

UNITY GAME DEVELOPMENT COURSES

Unit 3

**Unity Level Design, Post Processing Stack, User
Interface**

TOPICS

- **Game Level Design using ProBuilder or other map editor**
- **Post Processing Unity : HDRP**
- **3D Object Creation**
 - **Rendering**
 - **Texture**
 - **Lighting**

GROUP PROJECT

- At the end of our lesson, all of you need to create a group project in 2 people to create a virtual reality game and give a presentation to us.
1. Introduction: Briefly introduce the project and its objectives.
 2. Game Concept: Discuss the concept of the game and what makes it unique. Highlight the key features that we plan to include in the game.
 3. Gameplay Mechanics: Discuss the gameplay mechanics, including how the player will interact with the game world and the controls they will use.
 4. Graphics and Sound Design: Discuss the visual and audio elements of the game, including the art style and the sound effects and music.
 5. Technical Implementation: Discuss the technical details of how we plan to develop the game, including the software and hardware that we will use.
 6. Challenges and Solutions: Discuss any challenges that we anticipate facing during the development process and our proposed solutions to overcome them.

QUICK PROTOTYPING TOOLS

- ProBuilder -

<https://unity3d.com/unity/features/worldbuilding/probuilder>

- **Realtime CSG** -

https://assetstore.unity.com/packages/tools/modeling/realtim_e-csg-69542

- **MAST - Modular Asset Staging Tool** -

<https://assetstore.unity.com/packages/tools/level-design/mast-modular-asset-staging-tool-154939>

INTRODUCTION

- **ProBuilder**

- ProBuilder is a unique hybrid of 3D modeling and level design tools, optimized for building simple geometry but capable of detailed editing and UV unwrapping as needed.
- It can quickly prototype structures, complex terrain features, vehicles and weapons, or make custom collision geometry, trigger zones, and nav meshes.



- <http://www.procure3d.com/probuilder/>

PROBUILDER

- Introduction
- Create Your First Brush
- Brush Tools
 - Free mode
 - Resize mode
 - Vertex mode
 - Edge mode
 - Face mode
 - Clip mode
- Build Settings

LABS EXCERCISE

- Making an car using ProBuilder
- Document : Designing Assets in Unity with ProBuilder
- Intro to HD Render Pipeline(HDRP) in Unity 2019.3 (Updated Workflow)
- <https://www.youtube.com/watch?v=VD5Qr4Rt7-Q>
- https://www.youtube.com/watch?v=HqaQJfuK_u8

ONLINE

- Making an FPS Level with ProBuilder in Unity 2019! (Tutorial - Beginner Friendly)
- <https://www.youtube.com/watch?v=EiWItni5JmA>
- Intro to HD Render Pipeline(HDRP) in Unity 2019.3 (Updated Workflow)
- <https://www.youtube.com/watch?v=VD5Qr4Rt7-Q>
- https://www.youtube.com/watch?v=HqaQJfuK_u8

INTRODUCTION OF POST PROCESSING



Scene with no post-processing



Scene with post-processing

INTRODUCTION OF POST PROCESSING

Render pipeline	Post-processing support
Built-in Render Pipeline	The Built-in Render Pipeline does not include a post-processing solution by default. To use post-processing effects with the Built-in Render Pipeline, download the Post-Processing Version 2 package. For information on using post-processing effects in the Built-in Render Pipeline, see the Post-Processing Version 2 documentation .
Universal Render Pipeline (URP)	URP includes its own post-processing solution, which Unity installs when you create a Project using a URP Template. For information on using post-processing effects in URP, see the URP post-processing documentation .
High Definition Render Pipeline (HDRP)	HDRP includes its own post-processing solution, which Unity installs when you create a Project using an HDRP Template. For information on using post-processing effects in HDRP, see the HDRP post-processing documentation .

TOPICS

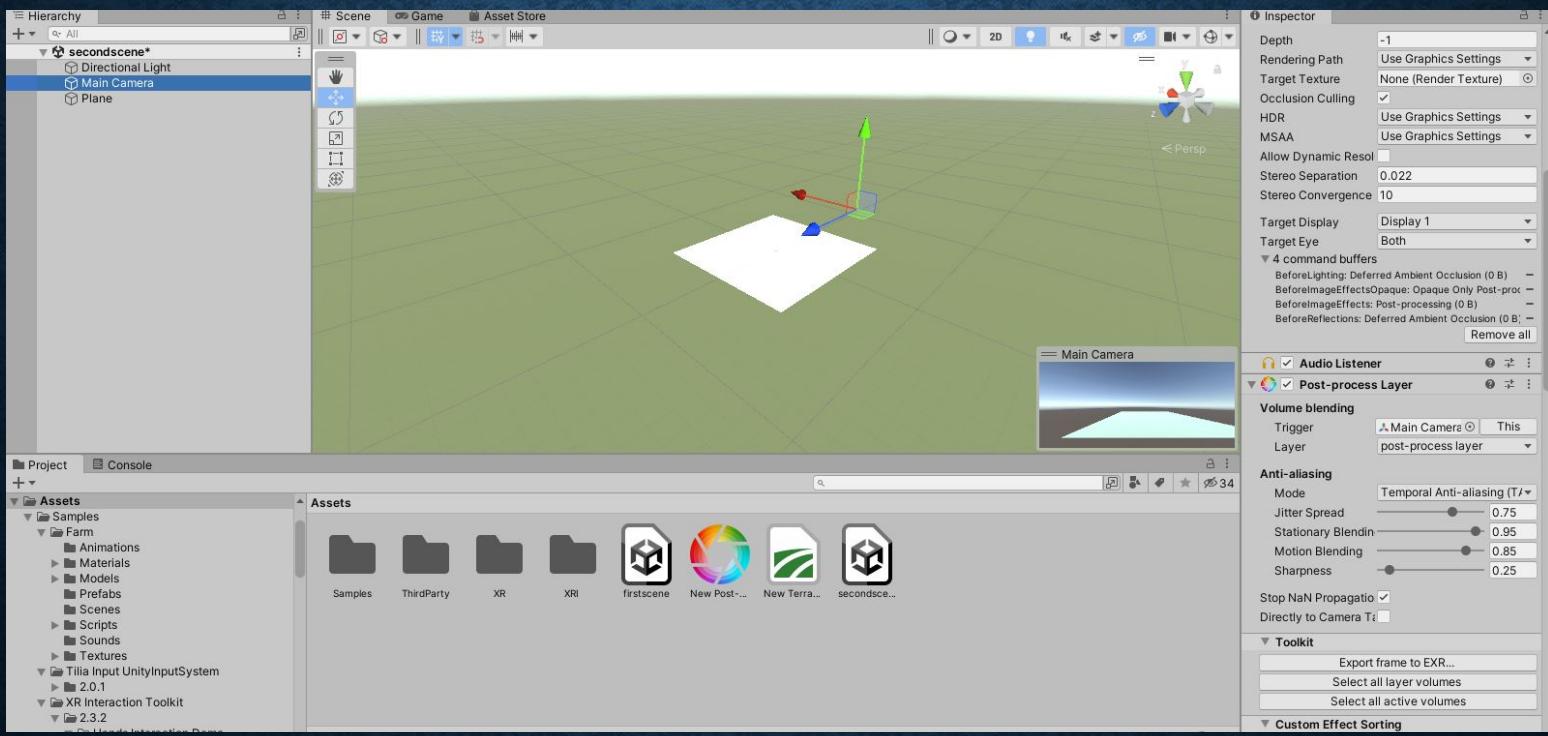
- Game Art Principles
- Post-processing stack
- Depth of Field
- Color Grading
- Tonemapping
- Ambient Occlusion

EFFECTS

- Ambient Occlusion
- Anti-aliasing
- Auto Exposure
- Bloom
- Chromatic Aberration
- Color Grading
- Deferred Fog
- Depth of Field
- Grain
- Lens Distortion
- Motion Blur
- Screen Space Reflections
- Vignette

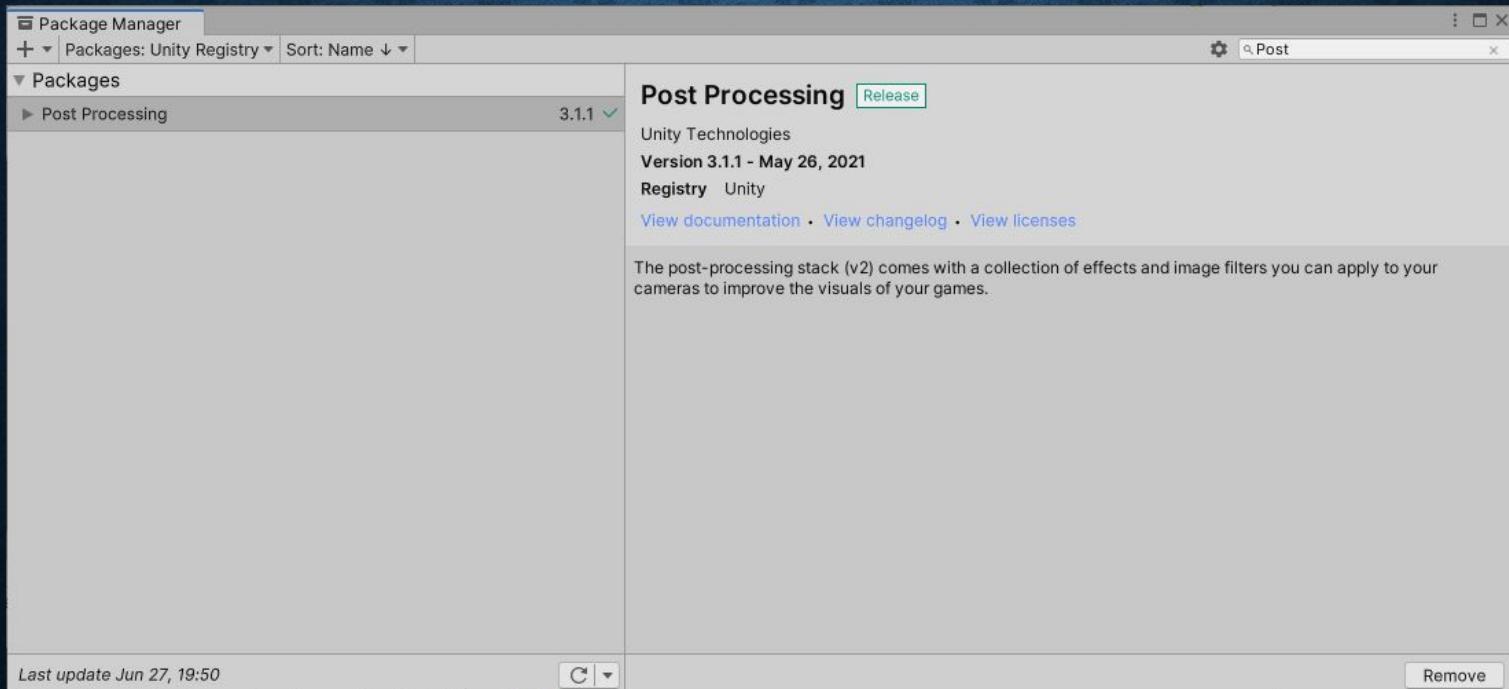
TERRAIN PROJECT

- Create a new project including the following object
 - plane



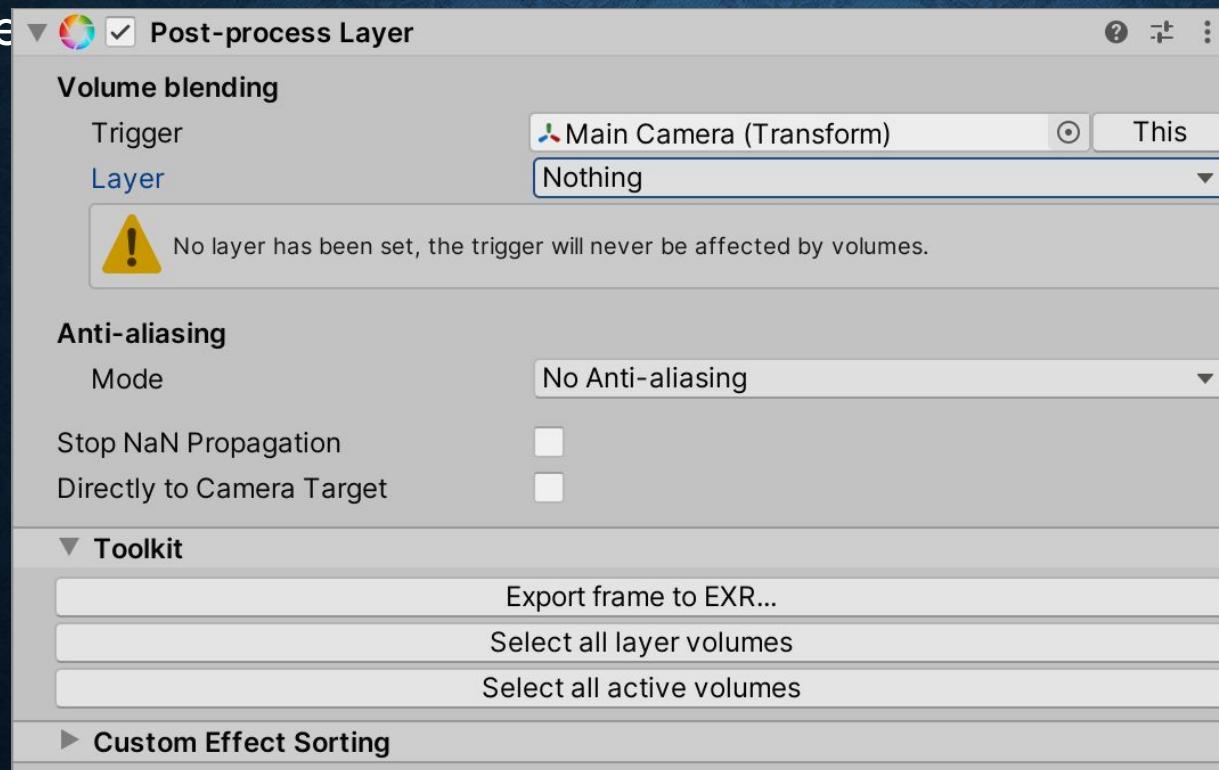
POST-PROCESSING STACK

- Install package “PostProcessing”
 - **Window > Package Manager > All packages > Post Processing**



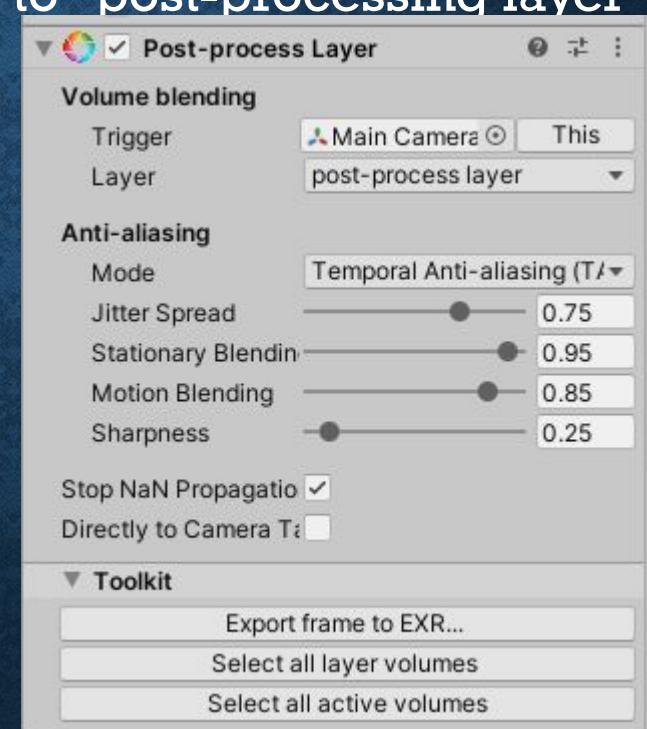
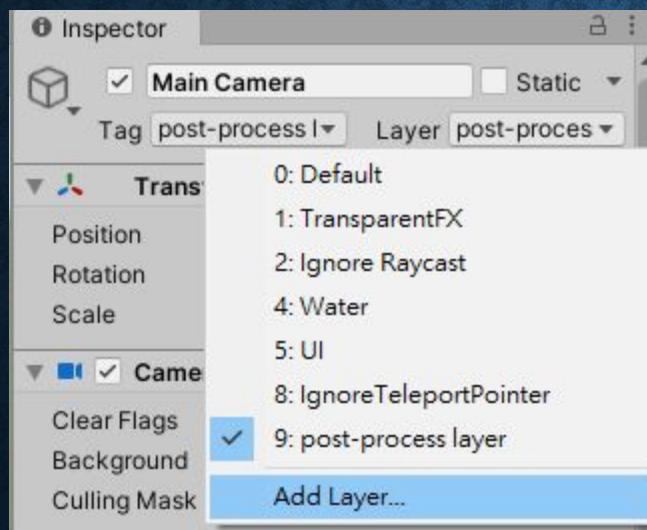
POST-PROCESSING STACK

- To enable post-processing in your scene, add the **Rendering > Post Process Layer** component to the Main Camera



POST-PROCESSING STACK

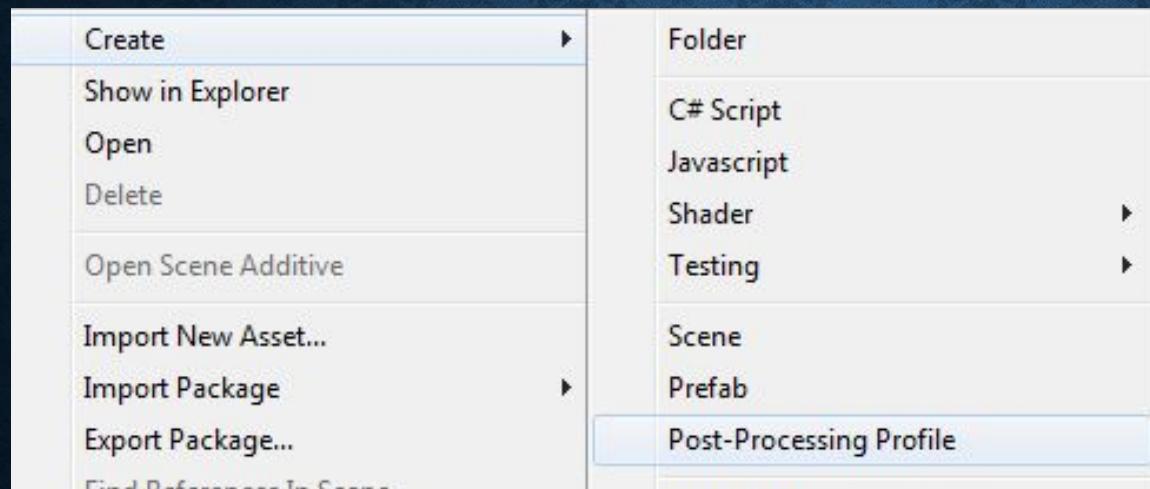
- In the Main Camera Object, create a new layer named “post-process layer”
- In the Post-process layer name Layer to “post-processing layer”



POST-PROCESSING STACK

- Right-click in your project window and select

Create > Post-Processing Profile.



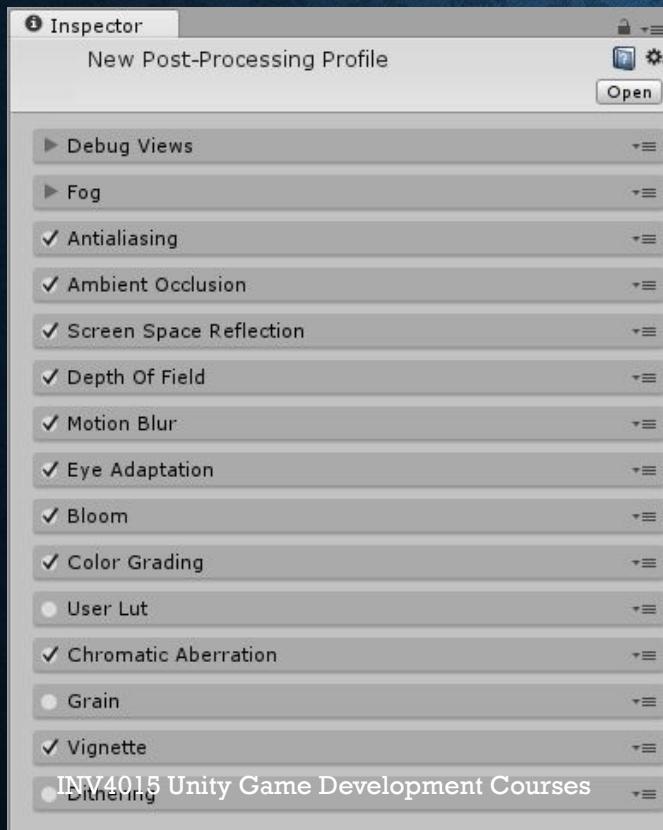
- Use the menu **Assets > Create > Post-Processing Profile.**
- This will create a new asset in your project.

POST-PROCESSING STACK

- Post-Processing Profiles are project assets and can be shared easily between scenes / cameras, as well as between different projects or on the Asset Store. This makes creating presets easier (ie. high quality preset for desktop or lower settings for mobile).

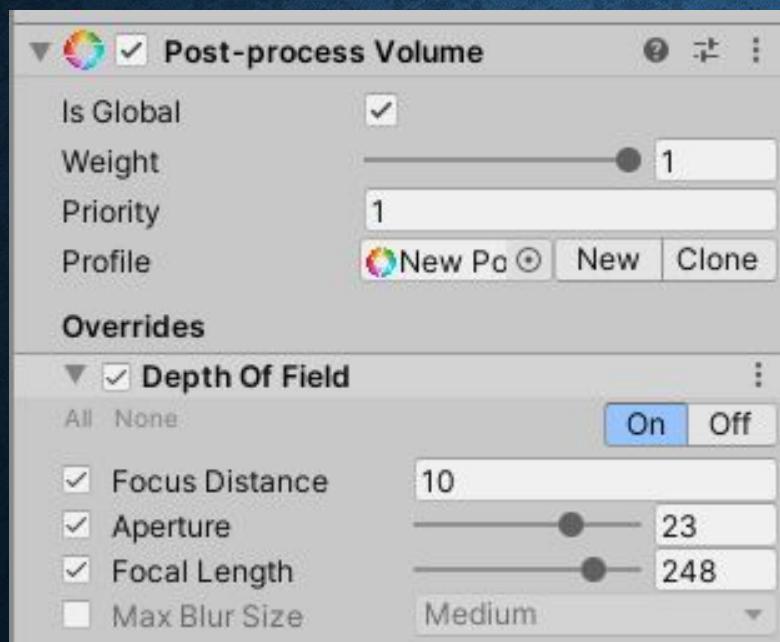
POST-PROCESSING STACK

- Selecting a profile will show the inspector window for editing the profile settings.



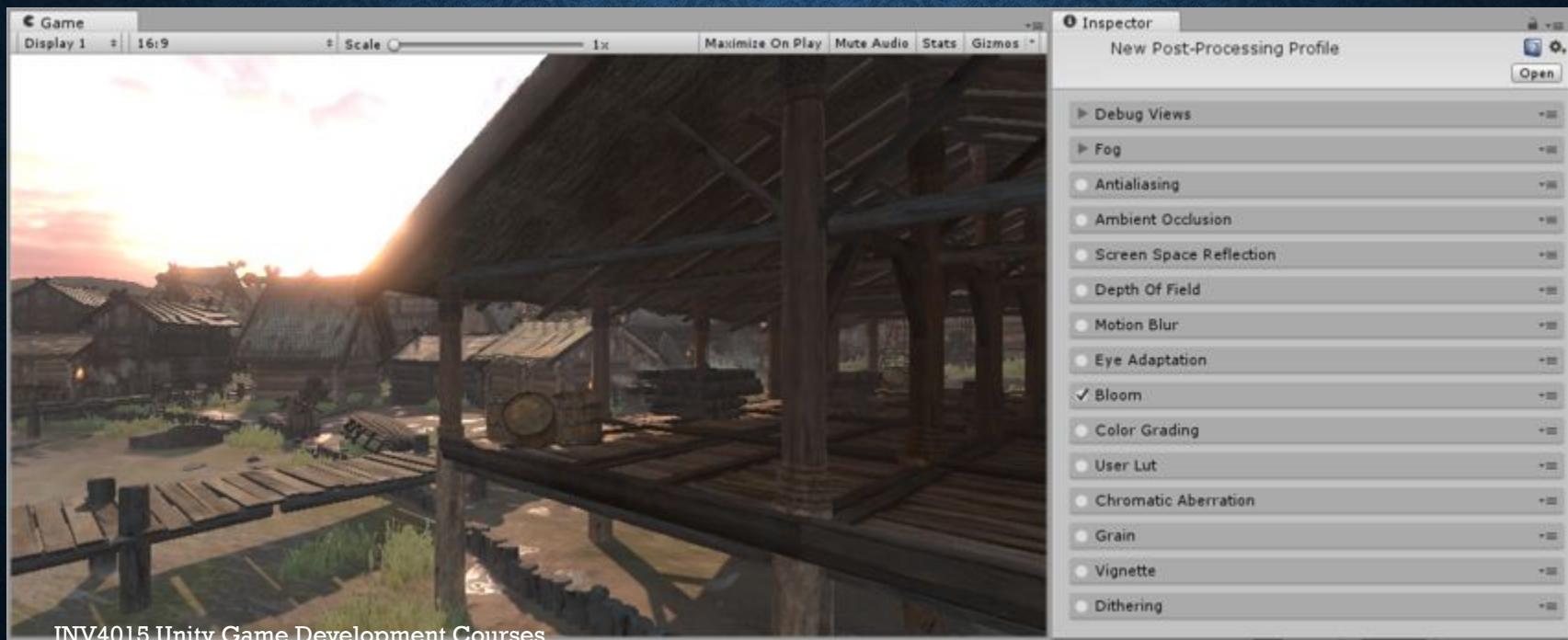
POST-PROCESSING STACK

- In the Main Camera, add new component “Post-process Volume”
- Enable “Is Global”, change profile to “New Post-processing Profile”



POST-PROCESSING STACK

- With the profile selected, you can use the checkbox on each effect in the inspector to enable or disable individual effects.



POST-PROCESSING STACK

- Post-processing stack provide the following features:

- Fog, Antialiasing, Ambient Occlusion
- Screen Space Reflection, Depth of Field
- Motion Blur, Eye Adaption, Bloom
- Color Grading, User Cut, Chromatic Aberration
- Grain, Vignette, Dithering

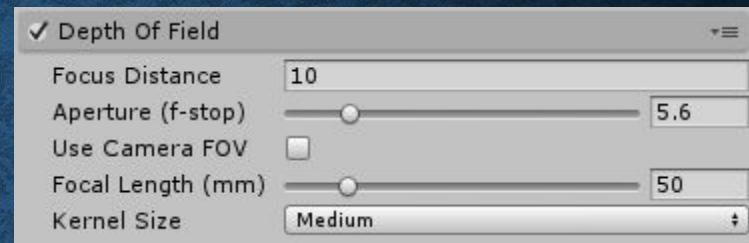
DEPTH OF FIELD

- Depth of Field is a common post-processing effect that simulates the focus properties of a camera lens.



DEPTH OF FIELD

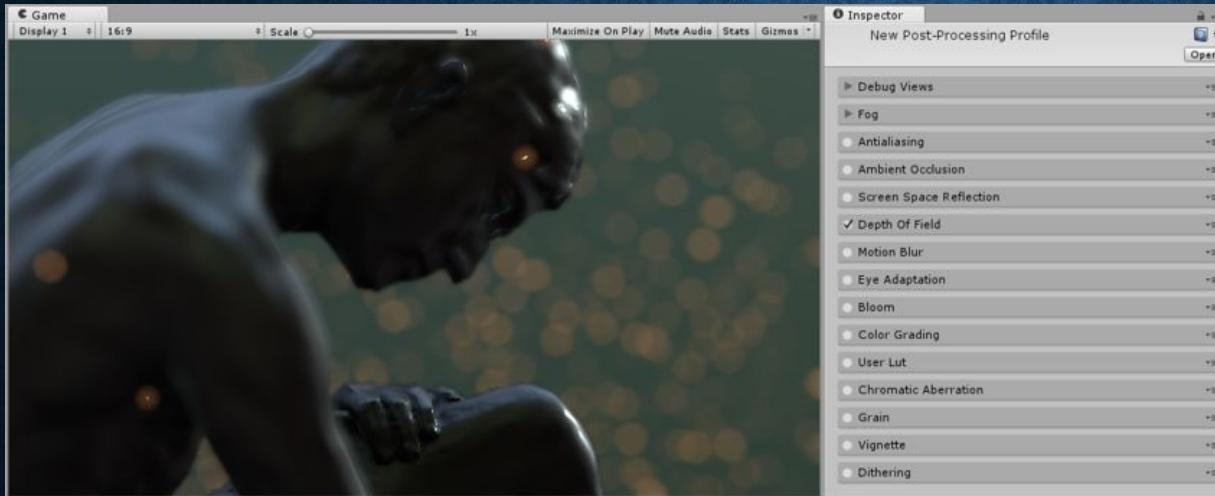
- Setting of Depth of Field :



Property:	Function:
Focus Distance	Distance to the point of focus.
Aperture	Ratio of the aperture (known as f-stop or f-number). The smaller the value is, the shallower the depth of field is.
Focal Length	Distance between the lens and the film. The larger the value is, the shallower the depth of field is.
Use Camera FOV	Calculate the focal length automatically from the field-of-view value set on the camera.
Kernel Size	Convolution kernel size of the bokeh filter, which determines the maximum radius of bokeh. It also affects the performance (the larger the kernel is, the longer the GPU time is).

DEPTH OF FIELD

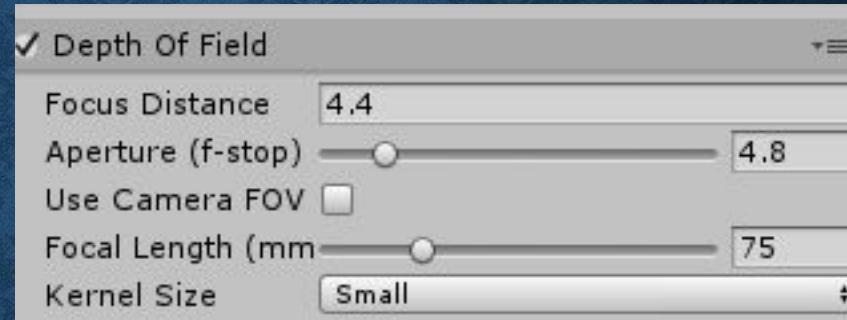
Example of Depth of Field effect :



DEPTH OF FIELD

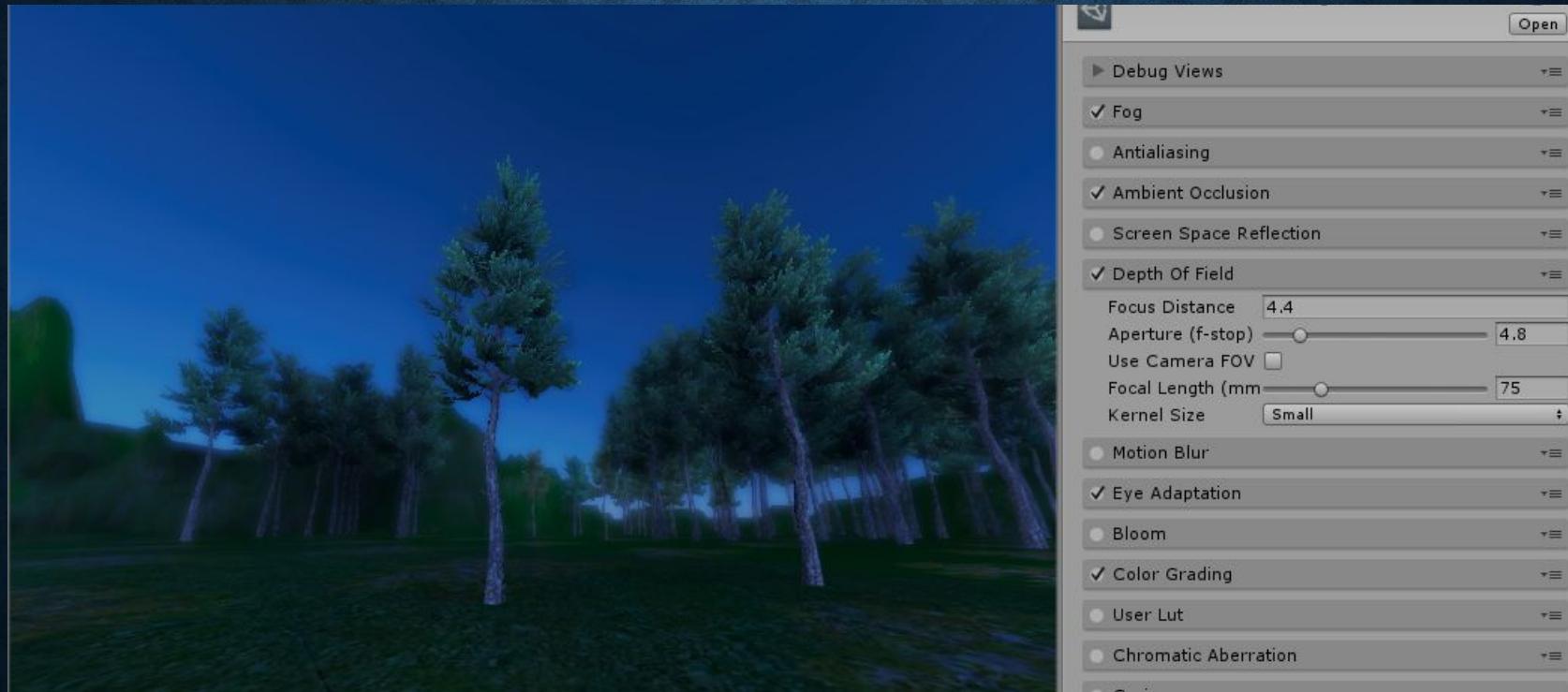
- Setup parameters

- Focus Length
 - 4.4
- Aperture (f-stop)
 - 4.8
- Focal Length
 - 75
- Use Camera FOV
 - No
- Kernel Size
 - Small



DEPTH OF FIELD

- This will produce the following screen.

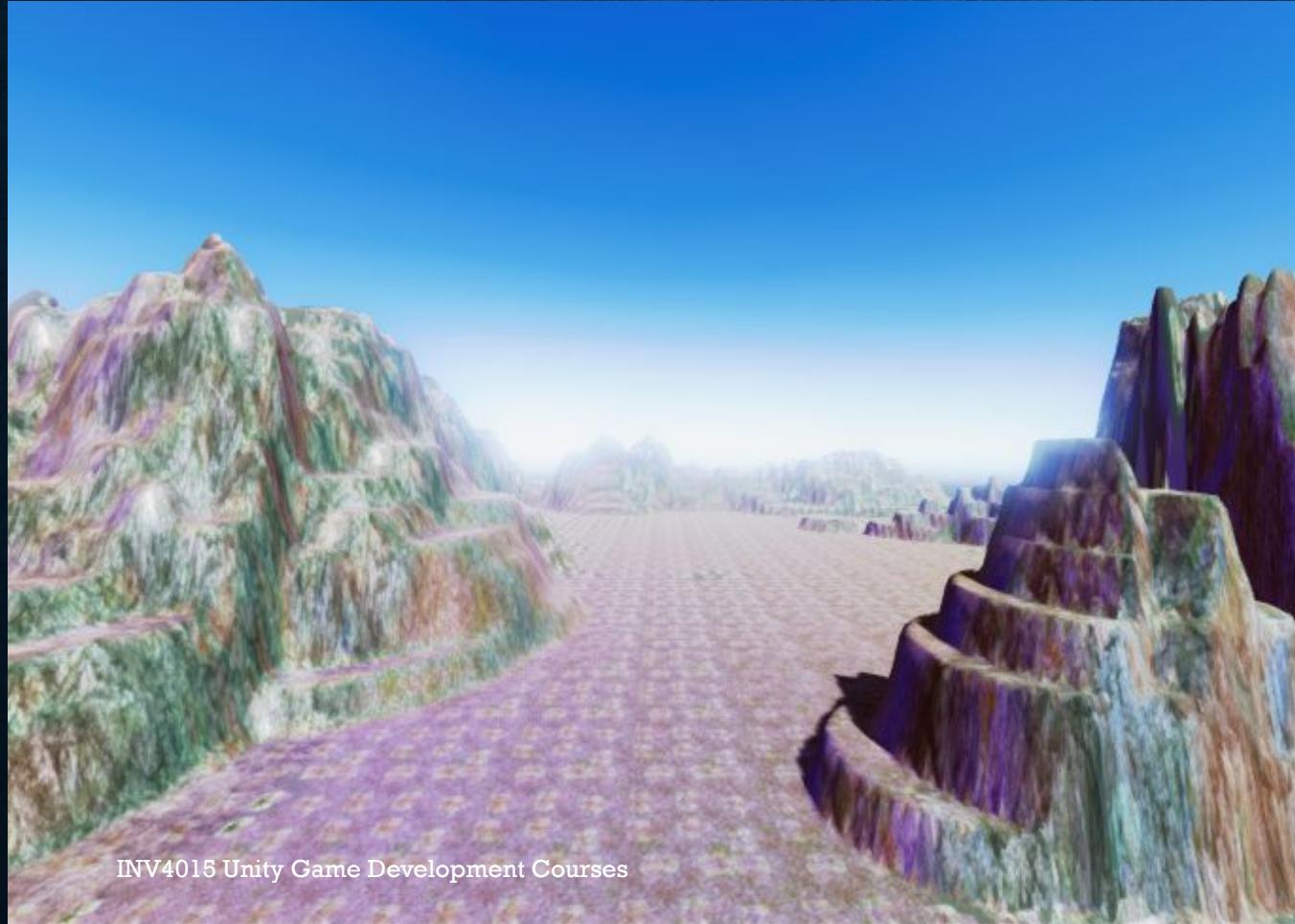


COLOR GRADING

- Color Grading is the process of altering or correcting the color and luminance of the final image. You can think of it like applying filters in software like Instagram.
- The Color Grading tools included in the post-processing stack are fully real-time HDR tools and internal processing is done in the ACES color-spaces.

COLOR GRADING

Example of Color Grading:



COLOR GRADING

- The Color Grading tools supplied in the post-processing stack come in five sections:

- Tonemapping
- Basic
- Channel Mixer
- Trackballs
- Grading Curves

- **Requirements**

- RGBAHalf Texture Format
- Shader model 3
- Reference on Graphics Emulation :
<https://docs.unity3d.com/Manual/GraphicsEmulation.html>

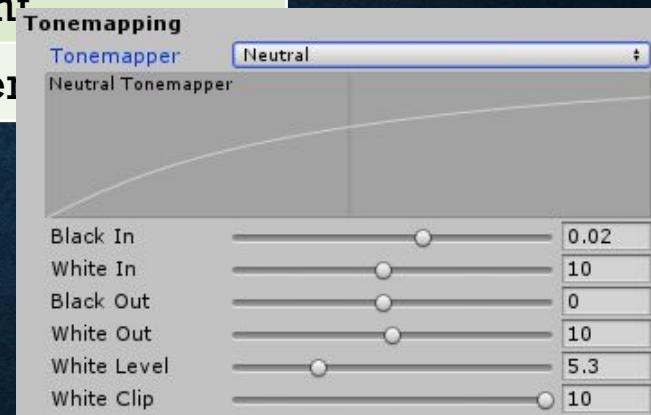
TONEMAPPING

- Tonemapping is the process of remapping HDR values of an image into a range suitable to be displayed on screen. Tonemapping should always be applied when using an HDR camera, otherwise values color intensities above 1 will be clamped at 1, altering the scenes luminance balance.
 - None (apply no tonemapping)
 - Neutral
 - Filmic (ACES)

NEUTRAL TONEMAPPER

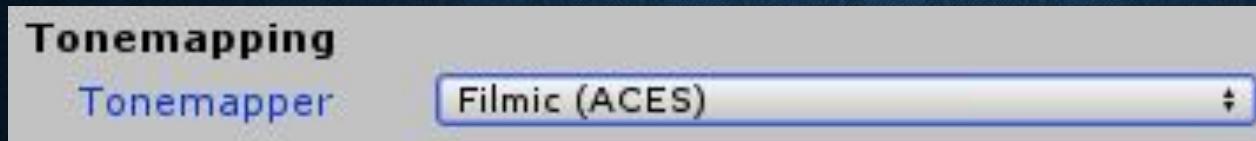
- Setting of Neutral Tonemapper:

Property:	Function:
Black In	Inner control point for the black point.
White In	Inner control point for the white point.
Black Out	Outer control point for the black point.
White Out	Outer control point for the white point.
White Level	Pre-curve white point adjustment
White Clip	Post-curve white point adjustment



FILMIC (ACES) TONEMAPPER

- The Filmic (ACES) tonemapper uses a close approximation of the reference ACES tonemapper for a more filmic look
- It is more contrasted than Neutral and has an effect on actual color hue & saturation. This tonemapper is the simplest to use as it requires no user input to give a standard filmic look to your scene



BASIC COLOR GRADING

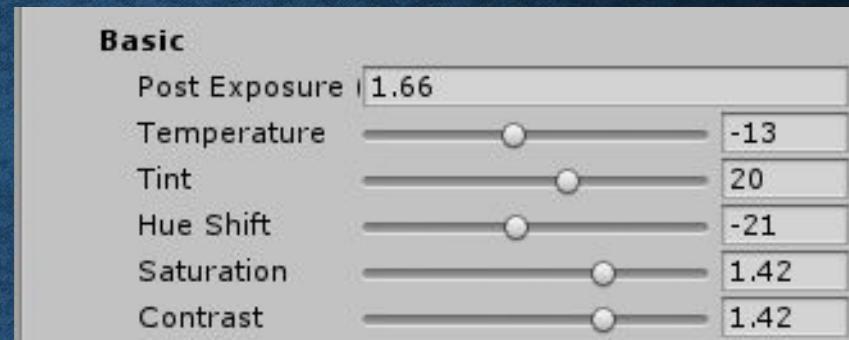
- Setting of Basic Color Grading:



Property:	Function:
Post Exposure	Adjusts the overall exposure of the scene in EV units. This is applied after HDR effect and right before tonemapping so it won't affect previous effects in the chain.
Temperature	Sets the white balance to a custom color temperature.
Tint	Sets the white balance to compensate for a green or magenta tint.
Hue Shift	Shift the hue of all colors.
Saturation	Pushes the intensity of all colors.
Contrast	Expands or shrinks the overall range of tonal values.

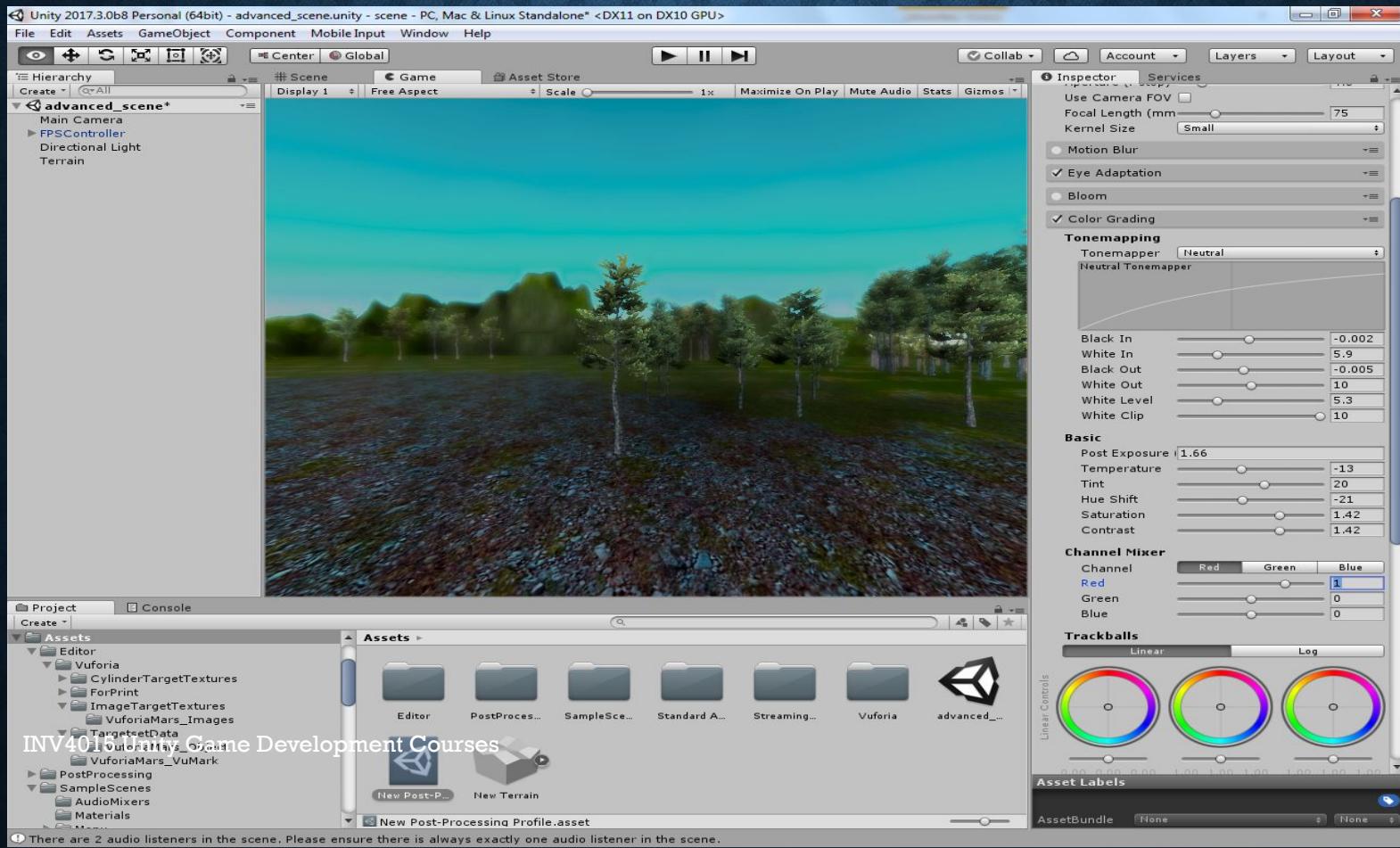
BASIC COLOR GRADING

- Setup parameters
 - Post Exposure
 - 1.66
 - Temperature
 - -13
 - Tint
 - 20
 - Hue Shift
 - -21
 - Saturation
 - 1.42
 - Contrast
 - 1.42



COLOR GRADING

- This will produce the following screen.



CHANNEL MIXER COLOR GRADING

- Setting of Channel Mixer Color Grading



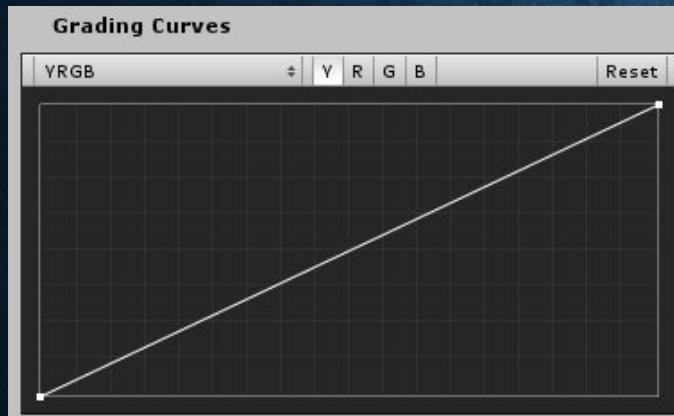
Property:	Function:
Channel	Select the output channel to modify
Red	Modify the influence of the red channel within the overall mix
Green	Modify the influence of the green channel within the overall mix
Blue	Modify the influence of the blue channel within the overall mix

GRADING CURVES COLOR GRADING

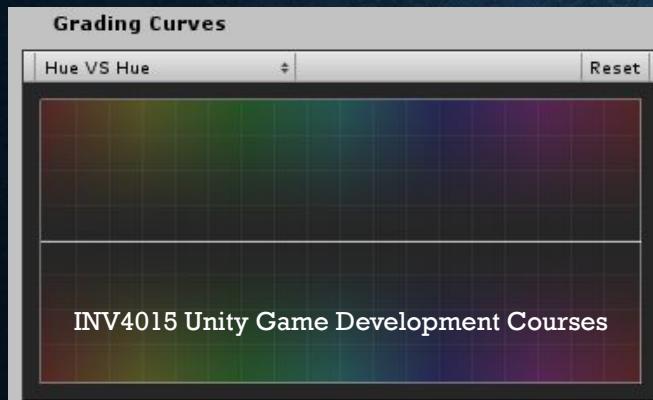
- Grading Curves (also known as ~~Curves~~ Grading Curves) are an advanced way to adjust specific ranges in hue, saturation or luminosity in your image.
- You can achieve the effects of specific hue replacement, desaturating certain luminosities.
- Five Grading Curve types are supplied in the post-processing stack:
 - YRGB
 - Hue vs Hue
 - Hue vs Sat
 - Sat vs Sat
 - Lum vs Sat

GRADING CURVES COLOR GRADING

- YRGB Curve:

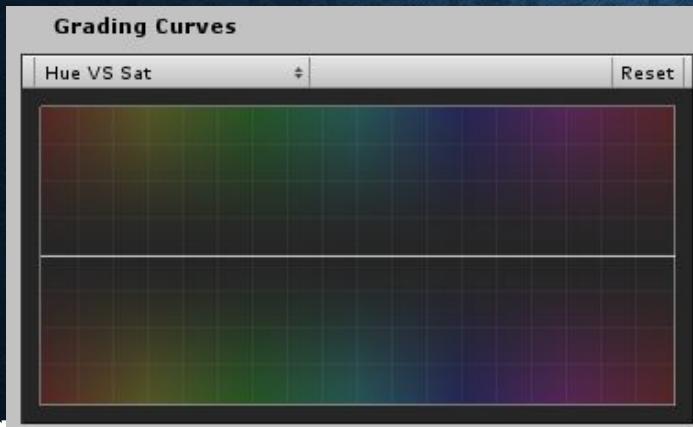


- Hue vs Hue Curve

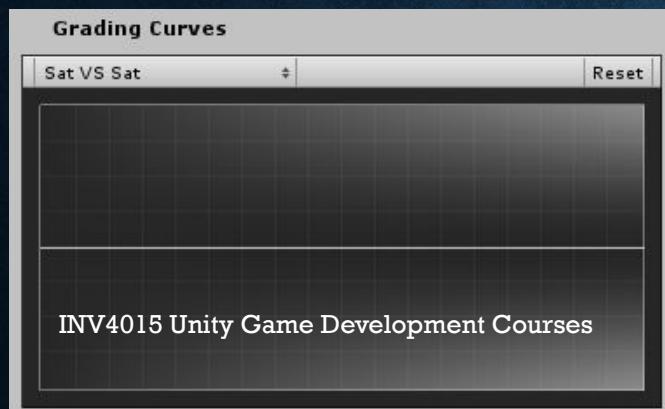


GRADING CURVES COLOR GRADING

- Hue vs Sat Curve:

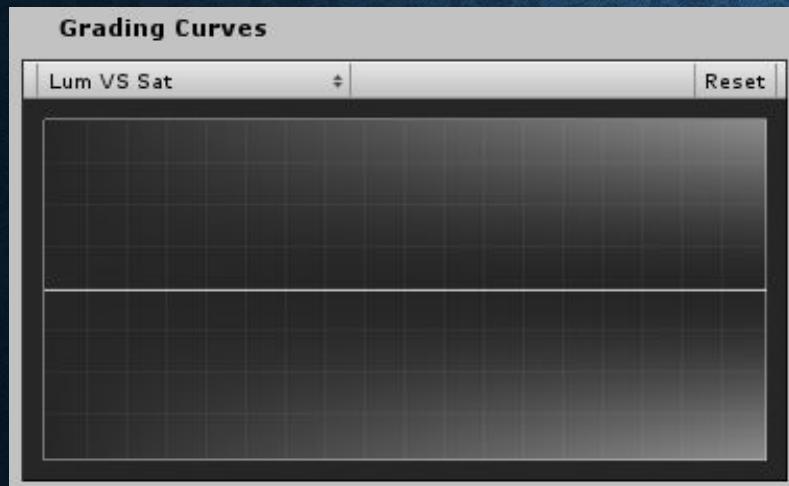


- Sat vs Sat Curve



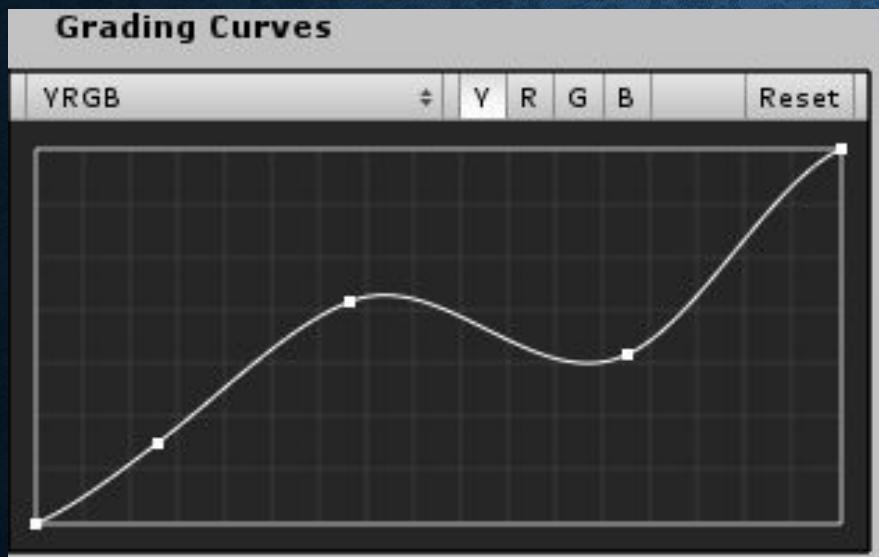
GRADING CURVES COLOR GRADING

- Lum vs Sat Curve :



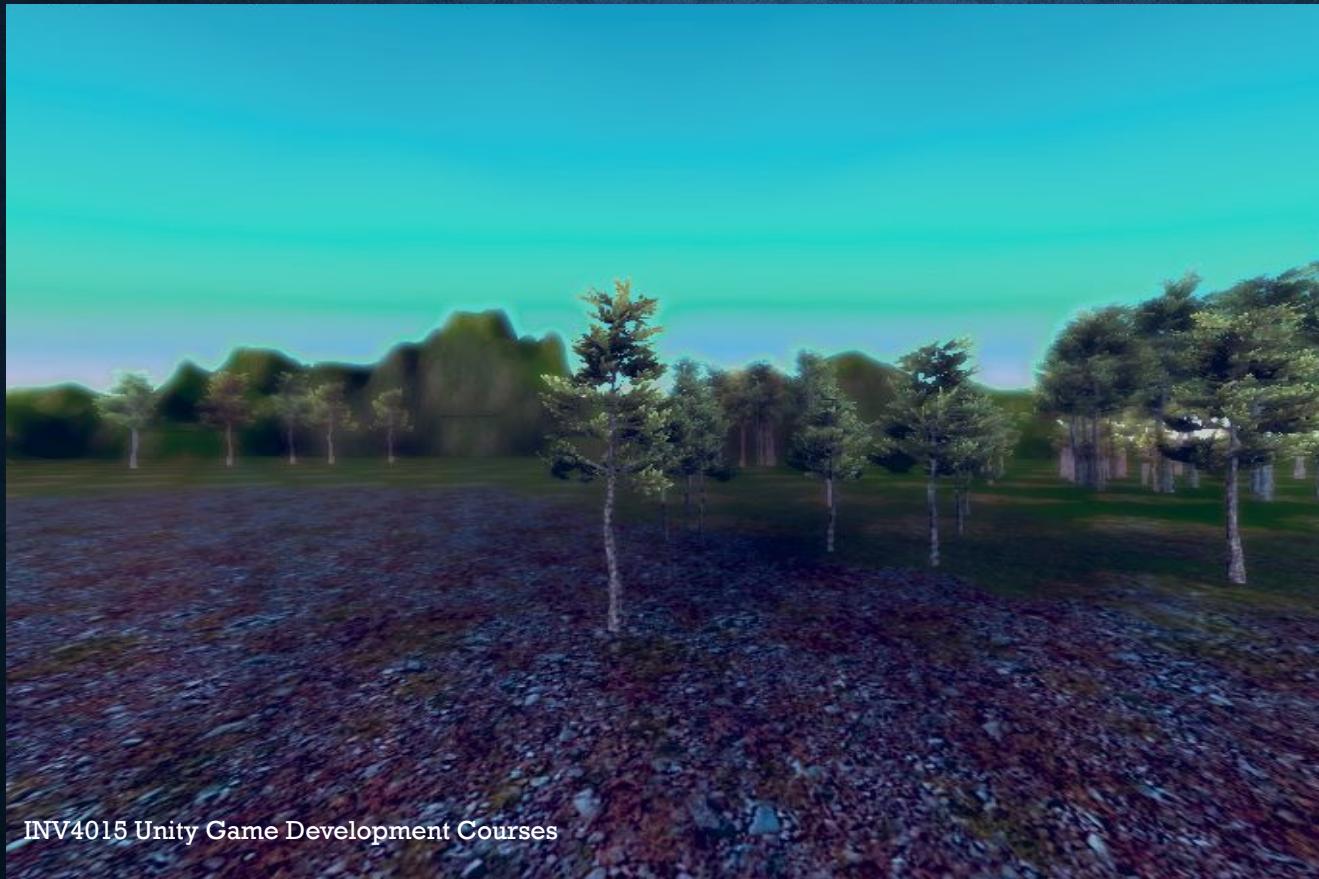
GRADING CURVES COLOR GRADING

- Setup parameters



GRADING CURVES COLOR GRADING

- This will produce the following screen.



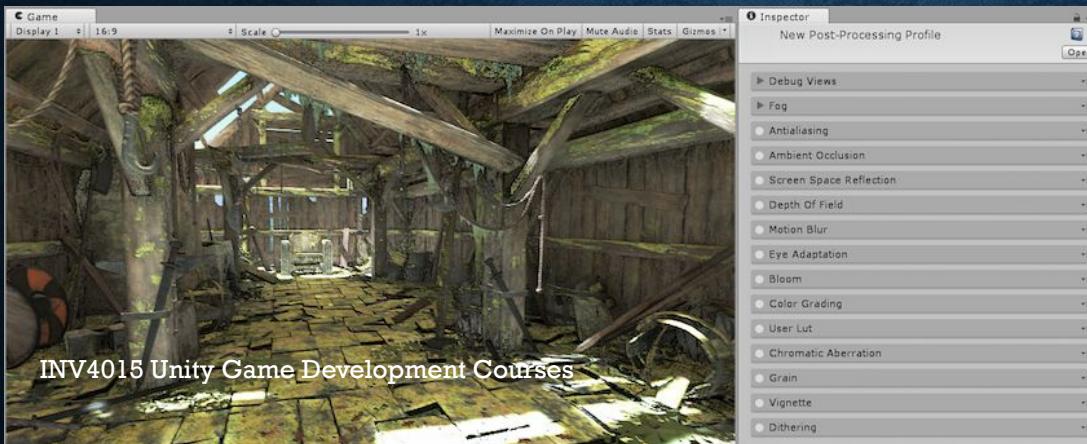
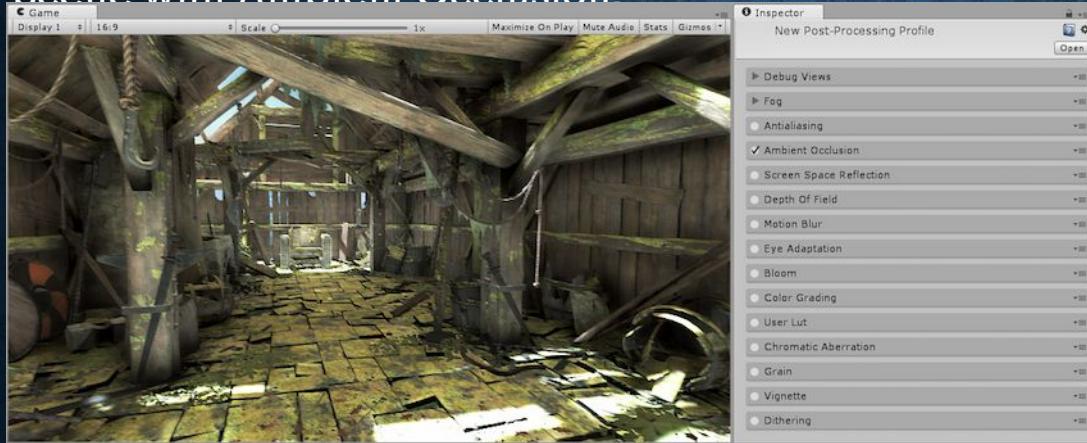
AMBIENT OCCLUSION

- The Ambient Occlusion post-processing effect approximates Ambient Occlusion in real time as a full-screen post-processing effect.
- It darkens creases, holes, intersections and surfaces that are close to each other. In real life, such areas tend to block out or occlude ambient light, hence they appear darker.
- **Requirements**
 - Depth & Normals texture
 - Shader model 3

AMBIENT OCCLUSION

Example of Ambient Occlusion:

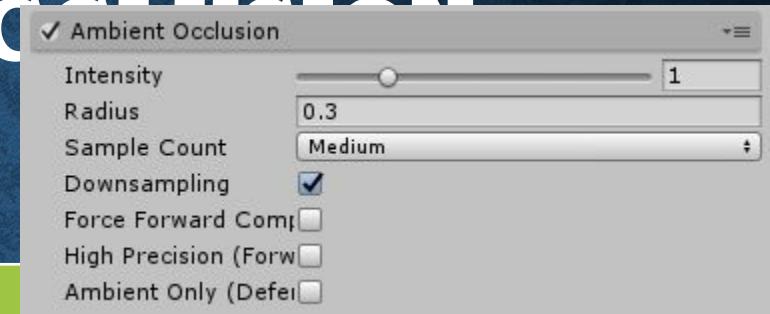
- Scene with Ambient Occlusion



INV4015 Unity Game Development Courses

AMBIENT OCCLUSION

- Setting of Ambient Occlusion:

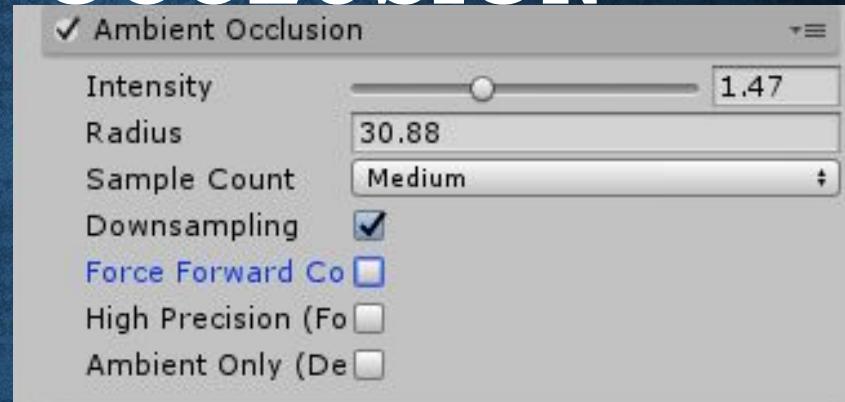


Property:	Function:
Intensity	Degree of darkness produced by the effect.
Radius	Radius of sample points, which affects extent of darkened areas.
Sample Count	Number of sample points, which affects quality and performance.
Downsampling	Halves the resolution of the effect to increase performance at the cost of visual quality.
Force Forward Compatibility	Forces compatibility with Forward rendered objects when working with the Deferred rendering path.
High Precision (Forward)	Toggles the use of a higher precision depth texture with the forward rendering path (may impact performances). Has no effect with the deferred rendering path.
Ambient Only	Enables the ambient-only mode in that the effect only affects ambient lighting. This mode is only available with the Deferred rendering path and HDR rendering.

AMBIENT OCCLUSION

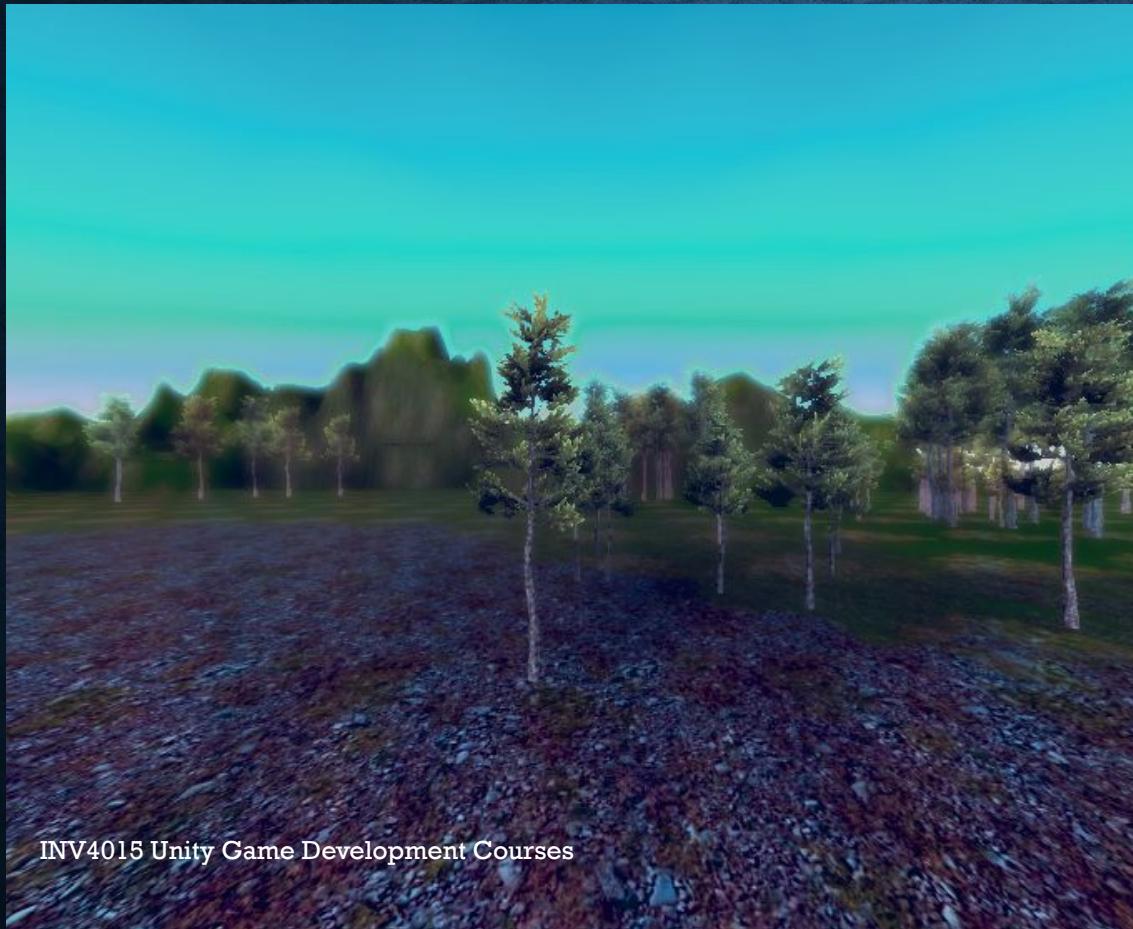
- Setup parameters

- Intensity
 - **1.47**
- Raduius
 - **30.88**
- Sample Count
 - **Medium**
- Downsampling
 - **Yes**
- Force Forward Compatibility
 - **null**
- High Precision (Forward)
 - **null**
- Ambient Only
 - **null**



AMBIENT OCCLUSION

- This will produce the following screen.



LABS EXCERCISE

- Conducting Post-Processing Stack
- Document : [Unit2_postprocessing_lab.docx](#)

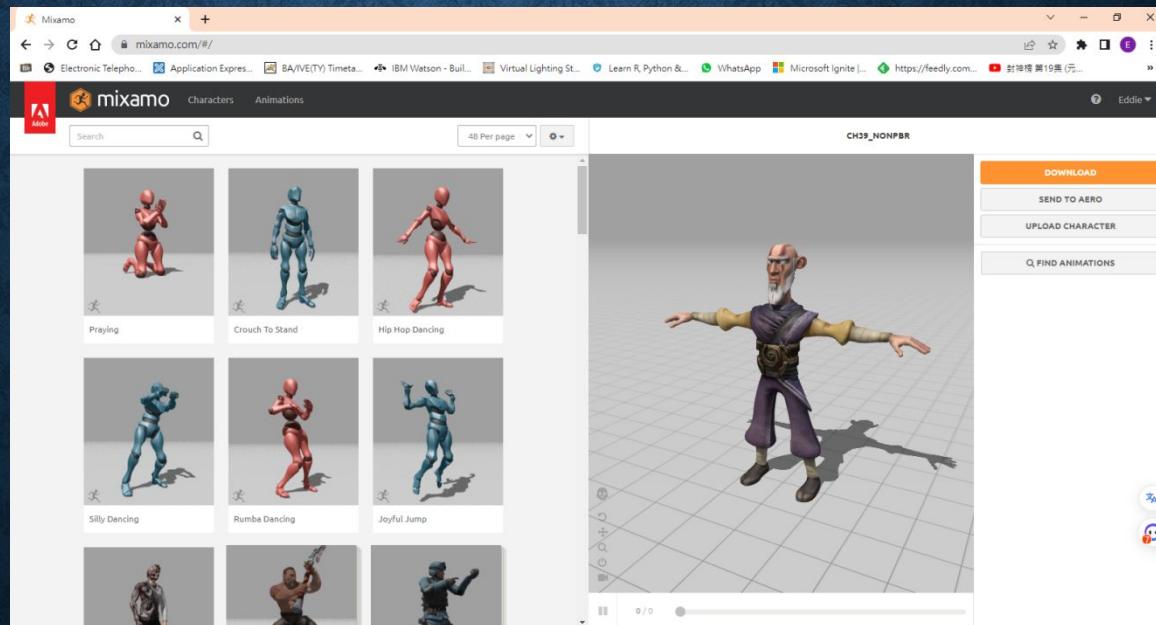
CUSTOMIZED CHARACTER

- Free 3D character models can be found
 - Mixamo
 - Customize character by Adobe Fuse
 - <http://www.mixamo.com>
 - Unity Assets Store
 - <https://www.assetstore.unity3d.com/en/>
 - TF3DM
 - <http://tf3dm.com/>



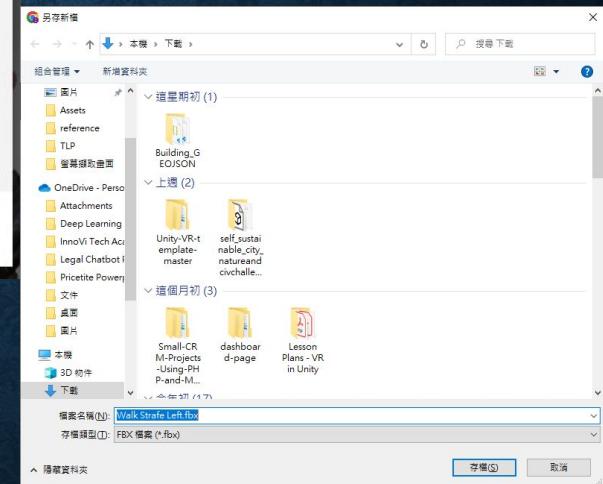
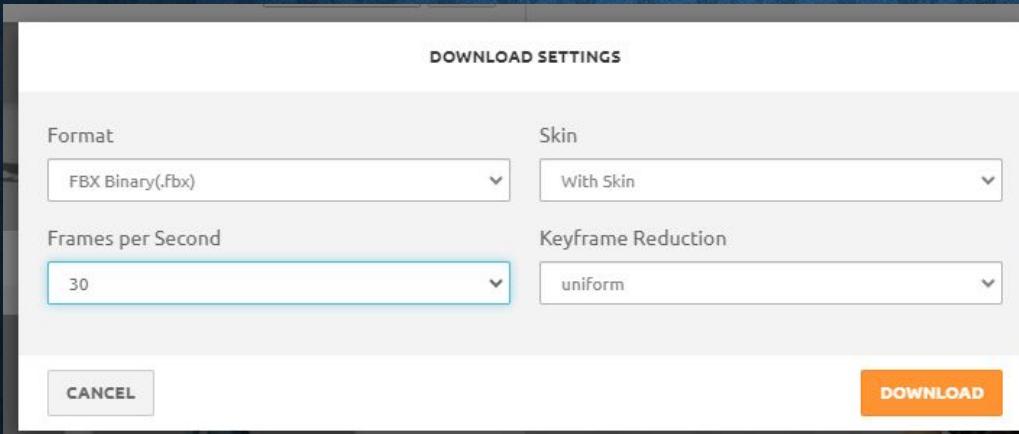
CUSTOMIZED CHARACTER

Go to Mixamo (<http://www.mixamo.com>) ->login using your gmail account



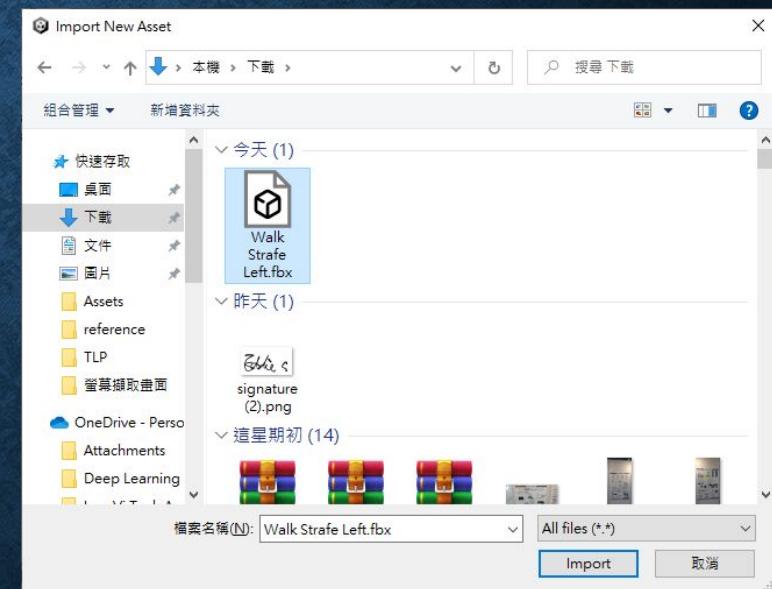
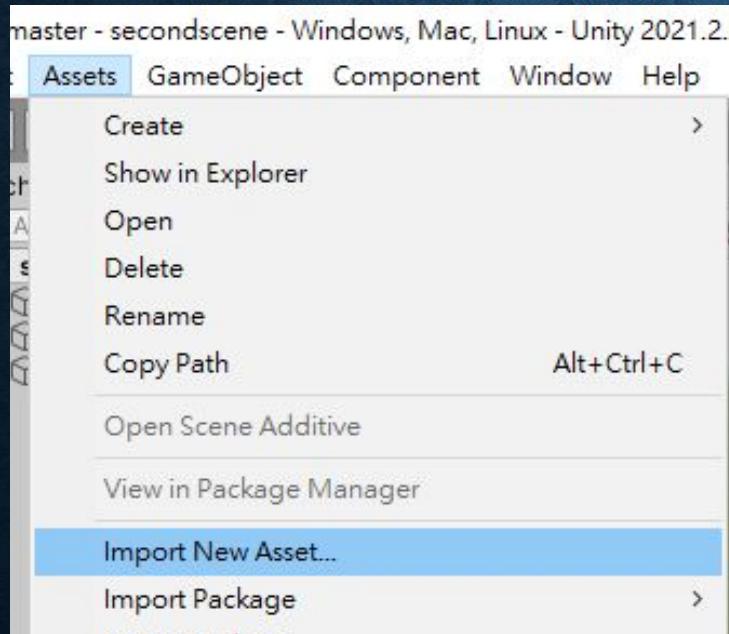
CUSTOMIZED CHARACTER

Click Download with the following details



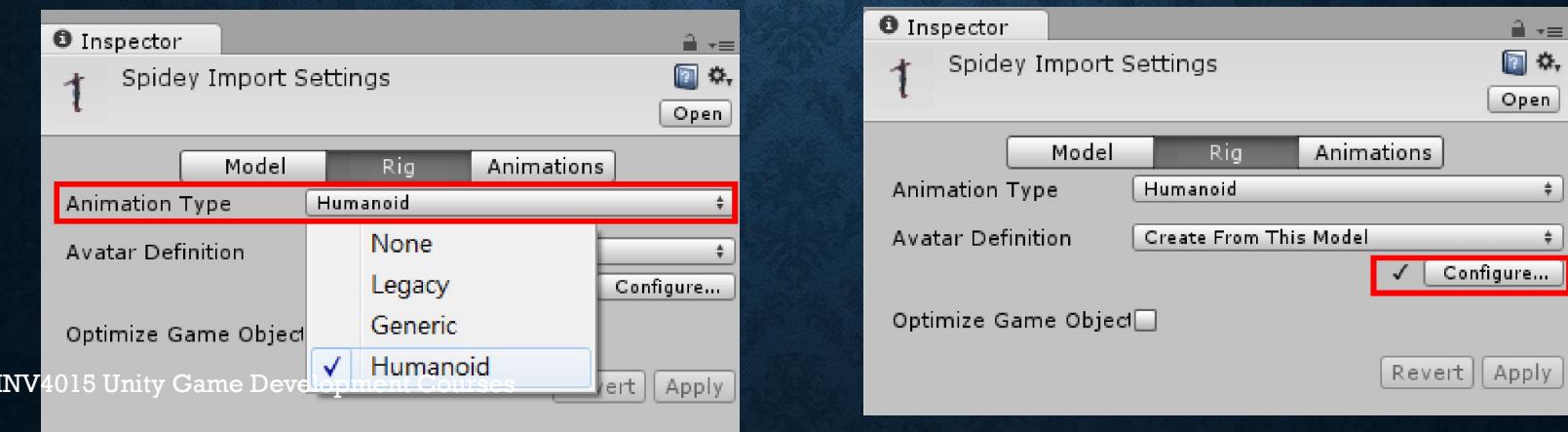
IMPORT CHARACTER TO UNITY

Click Assets -> Import New Asset



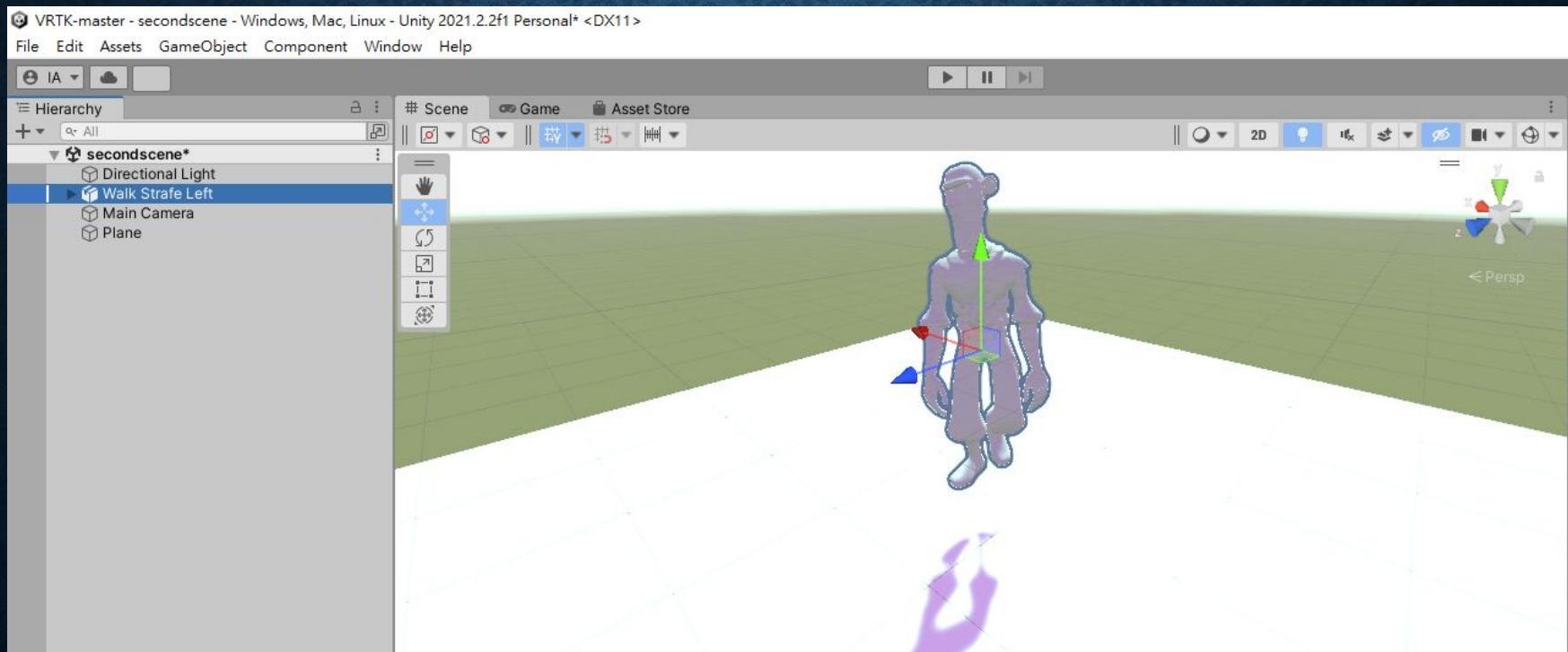
CUSTOMIZED CHARACTER

- The imported character has to use the “Humanoid” animation type for the Mecanim system
- A “tick” will be show if the character have the correct skeleton structure
- Otherwise have to be configured manually



CUSTOMIZED CHARACTER

- Drag the character model into unity

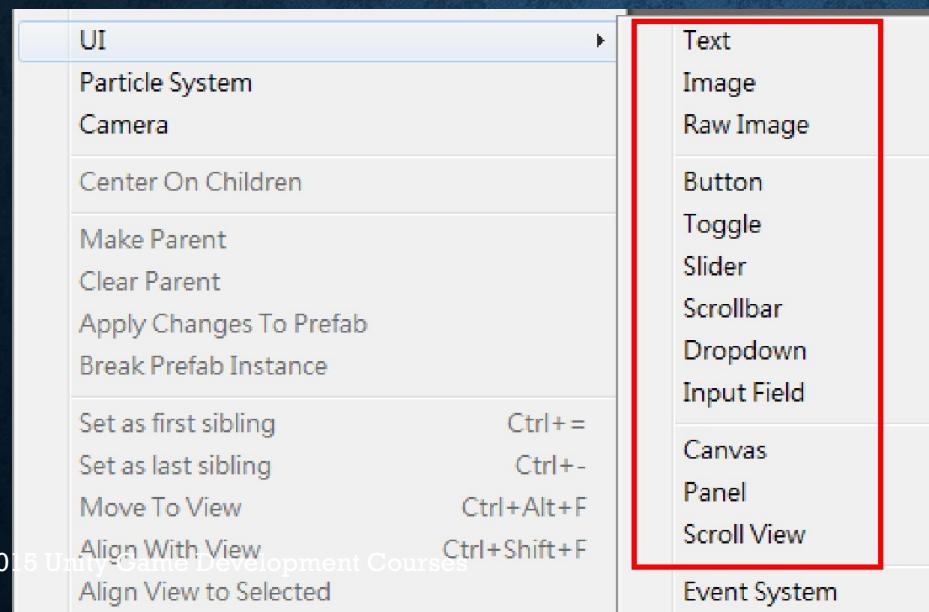


TOPICS

- User Interface
- Button and Image
- Slider Control
- Pause Menu

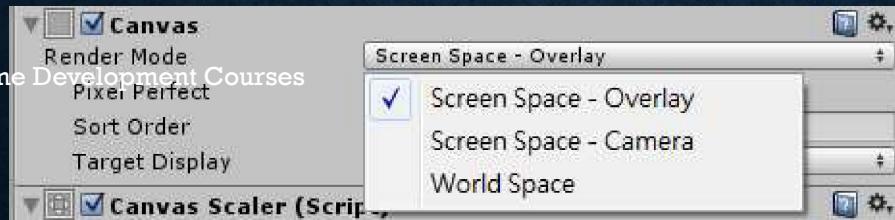
USER INTERFACE

- GUI element in UGUI system
- □ Can be created from
- – GameObject □ UI



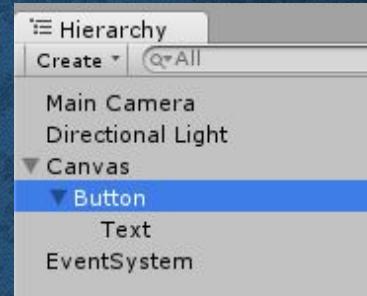
CANVAS

- Three Canvas Render Model
 - – Screen Space – Overlay
 - Render on top of the scene
 - – Screen Space – Camera
 - Placed given distance in front of the specified camera
 - – World Space
 - Behave as any other object in the scene



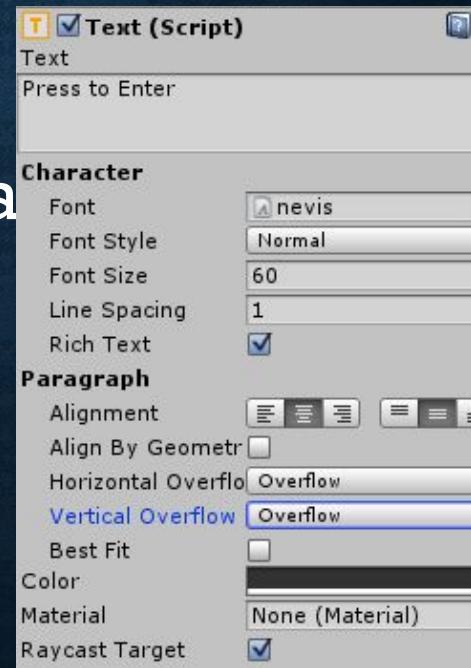
PANEL

- Create a GUI Button to display the button in 2D level
 - GameObject > UI -> Button



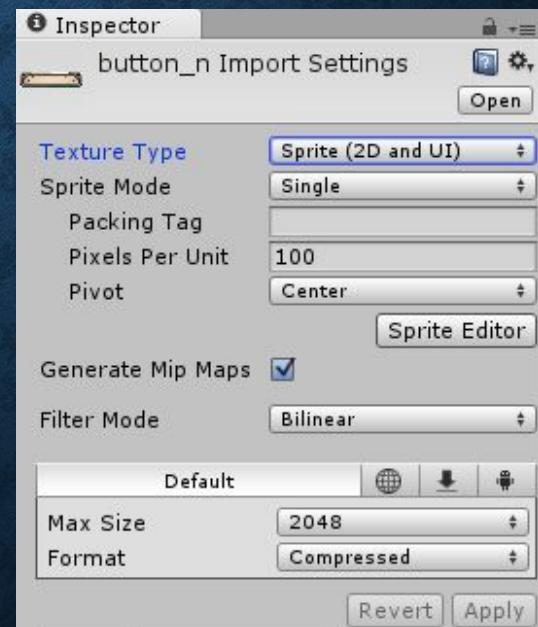
- Import the font “nevis”
- Change the Text value, Alignment and Overflow configuration on the text

Overflow configuration on the text



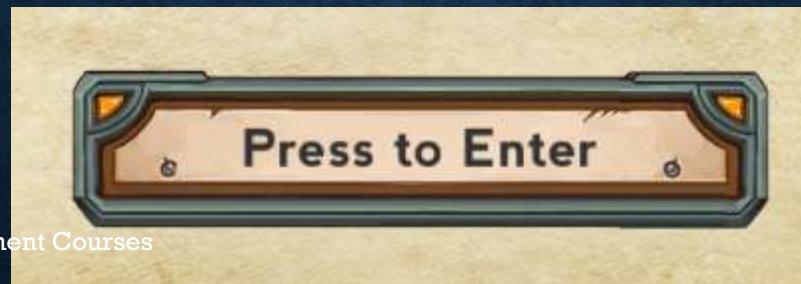
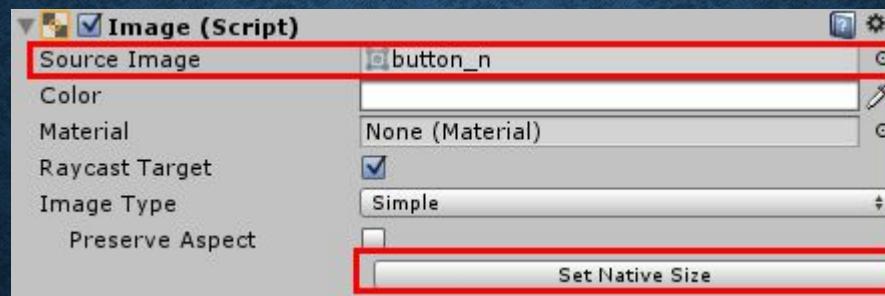
PANEL

- Panel is used as the background or menu backdrop
- By default it will occupied the full area of the canvas
- Every UI graphics must use the texture type “Sprite”
- Change the source image “button_n.png” texture type to “Sprite (2D and UI)



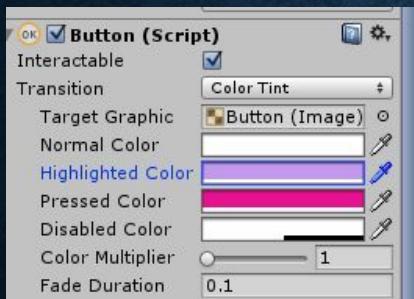
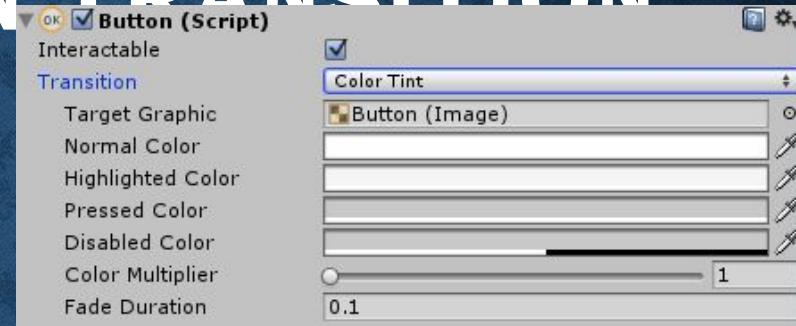
BUTTON

- Button always come with the text as the child
- Change the source image to your button image
- Use “Set Native Size” to change the width and height to the resolution of the button image



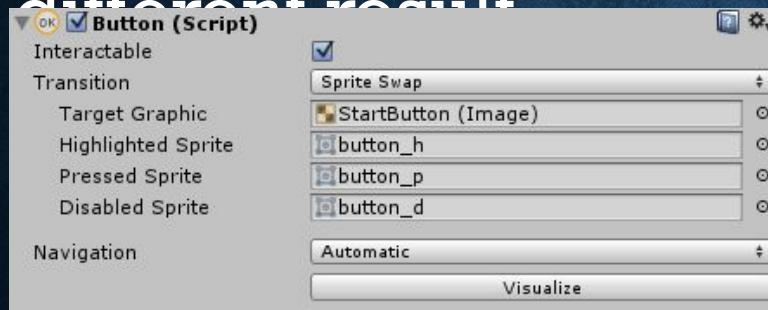
BUTTON TRANSITION

- Color Tint
 - Tint button color
- Change the colorTint “Highlighted color”, “Pressed Color” or “Disabled Color” and see different result.



BUTTON TRANSITION

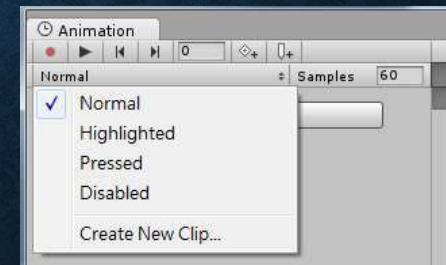
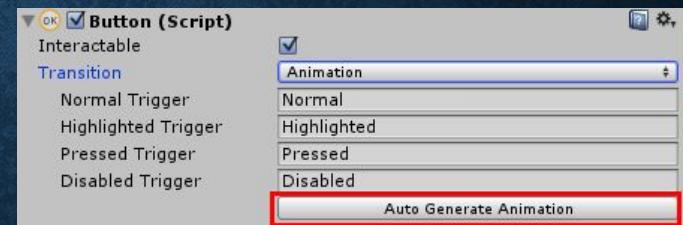
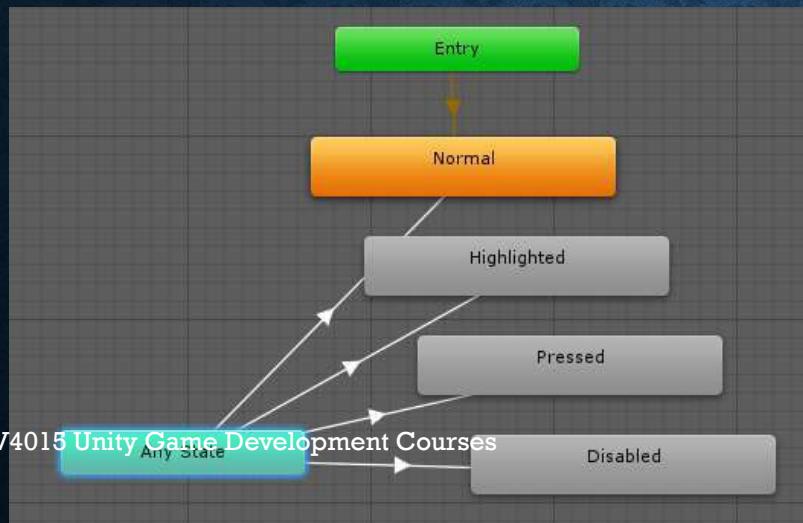
- Sprite Swap
 - Swap different button images
- Change the sprite “Highlighted Sprite”, “Pressed Sprite” or “Disabled Sprite” and see different result



BUTTON TRANSITION

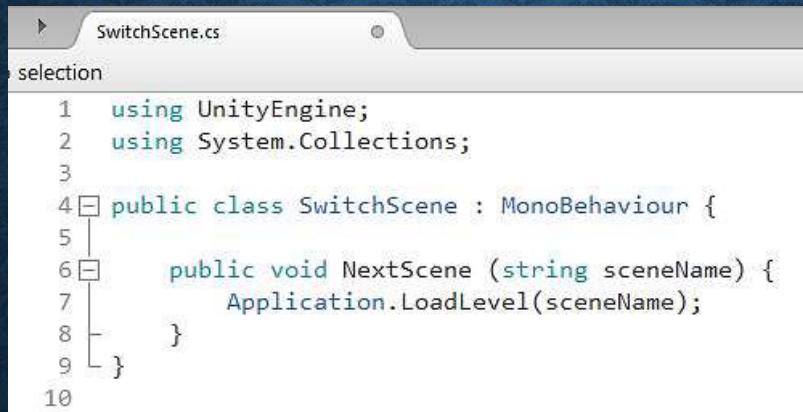
- Animation

- Play any animation
- Press “Auto Generate Animation” to create animator and animator controller automatically
- The animation name is called “Button”
- Create the animations for the 4 button state



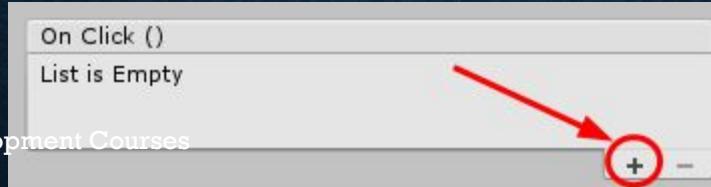
BUTTON EVENT

- Create the following script for switching scene
- Function must be set as public



```
SwitchScene.cs
selection
1  using UnityEngine;
2  using System.Collections;
3
4  public class SwitchScene : MonoBehaviour {
5
6      public void NextScene (string sceneName) {
7          Application.LoadLevel(sceneName);
8      }
9
10 }
```

- Select the button, from the button script add the On Click() event

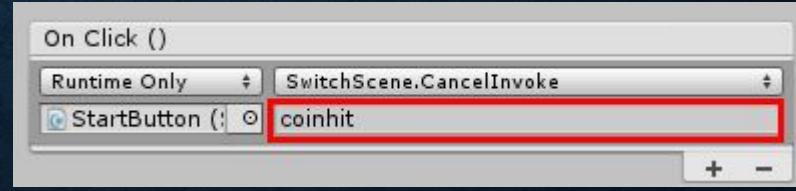
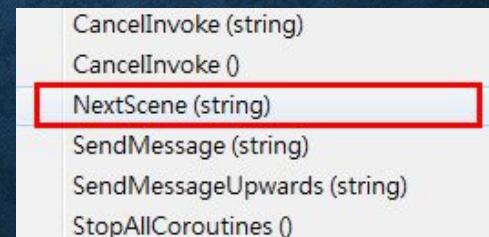
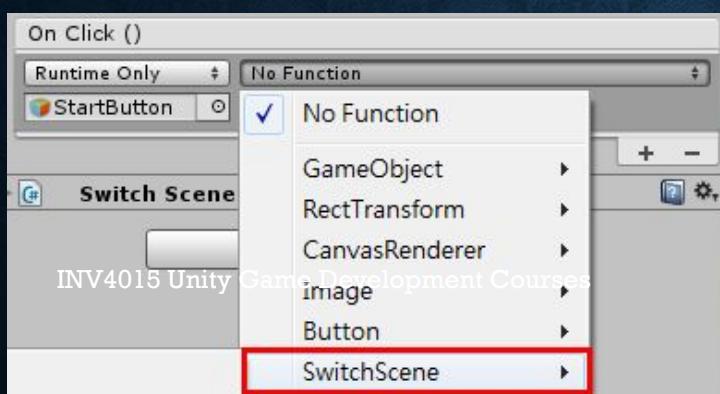


BUTTON EVENT

- Put the script to the button itself
- Drag the button to the “None(Object)”

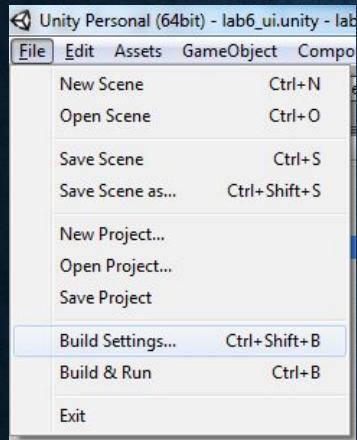


- Select the SwitchScene script and then the NextScene(string) function, type in the scene name parameter

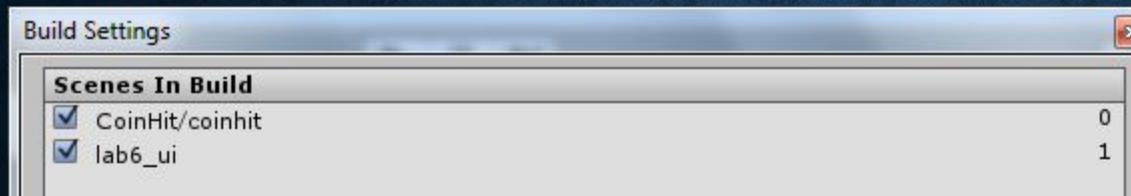


BUTTON EVENT

- Click File -> Build Setting



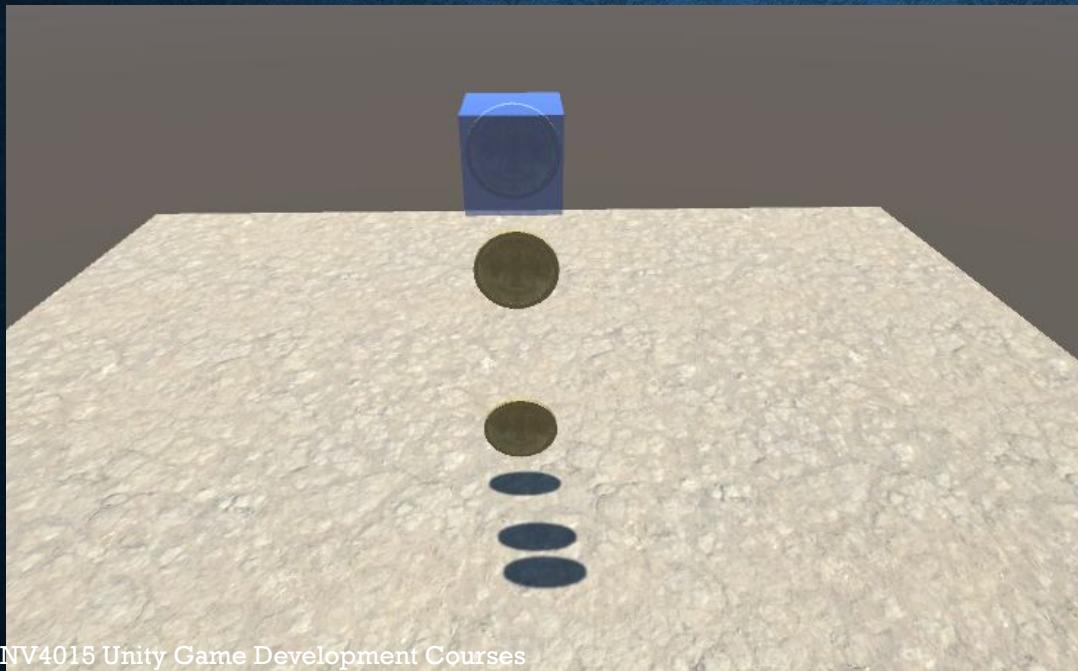
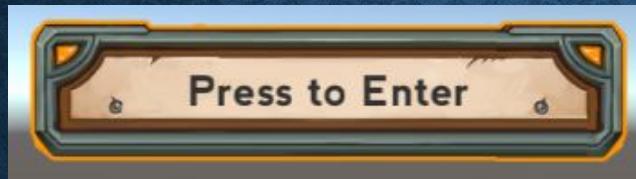
- Drag the scene "CoinHit" into the scenes



- Test it

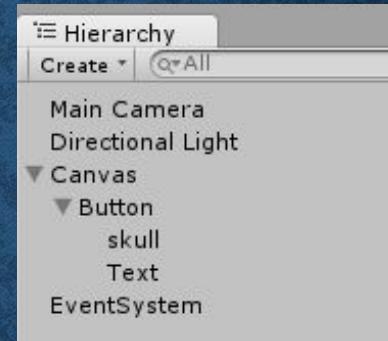
BUTTON EVENT

- When you click the button, the CoinHit scene will be shown.

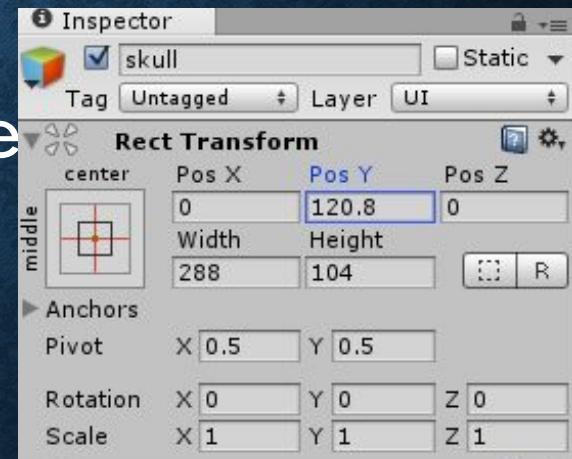


IMAGE

- Create a GUI Button to display the button in 2D level
 - GameObject > UI -> Image



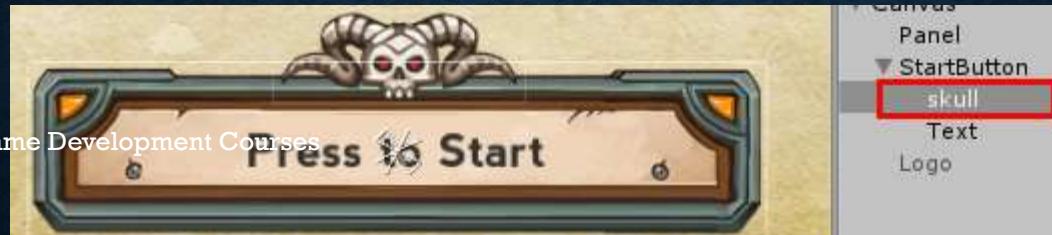
- Change the name to “skull”
- Change the Image coordinate to “120.8”



IMAGE

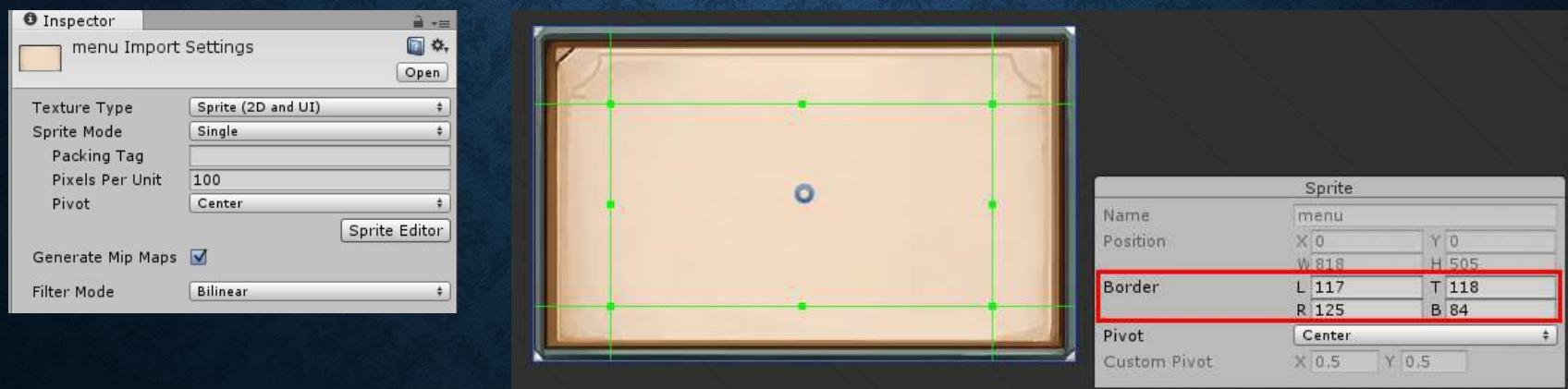
- Create an image “skull” for the button decoration

- Use native size
- The order of the hierarchy will affect the sorting of the display



SLICED IMAGE

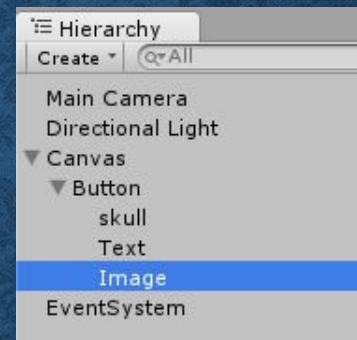
- Sprite can be sliced for creating extensible menu background
- Select the “menu.png”
- From the import settings, enter border size



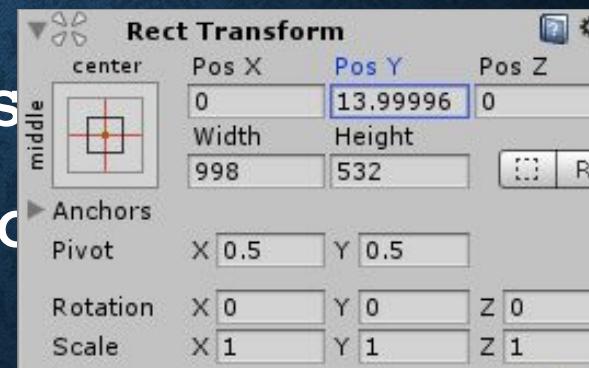
SLICED IMAGE

- Create a GUI Button to display the button in 2D level

- GameObject > UI -> Image

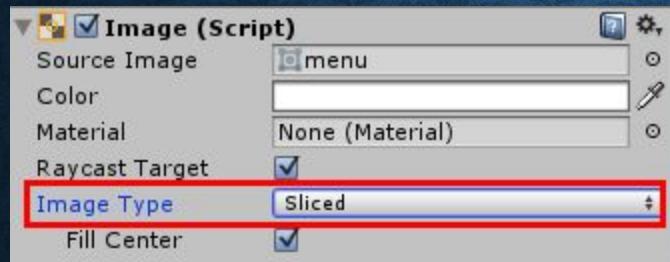


- Change the name to “menu”
- Change the Image coordinates to “13.99996”, Width “998” and Height “532”

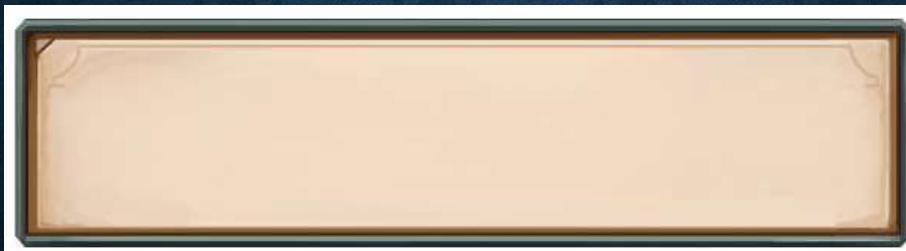


SLICED IMAGE

- Create the image using the sliced image
- Change the image type to sliced



- Adjust the width and height

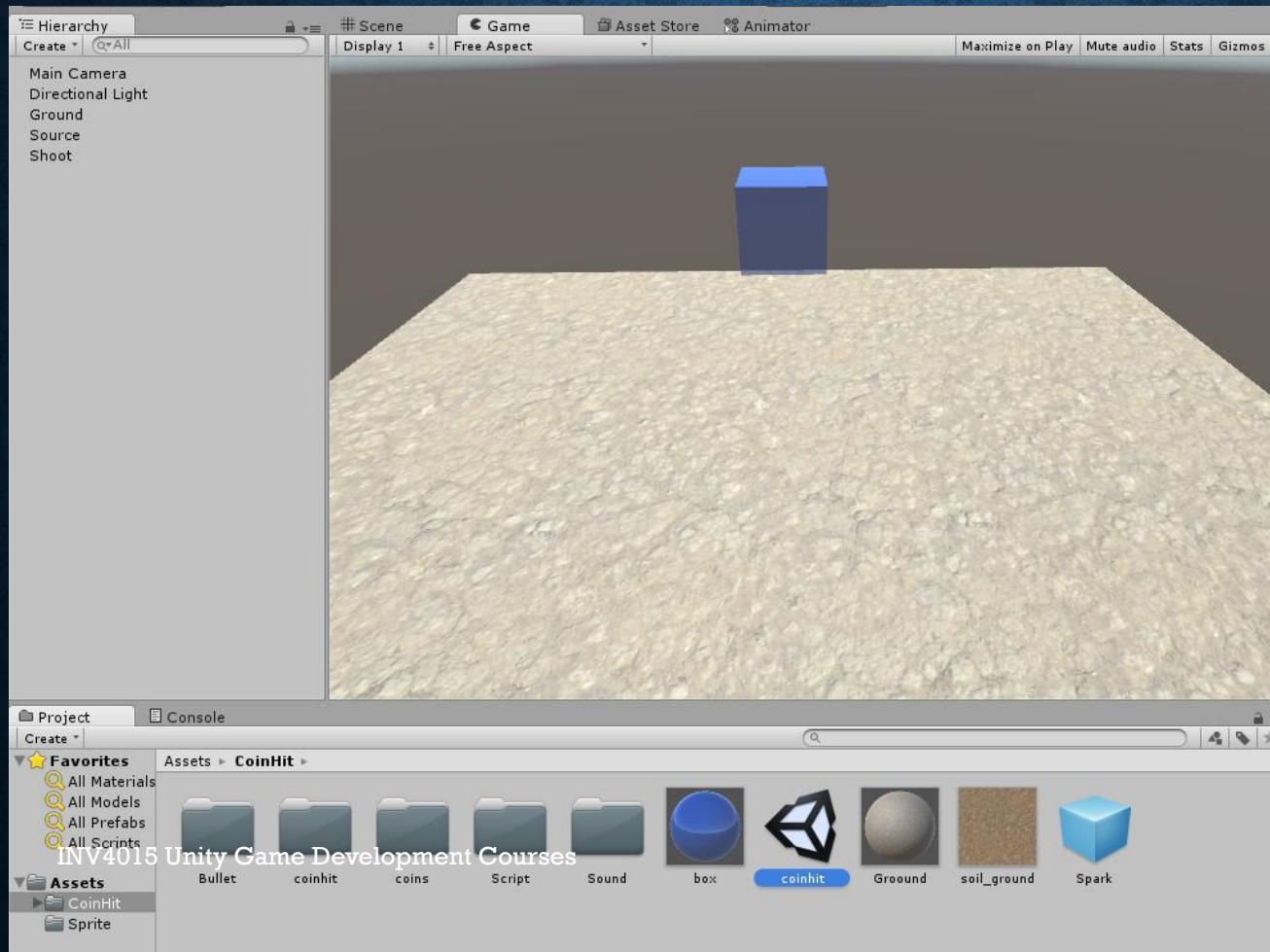


SLIDER

- Slider can be used to control different value of game object
 - E.g. Sound volume
 - E.g. Light Intensity
- Can also be used to display progress
 - E.g. Destination
 - E.g. Time bar
 - E.g. Hp bar

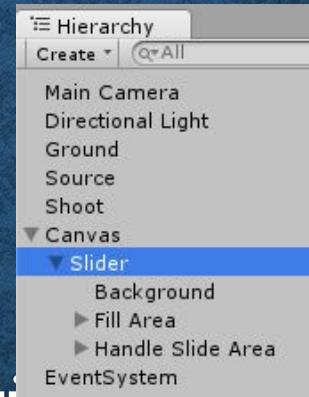
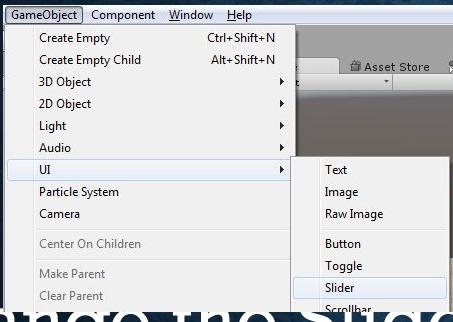
SLIDER

- Open the “coinhit” scene.

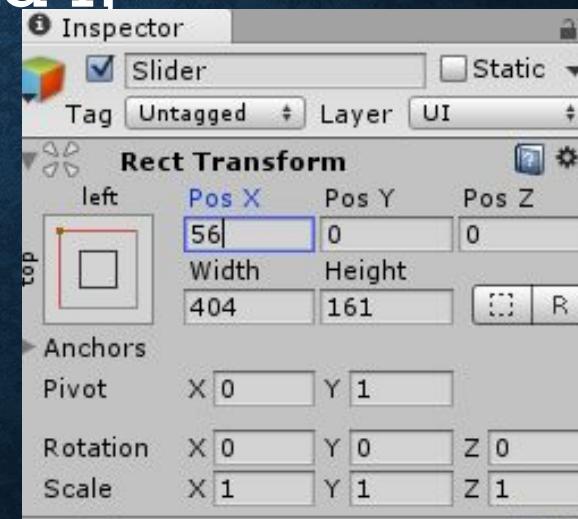


SLIDER SETUP

- Create a GUI Button to display the button in 2D level
 - GameObject > UI -> Slider

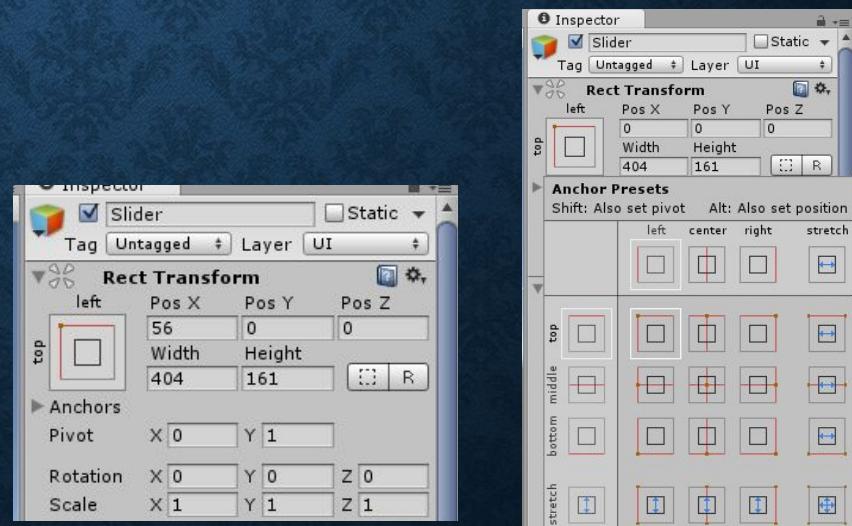
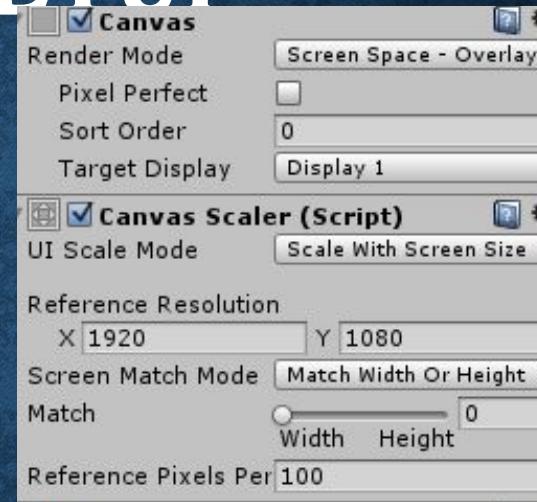


- Change the Slider position X and Y, Width and Height



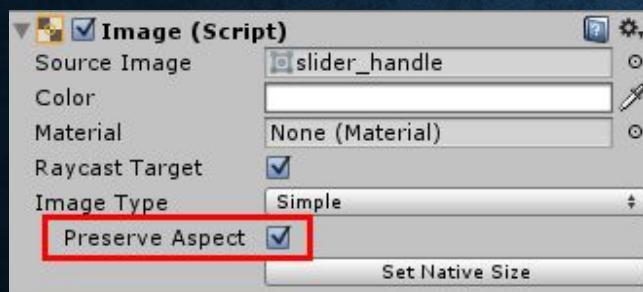
SLIDER SETUP

- Set the Canvas Render Mode
“Screen Space-Overlay”
- Set the Canvas Scaler resolution
to “1920 x 1080”
- Select Canvas -> Slider
 - Adjust the width and height
 - Use anchor preset to place the slider to the upper left corner



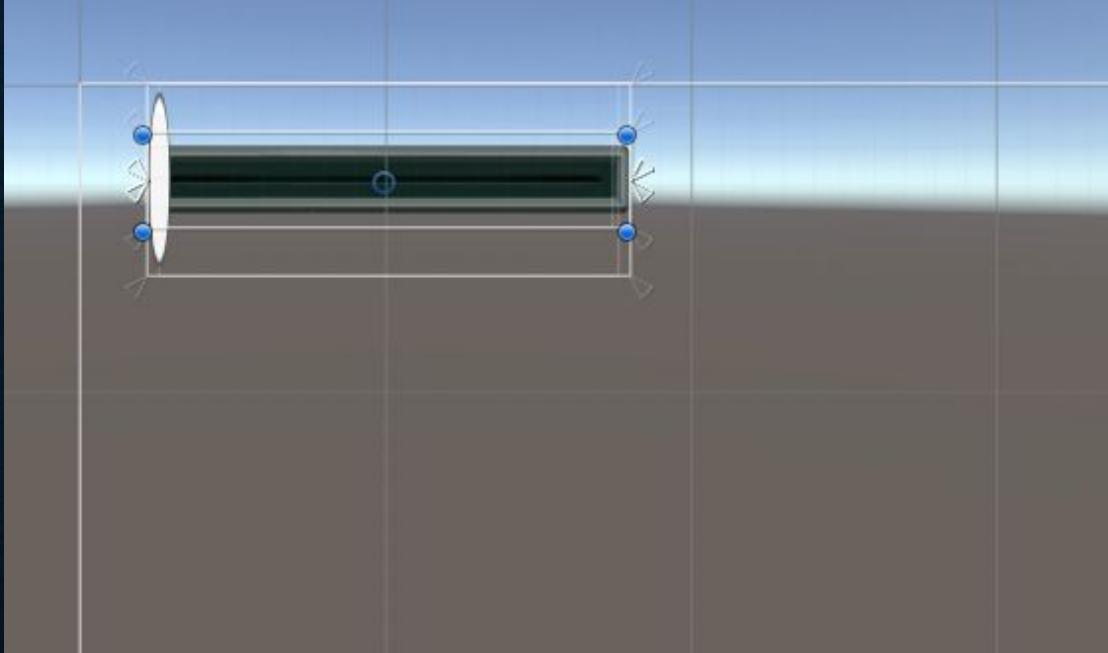
SLIDER SETUP

- Select Canvas ->Slider -> Background
- Adjust the width, height and pivot
- Change the background width and height
- Change the handle image and select “Preserve Aspect”, also change the scale accordingly



SLIDER SETUP

- Change the Handle Slide Area top and bottom

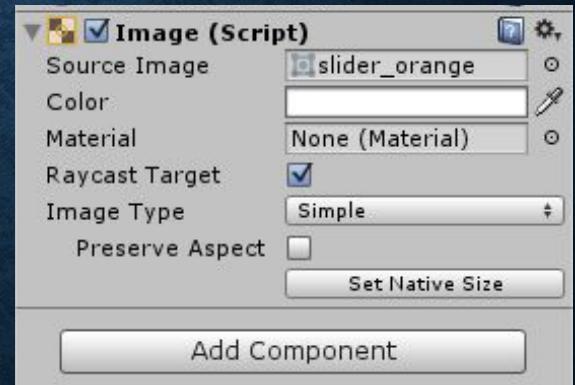


SLIDER SETUP

- Select Slider -> Fill Area -> Fill
- Adjust the width, height and pivot

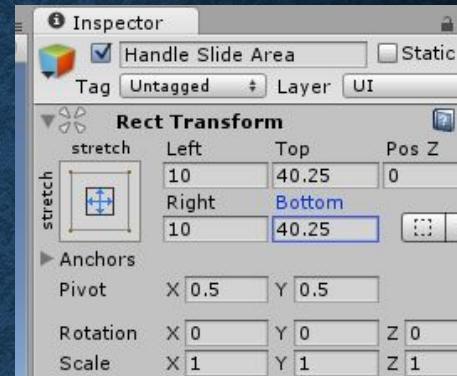


- Change the fill image “slider_orange” also change the scale accordingly



SLIDER SETUP

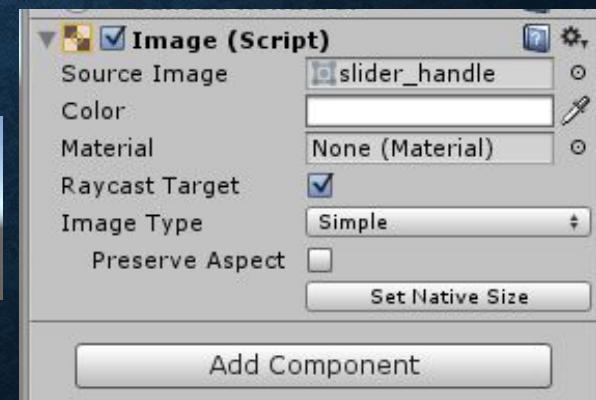
- Select Slider -> Handle Slide Area
- Adjust the width, height and pivot



- Select Slider -> Handle Slide Area-> Handle
- Change the fill image “slider_handle” also change the scale accordingly

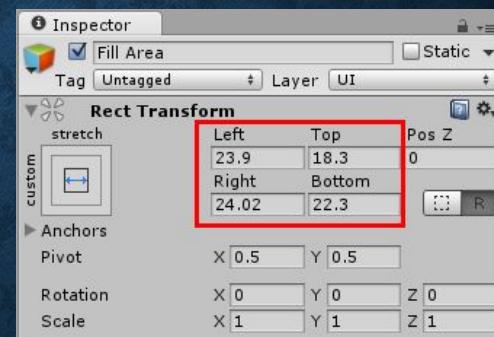
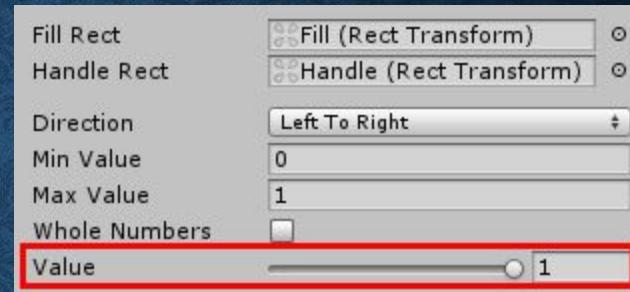


INV4015 Unity Game Development Courses



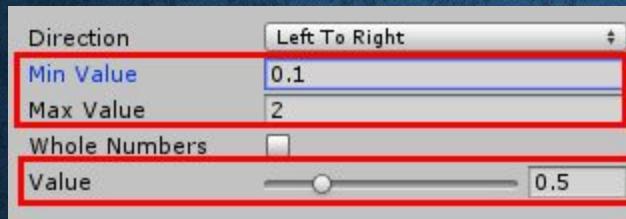
SLIDER SETUP

- Select the Slider script and change the value to 1
- Disable the handle
- Change the fill image (Sliced image)
- Select the Fill Area, change the left, right, top and bottom to fit the area of the slider background

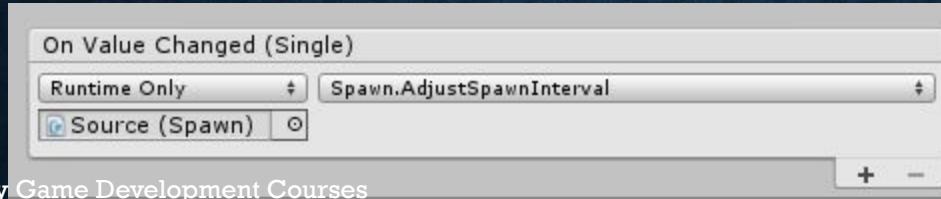


CONTROLLING SCRIPT

- Use the slider to control the coin spawn script
- Set the min, max and default value of the slider



- Add the On Value Changed event
- Drag the coin source to the (none)
- Select Spawn.AdjustSpawnInterval



CONTROLLING SCRIPT

- Coin spawn script

```
1  using UnityEngine;
2  using System.Collections;
3
4  public class Spawn : MonoBehaviour {
5
6      public GameObject item;
7      public float spawnInterval=0.5f;
8      float nextSpawnTime = 0;
9
10     void Update () {
11         if ((Time.time - nextSpawnTime) > 0) {
12             GameObject clone = Instantiate(item, transform.position, transform.rotation) as GameObject;
13             Physics.IgnoreCollision (clone.GetComponent<Collider>(), GetComponent<Collider>());
14             nextSpawnTime = Time.time + spawnInterval;
15         }
16     }
17
18     public void AdjustSpawnInterval (float interval) {
19         spawnInterval = interval;
20     }
21 }
```

CONTROLLING SCRIPT

- Now the slider can be used to change the spawn rate directly



CONTROLLING SCRIPT

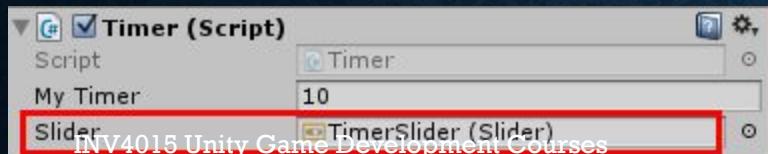
- To prevent firing while clicking the slider, as the following checking to the fire script

```
void Update () {
    if (Input.GetButtonDown("Fire1")
        && !UnityEngine.EventSystems.EventSystem.current.IsPointerOverGameObject()) {

        Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
        Debug.DrawRay(ray.origin, ray.direction*range, Color.green);
        RaycastHit hit;
        audio.PlayOneShot(hitSound);
        if (Physics.Raycast (ray, out hit, range)) {
```

PROGRESS BAR

- Duplicate the previous slider and name it TimeSlider, delete the Handle Slide Area
- Change the slider value to 0 and set the fill width to 0
- Place it at the button of the screen
- Create the Text for the timer display and add the timer script
- Drag the TimeSlider to the Timer script



PROGRESS BAR

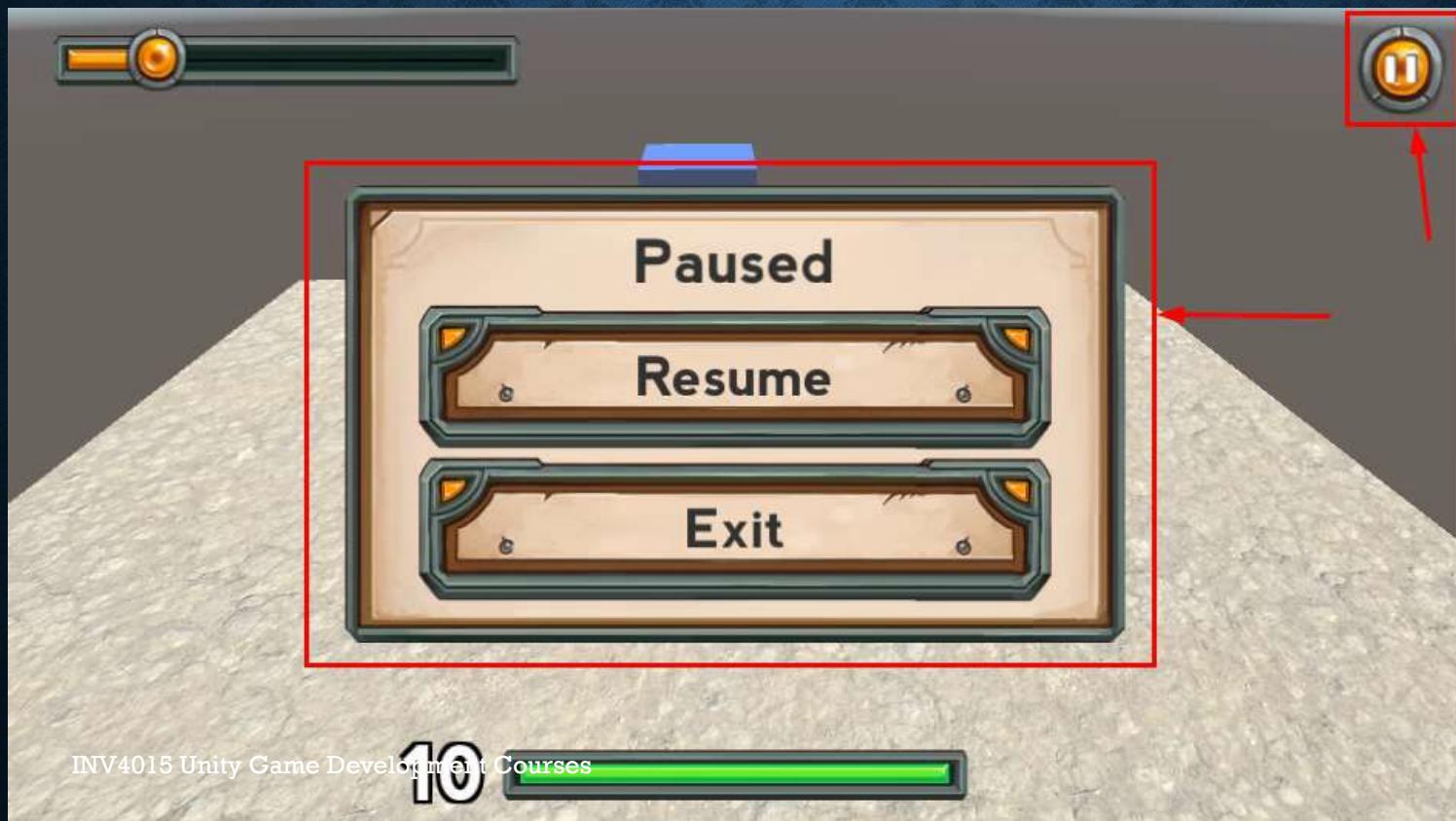
- Fill area will be reduced
- proportionally to the time left



```
Timer.cs
selection
2  using System.Collections;
3  using UnityEngine.UI;
4
5  public class Timer : MonoBehaviour {
6
7      public float myTimer=10;
8      public Slider slider;
9
10     Text text;
11
12     void Start () {
13         text = GetComponent<Text>();
14         slider.maxValue = myTimer;
15         slider.minValue = 0;
16     }
17
18     void Update () {
19         if (myTimer > 0)
20             myTimer -= Time.deltaTime;
21         else {
22             myTimer = 0;
23         }
24         slider.value = myTimer;
25         text.text = (myTimer).ToString("00");
26     }
27 }
```

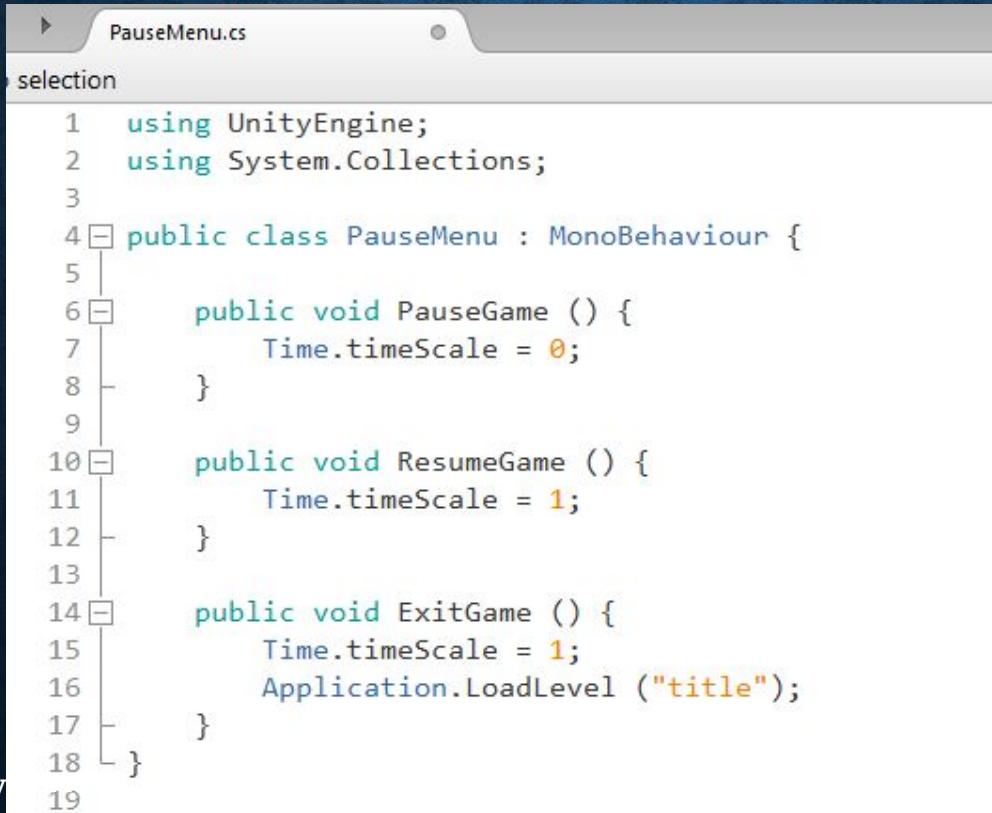
PAUSE MENU

- Use the image and button to create the following menu and pause button



PAUSE MENU

- Disable the pause menu and add the PauseMenu script to it

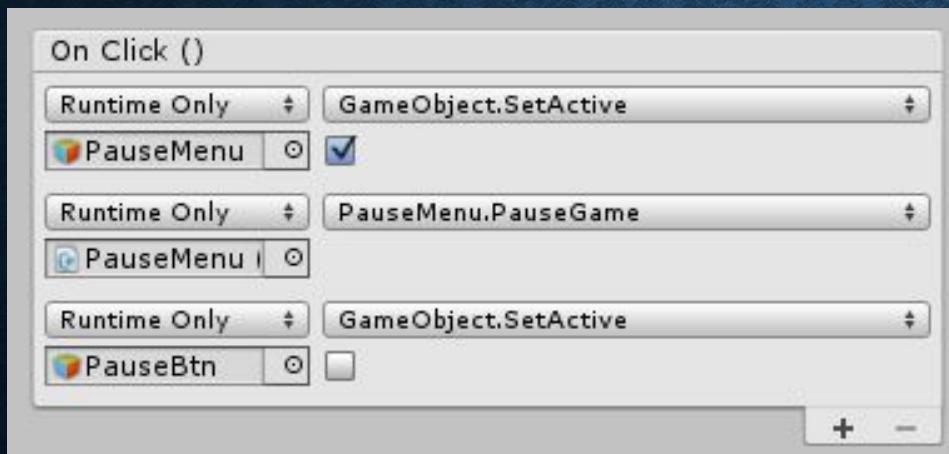


The image shows a code editor window with the file "PauseMenu.cs" open. The code defines a class "PauseMenu" that inherits from "MonoBehaviour". It contains three methods: "PauseGame", "ResumeGame", and "ExitGame". The "PauseGame" method sets Time.timeScale to 0, effectively pausing the game. The "ResumeGame" method sets Time.timeScale to 1, resuming the game. The "ExitGame" method sets Time.timeScale to 1 and loads a level titled "title".

```
1  using UnityEngine;
2  using System.Collections;
3
4  public class PauseMenu : MonoBehaviour {
5
6      public void PauseGame () {
7          Time.timeScale = 0;
8      }
9
10     public void ResumeGame () {
11         Time.timeScale = 1;
12     }
13
14     public void ExitGame () {
15         Time.timeScale = 1;
16         Application.LoadLevel ("title");
17     }
18 }
19
```

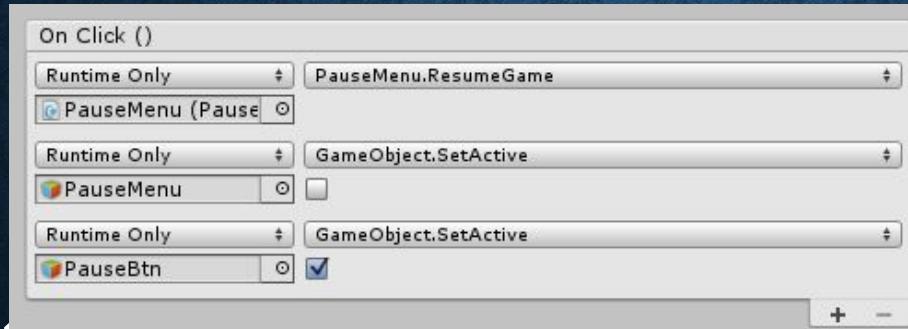
PAUSE MENU

- Select the pause button and add 3 On Click () Event
 1. PauseMenu -> GameObject -> SetActive -> True
 2. PauseMenu -> PauseMenu -> PauseGame ()
 3. PauseBtn -> GameObject -> SetActive -> False



PAUSE MENU

- Select resume button and add 3 On Click () events
 1. PauseMenu -> PauseMenu -> PauseGame ()
 2. PauseBtn -> GameObject -> SetActive -> True
 3. PauseMenu -> GameObject -> SetActive -> False



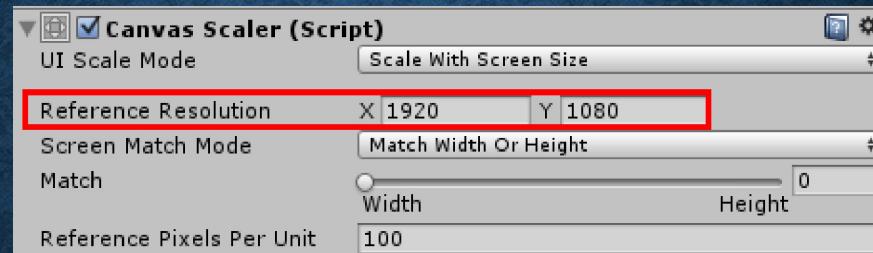
- Select exit button and add On Click () event

1. PauseMenu -> PauseMenu -> ExitGame ()

CANVAS SCALER

- Scale With Screen Size

- allow auto scaling of UI element according to the Reference Resolution



CANVAS SCALER

- Button and Image
- Panel
- Slider
- Progress Bar
- Pause

FUNCTIONS

- A piece of code that is called by name.
- Can be passed data to operate on (ie. The parameters) and can optionally return data (the return value)
- void is used for the return type if the function does not provide a result value to its caller

return type function name parameters

```
float HypotenuseLength(float sideALength, float sideBLength)
{
    return Mathf.Sqrt(sideALength * sideALength + sideBLength * sideBLength);
}
```

BASIC FUNCTION

- **void Start ()**
- – Any code inserted inside will be executed only once in
 - the lifetime when the game object is initially used
- **void Update ()**
- – Any code you insert inside will be executed every
 - frame
- **void FixedUpdate ()**
- – Any code you insert inside will be executed in
 - particular time interval
- – Usually for physics calculation

BASIC FUNCTION

- Example
 - – The following script will move the applied game object with 0.1 in the y axis every frame