'onglet layer de l'inspecteur, ajouter un nouveau layer nommé « grabbable »
- 7- Créer un nouveau script et nommer le « OculusTouchInteractions »
- 8- Copier ensuite le code suivant dans le script

¹ Figure 1 source : <https://i.ytimg.com/vi/mFFta9OszzA/maxresdefault.jpg>

```

private GameObject grabbedObject;
private bool grabbing;

public OVRInput.Controller controller;
public string buttonName;
public float grabRadius;
public LayerMask grabMask;

void GrabObject() {
    grabbing = true;

    RaycastHit[] hits;
    hits = Physics.SphereCastAll(transform.position, grabRadius,
    transform.forward, 0f, grabMask);

    if(hits.Length > 0)
    {
        int closestHit = 0;
        for(int i = 0; i < hits.Length; ++i)
        {
            if(hits[i].distance < hits[closestHit].distance) closestHit = i;
        }

        grabbedObject = hits[closestHit].transform.gameObject;
        grabbedObject.GetComponent<Rigidbody>().isKinematic = true;
        grabbedObject.transform.position = transform.position;
        grabbedObject.transform.parent = transform;
    }
}

void DropObject() {
    grabbing = false;
    if(grabbedObject != null)
    {
        grabbedObject.transform.parent = null;
        grabbedObject.GetComponent<Rigidbody>().isKinematic = false;

        grabbedObject.GetComponent<Rigidbody>().velocity =
        OVRInput.GetLocalControllerVelocity(controller);
        grabbedObject.GetComponent<Rigidbody>().angularVelocity =
        OVRInput.GetLocalControllerAngularVelocity(controller);

        grabbedObject = null;
    }
}

// Update is called once per frame
void Update () {
    if(!grabbing && Input.GetAxis(buttonName) == 1) GrabObject();
    if (grabbing && Input.GetAxis(buttonName) < 1) DropObject();
}

```

- 9- Ajouter le script à l'objet hand_left et hand_right
- 10- Ajuster les paramètres du script pour que la variable controller soit liée au bon Touch (R touch ou L Touch), pour la variable Button Name mettre « RHandTrigger » ou « LHandTrigger » et mettre le grab mask à « grabbable »
- 11- Appuyer sur Play, vous devriez être en mesure de prendre l'objet que vous avez créé.