UI UX Advanced
Unity UI for Happy People

Yvens Serpa

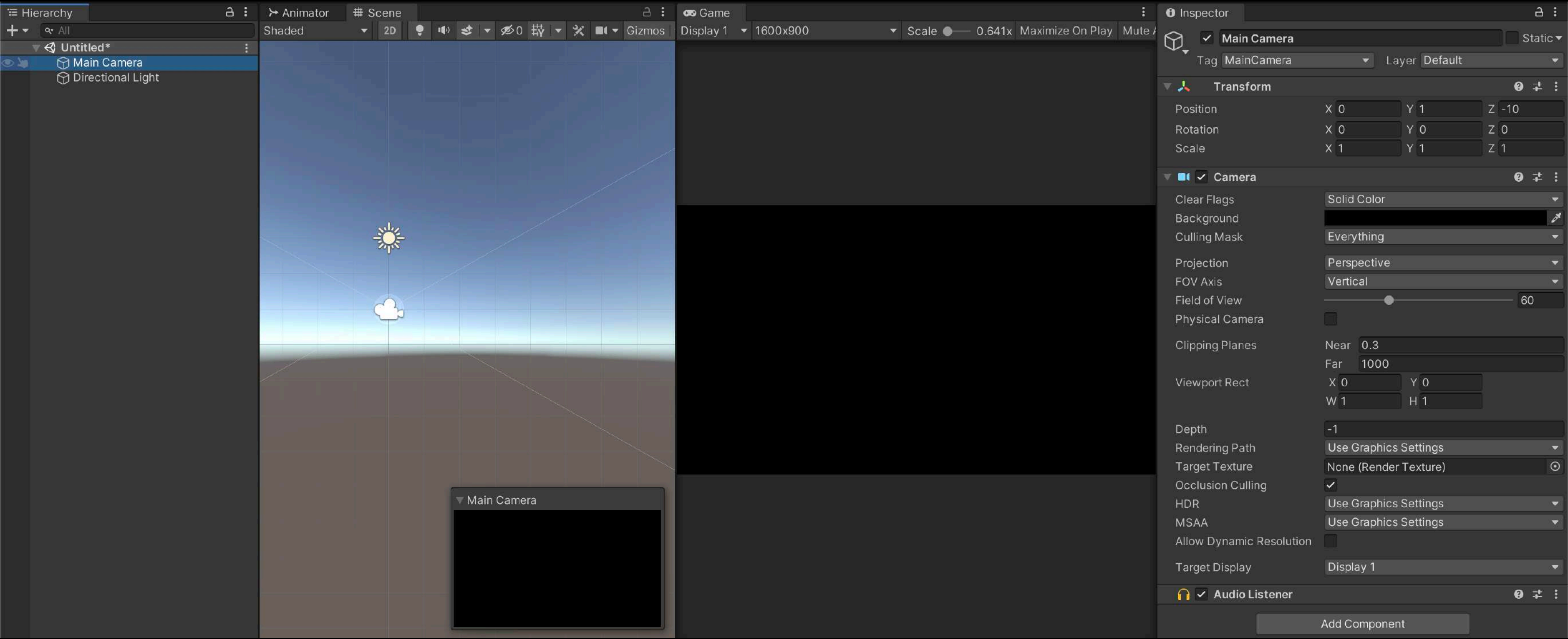
Introduction

- Yvens Rebouças Serpa
- y.reboucasserpa@saxion.nl
  - or YRE03
- From Fortaleza, Ceará - Brazil ☺
- Background in Computer Sciences, especially Computer Graphics (OpenGL with C++)
  - Scientific Research on Games and CG techniques

Unity UI

- Based on Canvas Objects
- Easier to organize using **panels**
- Focus on anchors to properly set elements for different aspects and resolutions

- Goals:
  - Use the Editor/Inspector as much as possible
  - Few scripts to generalize behavior
  - Animations for the magic
- Examples using Unity 2020.1



Main Camera
Static
Tag MainCamera
Layer Default

Transform

Position	X 0	Y 1	Z -10
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera

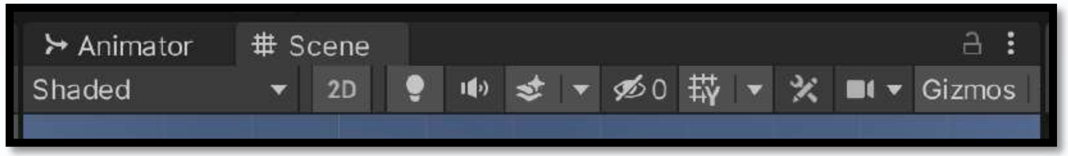
Clear Flags	Solid Color
Background	⬛
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	
Target Display	Display 1

Audio Listener

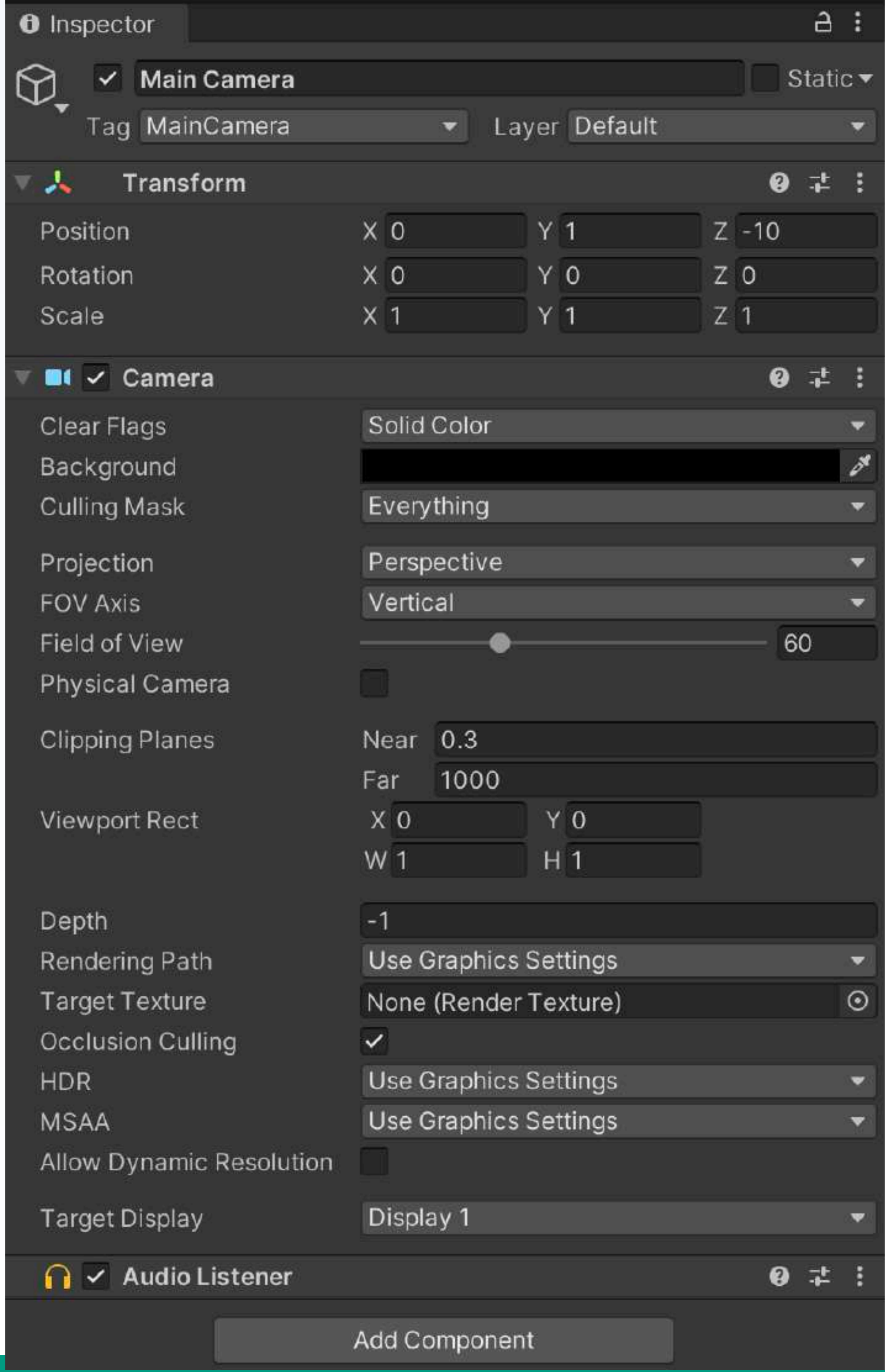
Add Component

Setting up the Camera

- Change the Camera Settings to have a solid color background
  - Not necessarily black
- Mark Unity to use the 2D view



- Camera changes are only applied to the Game Window


Inspector

Main Camera ✓ — Static

Tag MainCamera — Layer Default

Transform

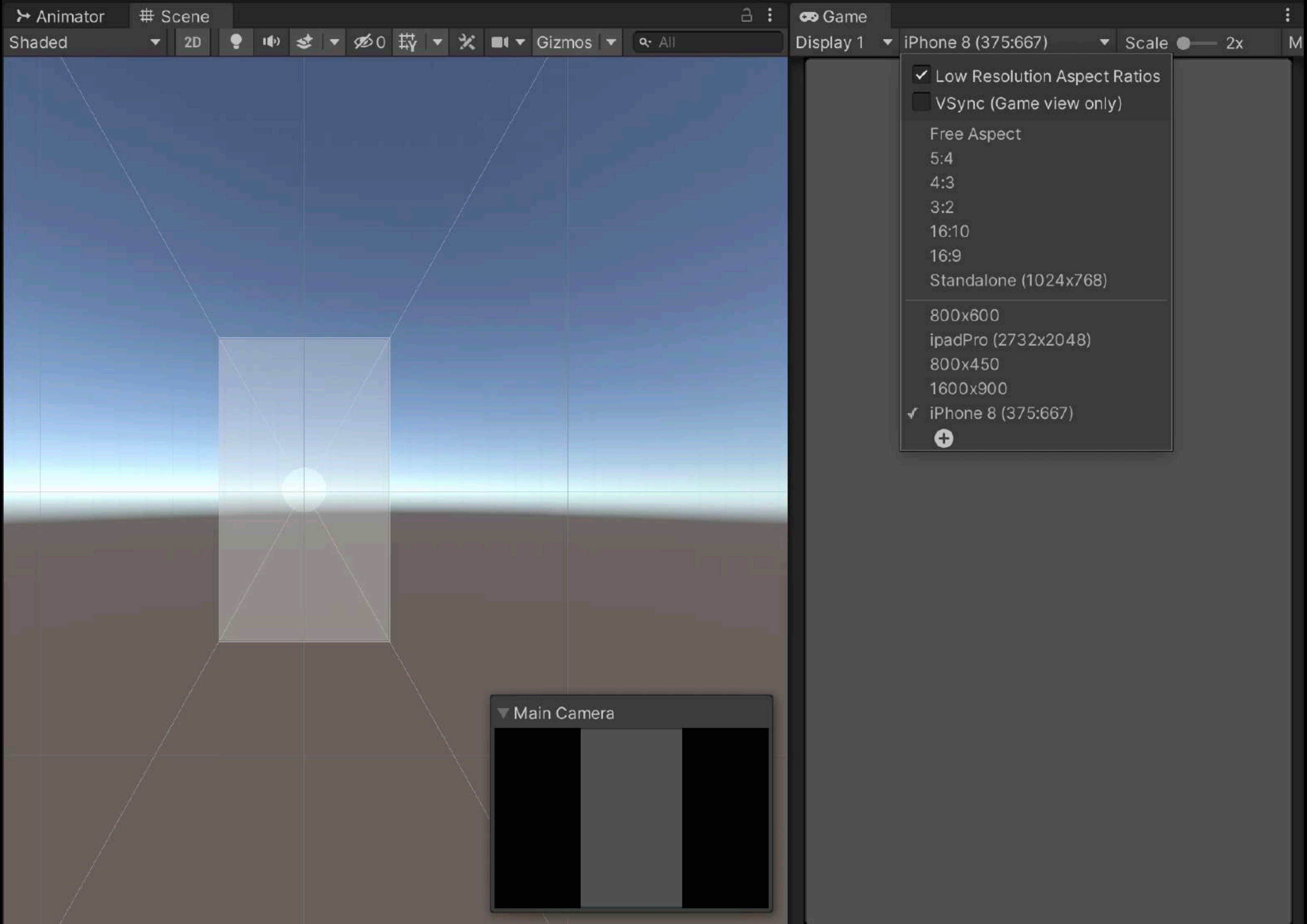
Position	X 0	Y 1	Z -10
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Camera ✓

Clear Flags	Solid Color
Background	■
Culling Mask	Everything
Projection	Perspective
FOV Axis	Vertical
Field of View	60
Physical Camera	
Clipping Planes	Near 0.3
	Far 1000
Viewport Rect	X 0 Y 0
	W 1 H 1
Depth	-1
Rendering Path	Use Graphics Settings
Target Texture	None (Render Texture)
Occlusion Culling	✓
HDR	Use Graphics Settings
MSAA	Use Graphics Settings
Allow Dynamic Resolution	
Target Display	Display 1

Audio Listener ✓

Add Component



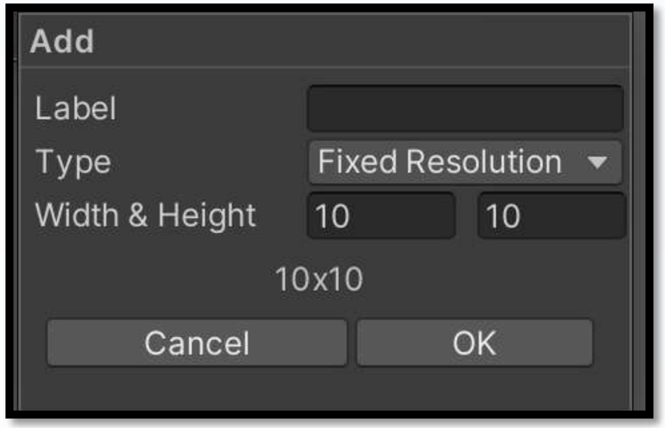
- [✓] Low Resolution Aspect Ratios
- [ ] VSync (Game view only)

Free Aspect
5:4
4:3
3:2
16:10
16:9
Standalone (1024x768)

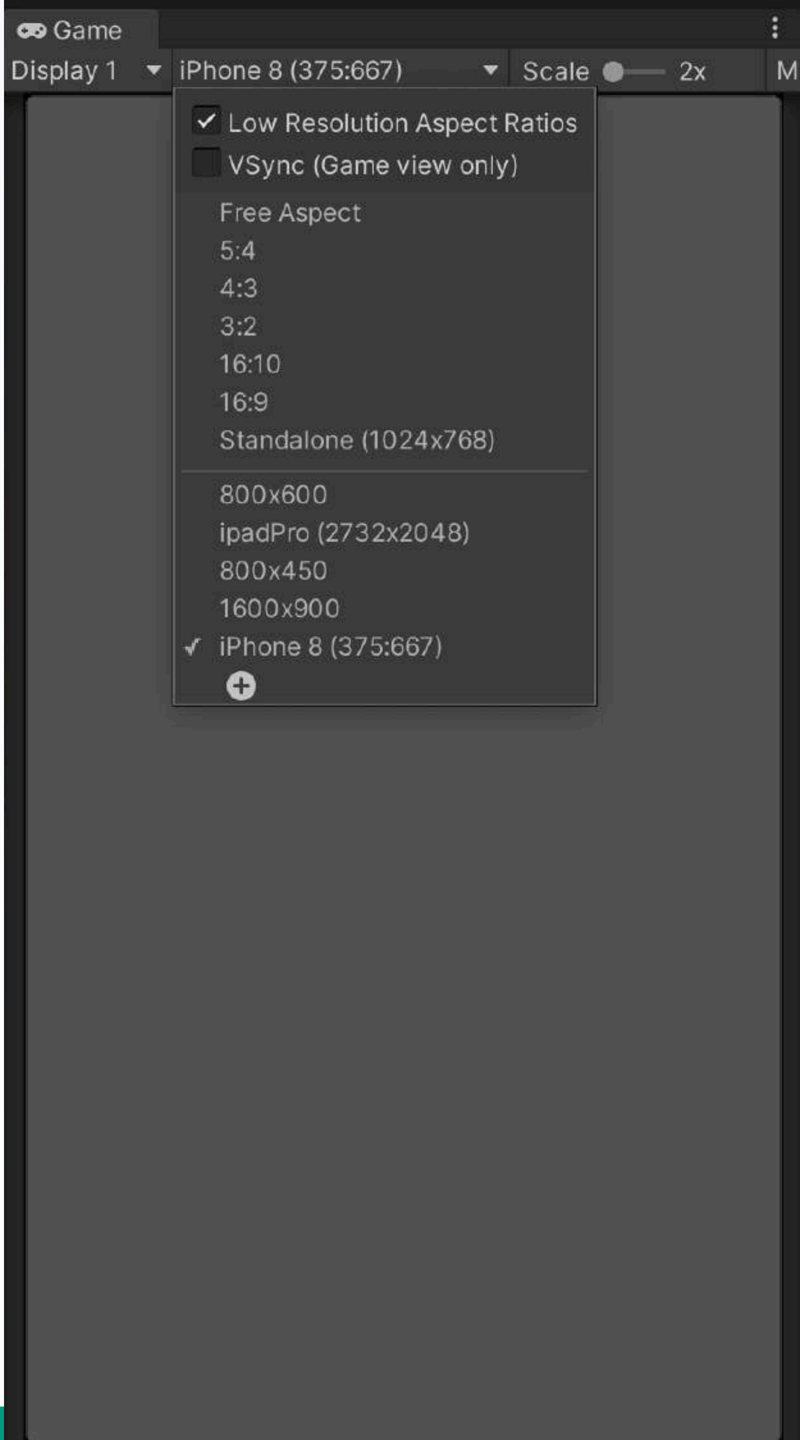
800x600
ipadPro (2732x2048)
800x450
1600x900
✓ iPhone 8 (375:667)

Setting up the Game Window

- Setup the game window to match your end device's aspect ratio
  - Can use either aspect ratio or fixed resolution


Add
Label
Type: Fixed Resolution
Width & Height: 10 10
10x10

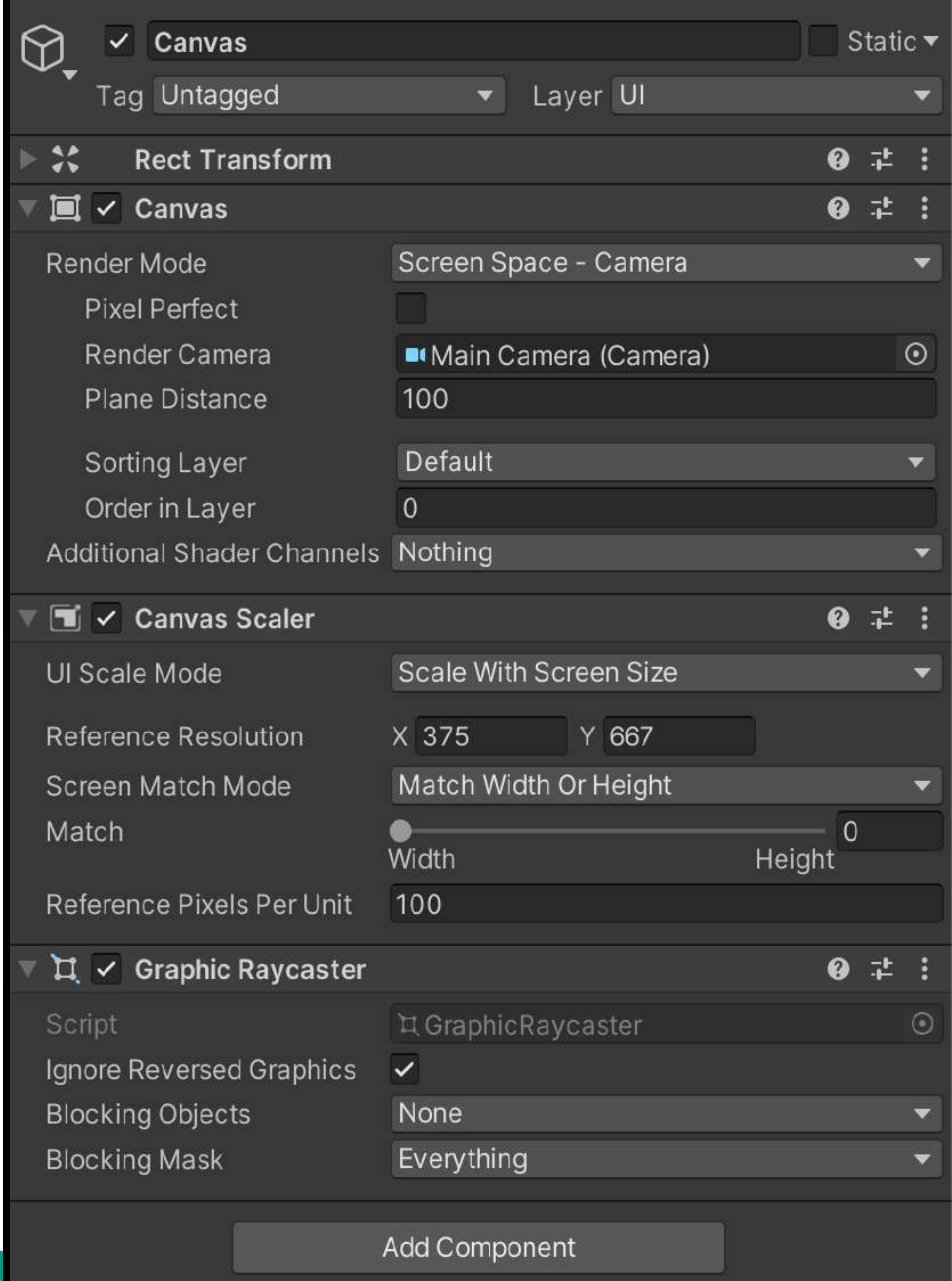
- This step is the basis for the UI configurations



Canvas

- The Canvas is the game object in which the UI is rendered and displayed
  - UI elements are children of the Canvas object

- Practical Notes:
  - Set the canvas to Screen Space – Camera
    - Select the main camera
  - Canvas Scaler for Scale With Screen Size
    - Set the reference resolution based on your screen resolution or aspect ration



Canvas — Static

Tag: Untagged — Layer: UI

Rect Transform

Canvas

Render Mode	Screen Space - Camera
Pixel Perfect	
Render Camera	Main Camera (Camera)
Plane Distance	100
Sorting Layer	Default
Order in Layer	0
Additional Shader Channels	Nothing

Canvas Scaler

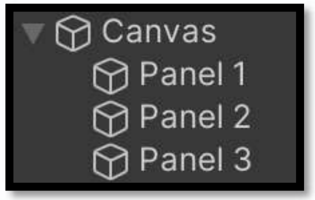
UI Scale Mode	Scale With Screen Size
Reference Resolution	X 375 Y 667
Screen Match Mode	Match Width Or Height
Match	0 (Width – Height)
Reference Pixels Per Unit	100

Graphic Raycaster

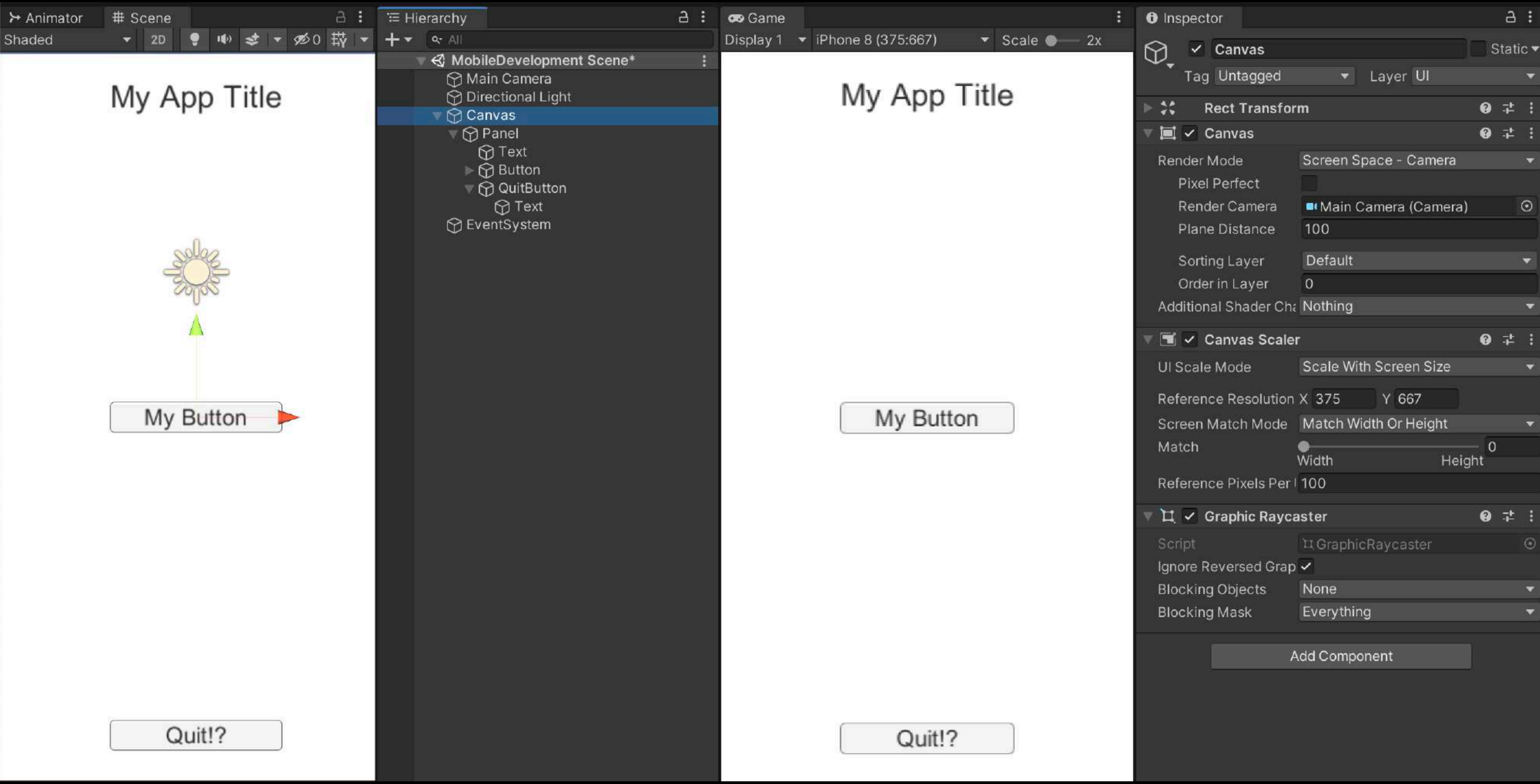
Script	GraphicRaycaster
Ignore Reversed Graphics	✓
Blocking Objects	None
Blocking Mask	Everything

A Few Comments on the Canvas

- You are **not** limited to one canvas per scene
  - Use as many as necessary
- Canvas update all its elements every time one change happens
  - Separate elements in different canvas if they have different update routines/cycles
  - A menu or background effect, *e.g.*
- Screen Space Camera allows elements to have a Z position
  - Closer/Further from the camera
- Elements are drawn in the same order as displayed in the hierarchy



  - Panel 3 is drawn on top of Panel 2
  - Panel 2 is drawn on top of Panel 1



Hierarchy

- MobileDevelopment Scene*
  - Main Camera
  - Directional Light
  - Canvas
    - Panel
      - Text
      - Button
      - QuitButton
        - Text
  - EventSystem

Inspector

Canvas — Static
Tag Untagged — Layer UI

Rect Transform

Canvas
Render Mode: Screen Space - Camera
Pixel Perfect:
Render Camera: Main Camera (Camera)
Plane Distance: 100
Sorting Layer: Default
Order in Layer: 0
Additional Shader Chı: Nothing

Canvas Scaler
UI Scale Mode: Scale With Screen Size
Reference Resolution: X 375 Y 667
Screen Match Mode: Match Width Or Height
Match: 0 (Width — Height)
Reference Pixels Per I: 100

Graphic Raycaster
Script: GraphicRaycaster
Ignore Reversed Grap: ✓
Blocking Objects: None
Blocking Mask: Everything

Add Component

Panels, Buttons, and Texts

- UI applications will be built using mostly Panels, Buttons and Texts
  - Panels: Contain content
  - Texts: Contain a text
  - Buttons: Are interactable
    - Also have a Text

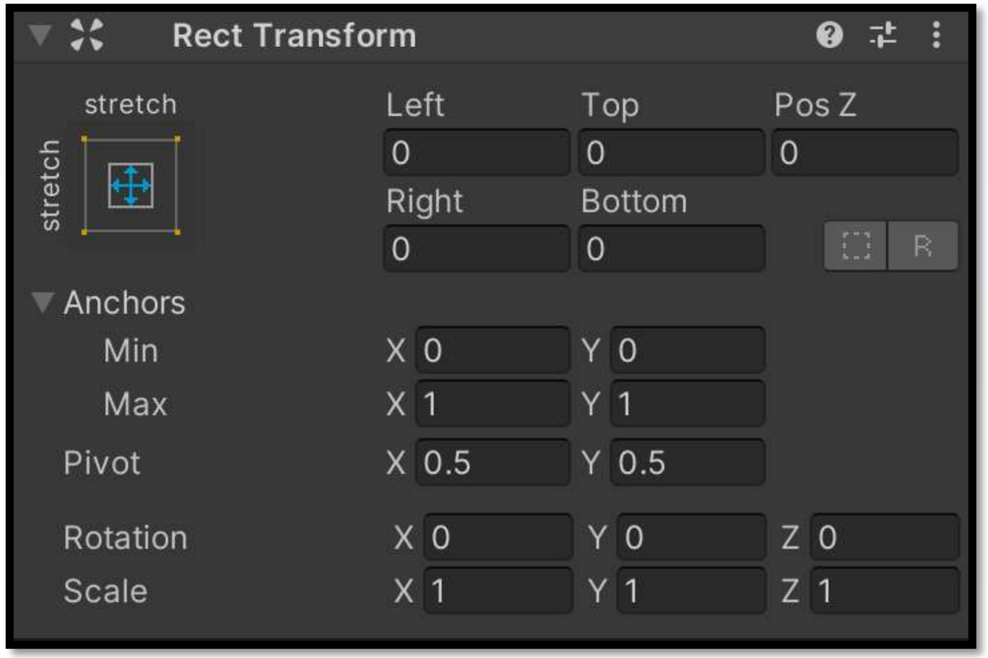
- These elements are all Game Objects and can be treated as such
  - Scripts, Components, etc.

- The main difference from these components to regular Game Objects from a 3D application is that they have a **RectTransform** instead of a regular **Transform**

- RectTransform is the 2D equivalent of a Transform and represents a rectangle that can contain an UI Element

RectTransforms and Anchor Points

- RectTransforms can be placed on the Canvas to position elements
  - This is bound to how the anchor points are established in the scene
- Anchor Points control how the element will be anchored in the UI, relative to its Parent



Rect Transform

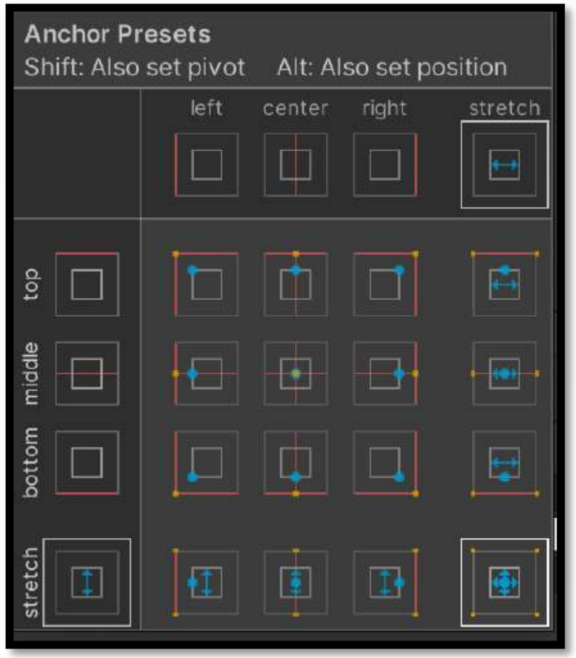
	Left	Top	Pos Z
stretch / stretch	0	0	0
	Right	Bottom	
	0	0	

Anchors
- Min: X 0, Y 0
- Max: X 1, Y 1

Pivot: X 0.5, Y 0.5

Rotation: X 0, Y 0, Z 0

Scale: X 1, Y 1, Z 1



We Most Use Only 2 of Them

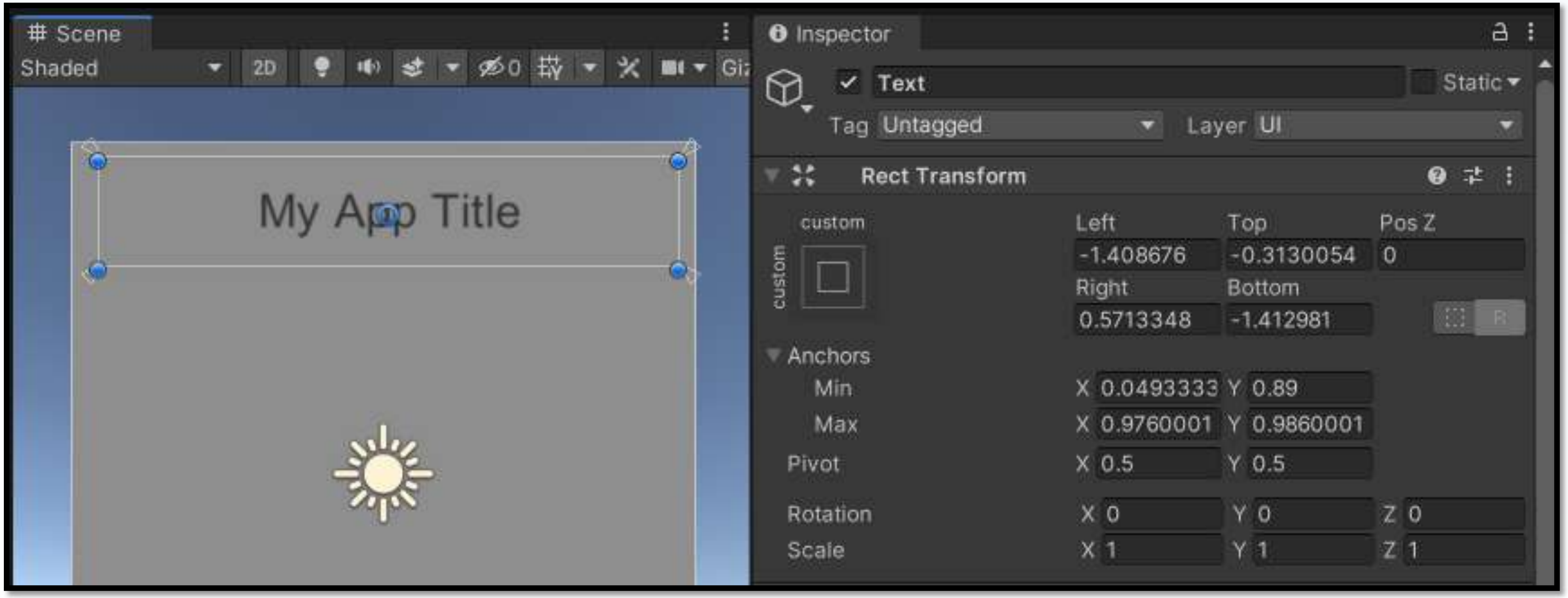
- Most commonly, to ensure that the UI elements stay in their correct place, we tend to use only two of the anchors presets
    - **Strecht x Stretch**: that completely expands the child object to spam over the parent object
    - **Custom**: in which we manually place the anchors on the screen

- Custom allows for better control

- Two approaches for setting custom anchors:
    - Use the **T shortcut** (Rect Tool) and adjust the UI element first, then move the anchor points to their correct place*
    - Move the anchor points directly holding SHIFT to apply the changes directly to the object

* Using the T approach might require a RectTransform clean up

Using the Rect Tool for Placement



Text
Static
Tag Untagged
Layer UI

custom

Left	Top	Pos Z
-1.408676	-0.3130054	0
Right	Bottom	
0.5713348	-1.412981	

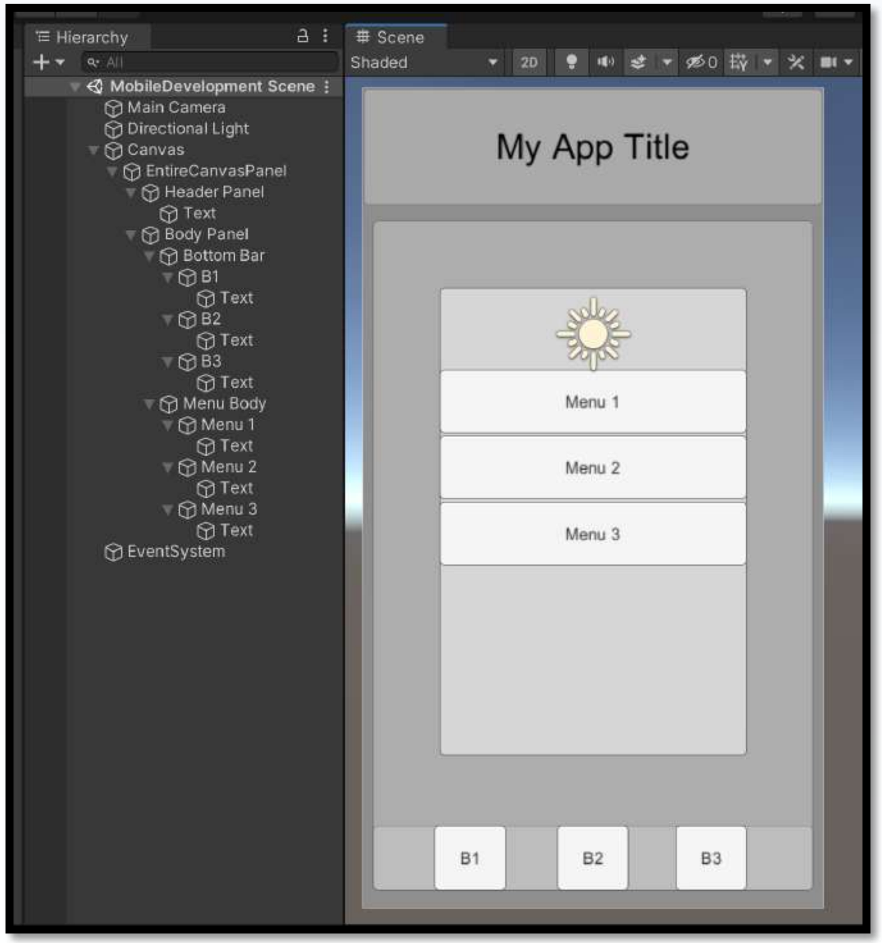
Anchors

	X	Y	Z
Min	0.0493333	0.89	
Max	0.9760001	0.9860001	
Pivot	0.5	0.5	
Rotation	0	0	0
Scale	1	1	1

- Anchors' range from 0 to 1 (0% to 100%), and are relative to the Parent

Setting Up The Layout

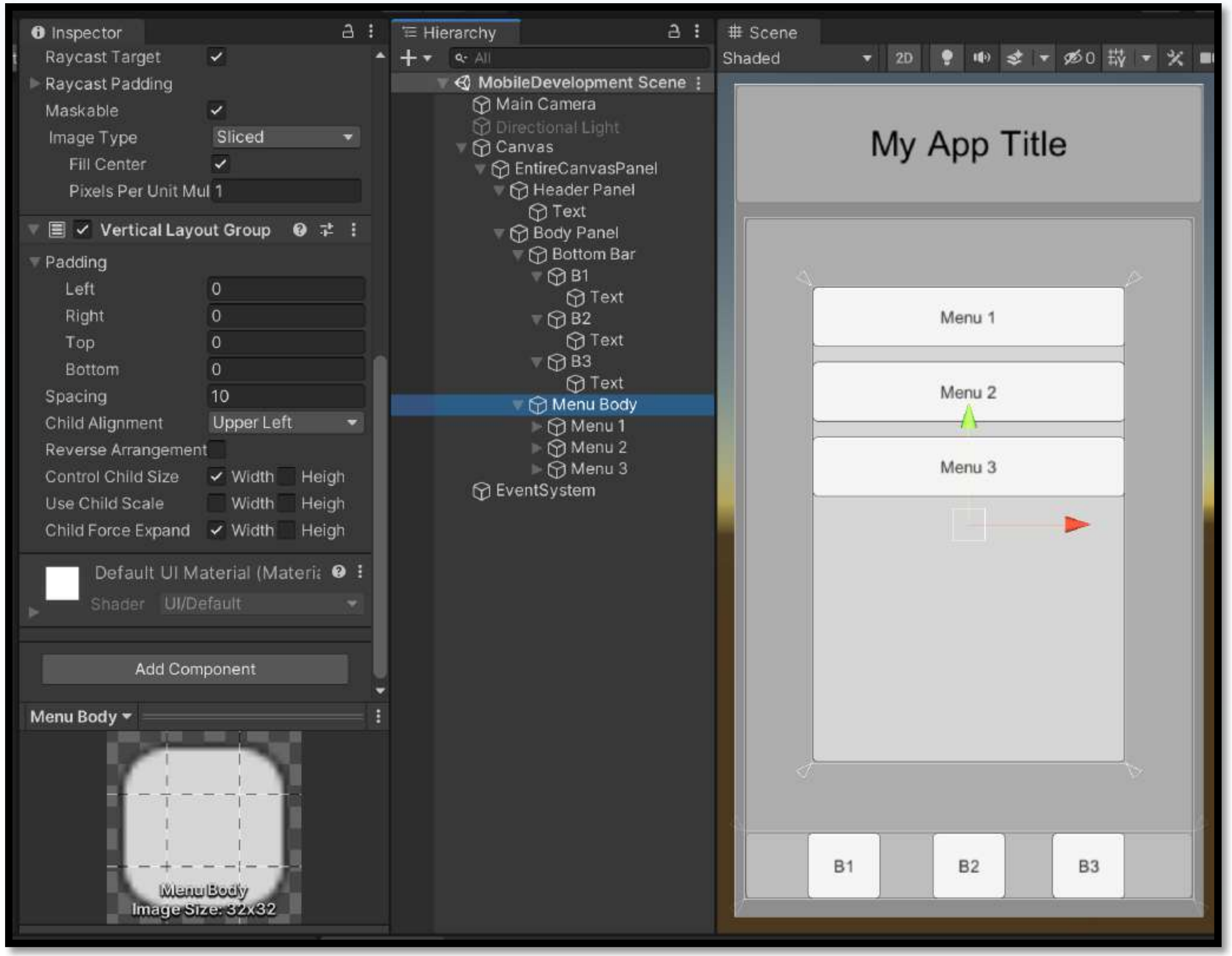
- Organize your components to create the desired layout
  - Nest components to form a hierarchy
  - Anchors are relative to the parent objects
- Focus on layout first
  - Fix the positions and details after the major elements are in place



Layout Groups

- Instead of fixing the position for all elements, you can use Layout Groups:
  - Vertical and Horizontal layout

- Control:
  - Padding
  - Spacing
  - Alignment
  - Size (Width and Height)
  - Order



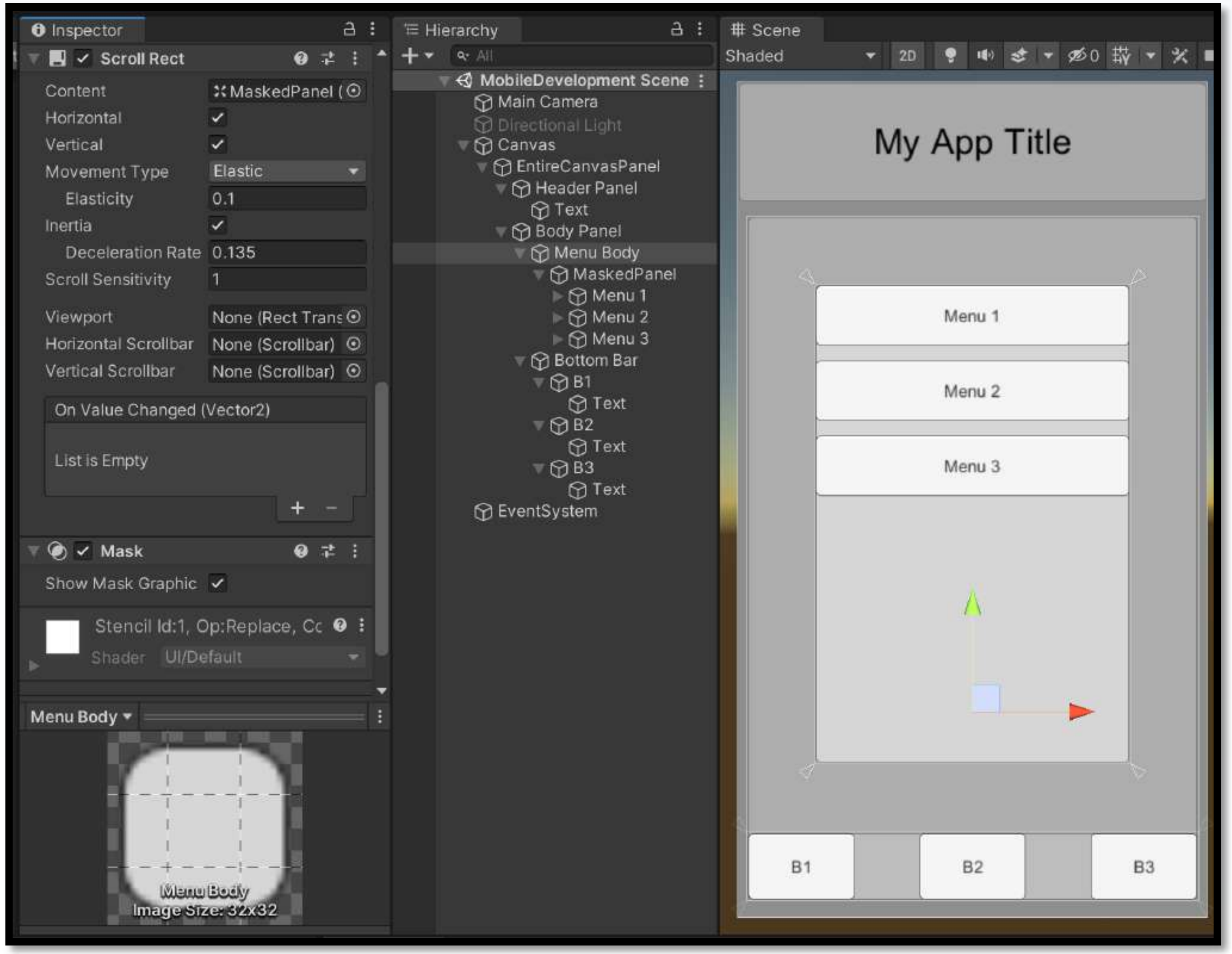
```
Inspector
Raycast Target            ✓
Raycast Padding
Maskable                  ✓
Image Type                Sliced
  Fill Center             ✓
  Pixels Per Unit Mul     1
Vertical Layout Group     ✓
Padding
  Left                    0
  Right                   0
  Top                     0
  Bottom                  0
Spacing                   10
Child Alignment           Upper Left
Reverse Arrangement
Control Child Size        ✓ Width    Heigh
Use Child Scale             Width    Heigh
Child Force Expand        ✓ Width    Heigh
Default UI Material (Materi
Shader                    UI/Default
Add Component
Menu Body
Menu Body
Image Size: 32x32
```

```
Hierarchy
MobileDevelopment Scene
  Main Camera
  Directional Light
  Canvas
    EntireCanvasPanel
      Header Panel
        Text
      Body Panel
        Bottom Bar
          B1
            Text
          B2
            Text
          B3
            Text
        Menu Body
          Menu 1
          Menu 2
          Menu 3
  EventSystem
```

Scroll and Mask

- Sometimes we do not know how many elements will be listed, but our screen is still limited
  - We can use a Scroll Rect component to allow a Rect Transform to be scrollable

- We need to change the hiearchy
  - Scrollable needs to be nested
  - Scrollable needs to be bigger
  - Scrollable needs to be masked



Inspector

Scroll Rect

Content	MaskedPanel (
Horizontal	✓
Vertical	✓
Movement Type	Elastic
Elasticity	0.1
Inertia	✓
Deceleration Rate	0.135
Scroll Sensitivity	1
Viewport	None (Rect Trans
Horizontal Scrollbar	None (Scrollbar)
Vertical Scrollbar	None (Scrollbar)

On Value Changed (Vector2)

List is Empty

Mask

Show Mask Graphic	✓

Stencil Id:1, Op:Replace, Cc

Shader UI/Default

Menu Body

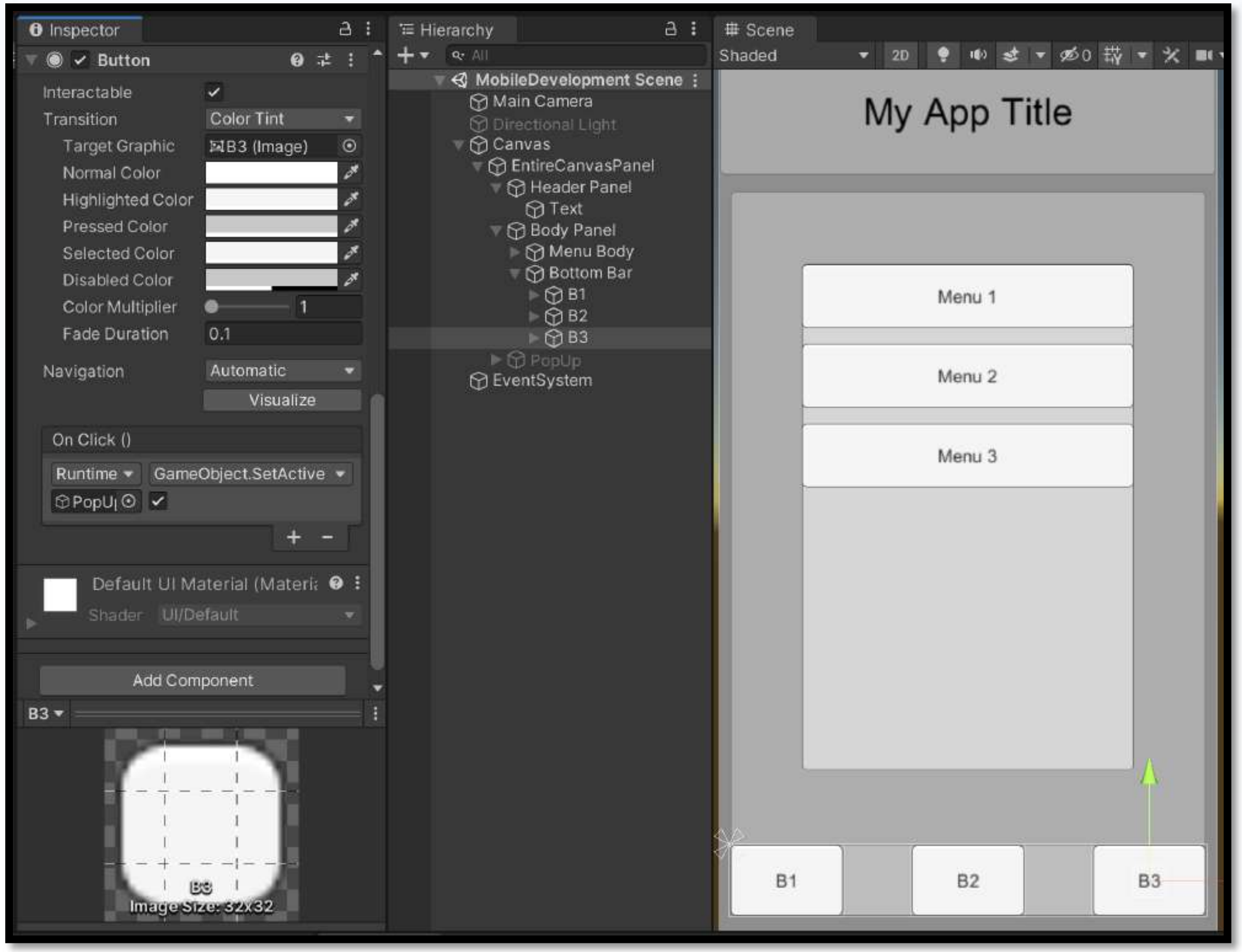
Menu Body
Image Size: 32x32

Hierarchy

```
MobileDevelopment Scene
  Main Camera
  Directional Light
  Canvas
    EntireCanvasPanel
      Header Panel
        Text
      Body Panel
        Menu Body
          MaskedPanel
            Menu 1
            Menu 2
            Menu 3
      Bottom Bar
        B1
          Text
        B2
          Text
        B3
          Text
  EventSystem
```

Use Buttons to Activate/Deactivate

- Buttons have a OnClick event that will be executed when the button is clicked
- You can use this functionality to activate/deactivate game objects
  - GameObject.SetActive
- Simplest alternative to make the navigation for your app



Inspector

Button

Interactable ✓
Transition Color Tint
Target Graphic B3 (Image)
Normal Color
Highlighted Color
Pressed Color
Selected Color
Disabled Color
Color Multiplier 1
Fade Duration 0.1

Navigation Automatic
Visualize

On Click ()
Runtime GameObject.SetActive
PopU ✓

Default UI Material (Materi
Shader UI/Default

Add Component

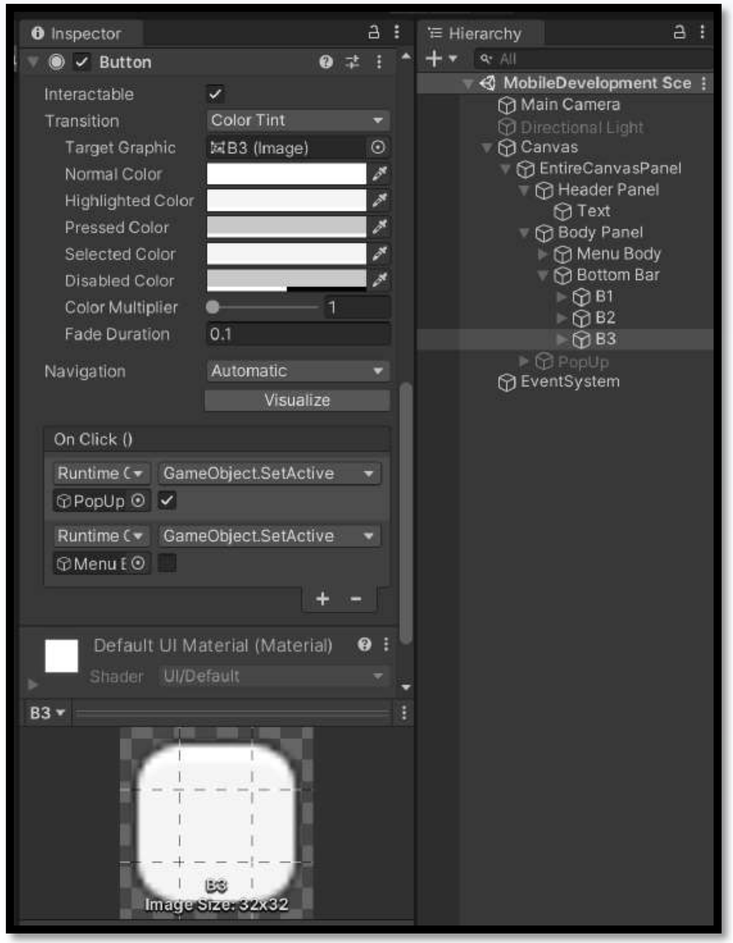
B3
B3
Image Size: 32x32

Hierarchy

MobileDevelopment Scene
Main Camera
Directional Light
Canvas
EntireCanvasPanel
Header Panel
Text
Body Panel
Menu Body
Bottom Bar
B1
B2
B3
PopUp
EventSystem

Use Buttons for Multiple Actions

- The OnClick function is not limited to just one event
  - You can use multiple
- It is very common to resort to this for more complex behavior:
  - Activate/Deactive objects
  - Play a sound
  - Activate an animation
  - Play a timeline event



Interactable ✓
Transition: Color Tint
Target Graphic: B3 (Image)
Normal Color
Highlighted Color
Pressed Color
Selected Color
Disabled Color
Color Multiplier: 1
Fade Duration: 0.1
Navigation: Automatic

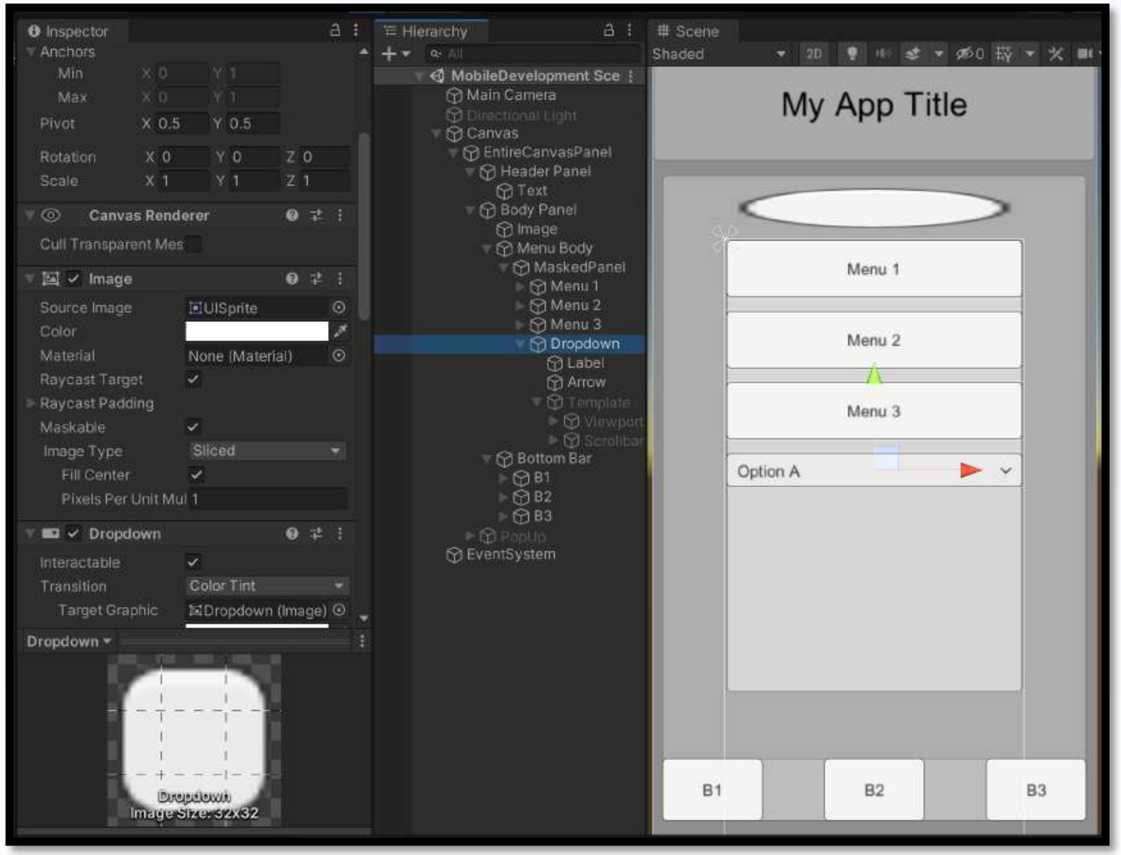
On Click ()
Runtime C | GameObject.SetActive
PopUp ✓
Runtime C | GameObject.SetActive
Menu E

Default UI Material (Material)
Shader: UI/Default

Other Elements with Rect Transforms

- There are other elements with Rect Transforms, but they are more situationals

- Worth mentioning:
  - Image: To display an image (sprite)
  - Dropdow: To display options
  - Input Field: To input text



Inspector

Anchors			
Min	X 0	Y 1	
Max	X 0	Y 1	
Pivot	X 0.5	Y 0.5	
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Canvas Renderer

Cull Transparent Mes	

Image

Source Image	UISprite
Color	
Material	None (Material)
Raycast Target	✓
Raycast Padding	
Maskable	✓
Image Type	Sliced
Fill Center	✓
Pixels Per Unit Mul	1

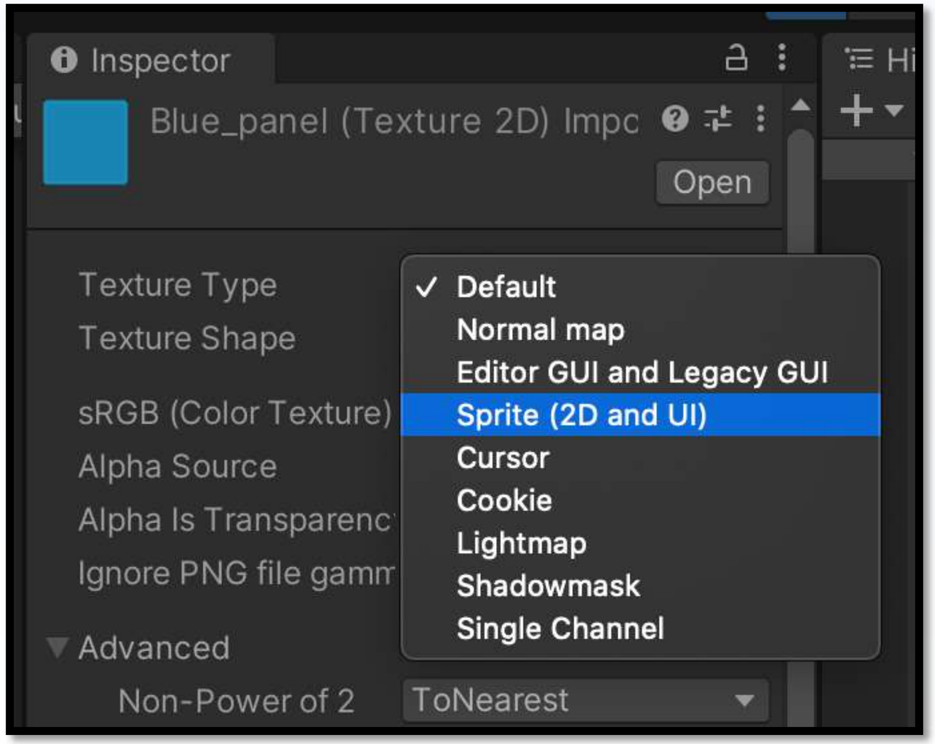
Dropdown

Interactable	✓
Transition	Color Tint
Target Graphic	Dropdown (Image)

Dropdown
Image Size: 32x32

Importing Images and Making them Sprites

- Importing images to Unity will mark them as Default type
- Unity UI elements need Sprites instead of Default images
- Select the image you want to use, change the type to Sprite and click Apply



Inspector

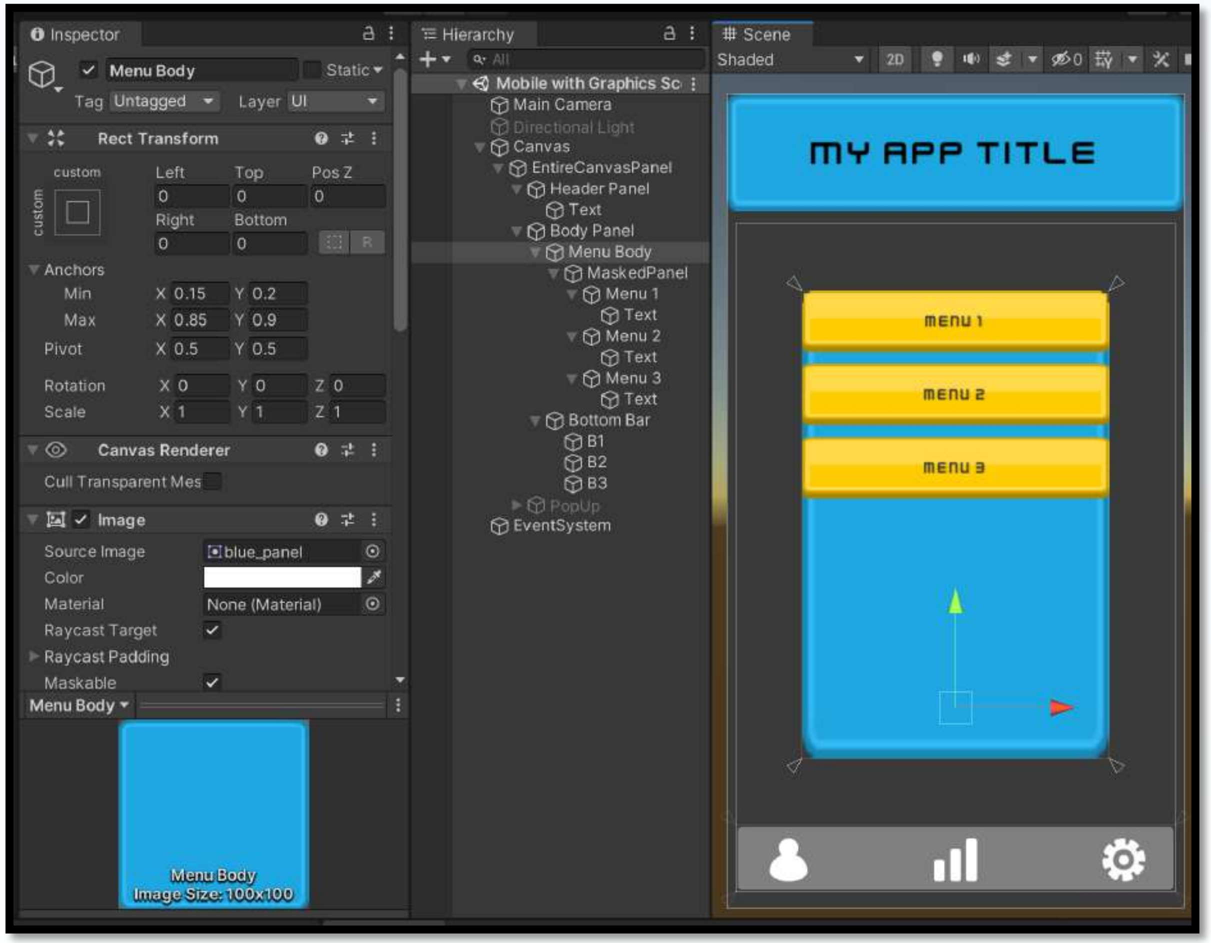
Blue_panel (Texture 2D) Impc

Open

Texture Type
Texture Shape
sRGB (Color Texture)
Alpha Source
Alpha Is Transparenc
Ignore PNG file gamm
Advanced
Non-Power of 2 ToNearest

Panels, Buttons, and Images

- Panels and Buttons have an Image component attached to them
  - You can change how they look
- In fact, you can add an image component to Rect Transforms, as well as use Image objects to act as Buttons and Panels
  - For Buttons, you need to add a Button Component



Inspector

Menu Body — Static

Tag: Untagged — Layer: UI

Rect Transform

custom

Left	Top	Pos Z
0	0	0
Right	**Bottom**	
0	0	

Anchors

Min	X 0.15	Y 0.2	
Max	X 0.85	Y 0.9	
Pivot	X 0.5	Y 0.5	
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Canvas Renderer

Cull Transparent Mes

Image

Source Image	blue_panel
Color	
Material	None (Material)
Raycast Target	✓
Raycast Padding	
Maskable	✓

Menu Body

Menu Body
Image Size: 100x100

Hierarchy

- Mobile with Graphics Sc
  - Main Camera
  - Directional Light
  - Canvas
    - EntireCanvasPanel
      - Header Panel
        - Text
      - Body Panel
        - Menu Body
          - MaskedPanel
            - Menu 1
              - Text
            - Menu 2
              - Text
            - Menu 3
              - Text
        - Bottom Bar
          - B1
          - B2
          - B3
    - PopUp
  - EventSystem

Scene

Shaded

Buttons with Images have States

- You can make buttons change their sprite given its state transitions: Sprite Swap
  - Highlighted
  - Pressed
  - Selected
  - Disabled

- And you can also just use the regular Color Tint option

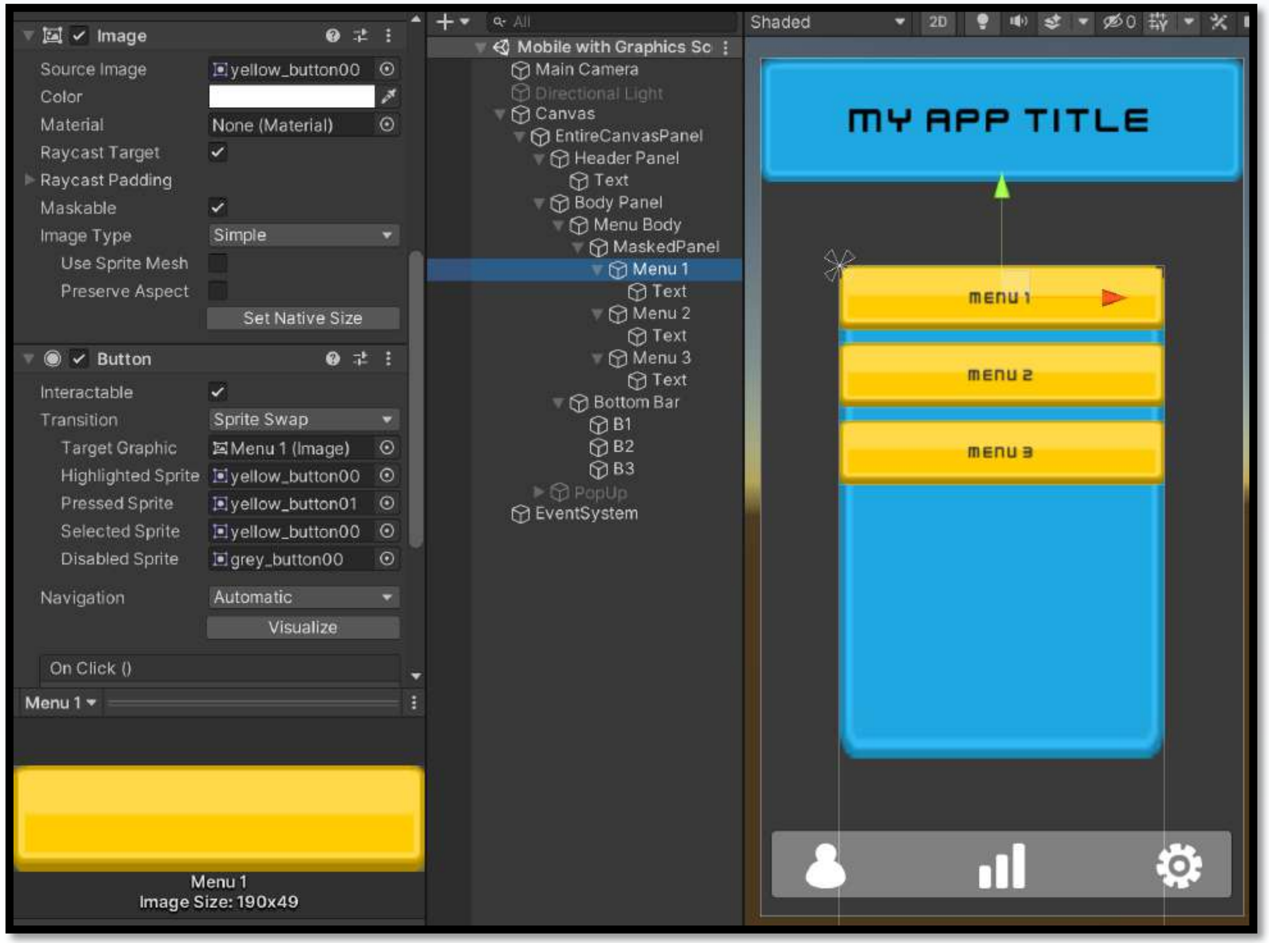


Image
Source Image: yellow_button00
Color
Material: None (Material)
Raycast Target: ✓
Raycast Padding
Maskable: ✓
Image Type: Simple
Use Sprite Mesh
Preserve Aspect
Set Native Size

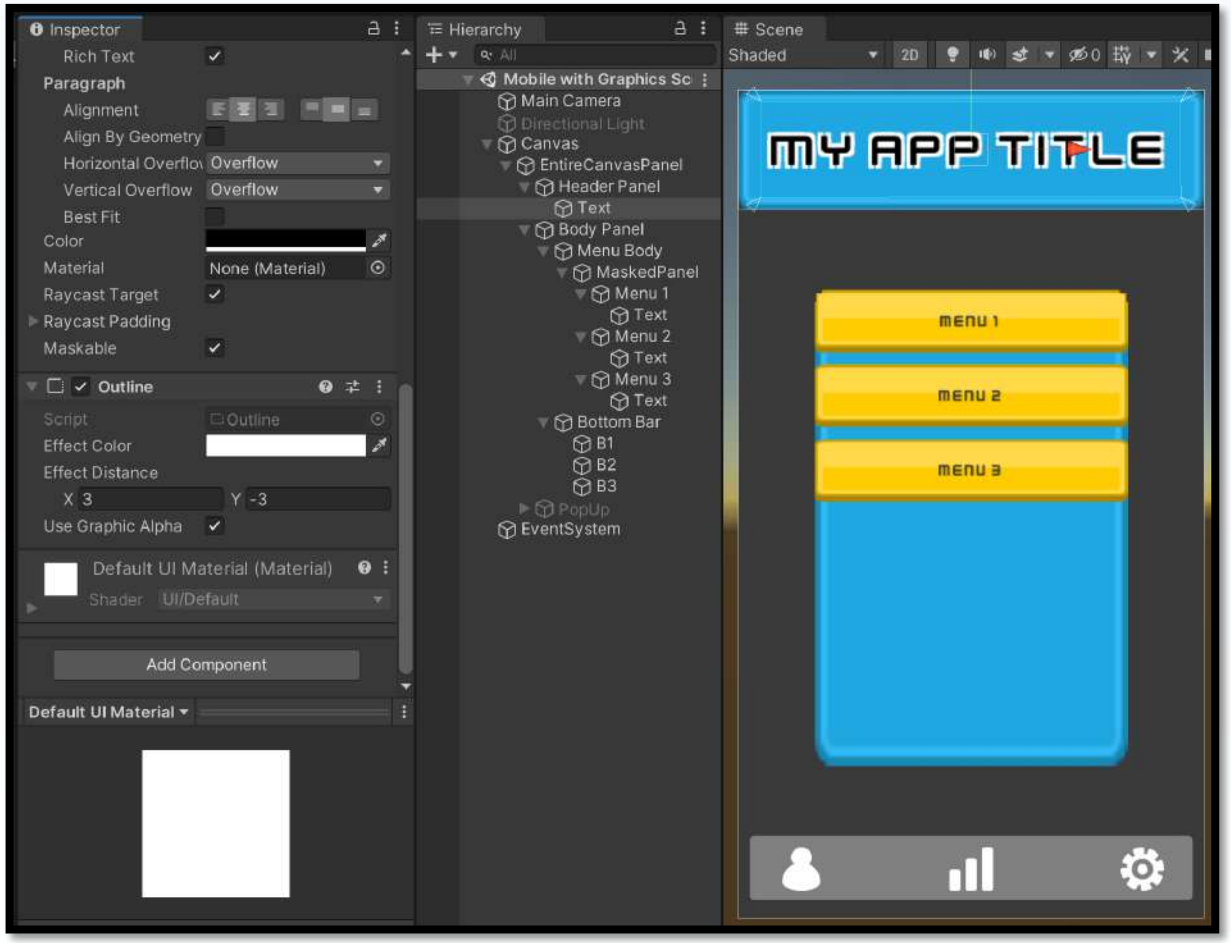
Button
Interactable: ✓
Transition: Sprite Swap
Target Graphic: Menu 1 (Image)
Highlighted Sprite: yellow_button00
Pressed Sprite: yellow_button01
Selected Sprite: yellow_button00
Disabled Sprite: grey_button00
Navigation: Automatic
Visualize

On Click ()

Menu 1
Image Size: 190x49

Outline

- There are other components that can help adding visuals to your prototype
- The most simple one (but also not that reliable) is the Outline
- Outlines the element with a given color and width (not robust)



Inspector

Rich Text ✓
Paragraph
Alignment
Align By Geometry
Horizontal Overflow: Overflow
Vertical Overflow: Overflow
Best Fit
Color
Material: None (Material)
Raycast Target ✓
Raycast Padding
Maskable ✓

✓ Outline
Script: Outline
Effect Color
Effect Distance: X 3 Y -3
Use Graphic Alpha ✓

Default UI Material (Material)
Shader: UI/Default

Add Component

Default UI Material

Questions?

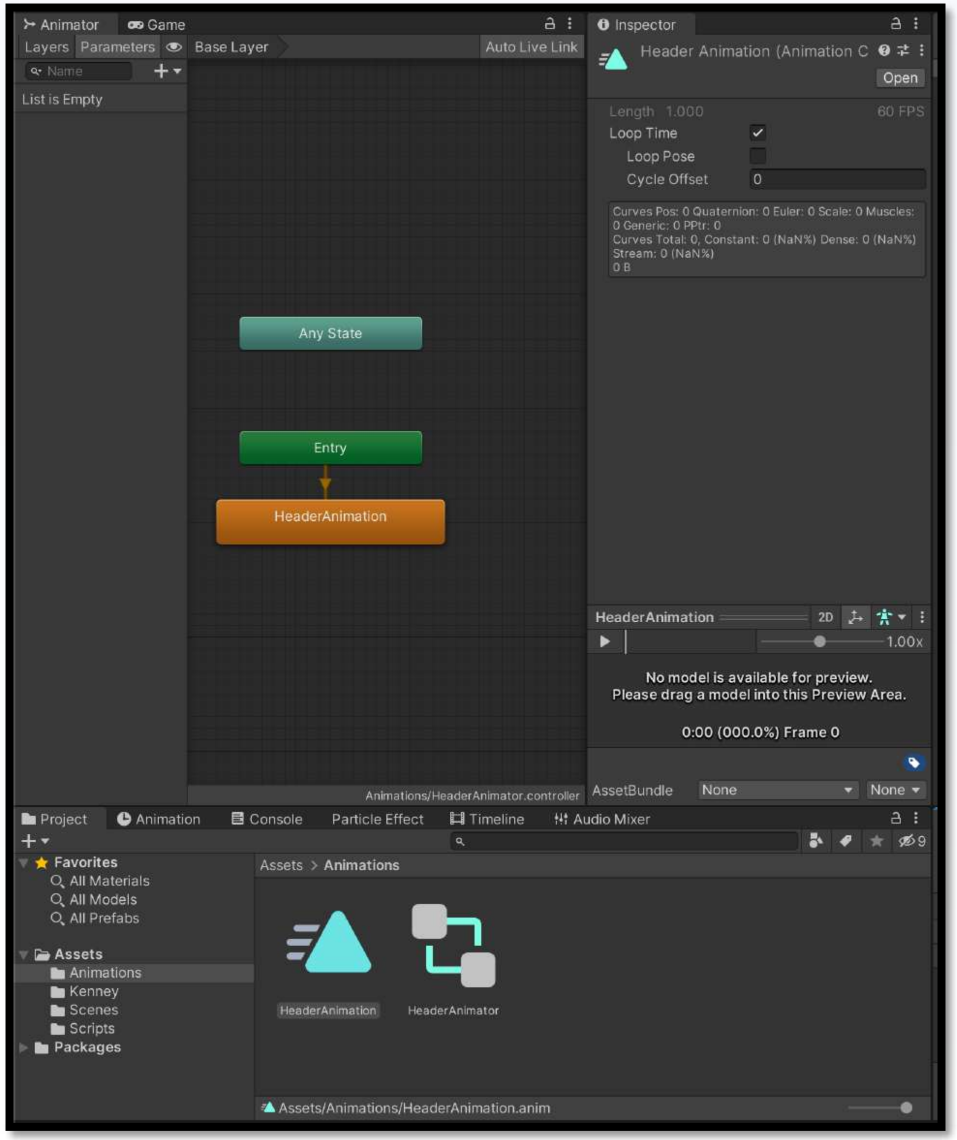
Any questions so far?

Should I repeat anything?

What about code!?

Adding life with Animations

- Unity allows the creation of animations within the engine
- We can use it to add life to the UI interface and also to control it
- To animate objects we need:
  - An Animator
  - Animations



Length 1.000 60 FPS
Loop Time ✓
Loop Pose
Cycle Offset 0

Curves Pos: 0 Quaternion: 0 Euler: 0 Scale: 0 Muscles: 0 Generic: 0 PPtr: 0
Curves Total: 0, Constant: 0 (NaN%) Dense: 0 (NaN%) Stream: 0 (NaN%)
0 B

No model is available for preview.
Please drag a model into this Preview Area.

0:00 (000.0%) Frame 0

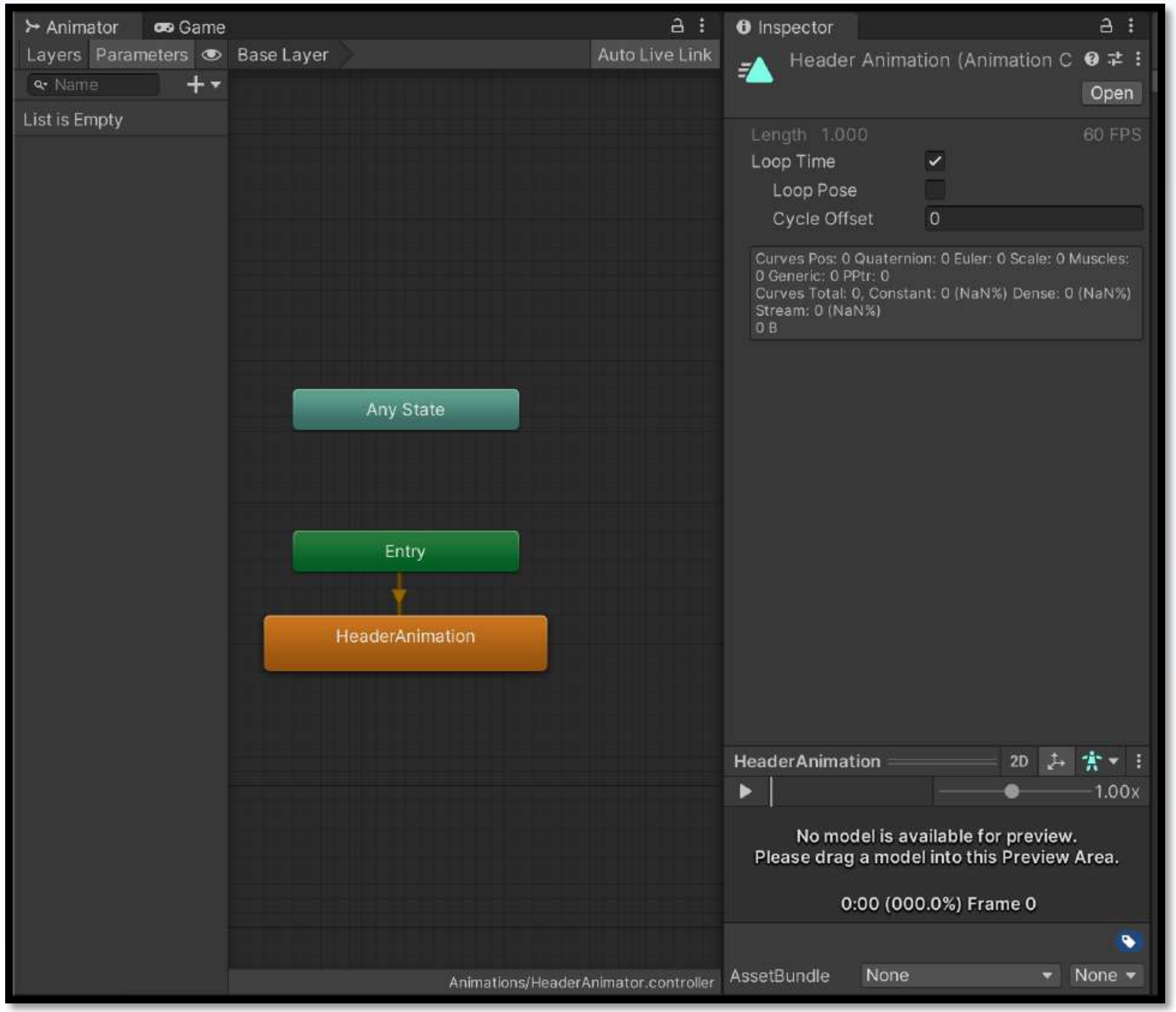
Animations/HeaderAnimator.controller

AssetBundle None None

Assets/Animations/HeaderAnimation.anim

Looping Animations and Animator Graph

- Animations have properties
  - Such as if it Loops or not
- The animator is a web of animations connected
  - Technically, a Graph
- Animations transition to others and from states
  - Any State, Entry, Exit



Length 1.000 60 FPS
Loop Time ✓
Loop Pose
Cycle Offset 0

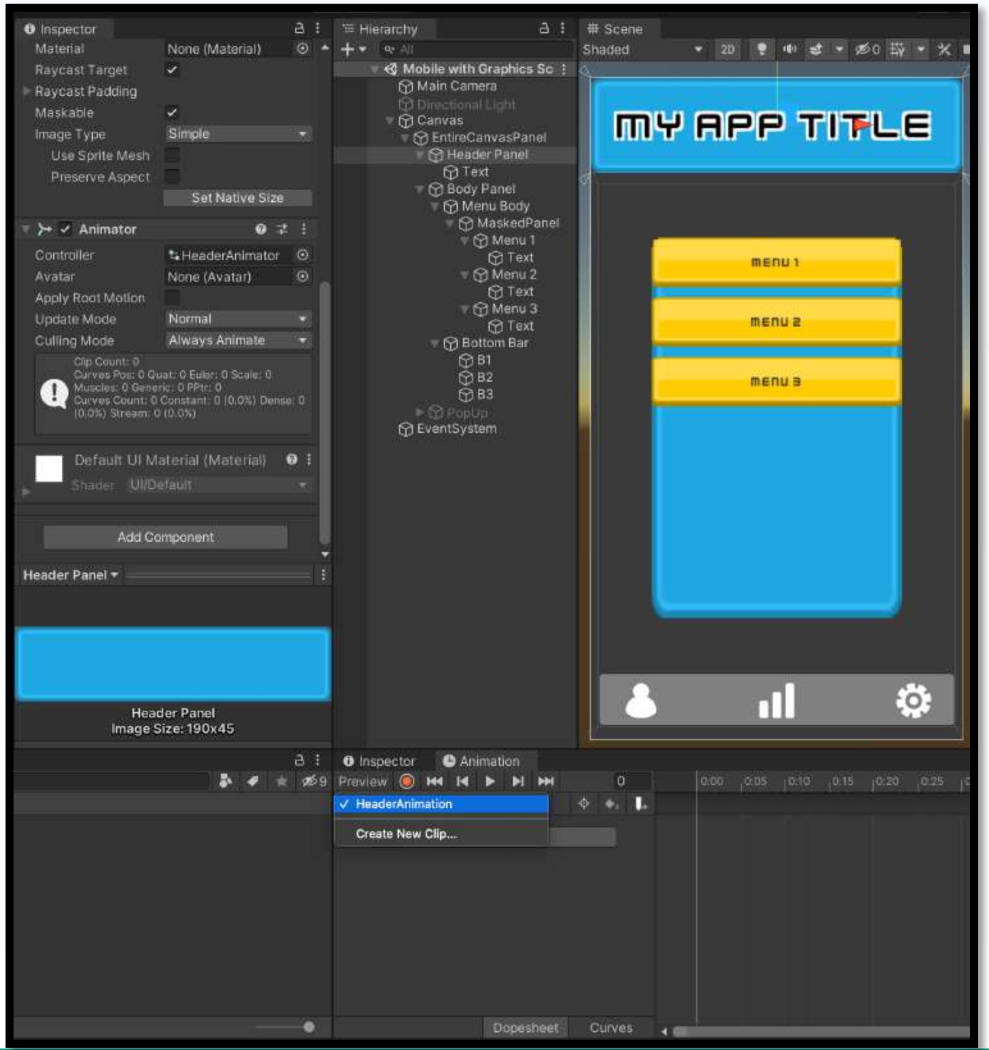
Curves Pos: 0 Quaternion: 0 Euler: 0 Scale: 0 Muscles: 0 Generic: 0 PPtr: 0
Curves Total: 0, Constant: 0 (NaN%) Dense: 0 (NaN%) Stream: 0 (NaN%)
0 B

HeaderAnimation 2D 1.00x
No model is available for preview.
Please drag a model into this Preview Area.
0:00 (000.0%) Frame 0

AssetBundle None None

Adding the Animator Component

- You need to add an Animator Component to the object that will be animated
  - All children objects are also subject to be animated by it
- The Animation Window allows the creation of animations
  - It already display all animations added to the Animator



Material: None (Material)
Raycast Target: ✓
Raycast Padding
Maskable: ✓
Image Type: Simple
Use Sprite Mesh
Preserve Aspect
Set Native Size

Animator ✓
Controller: HeaderAnimator
Avatar: None (Avatar)
Apply Root Motion
Update Mode: Normal
Culling Mode: Always Animate

Clip Count: 0
Curves Pos: 0 Quat: 0 Euler: 0 Scale: 0
Muscles: 0 Generic: 0 PPtr: 0
Curves Count: 0 Constant: 0 (0.0%) Dense: 0 (0.0%) Stream: 0 (0.0%)

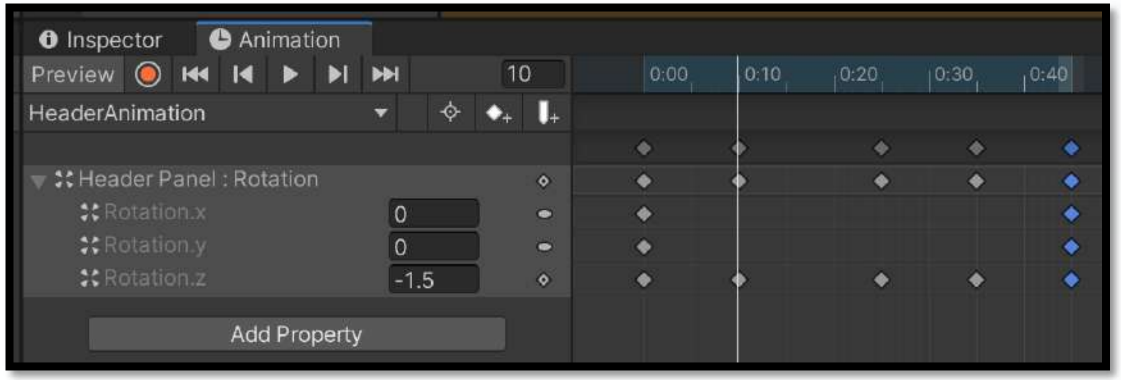
Default UI Material (Material)
Shader: UI/Default

Add Component

Header Panel
Image Size: 190x45

Making Animations

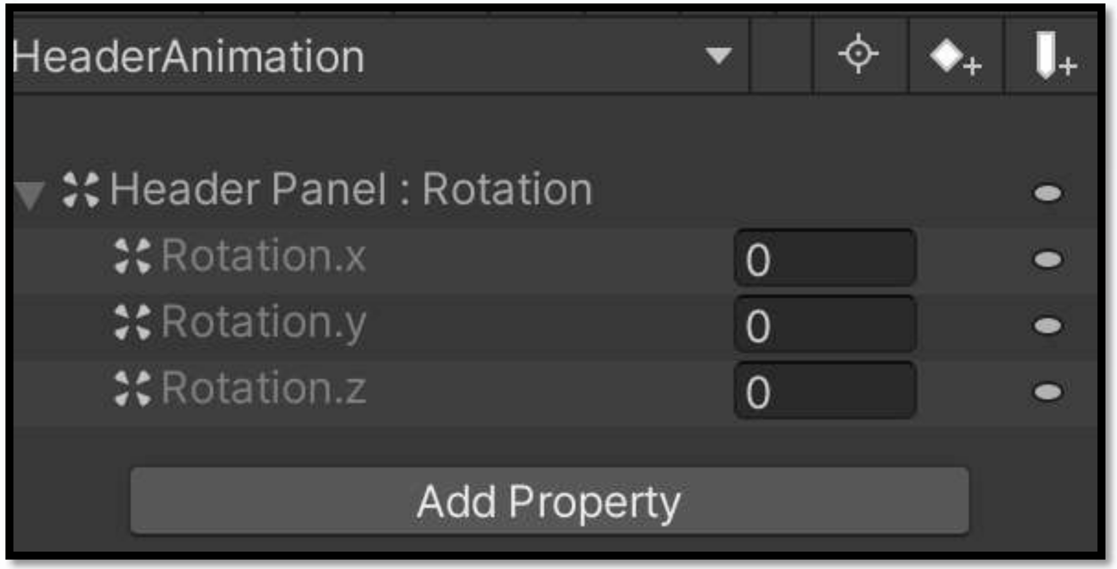
- Animations in Unity work as most of other animation software
  - Create keyframes
  - Keyframes are interpolated
- Access to the curve editor for how values are being interpolated
- Add properties to be animated
- Unity also has the record button:
  - Starts to record to the selected key frame all changes made to the object
  - It also detects changes in the chidren objects



Rotation.x	0
Rotation.y	0
Rotation.z	-1.5

Beware Renaming and Reordering!

- Unity's Animation system uses the game objects' name to control them
- If you rename or reorder them, the animation might break and lose the references
- Either avoid renaming or reordering, or fix the animations manually



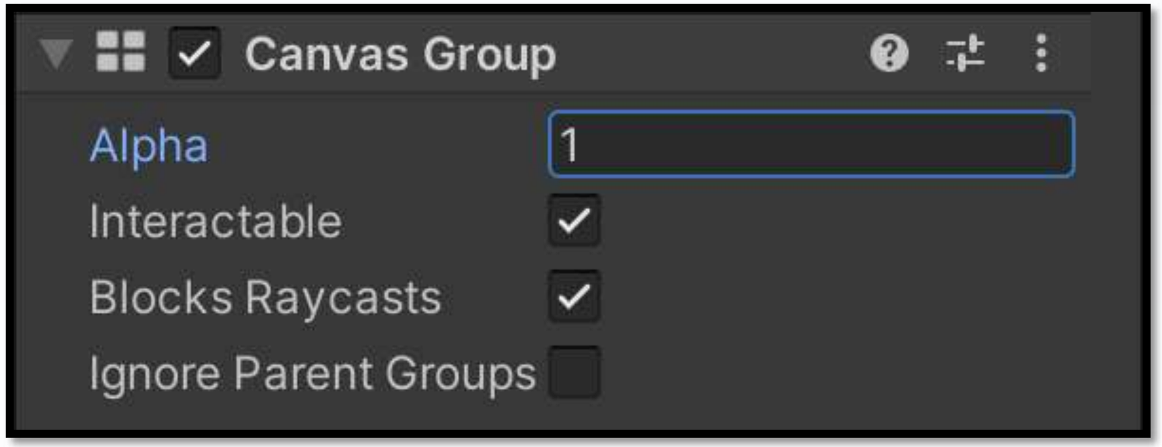
HeaderAnimation

Header Panel : Rotation

Rotation.x	0
Rotation.y	0
Rotation.z	0

Canvas Group

- Canvas Group is another useful component for the UI
- It allows controlling the entire hierarchy from one single component
- Can use it to animate the alpha of all elements at once

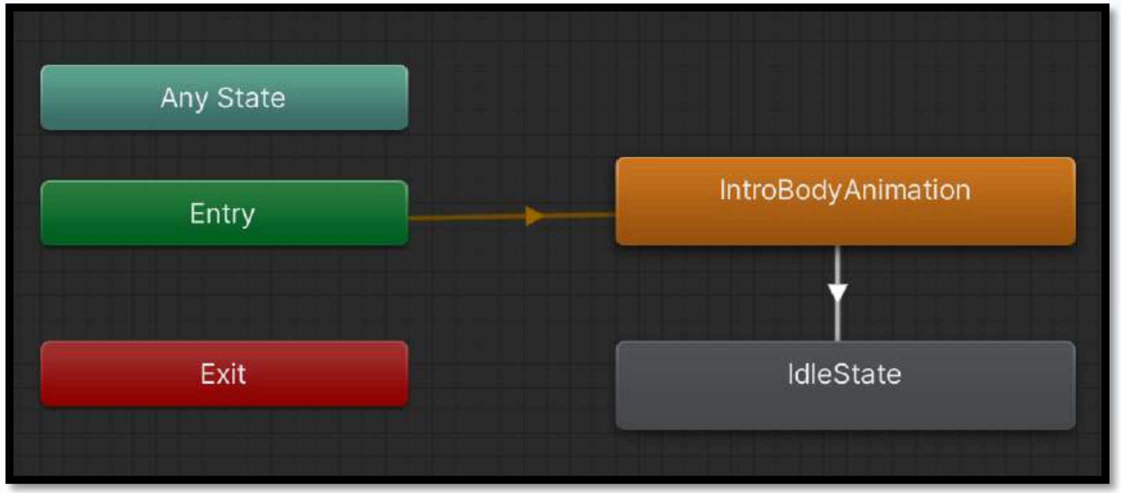


Canvas Group

Alpha	1
Interactable	✓
Blocks Raycasts	✓
Ignore Parent Groups	

Animator Transitions and Empty States

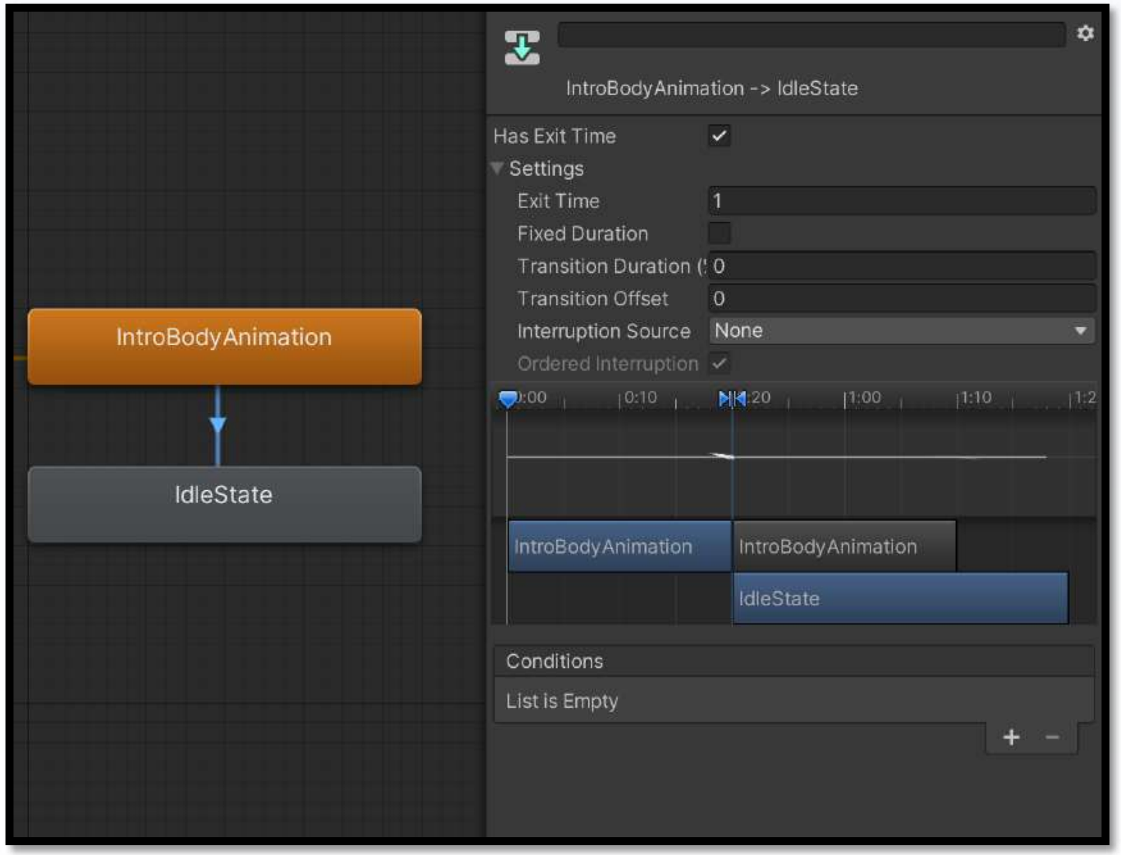
- The animator also allows for empty states
  - States without an animation
- This is particularly useful for idle states or inbetween states in which no animation or logic should be performed
- Transitions themselves control the conditions to be executed



Transitions

- Transition can be activated via conditions
  - Based on the Animator Parameters
  - Needs an ExitTime if there is no condition attached to it

- Can take over another animation
  - HasExitTime: take time before transitioning
  - Exit Time: normalized (1 = 100%)
  - Timeline can be used to manually adjust the transition



IntroBodyAnimation -> IdleState

Has Exit Time ✓
Settings
Exit Time 1
Fixed Duration
Transition Duration (: 0
Transition Offset 0
Interruption Source None
Ordered Interruption ✓

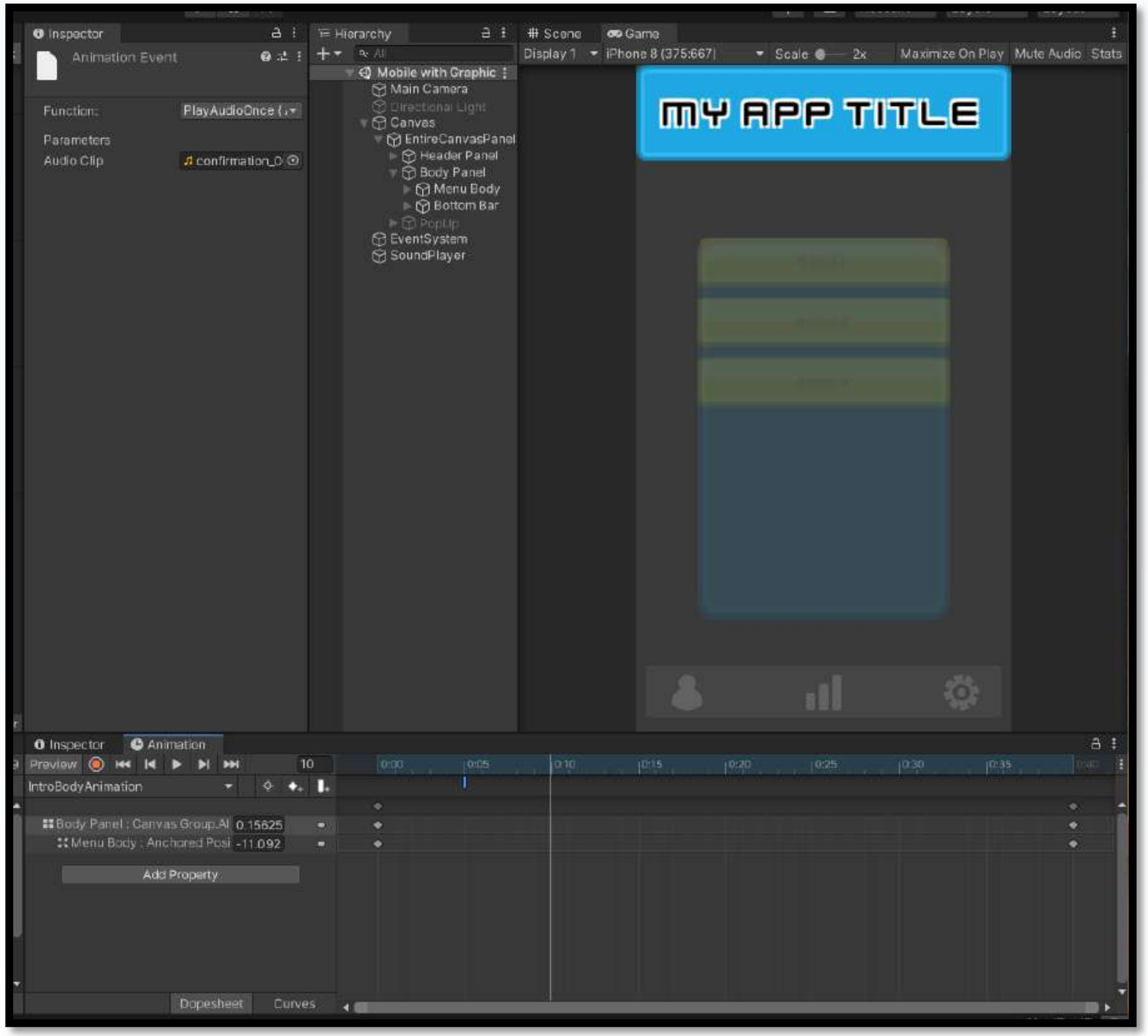
Conditions
List is Empty

Animations can Invoke Functions

- Animations can also trigger functions in Scripts
  - Scripts need to be attached to the Game Object holding the animator

- Interface is very simple
  - Only accepts functions with ONE parameter

- Cannot access outside scripts, except …


Animation Event

Function: PlayAudioOnce (

Parameters

Audio Clip: confirmation_0

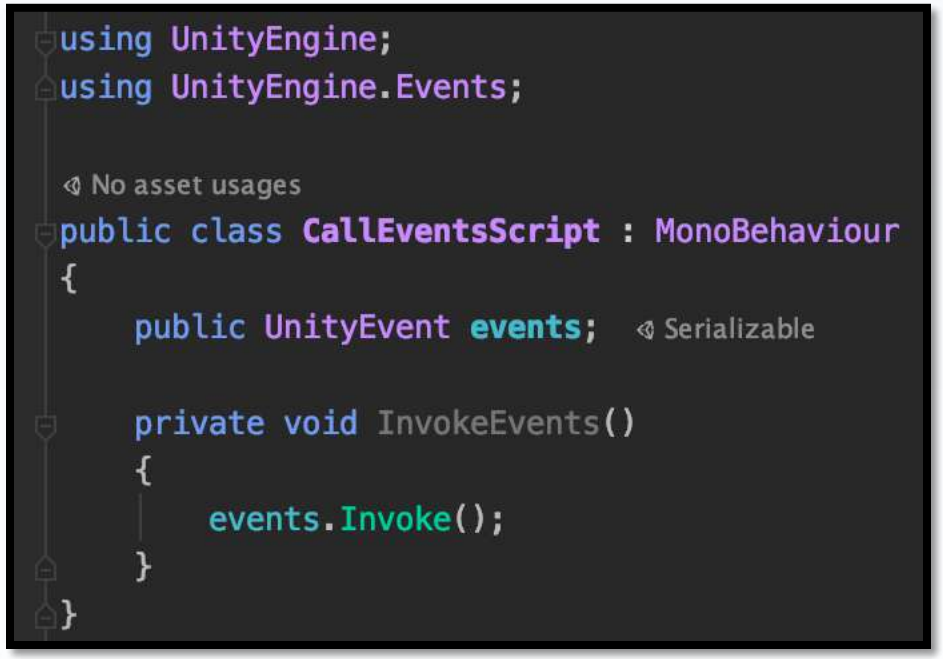
IntroBodyAnimation

Body Panel : Canvas Group.Al 0.15625

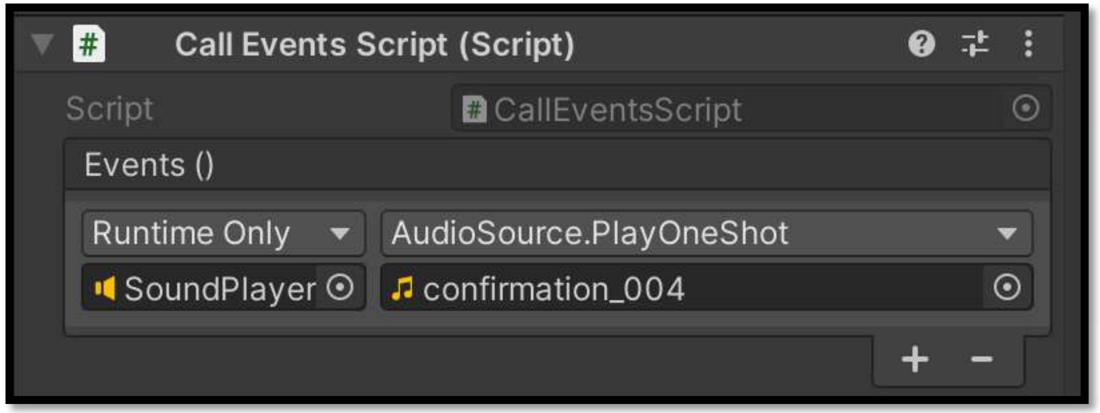
Menu Body : Anchored Posi -11.092

Unity Events

- Unity Events can mimic the interface from Unity Buttons
- Can hold references to other objects in the screen
- Can be invoked with a simple function call



```
using UnityEngine;
using UnityEngine.Events;

 No asset usages
public class CallEventsScript : MonoBehaviour
{
    public UnityEvent events;   Serializable

    private void InvokeEvents()
    {
        events.Invoke();
    }
}
```



Call Events Script (Script)

Script: CallEventsScript

Events ()

Runtime Only | AudioSource.PlayOneShot

SoundPlayer | confirmation_004

Questions?

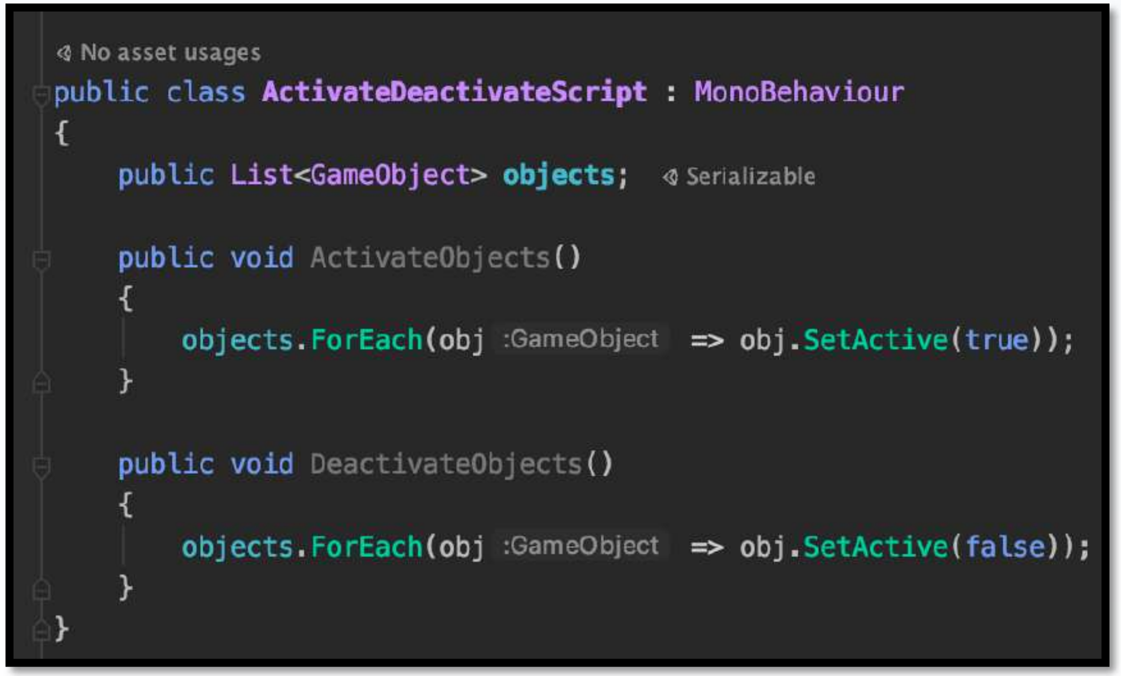
Any questions so far?

Should I repeat anything?

What about code [2]!?

Activate/Deactivate Script

- It is very common to have events or animations that also deactivate other (outside) objects
- This script holds a list of objects to be either activated/deactivate
- You cannot have more of these in the same object
  - Events will not recognize them

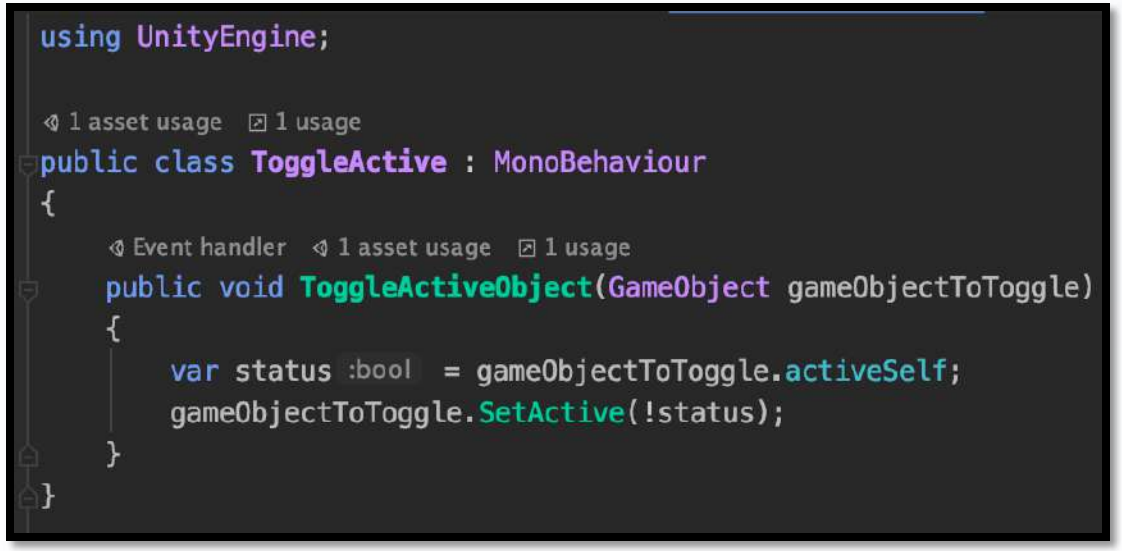


```
No asset usages
public class ActivateDeactivateScript : MonoBehaviour
{
    public List<GameObject> objects;   Serializable

    public void ActivateObjects()
    {
        objects.ForEach(obj :GameObject  => obj.SetActive(true));
    }

    public void DeactivateObjects()
    {
        objects.ForEach(obj :GameObject  => obj.SetActive(false));
    }
}
```

Toggle Object

- Sometimes we do not know the status of a game object, but we need to turn it on/off using the same button

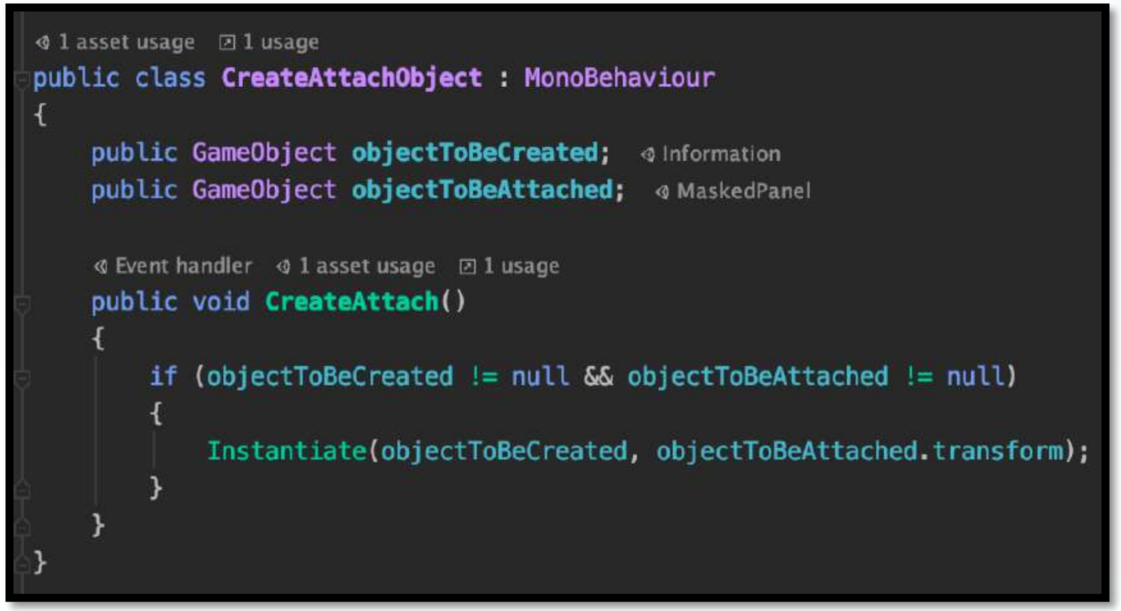
- The following code activate/deactive a game object based on its current status



```
using UnityEngine;

 1 asset usage   1 usage
public class ToggleActive : MonoBehaviour
{
     Event handler   1 asset usage   1 usage
    public void ToggleActiveObject(GameObject gameObjectToToggle)
    {
        var status :bool  = gameObjectToToggle.activeSelf;
        gameObjectToToggle.SetActive(!status);
    }
}
```

Create & Attach

- Most Apps have a functionality to add elements to a list
  - We already saw how to make a list
- Use a button and Prefabs to add more items to the list
  - Just need to make them a child object of the list

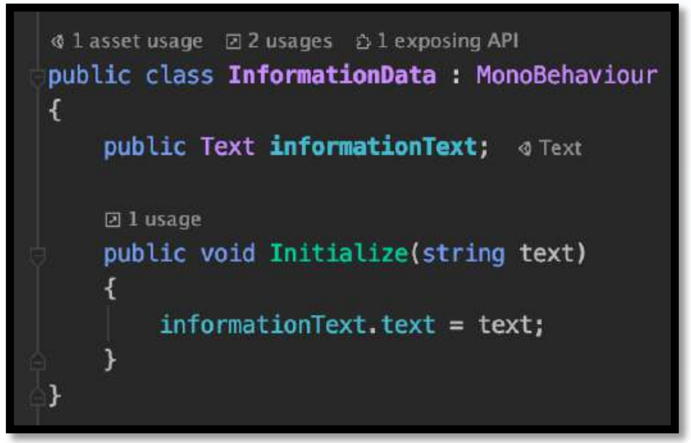


```
1 asset usage   1 usage
public class CreateAttachObject : MonoBehaviour
{
    public GameObject objectToBeCreated;   Information
    public GameObject objectToBeAttached;   MaskedPanel

    Event handler   1 asset usage   1 usage
    public void CreateAttach()
    {
        if (objectToBeCreated != null && objectToBeAttached != null)
        {
            Instantiate(objectToBeCreated, objectToBeAttached.transform);
        }
    }
}
```

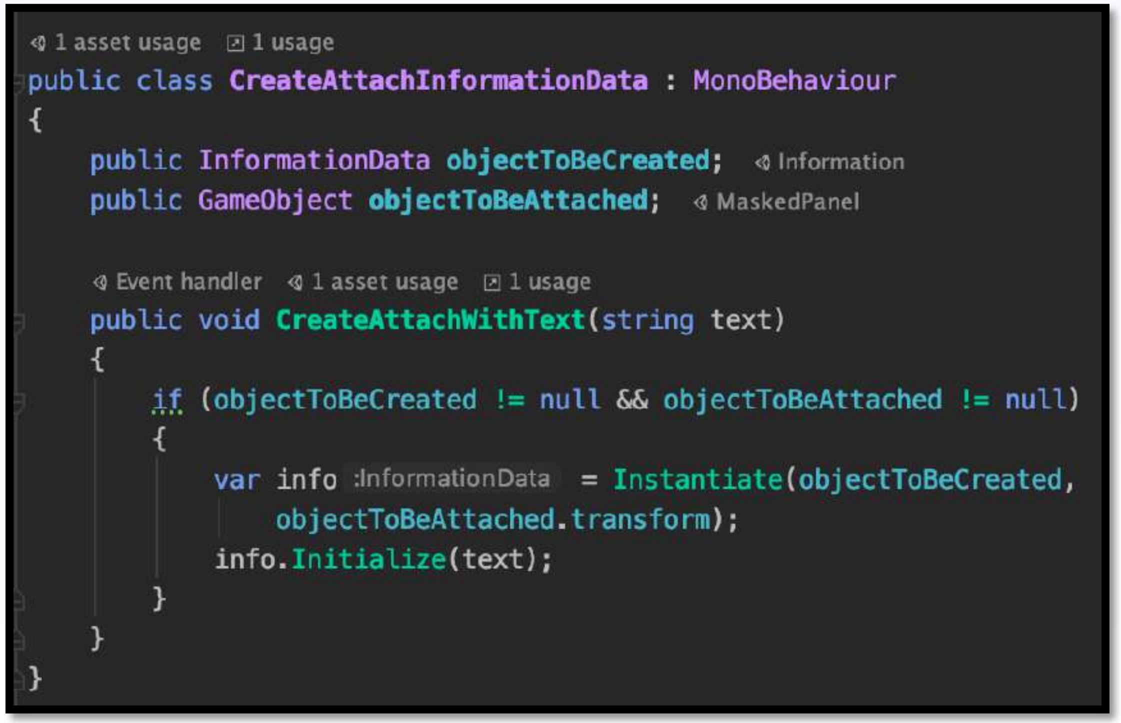
What If I Want to Initialize the Values

- You need a specific Create & Attach script that expects a Game Object with a script that will initialize the values

- Suppose this:



```
1 asset usage   2 usages   1 exposing API
public class InformationData : MonoBehaviour
{
    public Text informationText;  Text

    1 usage
    public void Initialize(string text)
    {
        informationText.text = text;
    }
}
```

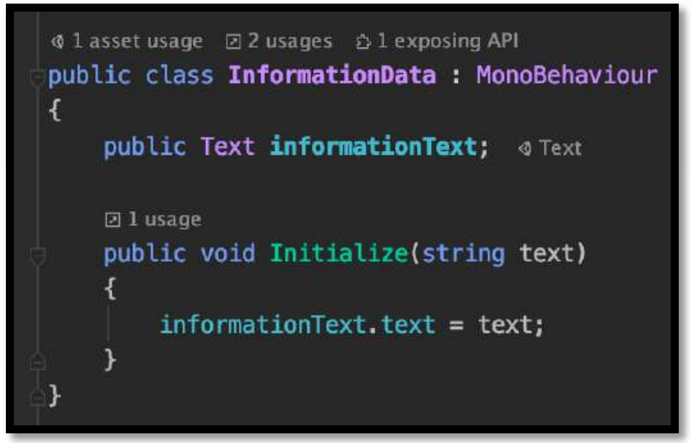


```
1 asset usage   1 usage
public class CreateAttachInformationData : MonoBehaviour
{
    public InformationData objectToBeCreated;  Information
    public GameObject objectToBeAttached;  MaskedPanel

    Event handler   1 asset usage   1 usage
    public void CreateAttachWithText(string text)
    {
        if (objectToBeCreated != null && objectToBeAttached != null)
        {
            var info :InformationData = Instantiate(objectToBeCreated,
                objectToBeAttached.transform);
            info.Initialize(text);
        }
    }
}
```

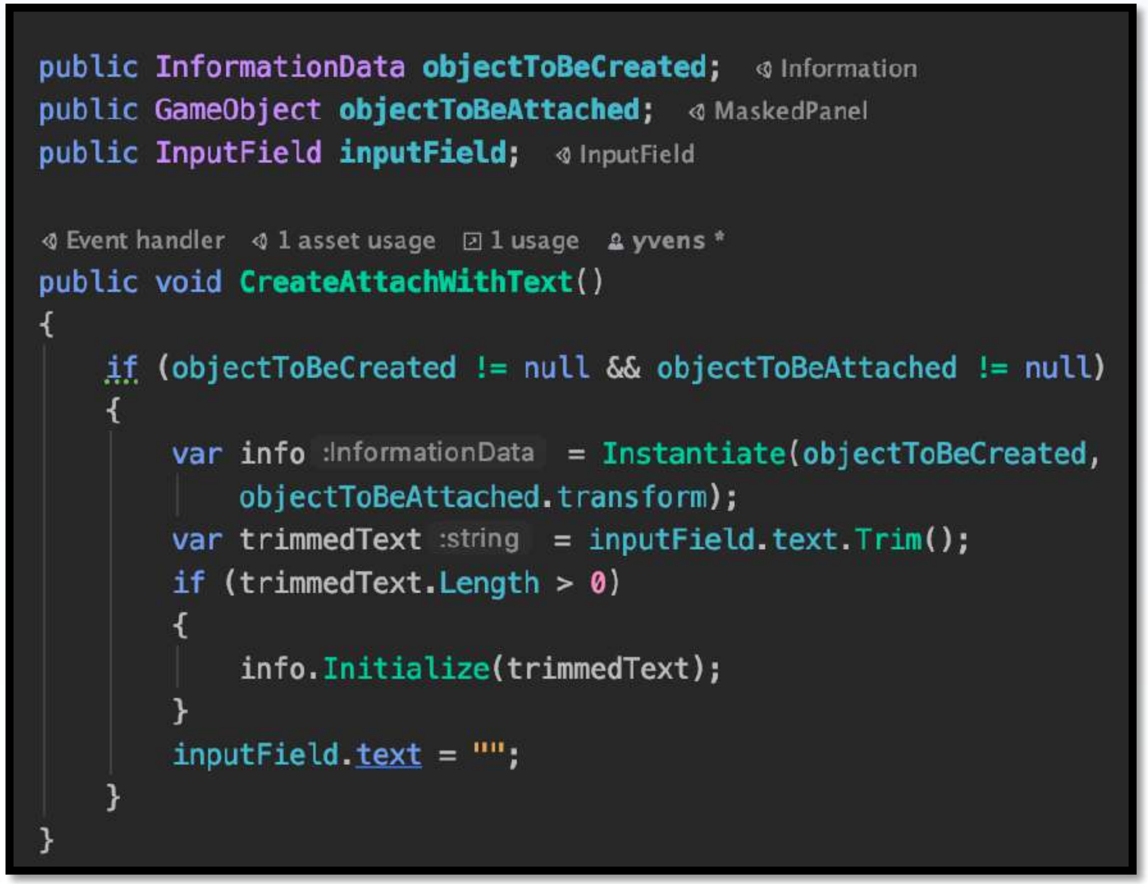
Initialize the Values using Input Field

- You can edit the script to also read data from other sources, such as InputFields or Dropdown

- Suppose this:



```
1 asset usage   2 usages   1 exposing API
public class InformationData : MonoBehaviour
{
    public Text informationText;   Text

    1 usage
    public void Initialize(string text)
    {
        informationText.text = text;
    }
}
```

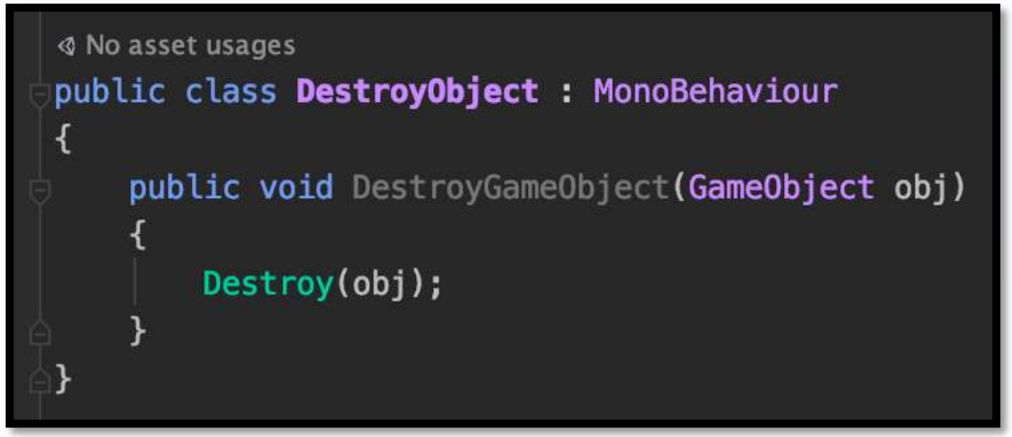


```
public InformationData objectToBeCreated;   Information
public GameObject objectToBeAttached;   MaskedPanel
public InputField inputField;   InputField

Event handler   1 asset usage   1 usage   yvens *
public void CreateAttachWithText()
{
    if (objectToBeCreated != null && objectToBeAttached != null)
    {
        var info :InformationData = Instantiate(objectToBeCreated,
            objectToBeAttached.transform);
        var trimmedText :string = inputField.text.Trim();
        if (trimmedText.Length > 0)
        {
            info.Initialize(trimmedText);
        }
        inputField.text = "";
    }
}
```

Destroying/Removing Objects

- For elements in a list, simply destroying them will be enough to remove them from the list

- Since they are already in order, you can use this without bothering about organizing the list or sorting it



```
No asset usages
public class DestroyObject : MonoBehaviour
{
    public void DestroyGameObject(GameObject obj)
    {
        Destroy(obj);
    }
}
```

Questions?

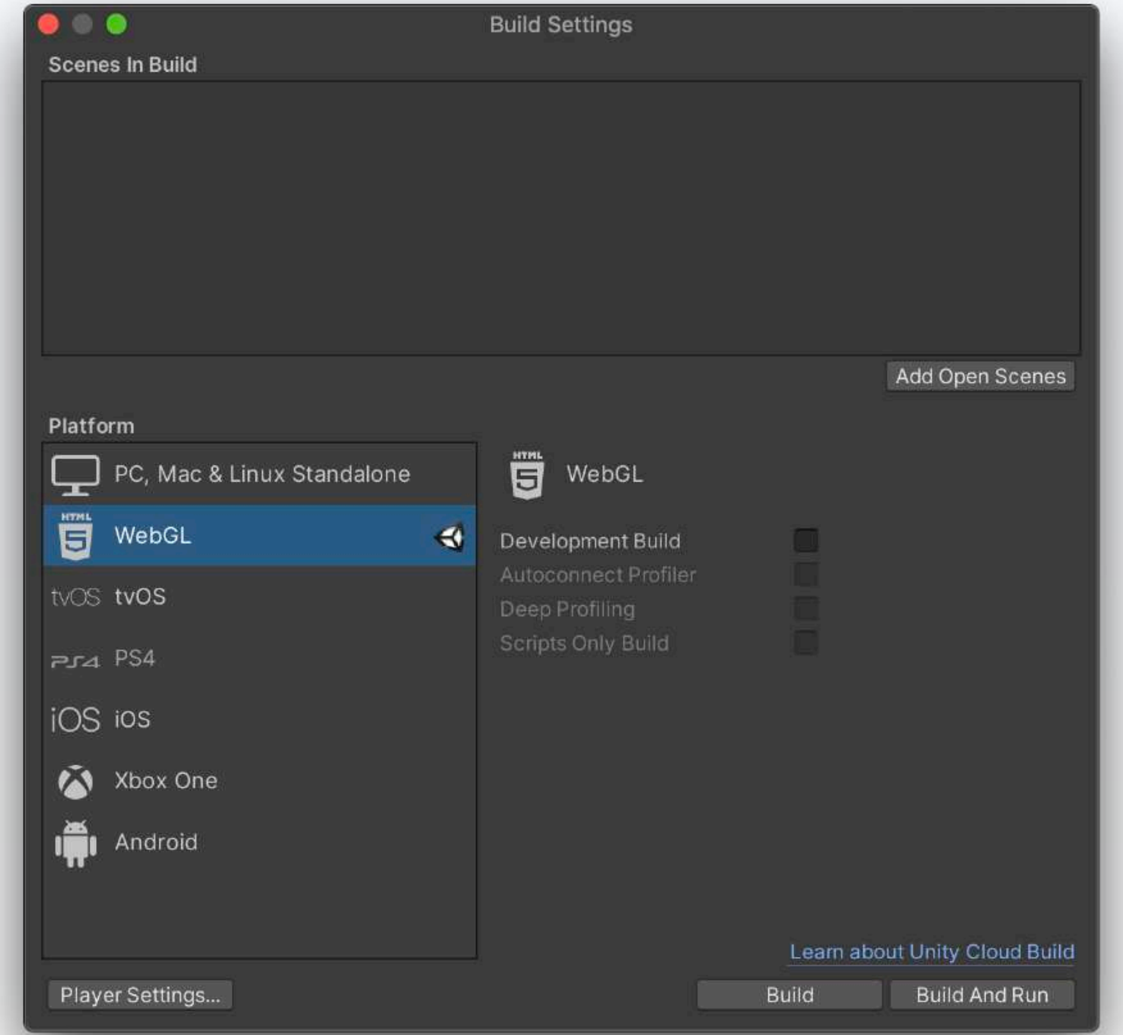
Questions?

This should cover most of the UI basic functionalities

What else do you want to know?

Preparing the Web Build

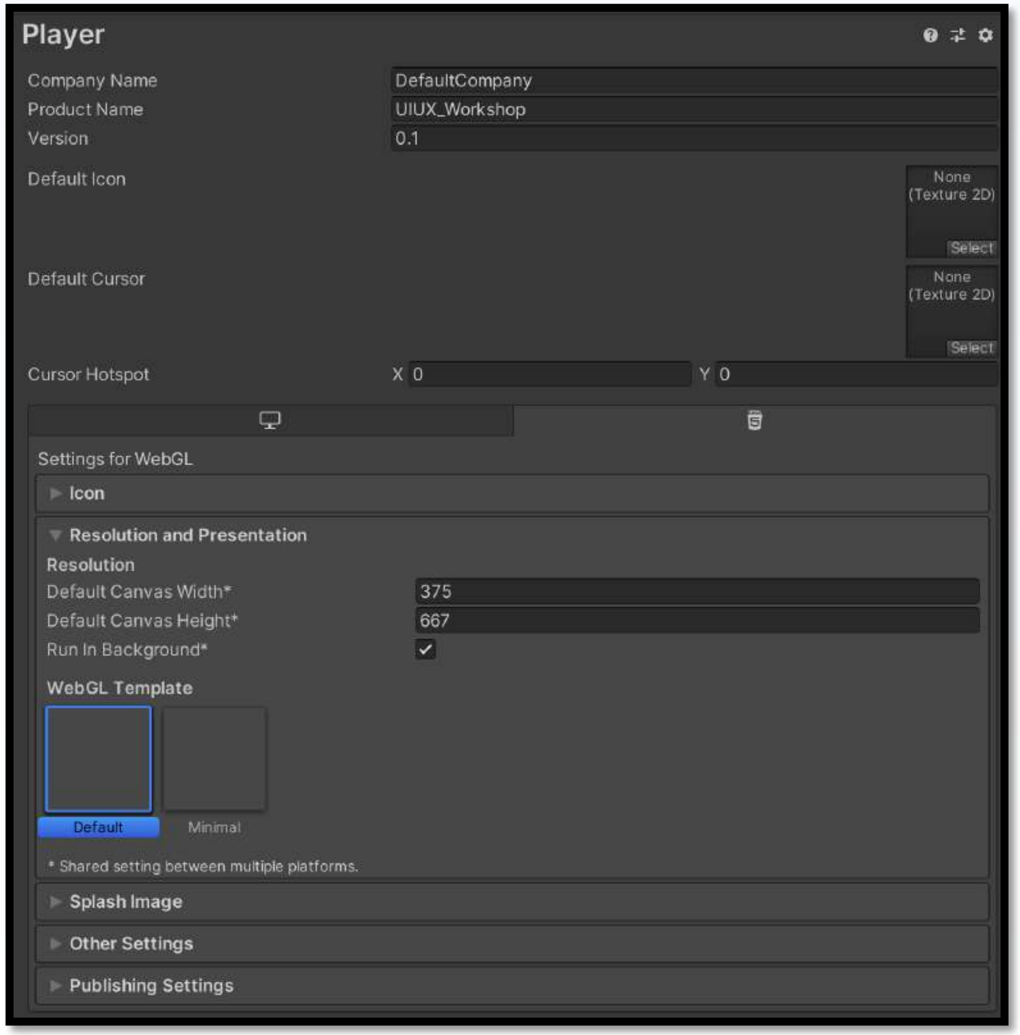
- Change the build type to Web
- You must download the Web Package from the Unity Hub
- You can download the other types if you want to deploy for a device or another OS



Adjust the Resolution in the Player Settings

- Change the default canvas width and default canvas height

- Many host websites (and Unity build itself) will use these values



Player

Company Name	DefaultCompany
Product Name	UIUX_Workshop
Version	0.1
Default Icon	None (Texture 2D)
Default Cursor	None (Texture 2D)
Cursor Hotspot	X 0 Y 0

Settings for WebGL

Icon

Resolution and Presentation

Resolution

Default Canvas Width*	375
Default Canvas Height*	667
Run In Background*	✓

WebGL Template

Default Minimal

* Shared setting between multiple platforms.

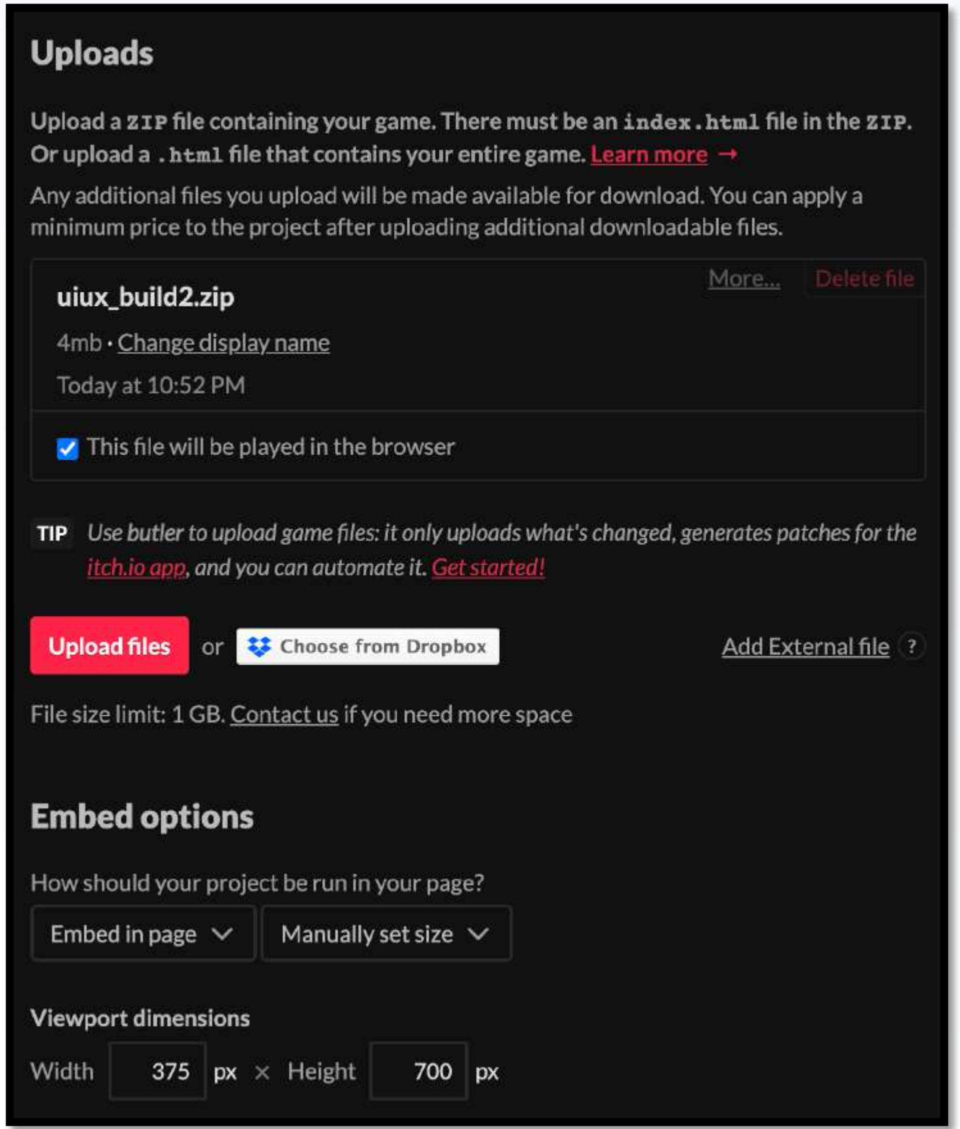
Splash Image

Other Settings

Publishing Settings

Hosting on Itch.IO

- Free hosting and easy access
- Host and play the unity build directly (upload as a .zip file)
  - Simply zip the entire build folder
- Make sure to adjust the viewport dimensions
  - I had a few issues with it



Uploads

Upload a `ZIP` file containing your game. There must be an `index.html` file in the `ZIP`. Or upload a `.html` file that contains your entire game. Learn more →

Any additional files you upload will be made available for download. You can apply a minimum price to the project after uploading additional downloadable files.

More... Delete file

uiux_build2.zip

4mb · Change display name

Today at 10:52 PM

☑ This file will be played in the browser

TIP *Use butler to upload game files: it only uploads what's changed, generates patches for the itch.io app, and you can automate it. Get started!*

Upload files or Choose from Dropbox Add External file ?

File size limit: 1 GB. Contact us if you need more space

Embed options

How should your project be run in your page?

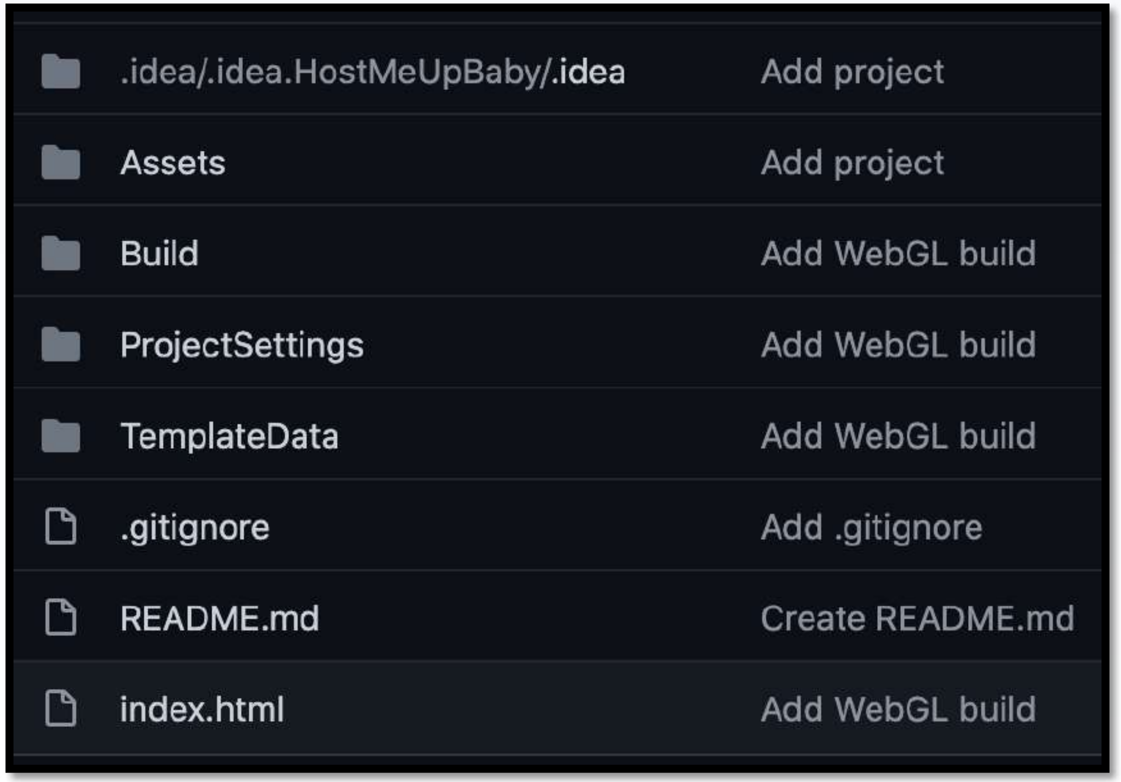
Embed in page ∨ Manually set size ∨

Viewport dimensions

Width 375 px × Height 700 px

Hosting on GitHub

- It is also possible to host your app directly from the repository folder:
- https://github.com/YvensFaos/HostMeUpBaby
- Tutorial: https://medium.com/@aboutin/host-unity-games-on-github-pages-for-free-2ed6b4d9c324



.idea/.idea.HostMeUpBaby/.idea	Add project
Assets	Add project
Build	Add WebGL build
ProjectSettings	Add WebGL build
TemplateData	Add WebGL build
.gitignore	Add .gitignore
README.md	Create README.md
index.html	Add WebGL build

You can acess this project:

Repository: https://tinyurl.com/y4h7u4wq

Direct Download: https://tinyurl.com/y2rfy6en

More Tips on Unity

- Two texts I wrote about techniques for faster prototyping in Unity:
  - https://tinyurl.com/yx8w6do5
  - https://tinyurl.com/y5839ac9

- Scene Transition using Animations
  - https://www.youtube.com/watch?v=CE9VOZivb3I&ab_channel=Brackeys

- Unity Timeline:
  - https://docs.unity3d.com/Packages/com.unity.timeline@1.5/manual/index.html

- Unity UI:
  - https://docs.unity3d.com/2020.1/Documentation/Manual/UIHowTos.html