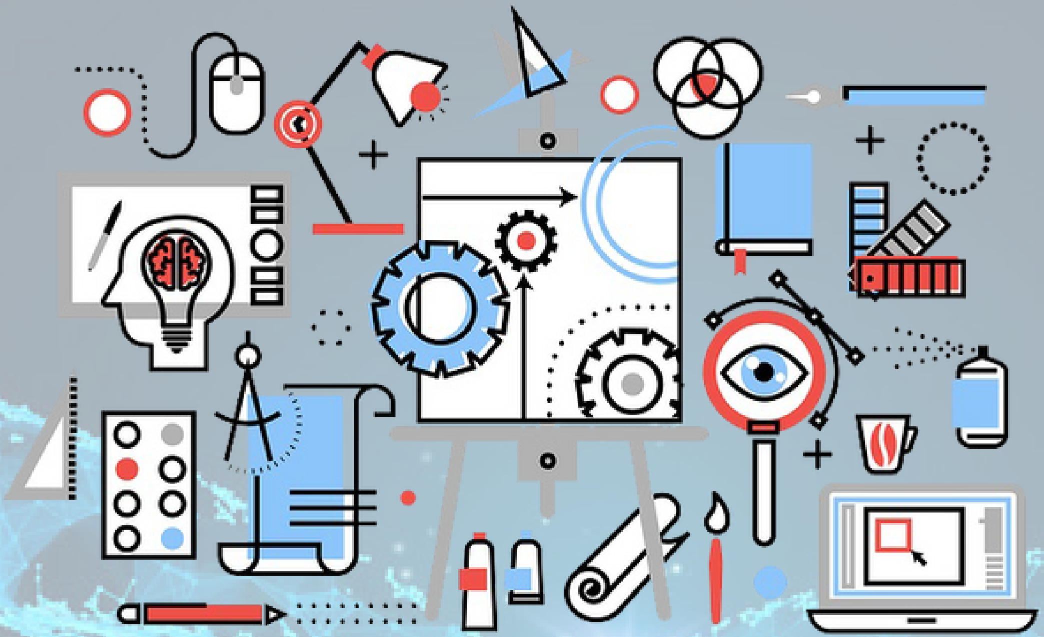
XRIG-IITM



AR/VR TUTORIAL#6

Contacts Details :

Maadhav Patel | +91 63536 47545
Pratik Zade | +91 82750 15550
Shubham Agrawal | +91 98184 37967
Harshal Gajbhiye | +91 95117 64800

Playing with Unity APIs

Unity has a lot of flexibility about the things you can do with it. While it has its inbuilt physics engine, you can still use scripts to manipulate or completely change it. You can also script your own render pipeline and tweak hdrp to make AAA-level graphics. But so much customizability while a good thing can also be confusing and overwhelming.

So Unity offers a lot of inbuilt functions, APIs,etc that make your life easy by taking away all the extra coding work. Letting you just call functions with a few words and simplifying tasks with proper logic. So today we will be playing with some of such functions which might serve you well in your tenure.

GameObject.SetActive

Most things in your Unity projects that appear in your hierarchy are known as 'GameObjects.' As the name suggests, they are the building blocks of your project—the bricks of your house or the pieces of your jigsaw puzzles. Manipulating them using scripts is an essential part of working with Unity.

SetActive can disable and enable a GameObject in runtime. Doing this can be helpful in many ways. Maybe you want something to be in your scene to be present but not influence any of the other objects? Do you want to stop using a game object during certain stages only to re-enable it again?
Many things are possible. Could you take a swing at it by doing the following task?

Make a scene with multiple camera angles/perspectives. Make a script that will switch between them using SetActive. You may switch between first-person and third-person perspectives or many camera angles using a key. Go a step further and make it random between several cameras. Go even further and make the switching a periodic function.

Transform

Transform is another such API class. It's used to store and manipulate an object's position, rotation, and scale. There are many public methods available to make use of it here. Please take a look at it and do the following task.

Make two scenes with two game objects.
In one scene, make one of the objects follow behind another object at a distance.

In another scene, make one GameObject revolve around another GameObject with a camera (This should act as the center of the revolution). Make a script that will make the GameObject with the camera i.e. the center one rotate around its axis slowly to face to revolving GameObject upon pressing a key. Try to make the rotation as smooth and non-instantaneous as possible.

Instantiate

Instantiate is a method used to create and load a predetermined asset/prefab into the scene upon being called. The advantage that this function offers is you can call it over and over again to create identical copies of any prefab you may need i.e. bullets. Also you will not need to create numerous copies of the same GameObject in the base build since instantiate creates copies in run time. This significantly debloats your base build and avoids hogging up unnecessary space.
A counter function called destroy can be used to destroy any GameObjects in the scene. This can help free up space and reduce resource consumption in run time. Also can be used to despawn assets after they’ve served their use (bullets may be destroyed after a while) to free up memory.

Using the above knowledge and Unity Documentations, create a scene where you use a gun to shoot bullets wherever the mouse points on the screen. Next create targets in the scene that you will shoot and fill the scene with them. Make use of Instantiate and Destroy to create bullets and targets as well as destroy them. You may try to make the targets randomly appear in the scene using instantiate if you want.