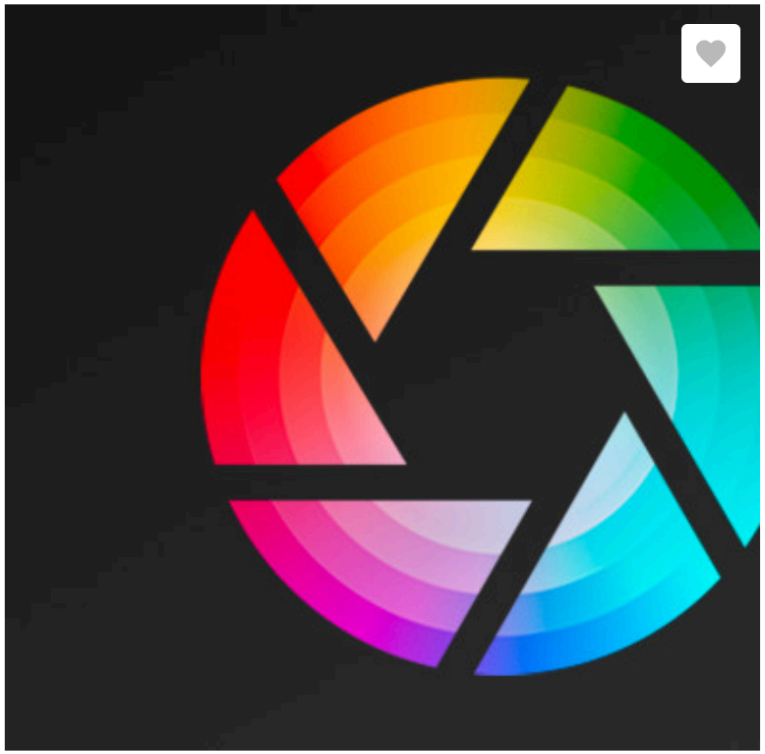
Inverting Colors (And Other Camera FX)

At least one of you has asked me about doing a ‘negative’ color effect, or inverting the colors in your scene. Unity has made this very easy in recent versions.

Download and import the **Post Processing Stack** add-on from the Unity Asset Store (This component is made by Unity Technologies)



UNITY TECHNOLOGIES

Post Processing Stack

FREE

★★★★★ 127 user reviews

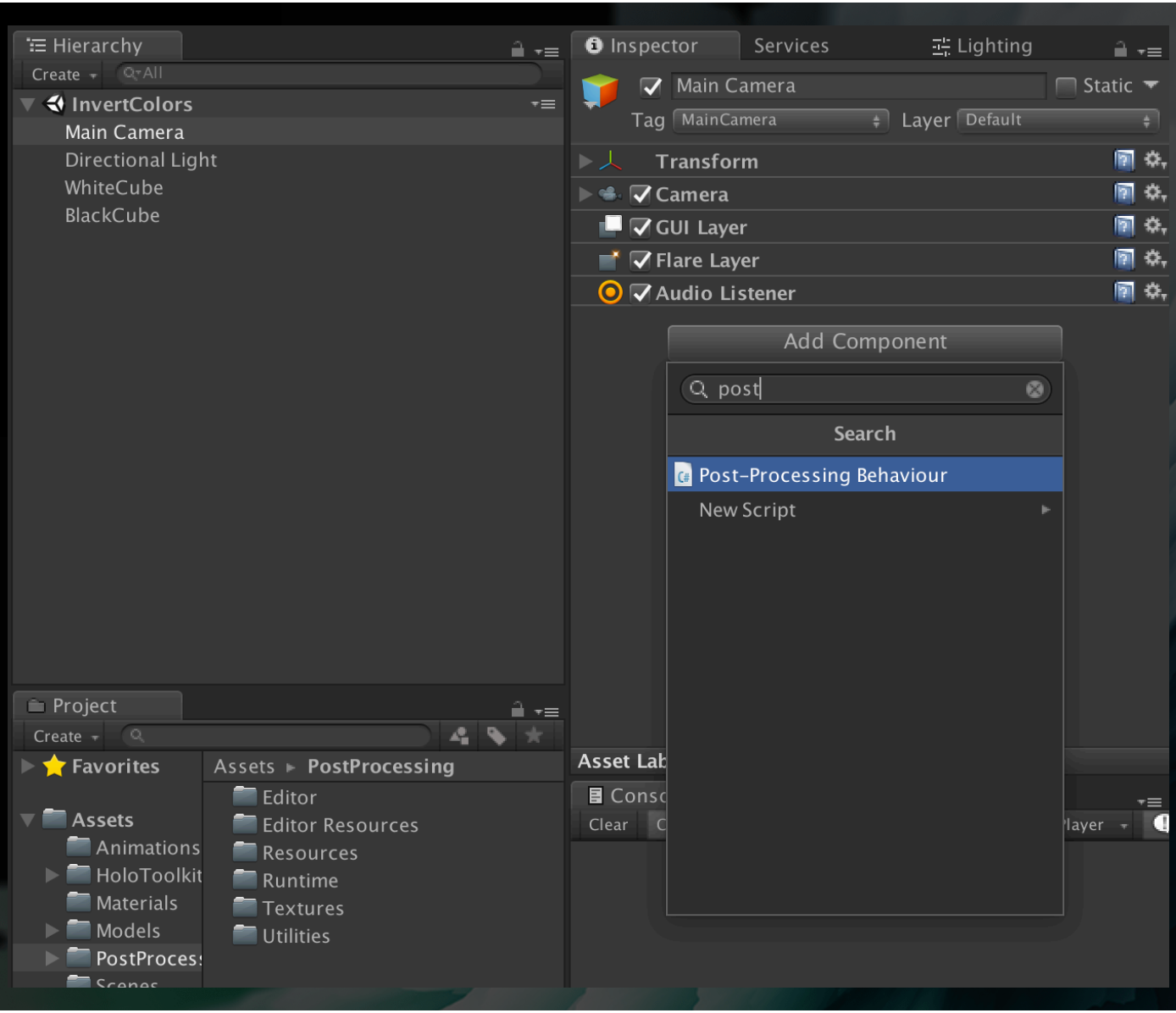
Open in Unity

Taxes/VAT calculated at checkout

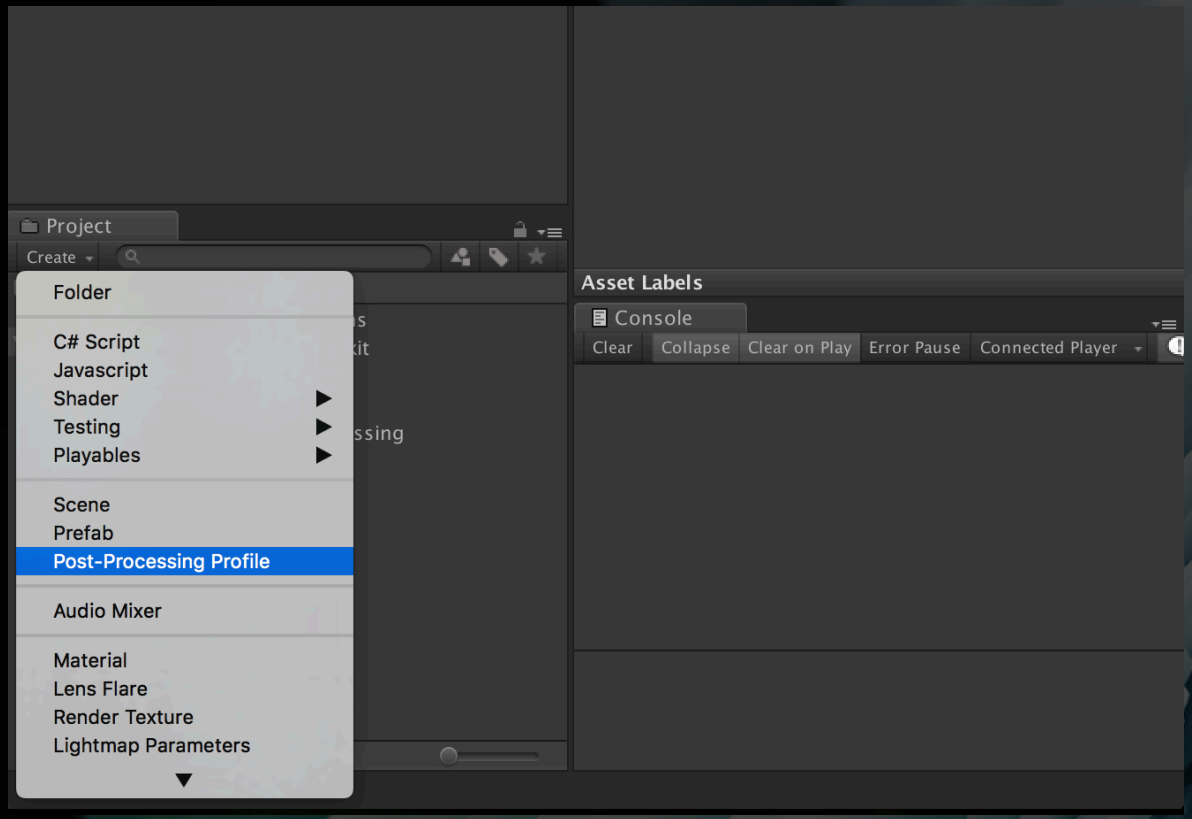
The new Unity post-processing stack is an über effect that combines a complete set of image effects into a single post-process pipeline. This has a few advantages :

- Effects are always configured in the correct order.
- It allows combination of many effects into a single pass.
- It features an asset-based configuration system for easy preset management.
- All effects are grouped together in the UI for a better user experience

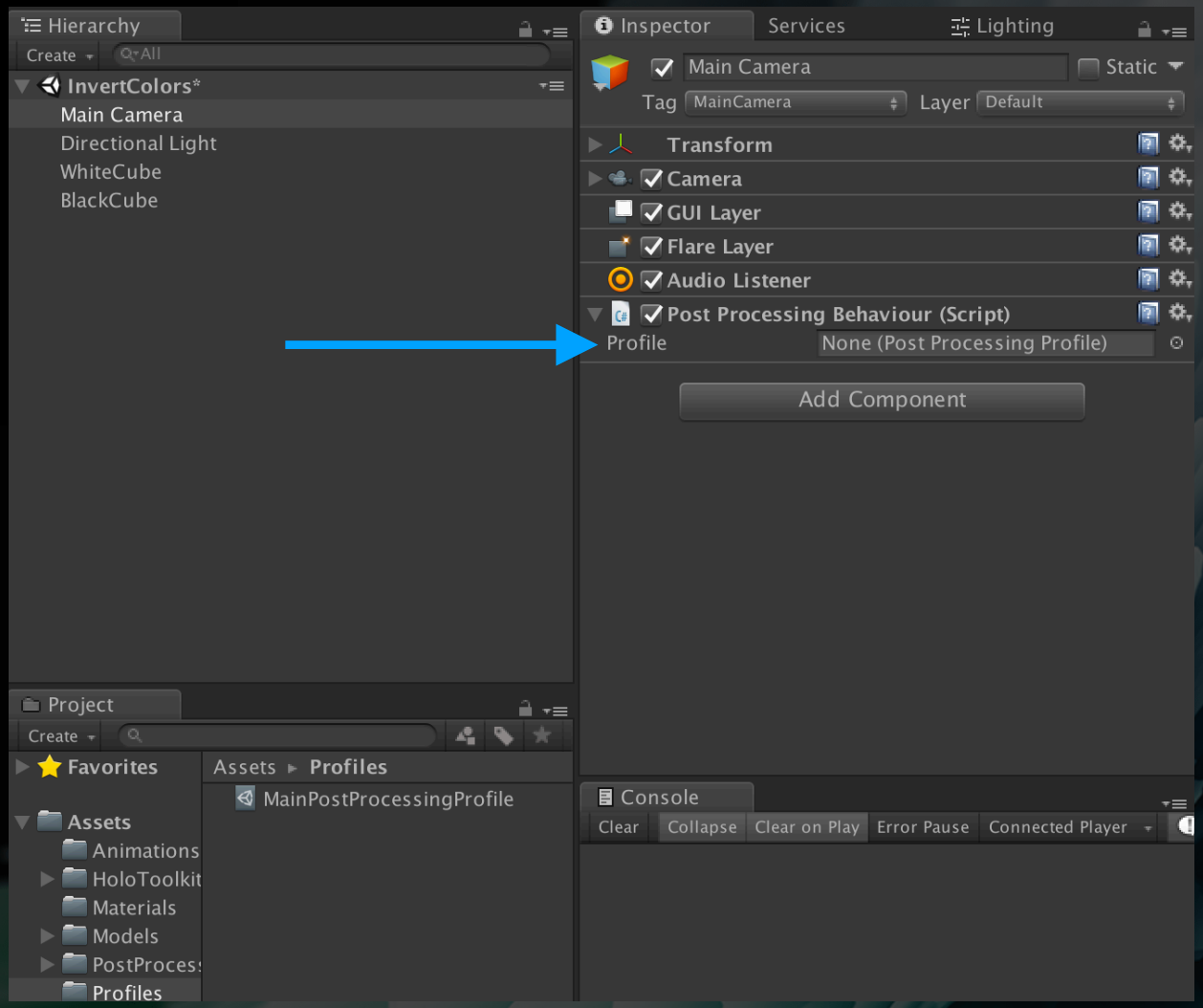
Select the camera in your scene that you would like to apply an effect to and add the **Post-Processing Behaviour** component to it.



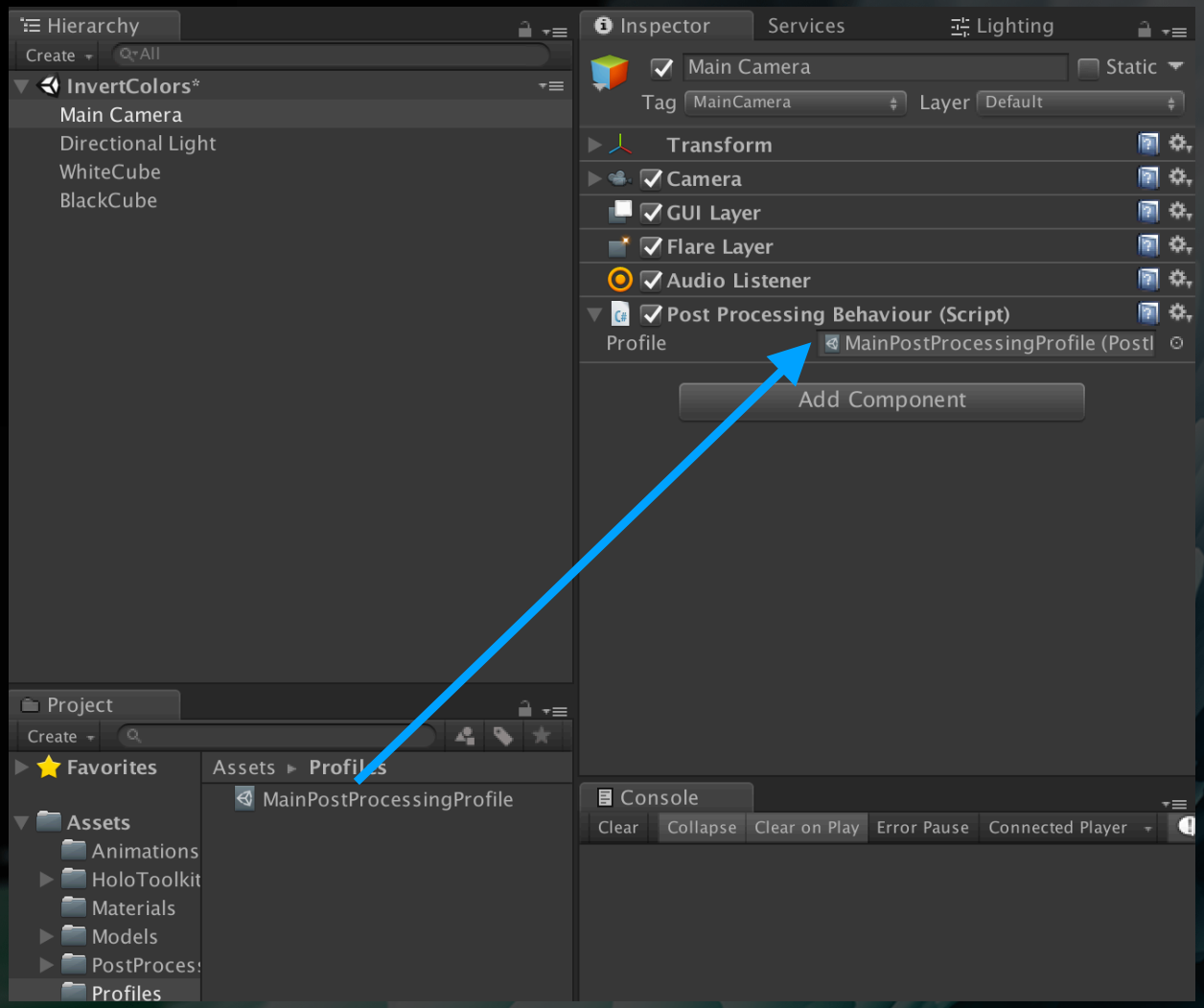
In your *Assets* folder, create a **Post-Processing Profile**.



Select your camera again and drag the new Profile you created in to the slot in the Behaviour that you added.

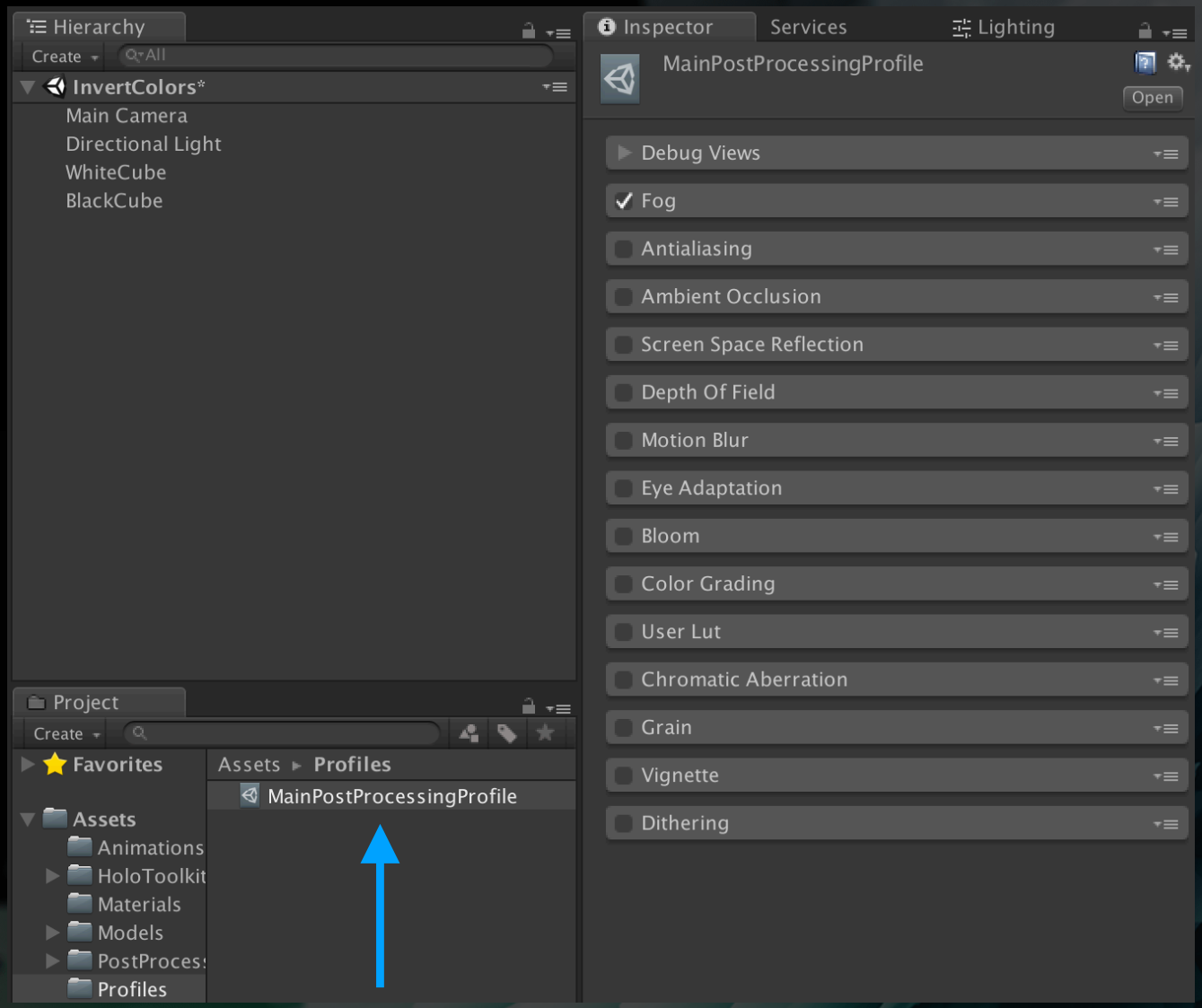


Select your camera again and drag the new Profile you created in to the slot in the Behaviour that you added.



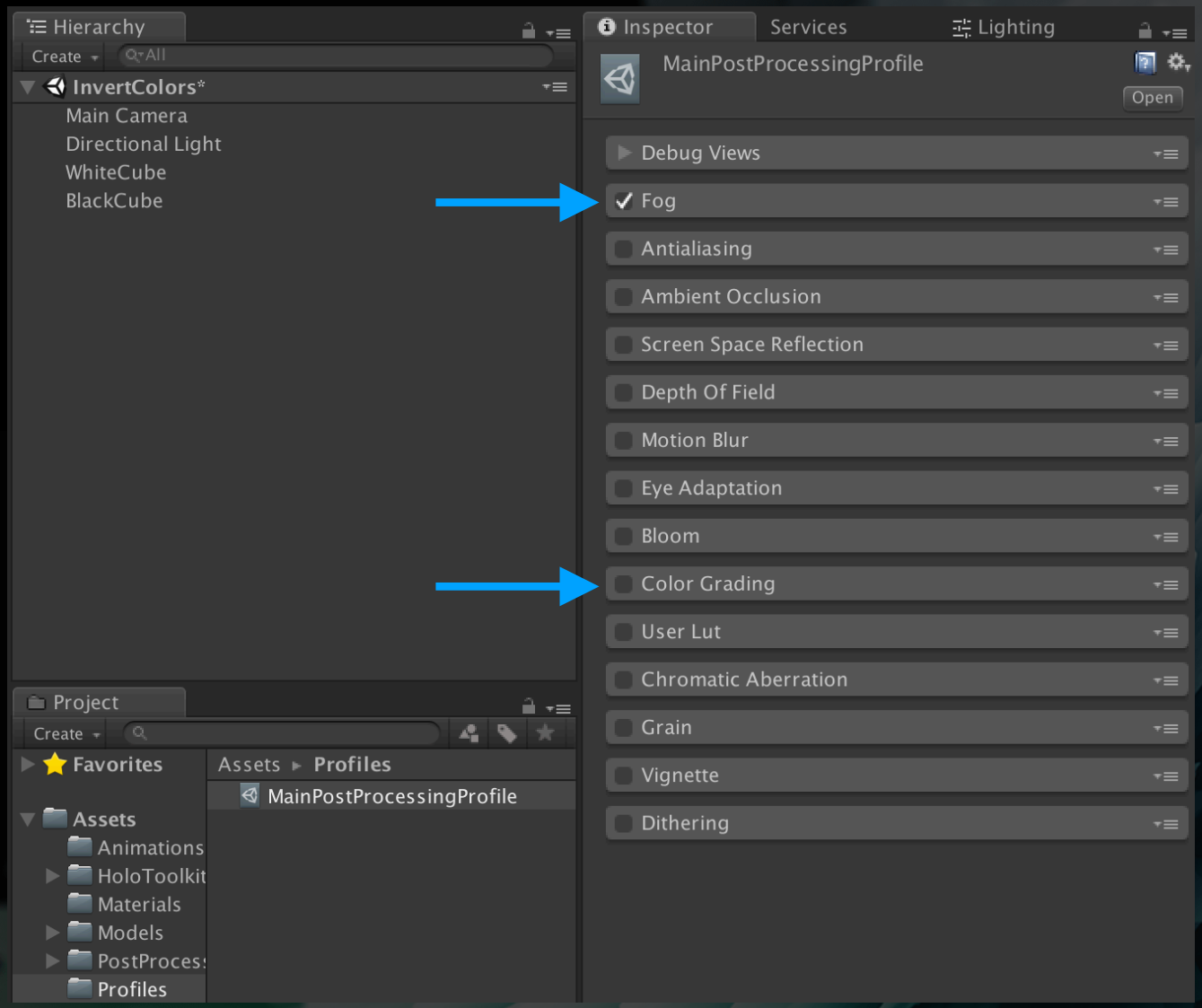
Click on the **Profile** in your Assets folder to bring it up in the inspector.

Each of these components are a visual effect you can add to your camera output!



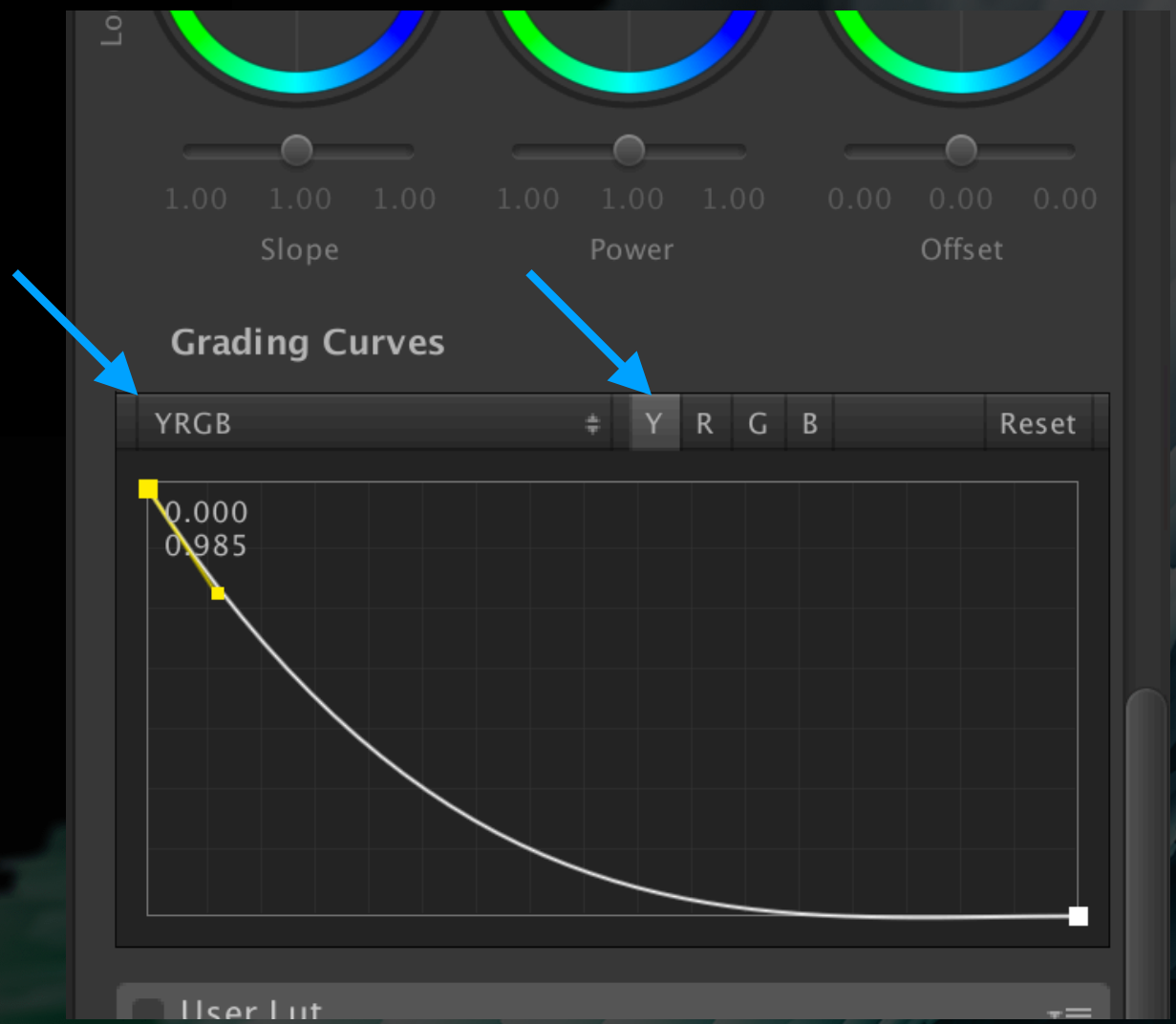
For now, we're going to focus on affecting the color in our scene.

Disable the **Fog** component (This is enabled by default) and enable the **Color Grading** component. Click on its name to open the tab and take a look at the parameters.



At the bottom of the **Color Grading** tab, adjust the Y-graph component of the YRGB curve (this is the one that is visible by default).

For an inverted (i.e ‘negative’) look, you will want the line to start at the top-left and end at the bottom-right. You can play with the amount that it curves until you see the desired effect.



We can now turn this effect on and off in our scripts.

Create a script called ToggleColorEffect.cs and add it to the Camera that has the Post-Processing Behaviour on it.

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

// Make sure to tell Unity that you will be using code
// defined in PostProcessing add-on by adding this line
// before your class definition:
using UnityEngine.PostProcessing;

public class ToggleColorEffect : MonoBehaviour {

  // This is just for the sake of example, to let us
  // toggle the effect on and off inside the Unity editor:
  public bool enableEffect = false;

  private PostProcessingProfile fxProfile;

  void Start() {

    // Find the 'prfile' element of the PostProcessingBehavior component
    // and assign it to the fxProfile variable
    fxProfile = GetComponent<PostProcessingBehaviour>().profile;
  }

  void Update() {
    if (enableEffect) {

      // This line allows you to enable the specific component that you
      // want, in this case 'color grading'
      fxProfile.colorGrading.enabled = true;

      // Remember that the dots represent that you are accessing something
      // that is a **inside** of the thing before the period. In this case
      // you are setting the variable 'enabled' of the component 'colorGrading'
      // inside of the 'fxProfile.'
    } else {
      fxProfile.colorGrading.enabled = false;
    }

  }
}
```

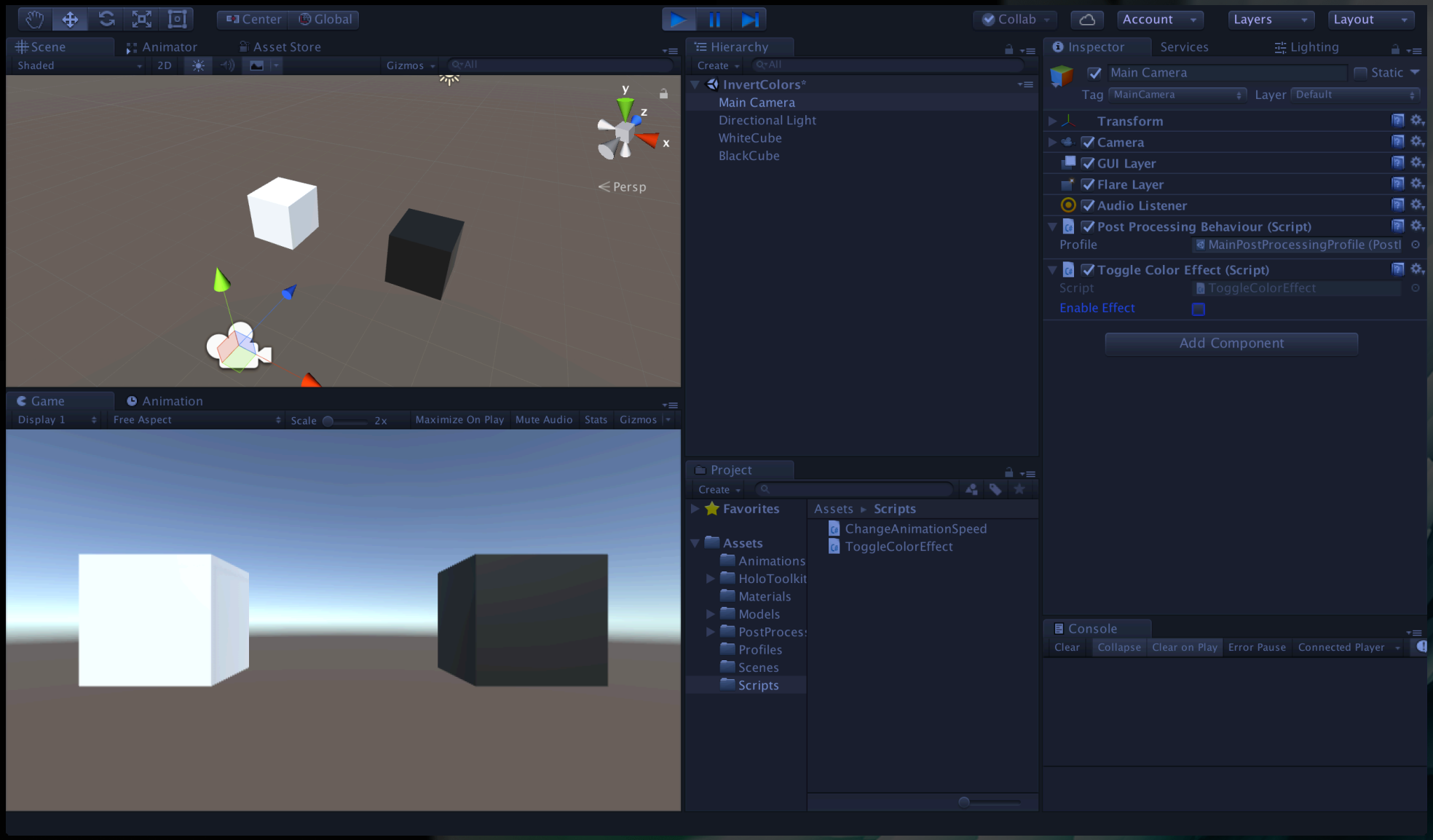
```
// Make sure to tell Unity that you will be using code
// defined in PostProcessing add-on by adding this line
// before your class definition:
using UnityEngine.PostProcessing;
```

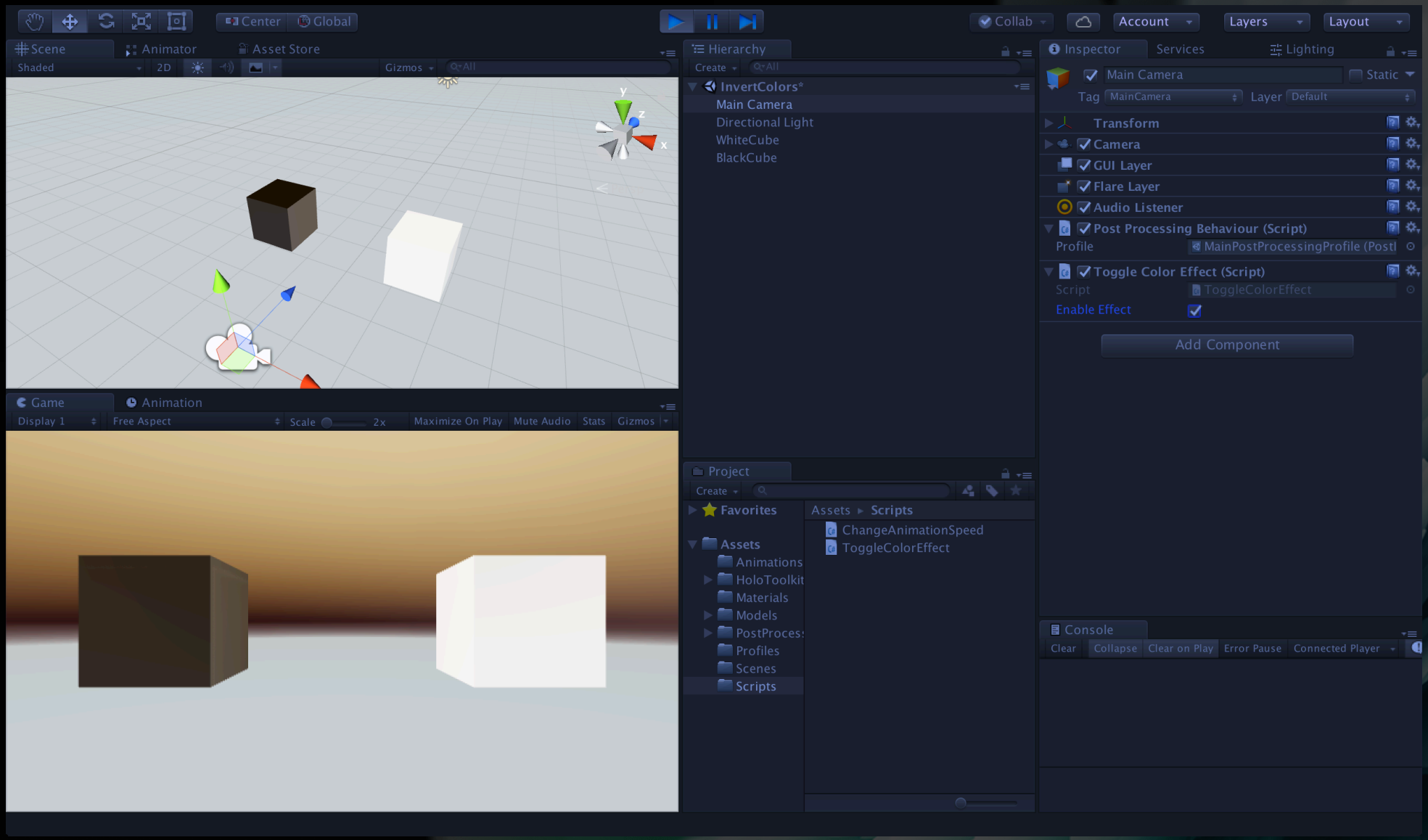
```
// Create a variable to hold a reference to
// your Post-Processing Profile
private PostProcessingProfile fxProfile;

void Start() {

  // Find the 'profile' element of the PostProcessingBehavior component
  // and assign it to the fxProfile variable
  fxProfile = GetComponent<PostProcessingBehaviour>().profile;
}
```

```
// This line allows you to enable the specific component that you
// want, in this case 'color grading'
fxProfile.colorGrading.enabled = true;
```


Center
Global
Collab
Account
Layers
Layout
Scene
Animator
Asset Store
Shaded
2D
Gizmos
Persp
Hierarchy
Create
InvertColors*
Main Camera
Directional Light
WhiteCube
BlackCube
Inspector
Services
Lighting
Main Camera
Static
Tag
MainCamera
Layer
Default
Transform
Camera
GUI Layer
Flare Layer
Audio Listener
Post Processing Behaviour (Script)
Profile
MainPostProcessingProfile (Postl
Toggle Color Effect (Script)
Script
ToggleColorEffect
Enable Effect
Add Component
Game
Animation
Display 1
Free Aspect
Scale
2x
Maximize On Play
Mute Audio
Stats
Gizmos
Project
Create
Favorites
Assets
Animations
HoloToolkit
Materials
Models
PostProces
Profiles
Scenes
Scripts
Assets ▸ Scripts
ChangeAnimationSpeed
ToggleColorEffect
Console
Clear
Collapse
Clear on Play
Error Pause
Connected Player


Center
Global
Collab
Account
Layers
Layout
Scene
Animator
Asset Store
Shaded
2D
Gizmos
Hierarchy
Create
InvertColors*
Main Camera
Directional Light
WhiteCube
BlackCube
Inspector
Services
Lighting
Main Camera
Static
Tag
MainCamera
Layer
Default
Transform
Camera
GUI Layer
Flare Layer
Audio Listener
Post Processing Behaviour (Script)
Profile
MainPostProcessingProfile (Postl
Toggle Color Effect (Script)
Script
ToggleColorEffect
Enable Effect
Add Component
Game
Animation
Display 1
Free Aspect
Scale
2x
Maximize On Play
Mute Audio
Stats
Gizmos
Project
Create
Favorites
Assets
Animations
HoloToolkit
Materials
Models
PostProces
Profiles
Scenes
Scripts
Assets ▸ Scripts
ChangeAnimationSpeed
ToggleColorEffect
Console
Clear
Collapse
Clear on Play
Error Pause
Connected Player