AR/VR Day 02 : Arizona VR

Shoot them all !

Prerequisites

Download the latest LTS version of Unity (2021.3)

Also install the two Android Build Support add-ons through Unity Hub

Exercice 00 : Setting up Unity

Let's create our first VR-ready scene.
Create a Unity project, then open the Build Settings:

- Click on Add Open Scenes to add your active scene.
- Select the Android platform, then Switch.

Now open the Project Settings panel -> XR plugin management. Install the package:

- Choose the open XR plugin, then confirm the reboot.
- In the OpenXR subtab, add the profile interaction matching the Oculus. Switch the render mode to multipass.
- Now back in the XR plugin management tab, select the Android subtab: add the plugin matching your Oculus.
- In the new Oculus tab, choose your target device (in this workshop, we will be compiling on an Oculus Quest 2).

Our scene is now ready to be compiled for Quest 2!

Exercice 01 : Adding the Input Systems

To detect inputs for our Quest 2, we need to install a package: XR interaction toolkit.

Go to Window -> Package manager :

- Switch from Packages: in project to Packages: Unity registry.
- In the Advanced project settings tab, tick "enable pre-release package".
- Back in the package manager, add a package using this git URL: com.unity.xr.interaction.toolkit
- Confirm the backup, then install the package's Default Input Actions.

There should now be an XR interaction toolkit in your Assets folder. Go to Assets > samples > XR interaction Toolkit > 2.0.0-pre6 > Default input action.

Add every default input action to your scene. To do this, click on an XRI object, then click "Add" in the Inspector.

Back in the project settings > Preset Manager, add two filters:

- « right » for the XRI Default Right Controller
- « left » for the XRI Default Left Controller

Exercice 02 : Player Spawn

In your Hierarchy, right-click an empty spot to add an XR Origin (VR).

An Interaction Manager object has appeared in your scene. Give it a new Input Action Manager component. In the action list, add the XRI default input actions.

You can now try to build your scene on your Quest 2.

Exercice 03 : Interacting with objects

Let's grant our two hands the ability to interact with nearby objects:

- Add a sphere collider to your scene. Tick the "is trigger" parameter in the collider component and shrink its radius.
- Add a gun to your scene. Add the box collider, rigidbody and XR Grab interaction components to it.

Build your scene. You should be able to grab the gun!

Exercice 04 : Fire Script

Create a « Fire » script and add it to your gun.

- Create three variables in your object class: one of type gameObject ("Bullet"), one of type Transform ("SpawnPoint"), and a float ("BulletSpeed").
- Don't forget to add the XR interaction toolkit library to your script ("using")
- In the "Start" function, store your grabbable component in a variable, then add a "FireBullet" listener that will serve for a new function.
- Create the FireBullet function that takes the parameter ActivateEventArg.
  In this function, instantiate your Bullet object, give it the position of your spawnPoint variable, then grant it a velocity using your BulletSpeed variable.
- Finally, don't forget to destroy your Bullet object after a couple seconds.

Go back to your Unity scene, and create a "Bullet" sphere. Give it a rigidbody (disable gravity), then change its collision type to Dynamic.

Create an Empty gameobject and place it at the tip of the barrel of your gun, to act as your bullets' spawn point.

Finally, assign these two objects to your gun's "Fire" script in the Inspector.

Exercice 05 : Target spawn

Create a target object: add a cylinder to your scene and change its material color to red.

Create a new "Spawn" script and add it to a new "Spawner" gameObject.

- Create an array to store all your spawn points (it is an array of positions).
- Also create a gameObject representing your object to instantiate (the target).
- Create a StartSpawning() coroutine that instantiates a Target once every few seconds at a random spawn point.

https://docs.unity3d.com/Manual/Coroutines.html

Edit your "Fire" script:

- Make it so when a bullet hits a target, that target is destroyed.

Use tags !

Exercice 06 : Bonus

You can now shoot with a gun in VR, congratulations!

With your extra time, you are now free to:

- Add a scoring system: with each target hit, increment a score integer.
- Create moving targets.
- Make your targets change colors when hit.
- Add audio to your scene.