January 29, 2021

Muhammad Taha Bin Farooq

Mood Games

Nilüfer, Bursa, Turkey

Documentation

Secret Panel

Table of Contents

[Introduction 2](#_Toc62775799)

[Prerequisite 2](#_Toc62775800)

[How to use Camera Secret Panel? 2](#_Toc62775801)

[Camera States 4](#_Toc62775802)

[How to create new camera state? 5](#_Toc62775803)

[How to change Camera State at Runtime? 5](#_Toc62775804)

# Introduction

The camera secret panel is a utility designed for game designer to decide what camera will look best in gameplay. This panel is specifically designed for third person camera and top-down camera.

# Prerequisite

* Odin Inspector

(You can import those packages from the Menu Button Mood Games => About)

# How to use Camera Secret Panel?

To use this utility a developer need to drag and drop the prefab in the canvas located in the prefab folder.

Path to the folder Game => Prefabs => Camera Setting => Camera Setting.prefab

You can see the button appear at the Bottom Right corner. Now, select the Cam Secret Panel from hierarchy you can see a script of CameraSetting.cs, in which most the variables are defined just need to tweak few variables base on given requirement. A developer needs to tweak only 3 variables.

1. Camera State.
   1. This is the Camera behavior these variables will apply changes and can changed base on the project to project.
2. Default Offset.
   1. This variable is the default position offset on which camera will maintain before game designer tweak (or I say what developer thinks is best for game).
3. Default Rotation.
   1. This is the default rotation offset on which camera will maintain before game designer tweak (or I say what developer thinks is best for game).



Figure 1 Camera Script variables

After that, **attach** SmoothCamera.cs with the camera object in the hierarchy.

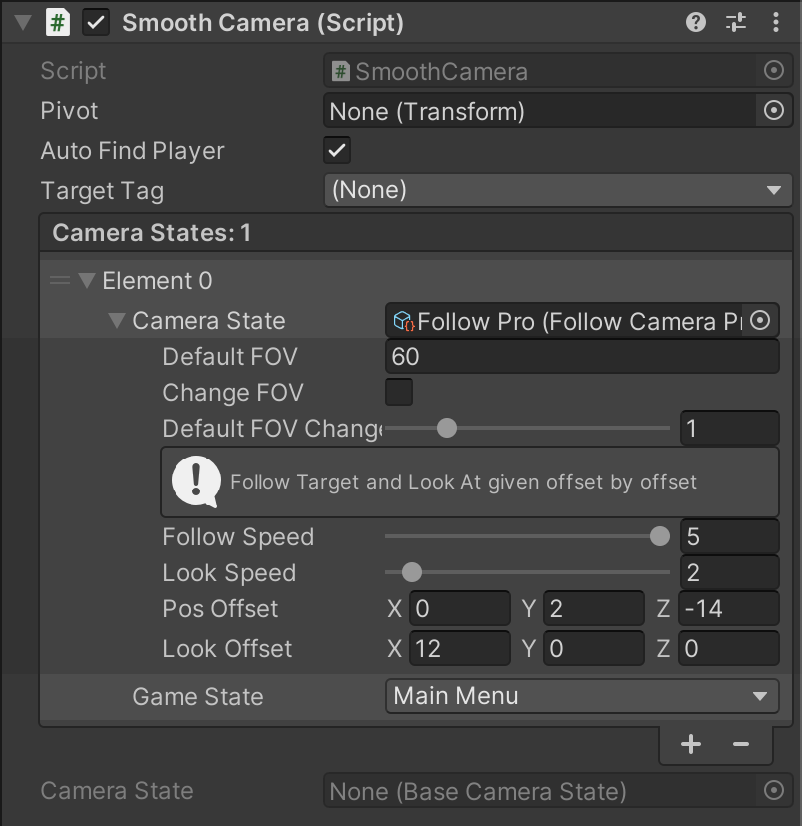


Figure 2 Smooth Camera look in inspector

In this picture you can see there is a **pivot** assign that from the respective child. A **bool** which detects player automatically by the **tag** **provided**. Below that there are camera states. Let me explain camera state.

## Camera States

The camera states are the state of camera which contain 2 information.

1. The state name.
2. The behavior of camera.

This is a very modular approach, because a developer defines state of camera and whenever player change to that camera will automatically smoothly lerp towards that state without writing extra code.

Currently there are 6 behavior defined in the package named below.

* Follow Camera Pro
* Follow Camera
* Follow Camera Rotation
* Follow Rotator Camera
* Rotator Camera
* Simple Follow

You can test all these behaviors and achieve desired result. These behaviors are actually [scriptable objects](https://docs.unity3d.com/Manual/class-ScriptableObject.html). You can create scriptable objects from the RightClick in Project Window.

Create => Utilities => Camera => XYZ State

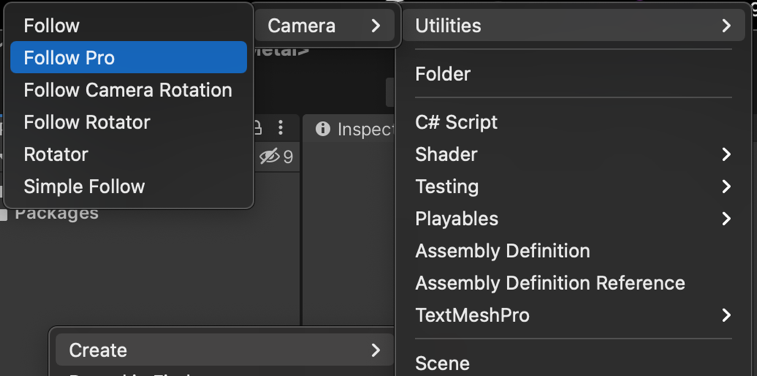


Figure 3 Path to create new state

# How to create new camera state?

It’s very easy and simple follow the given steps.

1. Create a new script (better if with the other camera behavior scripts just to make project more organized).
2. Open in the code editor and inherit from the BaseCameraState.
3. Add the given line above the class name. For more help please look at the other scriptable objects scripts.

[CreateAssetMenu(fileName = "DESIREDNAME", menuName = "Utilities/Camera/DESIREDNAME", order = 0)]

1. Now, implement the abstract function and write your functionality. For more help read any of the other script.

# How to change Camera State at Runtime?

It’s also very easy and simple too. As we all know camera will follow player, right?

So, we need to inherit player controller with the script provided in the package named as PlayerMovement.cs. And now we need to just change state by writing the following line.

PlayerState = PlayerState.XYZ

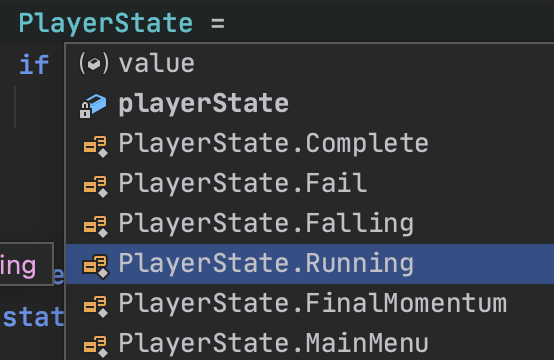


Figure 4 Writing Player State