



TP2 Réalité Augmentée

RA Interactive utilisant
le Plugin Vuforia

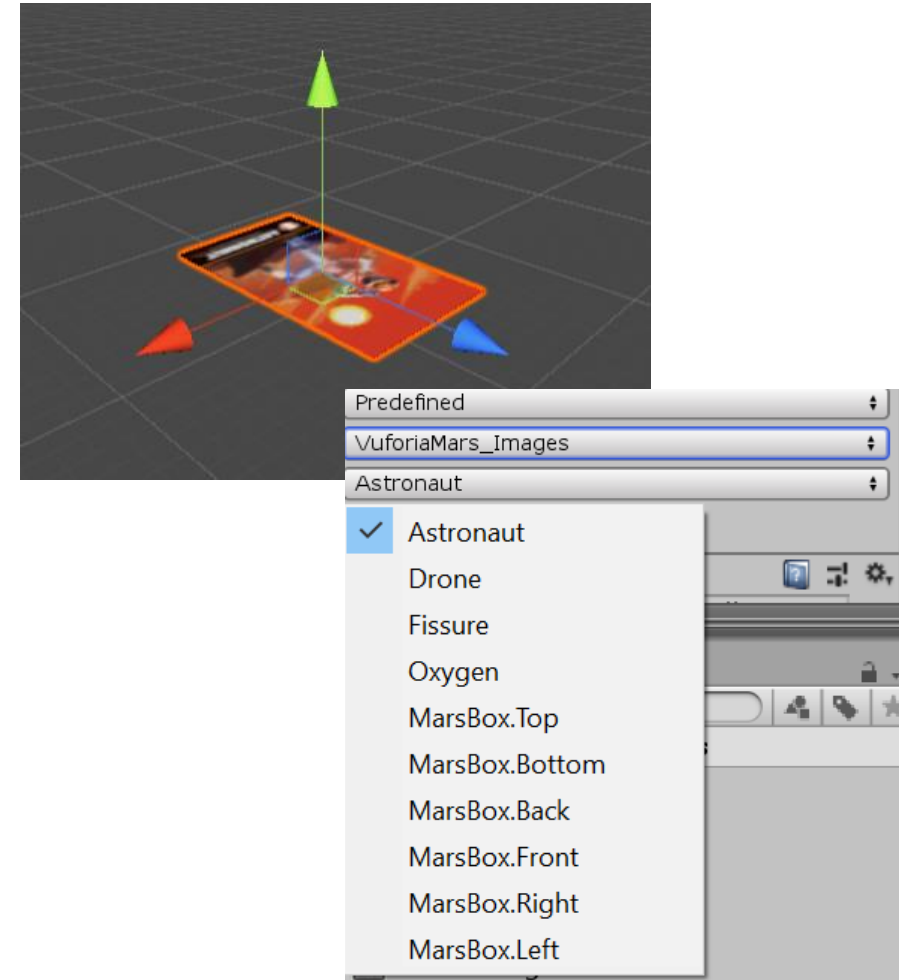
**Prof. Fakhreddine Ababsa,
Arts et Métiers ParisTech**

Prise en main du plugin Vuforia sous Unity 3D

Un guide d'utilisation du SDK Vuforia est détaillé sur la page web suivante :

<https://library.vuforia.com/articles/Training/getting-started-with-vuforia-in-unity.html>

1. Créer un nouveau projet Unity 3D
2. Importer le package Vuforia à partir de Vuforia Portal
3. Ajouter une AR Camera : GameObject → Vuforia → AR Camera
4. Supprimer Main Camera
5. Créer une ImageTarget : GameObject → Vuforia → Image.
6. Type : from database



Prise en main du plugin Vuforia sous Unity 3D

6. Ajouter une Licence Vuforia : AR Camera → Open Vuforia Configuration → Add Licence



License Manager

Create a license key for your application.

7. Créer un compte sur « Vuforia Developer Portal », puis se connecter

8. Develop → Licence Manager → Get Development Key : saisir un nom pour la Licence, puis confirmer

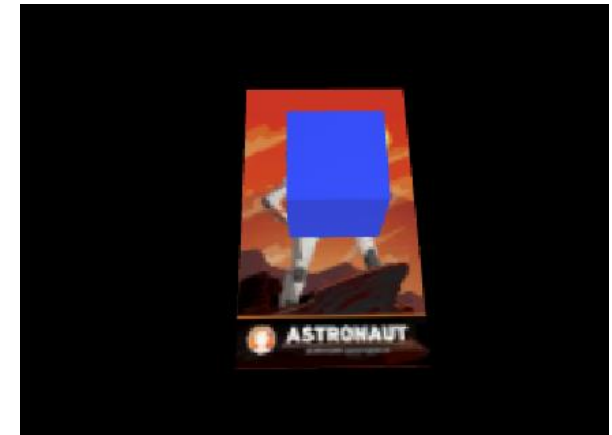
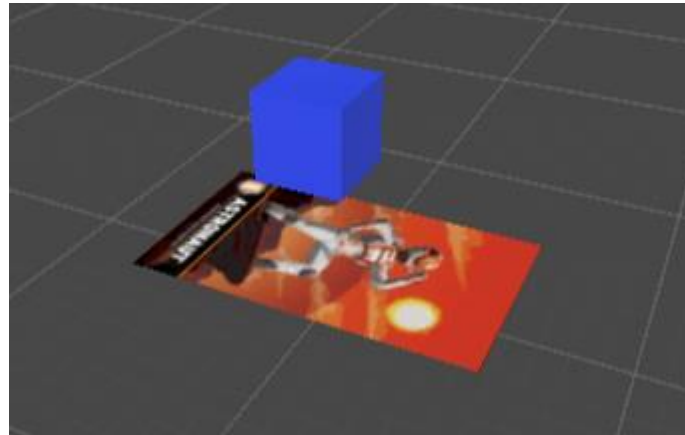
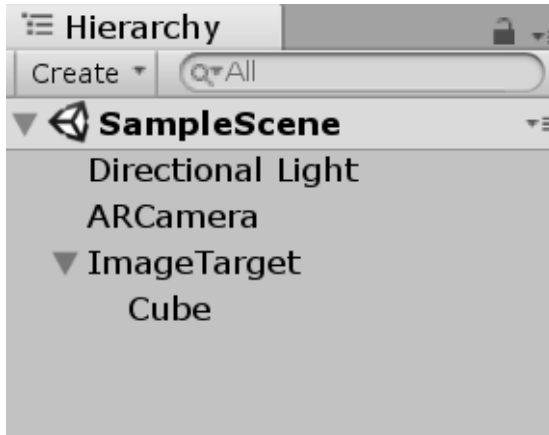
9. Cliquer dans la liste, sur la licence que vous venez de créer

10. Copier le code généré et coller dans votre application unity
→ App Licence Key

```
ARJ69PD/////AAABmWqnsP/x1ke3h+KhQcaR011iygDmATWcBRcD  
Z/cFcZNZNQNYs+J966PiDDJ62zPt6m6Y+rZD111SFbHfoacwRstn  
+CDRxWCwPFwBSldmxkbgYg38UCA4uI+kvnptHVuodWyYTDcPnwY  
GPWvWU3BrQQkOv7Q7UhgWVO4Yn2ZdZEu48VME9EYD49cAtmRu7my  
B6VL9LyEgElfpku9bf0OD2B3VyNGB4jNh8s4+c9uoIaKQxYB3mBw  
z43z111JDEERDjtbmrcuGrIxxd5VqvC9QGchbp77qfoGnudzm/tY  
u/8D99P08pFQJANn9mEU4g98+8X09s9PJXZS9mV6sP4XI90sepL  
j2rYrPWpEMd8jKEe
```

Prise en main du plugin Vuforia sous Unity 3D

11. Ajouter un GameObject de type Cube (bleu) et le mettre comme enfant de « Image Target » : réduire le scale et placer audessus de la cible.




12. Imprimer l'image de l'astronaute, à partir du site de Vuforia : Samples → printable target samples → Unity App → Image Targets → Download
13. Jouer la scène

Prise en main du plugin Vuforia sous Unity 3D

Personnalisation d'une Image Target

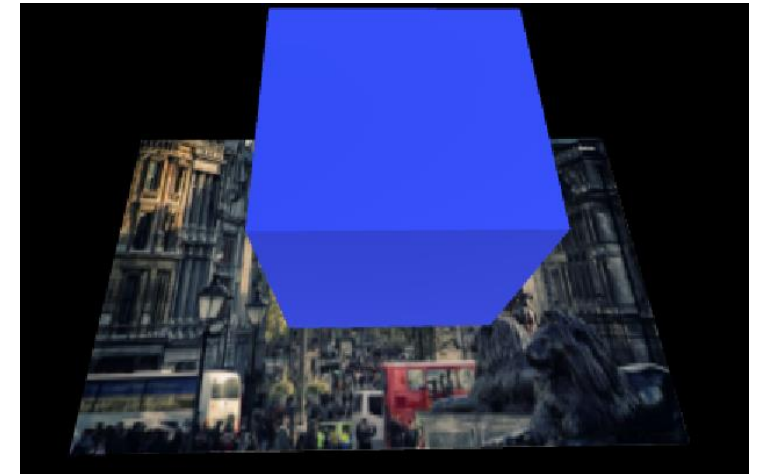
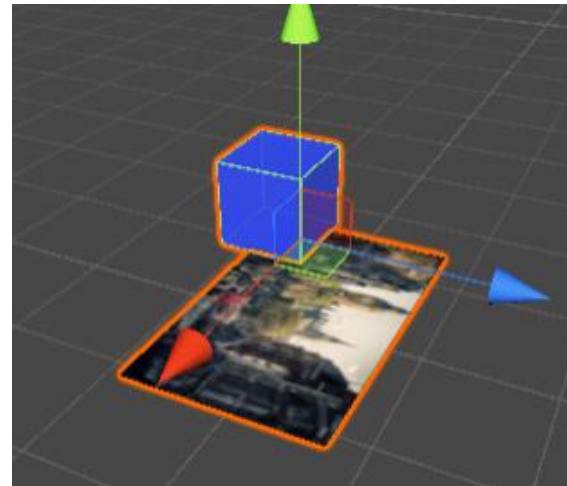
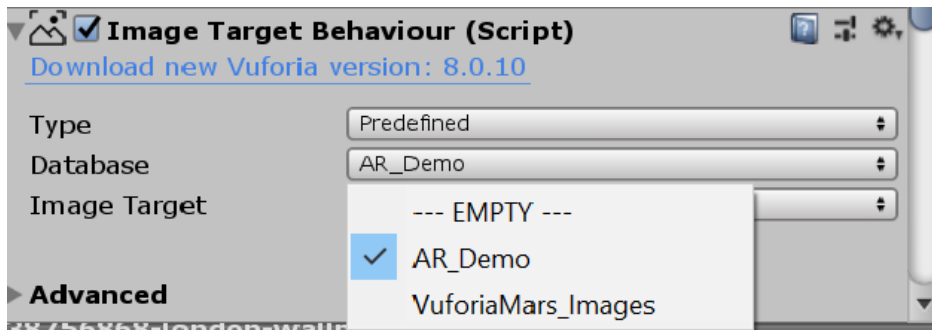
1. Aller sur votre compte Vuforia : Develop → Target Manager
2. Cliquer sur *Add Database* → Saisir un nom : ceci crée une base de données des images qui, à ce stade, est vide.
3. Cliquer sur *Add Target* : quatre types de cibles sont proposées → choisir *single image*
4. Cliquer sur *Browse* et choisir l'image Target que vous aurez préalablement téléchargée ou choisie.
5. Entrer dans le champs With, la largeur exacte de votre image (en mètre)

<input type="checkbox"/>	Target Name	Type	Rating	Status ▾
<input type="checkbox"/>	 38756868-london-wallpapers	Single Image	★★★★★	Active

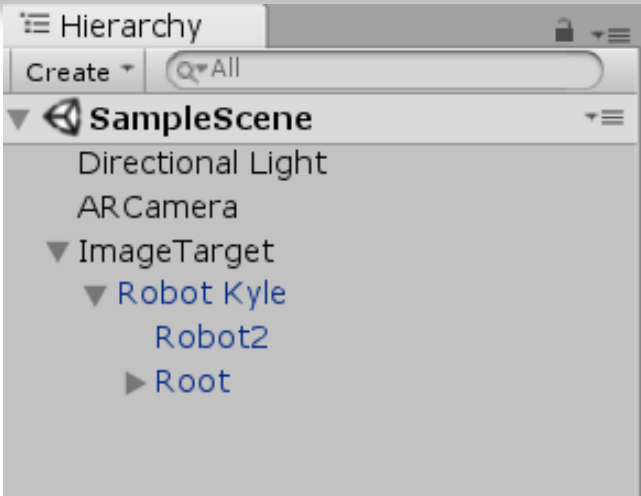


Prise en main du plugin Vuforia sous Unity 3D

6. Cliquer sur Download Database → Unity Editor
7. Ouvrir avec Unity → Import
8. Dans Hierarchy, cliquer sur Image Target → Data base → Choisir AR Demo → sélectionner votre image
9. Jouer la scène

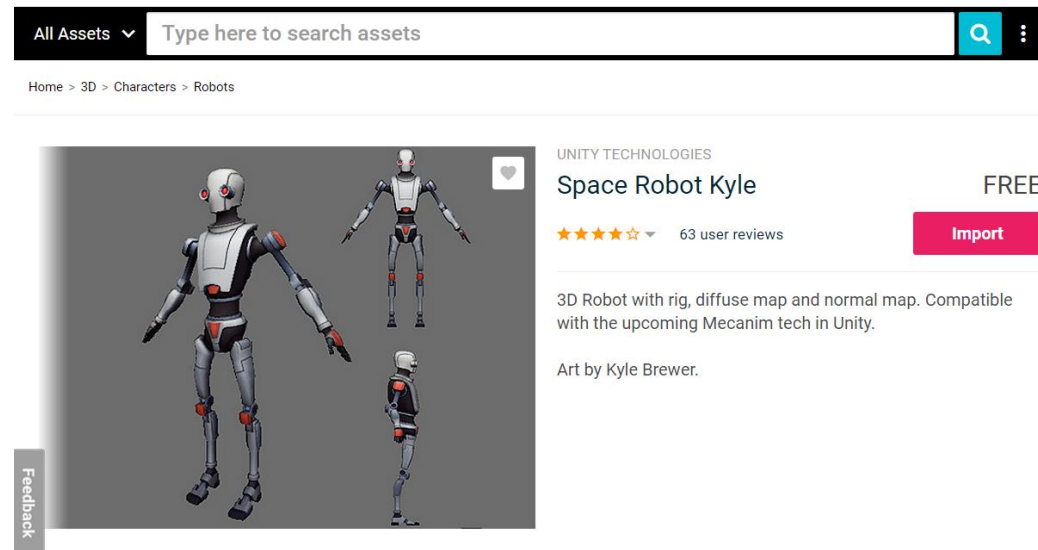
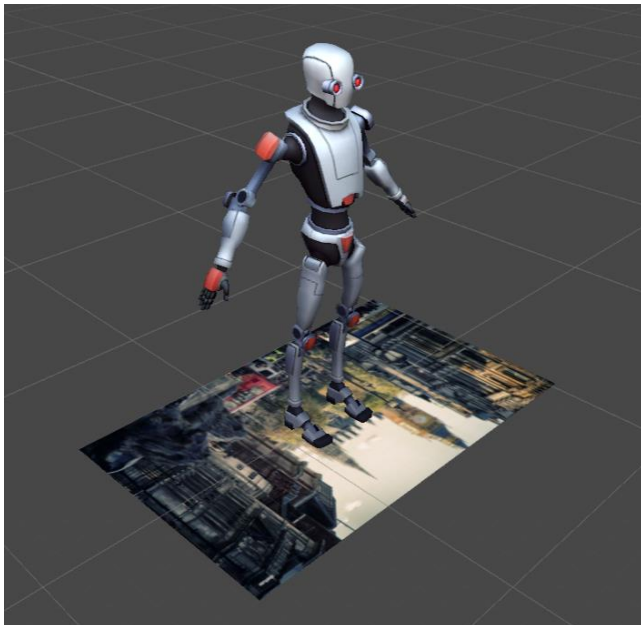


Exercice 1 (à valider)

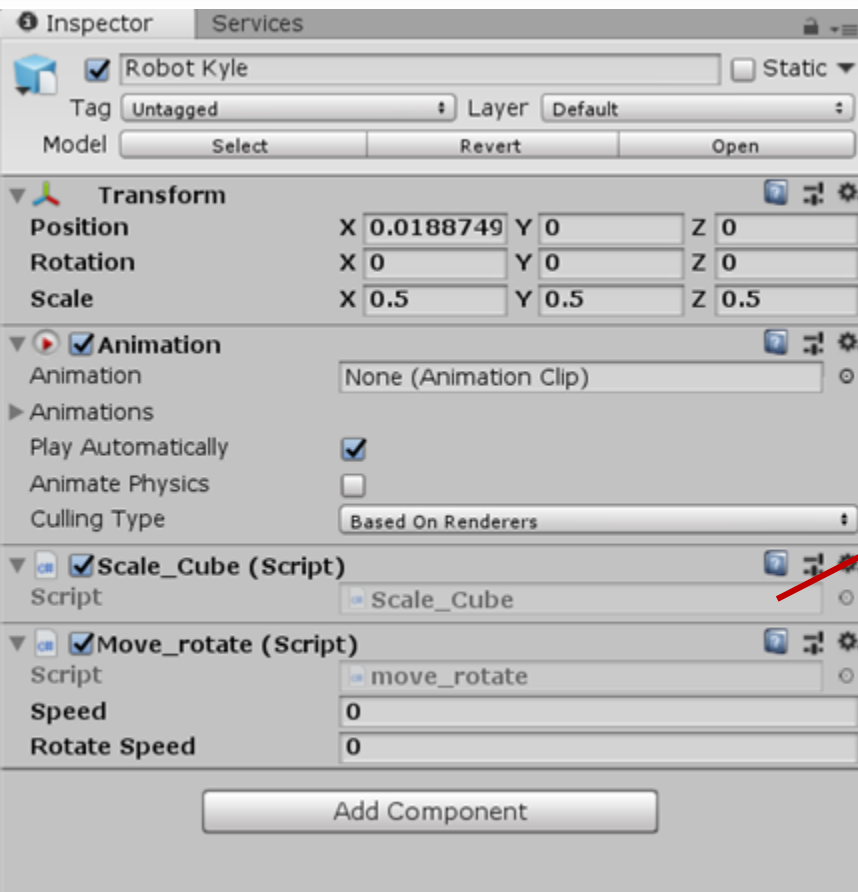


Suivi mono-cible + Interaction en RA

- Réaliser une application qui augmente une cible avec un modèle 3D (ex : Robot Kyle)
- En utilisant les touches du clavier (flèches, pavé numérique, etc.), proposer des interactions avec le modèle 3D, de type : translation, rotation, changement d'échelle, etc.



Exercice 1



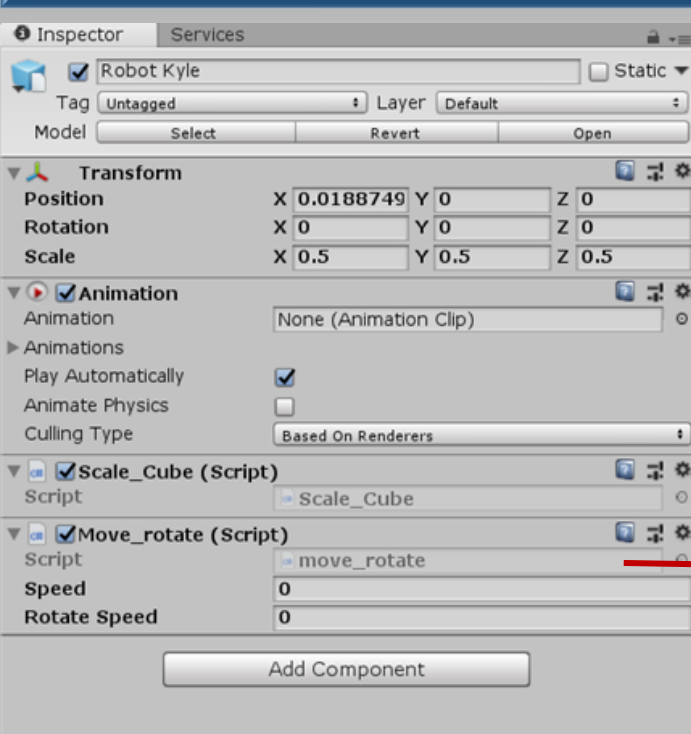
```
public class Scale_Cube : MonoBehaviour {

    // Use this for initialization
    void Start () {
    }

    // Update is called once per frame
    void Update () {
        // si la touche W est appuyee
        if (Input.GetKeyDown(KeyCode.W))
        {
            // si le cube a un scale superieur a 1 (ne pas effectuer un scale au dessus de 5)
            if (transform.localScale.x < 1 && transform.localScale.y < 1 && transform.localScale.z < 1)
            {
                // alors faire une transformation scale incrementee sur le gameobject
                transform.localScale += new Vector3(0.1f, 0.1f, 0.1f);
            }
        }

        // si la touche W est appuyee
        if (Input.GetKeyDown(KeyCode.X))
        {
            // si le cube a un scale superieur a 0 (ne pas effectuer un scale negatif)
            if (transform.localScale.x > 0 && transform.localScale.y > 0 && transform.localScale.z > 0)
            {
                // alors faire une transformation scale decrementee sur le gameobject
                transform.localScale -= new Vector3(0.1f, 0.1f, 0.1f);
            }
        }
    }
}
```


Exercice 1



```
public class move_rotate : MonoBehaviour
{
    public float speed; //vitesse a definir dans l'interface unity
    public float rotateSpeed; //vitesse de rotation a definir dans l'interface unity

    // Use this for initialization
    void Start()
    {
    }

    // Update is called once per frame
    void Update()
    {
        //si la touche fleche du haut appuyee, alors faire une transformation de type translation selon un vecteur "en avant"
        //avec la vitesse speed defini precedement
        if (Input.GetKey(KeyCode.UpArrow))
        {
            transform.Translate(Vector3.forward * Time.deltaTime * speed);
        }

        //si la touche fleche du bas appuyee, alors faire une transformation de type translation selon un vecteur "en arriere"
        //avec la vitesse defini precedement
        if (Input.GetKey(KeyCode.DownArrow))
        {
            transform.Translate(Vector3.back * Time.deltaTime * speed);
        }

        //si la touche fleche du bas appuyee, alors faire une transformation de type rotation sur x
        //avec la vitesse rotationSpeed defini precedement
        if (Input.GetKey(KeyCode.LeftArrow))
        {
            transform.Rotate(Vector3.down, Time.deltaTime * rotateSpeed);
        }

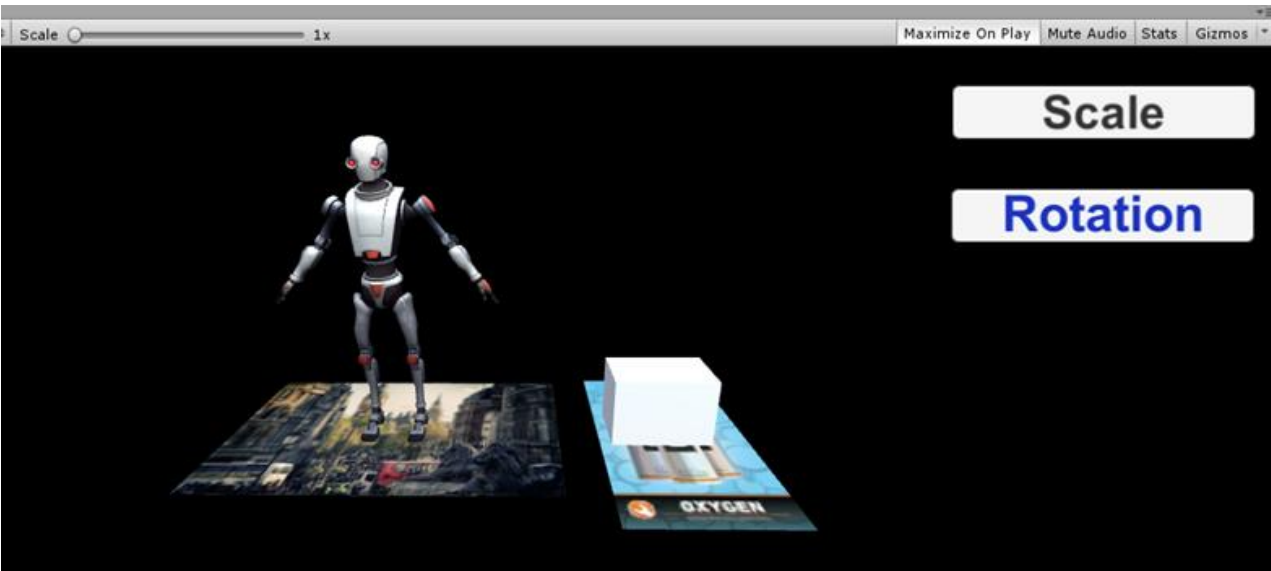
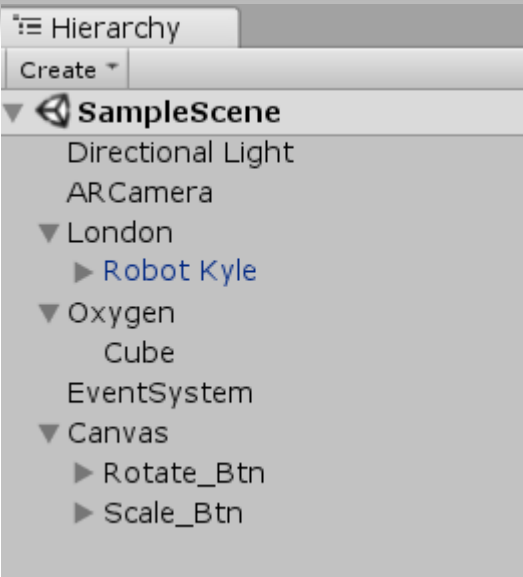
        //si la touche fleche du bas appuyee, alors faire une transformation de type rotation sur x
        //avec la vitesse rotationSpeed defini precedement
        if (Input.GetKey(KeyCode.RightArrow))
        {
            transform.Rotate(Vector3.up, Time.deltaTime * rotateSpeed);
        }
    }
}
```

Exercice 2 (à valider)

UI en RA - Vuforia sous Unity 3D

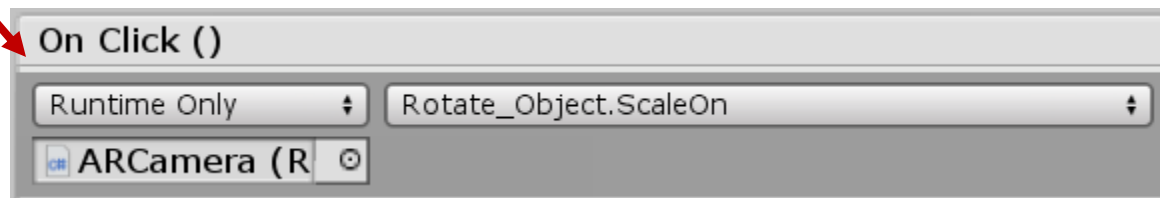
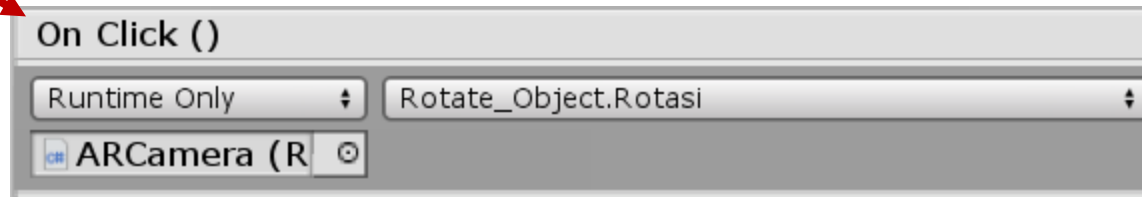
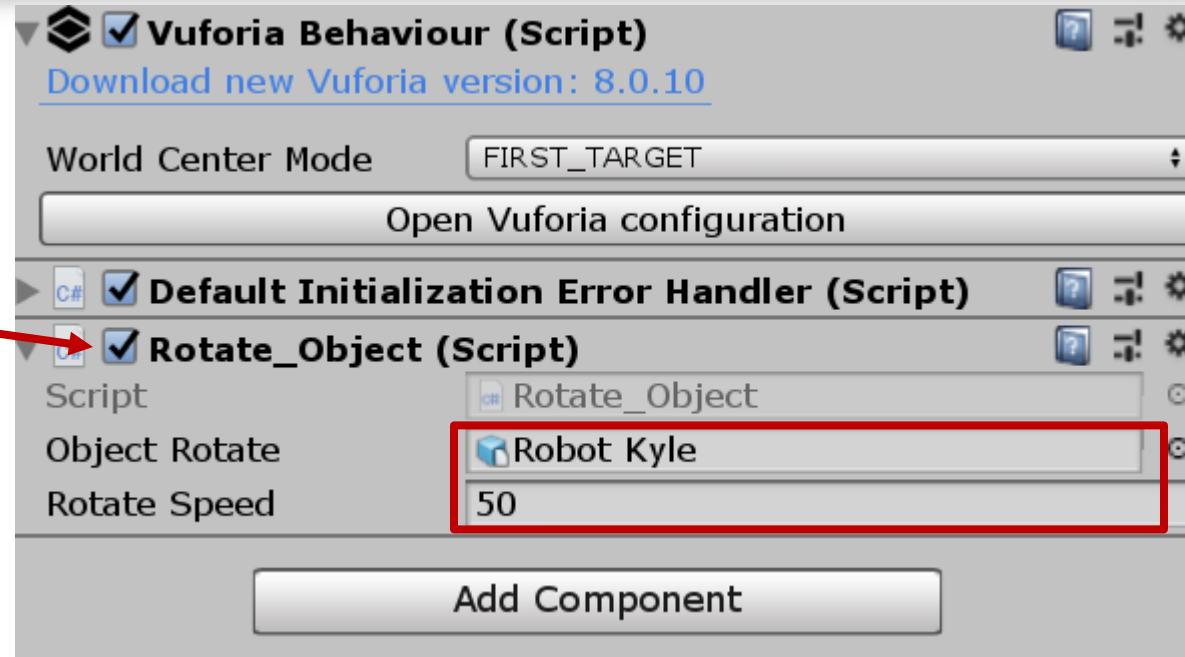
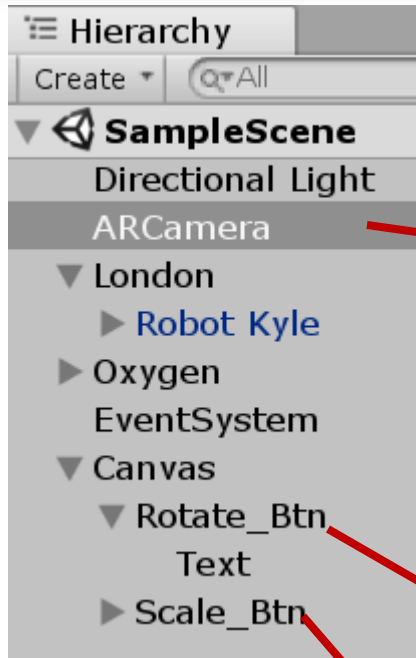
Créer une application de RA permettant de :

- Augmenter deux cibles avec deux modèles 3D: Robot Kyle + un cube 3D
- Changer l'échelle du robot Kyle grâce au bouton Scale
- Appliquer une rotation continue au robot Kyle grace au bouton Rotation



Exécuter l'application sur PC puis sur téléphone

Exercise 2



Exercise 2

Rotate_Object.cs

```
public class Rotate_Object : MonoBehaviour
{
    public GameObject objectRotate;
    public float rotateSpeed = 50f;
    bool rotateStatus = false;

    public void ScaleOn(){
        objectRotate.transform.localScale += new Vector3(0.1f, 0.1f, 0.1f);
    }

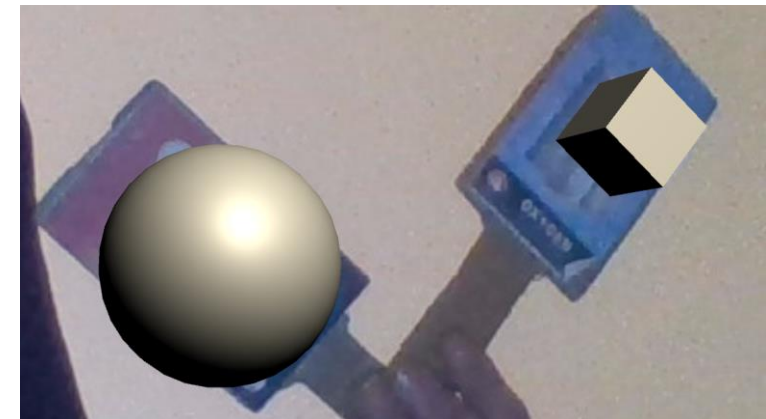
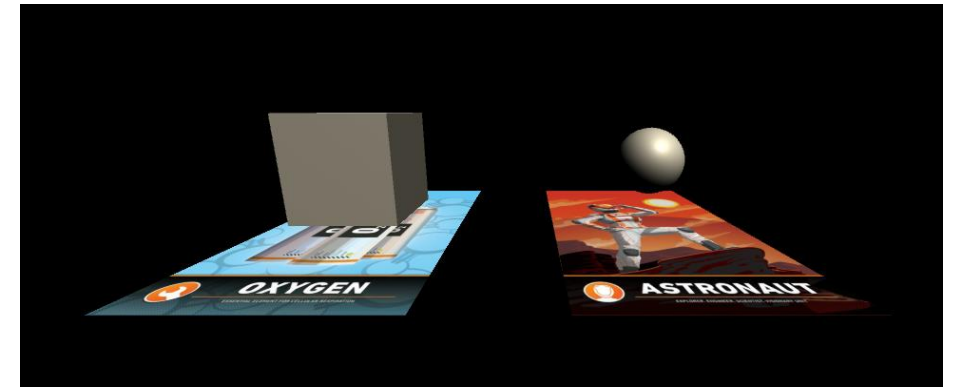
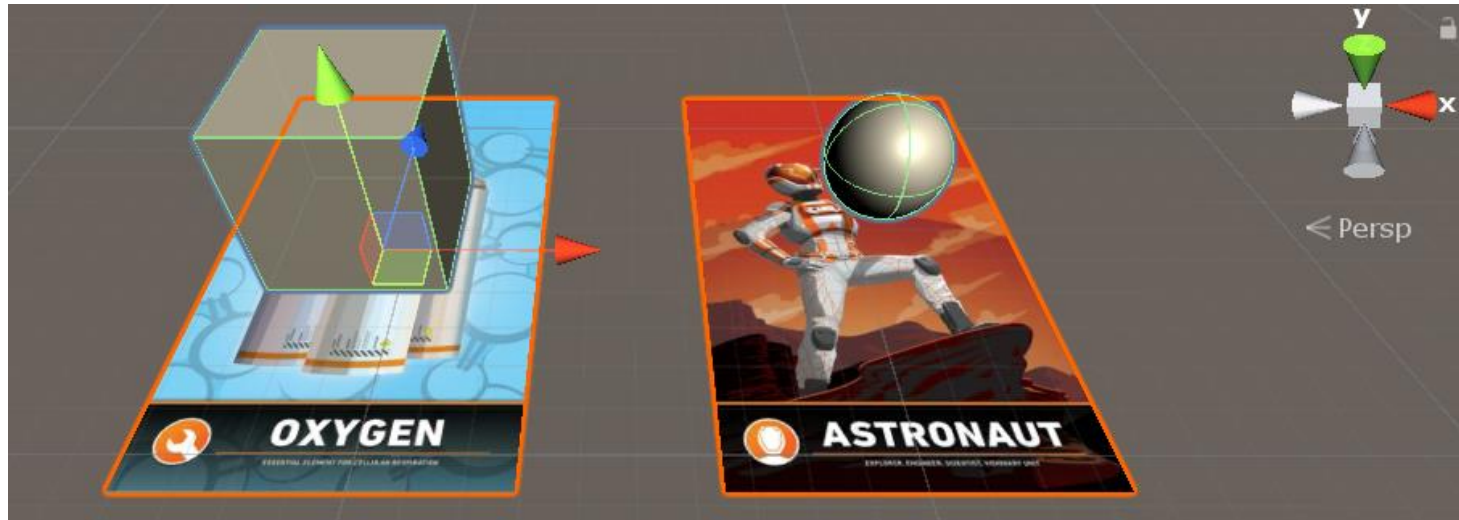
    public void Rotasi(){
        if (rotateStatus == false){
            rotateStatus = true;
        }
        else{
            rotateStatus = false;
        }
    }

    void Update()
    {
        if (rotateStatus == true){
            objectRotate.transform.Rotate(Vector3.up, Time.deltaTime * rotateSpeed);
        }
    }
}
```

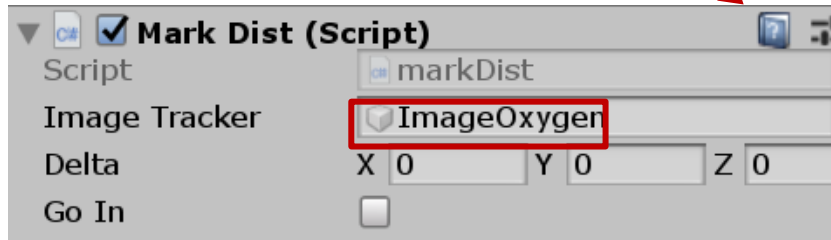
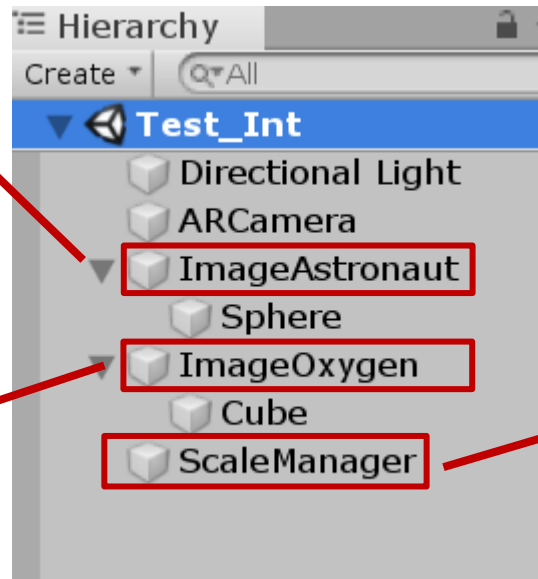
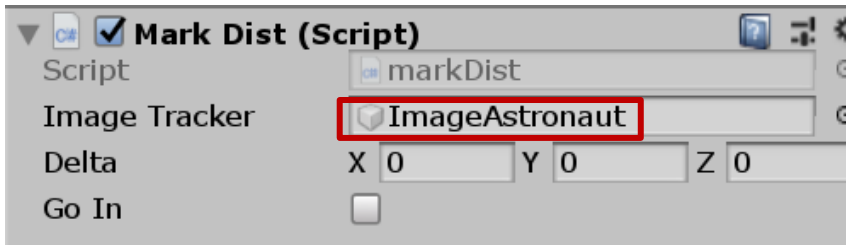
Exercice 3 (à valider)

Interaction naturelle en RA - Vuforia sous Unity 3D

Créer une application de RA permettant de changer, en temps réel, l'échelle de la Capsule en fonction de la distance qui sépare les deux cibles.



Exercise 3



Exercise 3

markDist.cs

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using Vuforia;
5
6  public class markDist : MonoBehaviour
7  {
8      public GameObject imageTracker;
9      public Vector3 delta;
10     public bool goIn = false;
11     void Update()
12     {
13         //var trackableImage = imageTracker.GetComponent<TrackableBehaviour>();
14         var trackableImage = imageTracker.GetComponent<ObserverBehaviour>();
15         //var statusImage = trackableImage.CurrentStatus;
16         var statusImage = trackableImage.TargetStatus.Status;
17         //if (statusImage == TrackableBehaviour.Status.TRACKED)
18         if (statusImage == Status.TRACKED)
19         {
20             goIn = true;
21             delta = Camera.main.transform.position - transform.position;
22         }
23         else
24         {
25             goIn = false;
26         }
27     }
28 }
```


Exercise 3

ScaleUpdate.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ScaleUpdate : MonoBehaviour
{
    public GameObject Image1 = null;
    public GameObject Image2 = null;
    Vector3 dist;
    float mod = 0;

    // Update is called once per frame
    void Update()
    {
        markDist SpherePosition = Image1.GetComponent<markDist>();
        markDist CubePosition = Image2.GetComponent<markDist>();
        dist = SpherePosition.delta - CubePosition.delta;
        if (SpherePosition.goIn == true && CubePosition.goIn == true)
        {
            mod = dist.magnitude;
            Image1.transform.GetChild(0).gameObject.transform.localScale = new Vector3(0.15f / mod, 0.15f / mod, 0.15f / mod);
            Debug.Log("module " + mod);
        }
    }
}
```


Exercice Autonomie (à valider)

Interfaces Tangibles en Réalité Augmentée avec Vuforia

Objectif :

Ce TP vise à introduire les interfaces tangibles en réalité augmentée en utilisant Vuforia et des marqueurs pour manipuler un objet 3D. Vous allez concevoir une application Unity où différents marqueurs auront des interactions spécifiques avec un cube virtuel. Le but est de permettre une interaction intuitive et immersive avec l'objet en réalité augmentée.

1. Configuration des Marqueurs Vuforia :

Créer les marqueurs nécessaires (six marqueurs) en les associant à leurs fonctions respectives.

2. Fonctions des Marqueurs :

- **Marqueur principal :**

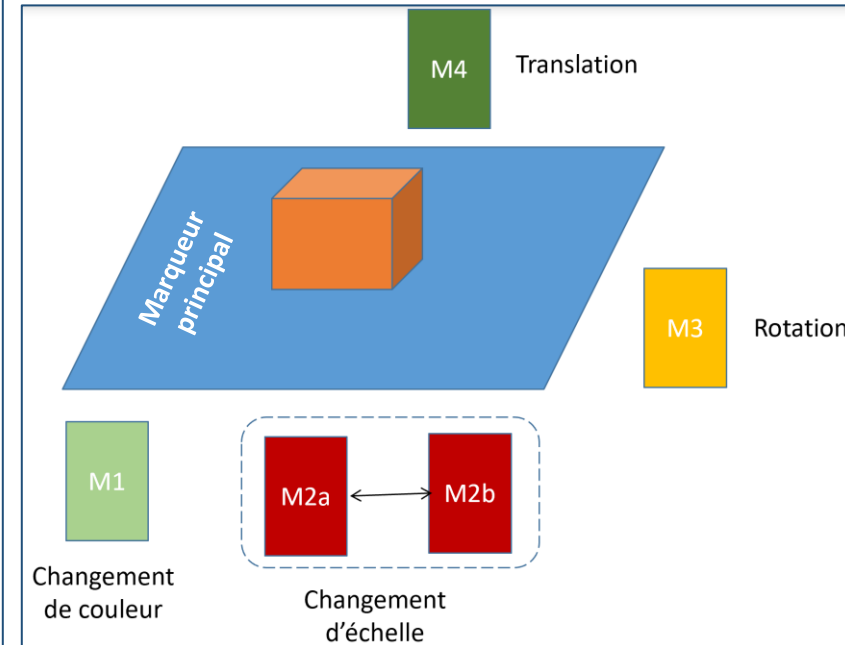
Ce marqueur affiche un cube 3D fixe lorsqu'il est détecté dans la scène. Le cube sera votre objet de base pour les manipulations.

- **Deuxième marqueur (changement de couleur) :**

Configurer ce marqueur pour modifier dynamiquement la couleur du cube lorsque le marqueur est détecté. Par exemple, alterne entre plusieurs couleurs prédéfinies à chaque détection.

- **Troisième marqueur (rotation du cube) :**

Programmer ce marqueur pour que le cube adopte la même orientation que celle du marqueur. Le cube devra tourner en temps réel en suivant les rotations du marqueur.



Exercice Autonomie (à valider)

- ***Quatrième marqueur (translation du cube) :***

Configurer ce marqueur pour que le cube se déplace dans l'espace en suivant les mouvements du marqueur. La position du marqueur déterminera la position relative du cube.

- ***Cinquième et sixième marqueurs (changement de taille) :***

Programmer ces deux marqueurs pour ajuster la taille du cube en fonction de la distance entre eux. Une augmentation ou diminution de la distance entre les marqueurs devra correspondre à un agrandissement ou une réduction du cube.

3. Rendu attendu :

- Le projet doit permettre une interaction fluide et intuitive avec le cube à travers les différents marqueurs.
- Pendant le TP, chaque fonctionnalité sera testée et validée.

4. Livrable final :

- Envoyer le package Unity contenant votre projet complet.
- Assurez-vous que votre code est bien structuré, nettoyé et commenté pour faciliter la compréhension et l'évaluation.