SORBONNE UNIVERSITÉ

POLYTECH SORBONNE

Domus VR

Sujet : Utilisation d'un casque VR Oculus Quest 2 programmé en Unity pour visualiser et commander depuis le casque une solution domotique

Julie FRAYSSE, Afif JEBALI, Mervis MUDRY, Théo PAILLIER

Encadrant : M. Pécheux

Sommaire

01

Contexte & **Origine** du projet

Domus VR : une **extension** du projet Maison Intelligente

Maison Intelligente

Protection de la vie privée.
Solution alternative complète.

+

Une expérience VR.

Une interface **moderne** et **interactive**

Réalité virtuelle

- Technologie innovante et plein essor
- Utilisation d’un casque pour une immersion dans un monde virtuel

Point de départ

Les capteurs/actionneurs

Communiquent avec le protocole **ZigBee**

- Interopérable
- Simplicité
- Faible coût
- Open source
- Basse consommation

Le serveur

Basé sur **Jeedom**

- Multi-protocoles
- Open Source
- Autonome
- Personnalisable
- Open source
- Gratuit

02

Architecture & **Intégration** de notre solution dans le projet plus large

Schéma fonctionnel

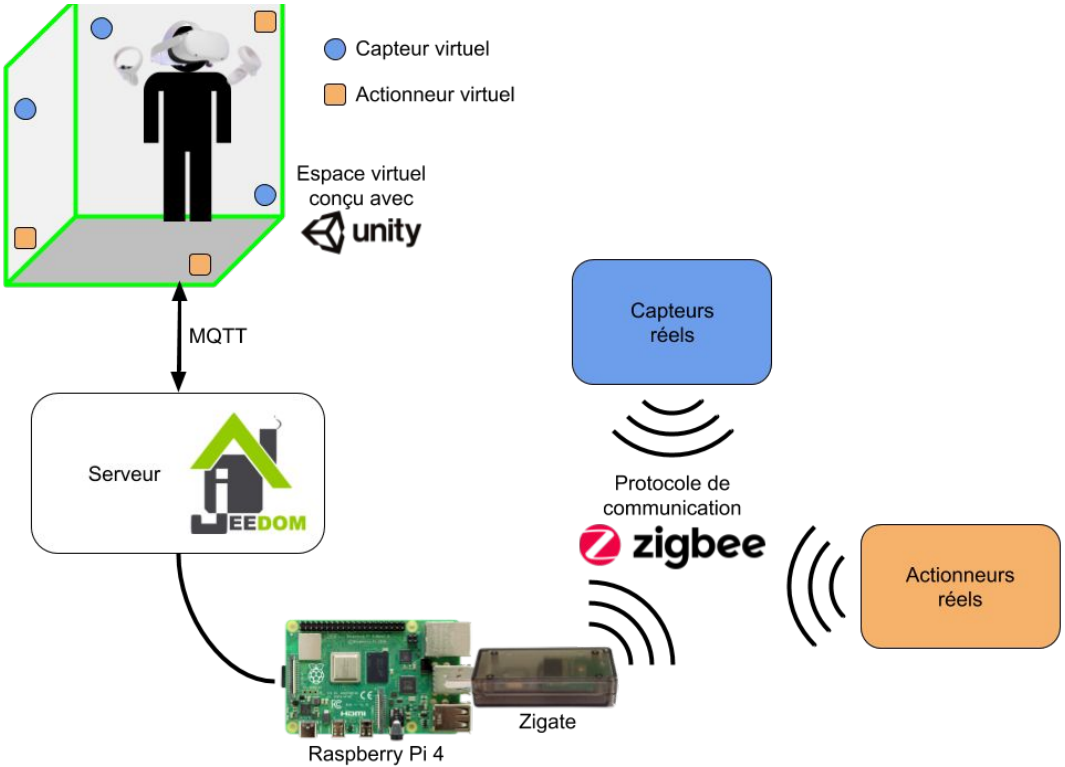
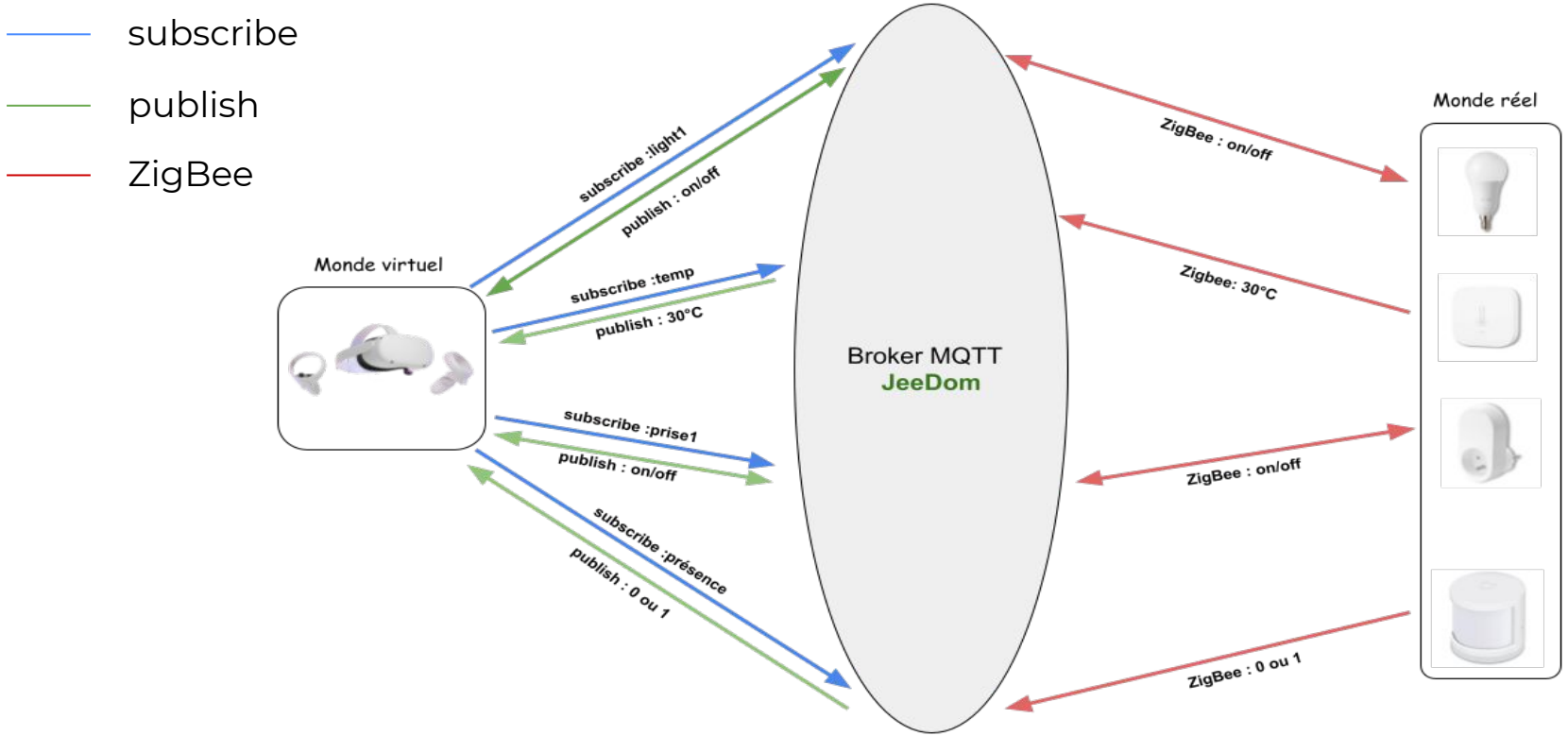


Schéma fonctionnel MQTT



Script TextMeshPro

```
public class Text_Pro_script : MonoBehaviour
{
    public TextMeshProUGUI textMesh;

    Message Unity | 0 références
    public void Start()
    {
        textMesh = GetComponent<TextMeshProUGUI>();
        textMesh.text = "In Text Pro Script's Start";
    }

    2 références
    public void TextUpdate(string s)
    {
        textMesh.text = s;
    }
}
```

Fonction utilisée pour chaque modification de texte

Script MQTT - La classe mqtt

```
[System.Serializable]
public class PrimaryButtonEvent : UnityEvent<bool> { }
public class mqtt :M2MqttUnityClient
//public class mqtt : MonoBehaviour
{
    private List<string> eventMessages = new List<string>();
```

Liste des messages MQTT

- **Déclaration des textes**
- **Récupération des Objets Textes via leurs noms**
- **Lancement de la connexion**

```
Text_Pro_script Text_1;
Text_Pro_script Text_2;

Text_1 = GameObject.Find("Text_1").GetComponent<Text_Pro_script>();
Text_2 = GameObject.Find("Text_2").GetComponent<Text_Pro_script>();

//Text.textMesh = "";
Text_1.TextUpdate("In MQTT's Start");
Text_2.TextUpdate("In MQTT's Start");

Debug.Log("mqtt Start()");
base.Start();
Debug.Log("base.Start() done");
base.OnConnecting();
```

Script MQTT

```
protected override void SubscribeTopics()
{
    client.Subscribe(new string[] { "ReadDomusVR" }, new byte[] { MqttMsgBase.QOS_LEVEL_EXACTLY_ONCE });
}

protected override void Update()
{
    base.Update(); // call ProcessMqttEvents()

    if (eventMessages.Count > 0)
    {
        foreach (string msg in eventMessages)
        {
            ProcessMessage(msg);
        }
        eventMessages.Clear();
    }
}
```

Script MQTT - Allumer/Eteindre une lampe

```
public void PublishLightOn()
{
    string topic = "WriteDomusVR";
    string txtPublish = "1";
    client.Publish(topic, System.Text.Encoding.UTF8.GetBytes(txtPublish), MqttMsgBase.QOS_LEVEL_EXACTLY_ONCE, true);
}
public void PublishLightOff()
{
    string topic = "WriteDomusVR";
    string txtPublish = "0";
    client.Publish(topic, System.Text.Encoding.UTF8.GetBytes(txtPublish), MqttMsgBase.QOS_LEVEL_EXACTLY_ONCE, true);
}
```

Script MQTT - PublishMessage

```
public void PublishMessage()
{
    Text_Pro_script Text_1;

    Text_1 = GameObject.Find("Text_1").GetComponent<Text_Pro_script>();


    //Text.textMesh = "";
    Text_1.TextUpdate("In Publish Message");

    if (LightState)
    {
        PublishLightOff();
    }
    else
    {
        PublishLightOn();
    }
    LightState = !LightState;
```

```
public bool LightState = false;
```

Script MQTT - Les boutons

```
List<InputDevice> devices = new List<InputDevice>();
InputDevices.GetDevicesWithCharacteristics(controllerCharacteristics, devices);

RightHand = devices[0];

RightHand.TryGetFeatureValue(CommonUsages.primaryButton, out bool PrimaryButtonValue);
RightHand.TryGetFeatureValue(CommonUsages.secondaryButton, out bool SecondaryButtonValue);
```

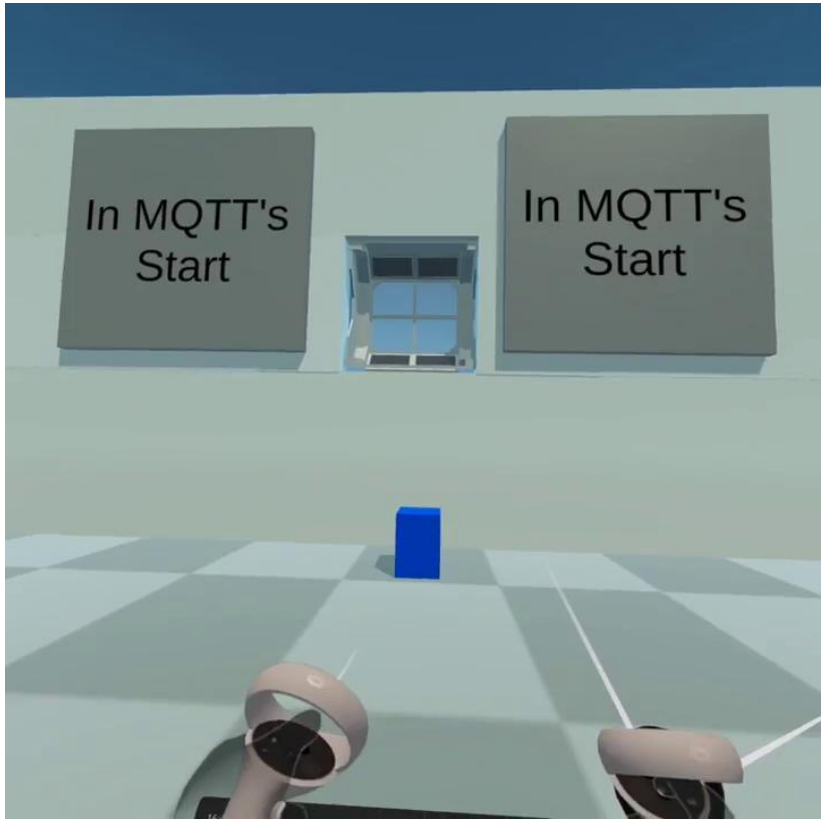
```
Text_1.TextUpdate(PrimaryButtonValue.ToString());
Text_2.TextUpdate(SecondaryButtonValue.ToString());
```

Script MQTT - Afficher/cacher un objet

```
public GameObject Panel;
public GameObject Text;
```

```
Panel.SetActive(!Panel.activeSelf);
Text.SetActive(!Text.activeSelf);
```

Script MQTT - Afficher/cacher un objet

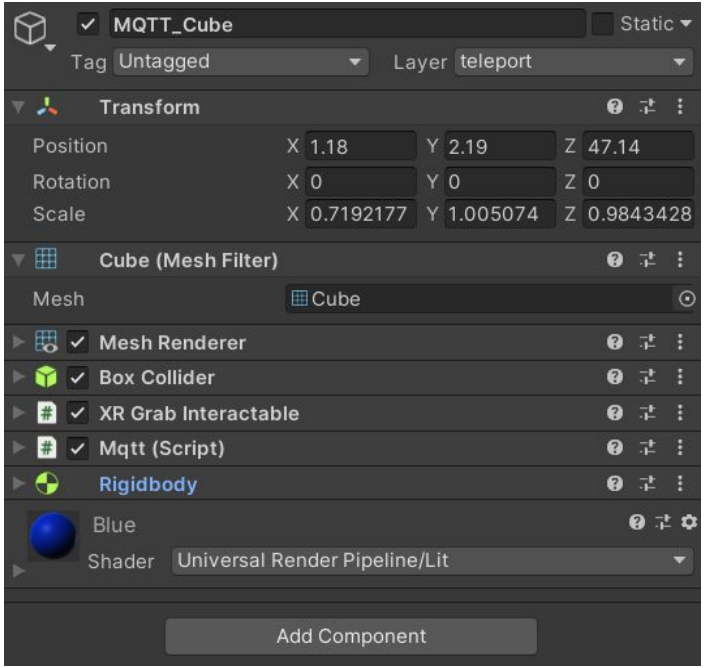


03

Création & **Importation** d’objets dans le casque

Création d'un objet capteur/actionneur

Capteur de température

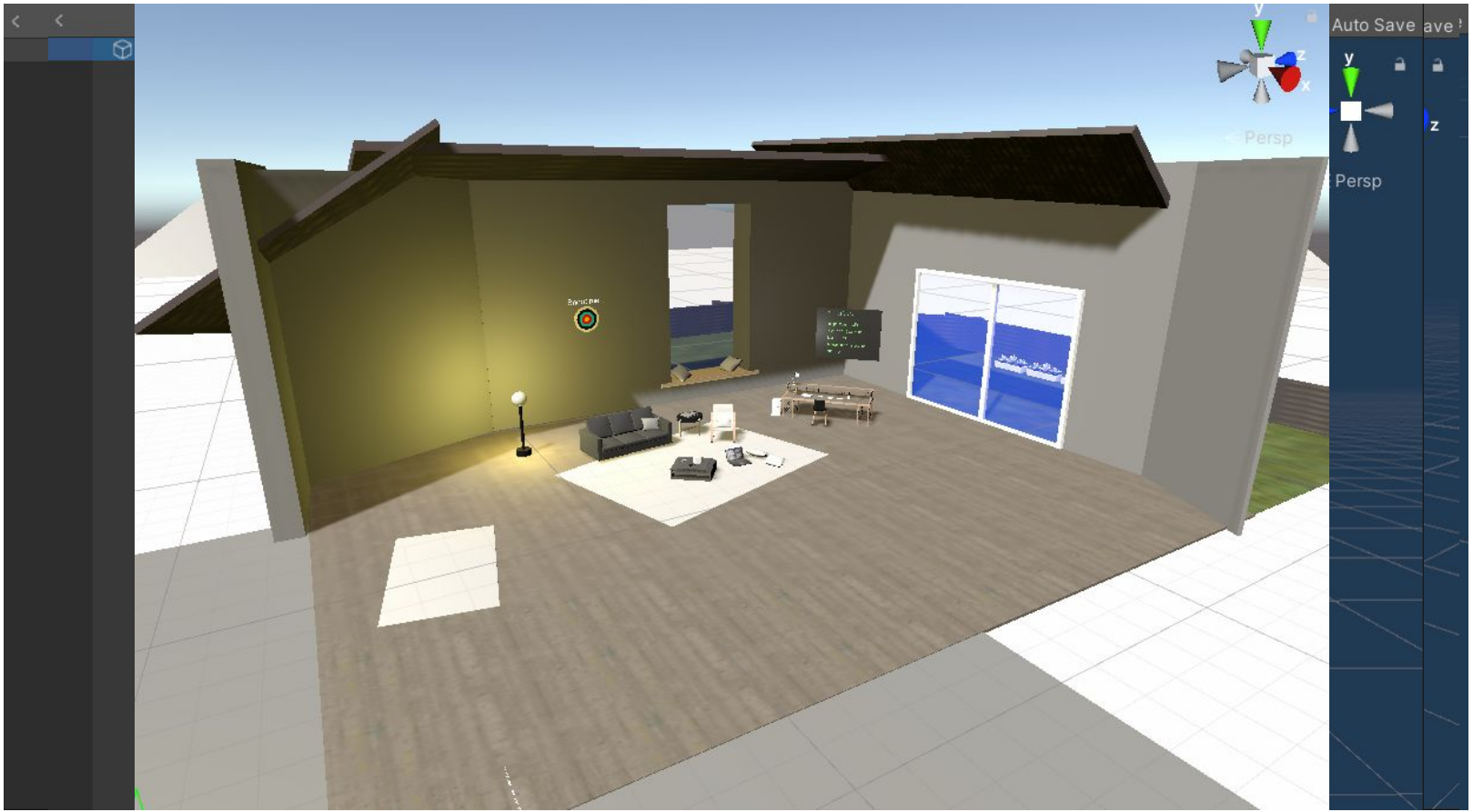


- Box Collider :
  - Permet de repérer la collision en donnant un "corps" à l'objet
- Solid Body :
  - Permet d'appliquer des lois physiques sur l'objet
- Script :
  - *Text_Pro_Script* : Script pour manipuler un text 3D
  - *Mqtt (publish)* : Script pour publier une donnée
- XR Interactable :
  - *Permet les interactions avec l'utilisateur*

Objectif :

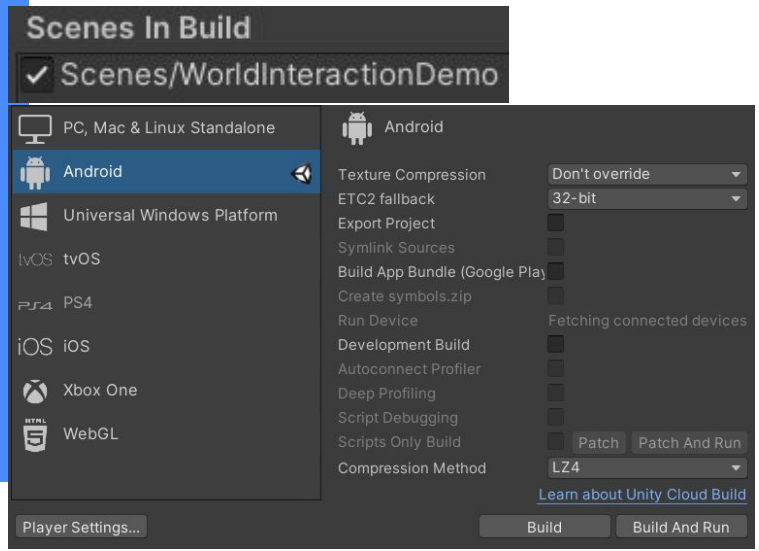
- Afficher la température reçu en MQTT.

Exemple d’objet



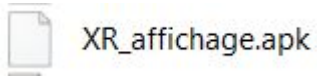
Importation notre application sur l'Oculus Quest 2

- Moteur de jeu multiplatform
- Langage : C#
- Environnement VR
  - Simule la présence d'un utilisateur dans un environnement artificiellement générée par des logiciels



- C'est une société américaine rachetée par Facebook
- Casque de réalité virtuel Oculus Quest 2

Fichier .apk :



- Comprend tous les fichiers nécessaires à l'installation de l'application sur l'oculus.
- Pour le système d'exploitation Android qui permet de faire fonctionner les smartphones, mais aussi pour l'Oculus Quest.

04

Démonstration

Objectifs

Contrôler un actionneur directement à partir du monde virtuel

- exemple : Allumer ou éteindre une lampe dans la vie réelle, grâce à un bouton dans le monde virtuel

Inversement : Dans le monde réel, grâce aux actionneurs, manipuler des objets dans le monde virtuel

- exemple : Allumer une lampe dans le monde virtuel, grâce à un actionneur dans le monde réel

Représenter les pièces d'une maison avec Unity dans le monde virtuel.

Afficher les informations des capteurs dans le monde virtuel.

Notre démo

Ce qu’il est possible de faire dans notre interface utilisateur :

- Évoluer dans une maison 3D
- Visualiser les valeurs des différents capteurs
- Tirer sur la lampe pour l’allumer/l'éteindre

Le petit + : Jouer à un jeu de tir


Shoot me!

05

Retour sur expérience

Découverte de l'environnement **Unity**

Découverte de **l'Oculus Quest 2**

Travail inter-groupe, interfaçage de notre solution **Domus VR** avec le projet **Maison Intelligente**

Découverte du développement d'application **VR**

Merci

Des questions ?