31263 / 32004
Intro to GamesDevelopment
Week 10

Dr. William Raffe
william.raffe@uts.edu.au
Senior Lecturer
School of Software, FEIT

Overview

- **Unity UI Overview**
- **Canvas**
  - **Overlay vs Camera vs World Space**
  - **Element Ordering**
- **Rect Transform**
  - **Anchors and Pivots**
  - **Stretching**
- **Multiple Screen Resolution**
- **Text**
- **Images**
- **Buttons**

Unity UI Overview

- Dedicated UI Designer and Developer job roles exist in game companies.
  - Menu screens are the first thing a player sees
  - In-game UI’s convey most non-environment information
  - <u>A bad UI is as deadly to a game as bad in-game controls!</u>

- The UI capabilities of Unity are vast and flexible
  - They have come a long way since early Unity days

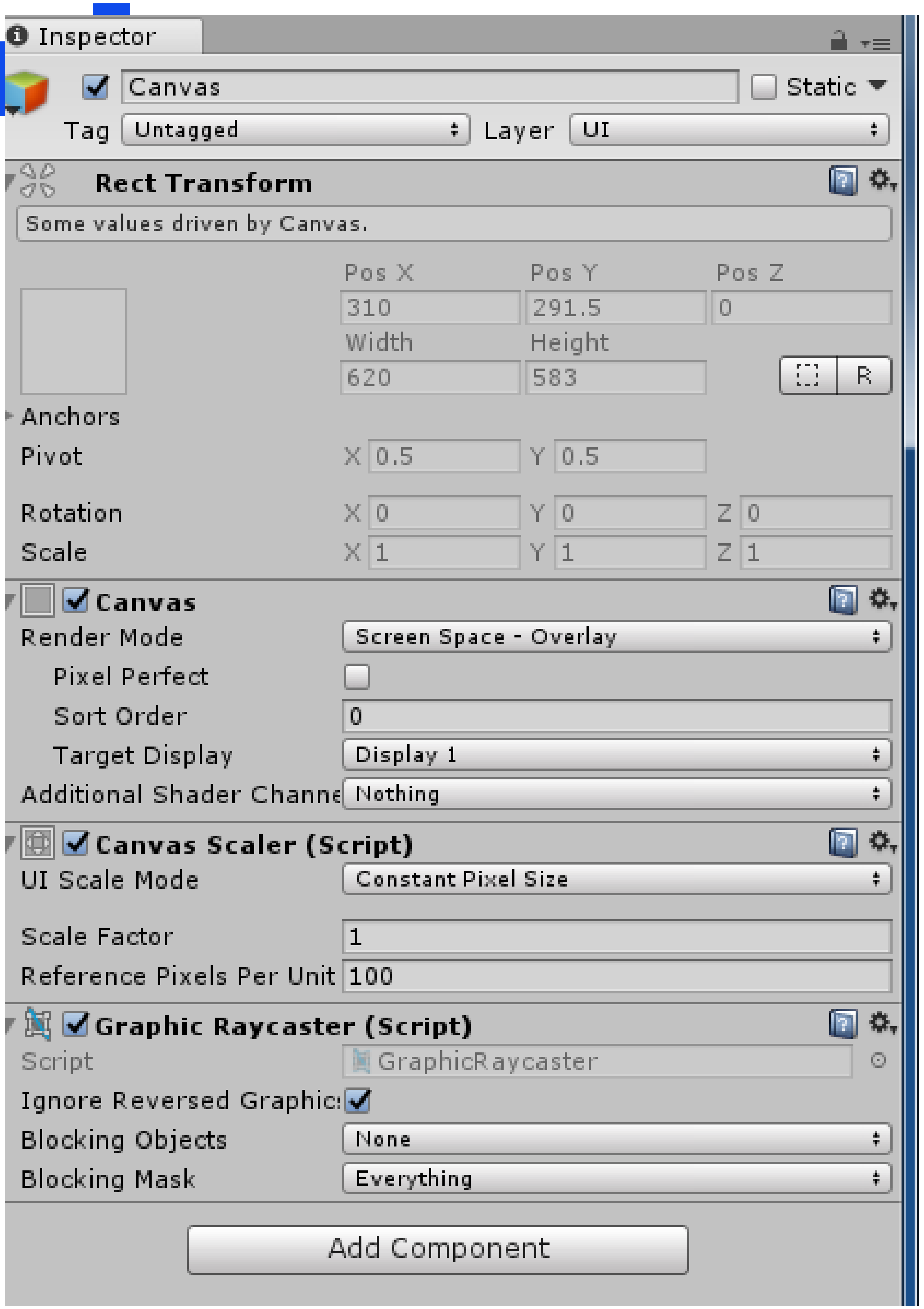
- Like **animation** and **audio**, we will only be scraping the surface

Canvas

- **Everything in a UI is placed under a GameObject with a Canvas component**
  - I.e. all UI elements are a child of a Canvas
  - Will be auto created if any other UI element is created and one doesn't exist.

- **One scene can have multiple canvases**
  - Best to minimize quantity where possible

- Creating a canvas will also create an EventSystem.
  - Handles UI Input
  - Mostly can be ignored for basic UI's


Inspector
Canvas — Static
Tag: Untagged — Layer: UI

Rect Transform
Some values driven by Canvas.

	Pos X	Pos Y	Pos Z
	310	291.5	0
	Width	Height	
	620	583	

Anchors
Pivot: X 0.5, Y 0.5
Rotation: X 0, Y 0, Z 0
Scale: X 1, Y 1, Z 1

Canvas
Render Mode: Screen Space - Overlay
Pixel Perfect: ☐
Sort Order: 0
Target Display: Display 1
Additional Shader Channe: Nothing

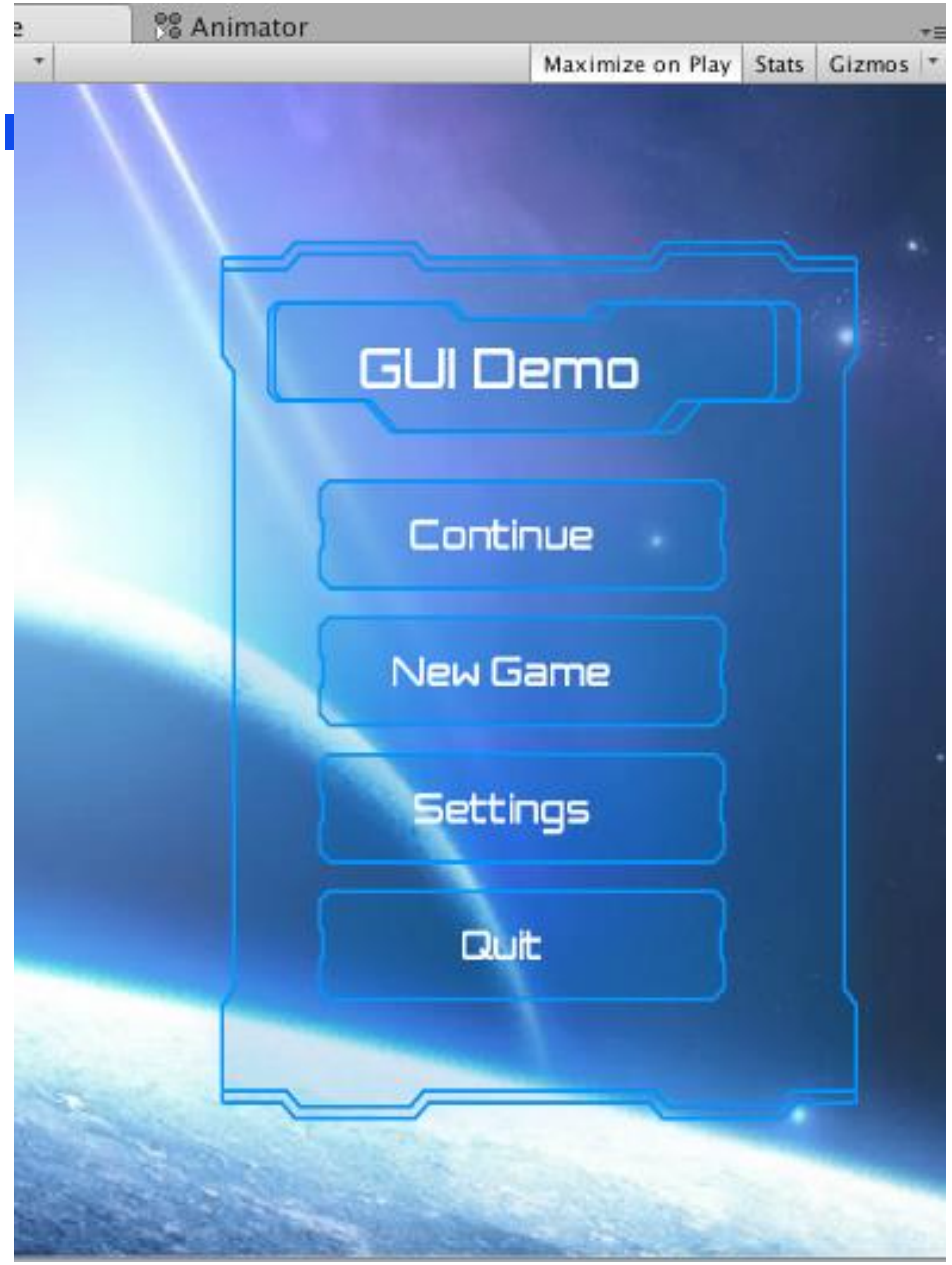
Canvas Scaler (Script)
UI Scale Mode: Constant Pixel Size
Scale Factor: 1
Reference Pixels Per Unit: 100

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

Canvas – Screen Space - Overlay

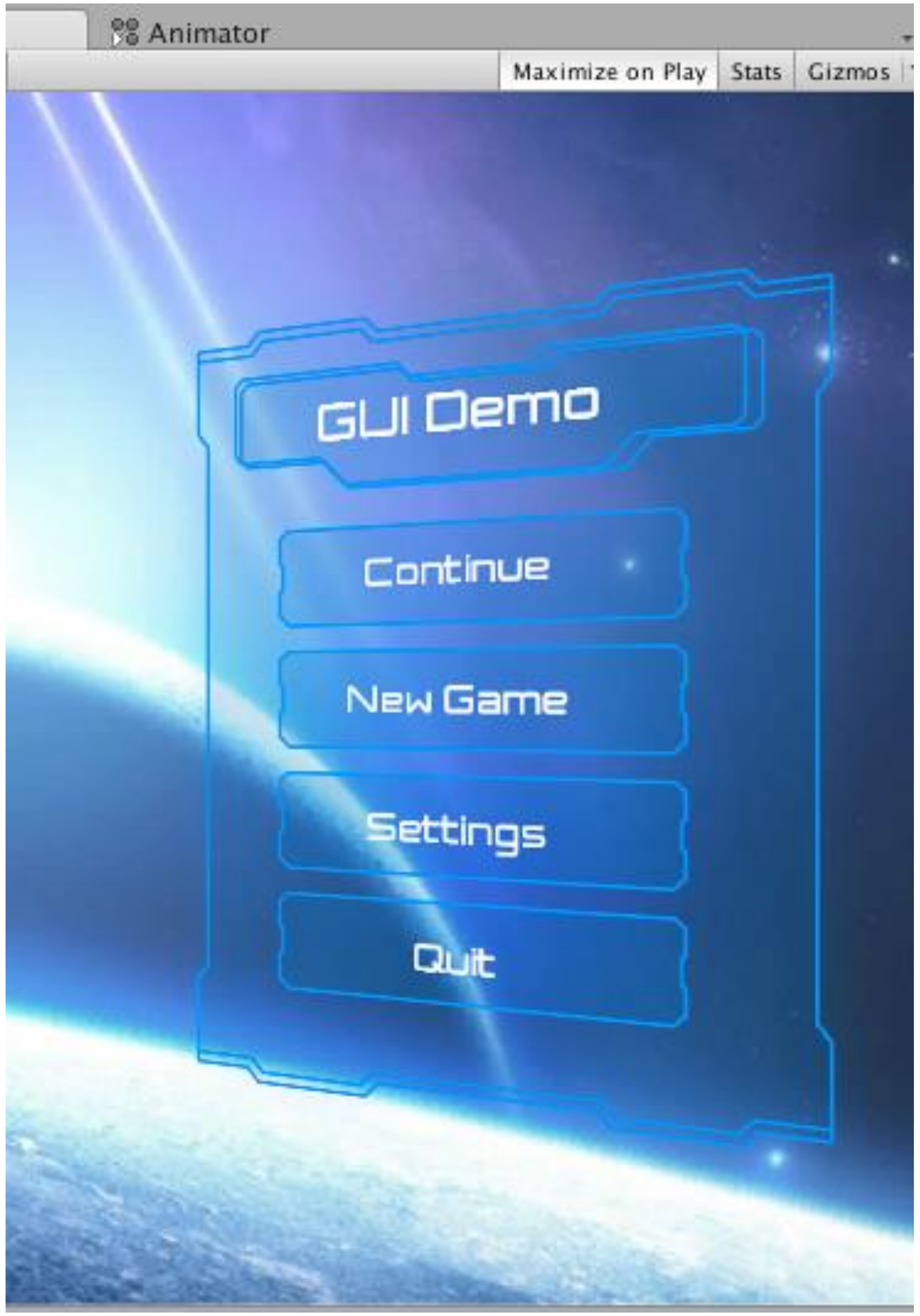
- The canvas covers the **entire screen**.
  - I.e. has the same dimensions as the screen it is viewed on.

- Has no depth
  - Everything is flat (no perspective)
  - Z coordinate is meaningless
  - Rotation around y axis is meaningless



Canvas – Screen Space - Camera

- The canvas covers **one camera.**
  - Has same dimensions as camera view.
- <u>If the camera is in Perspective mode</u>
  - Then the canvas will have perspective.
- Otherwise quite similar to Screen space - Overlay

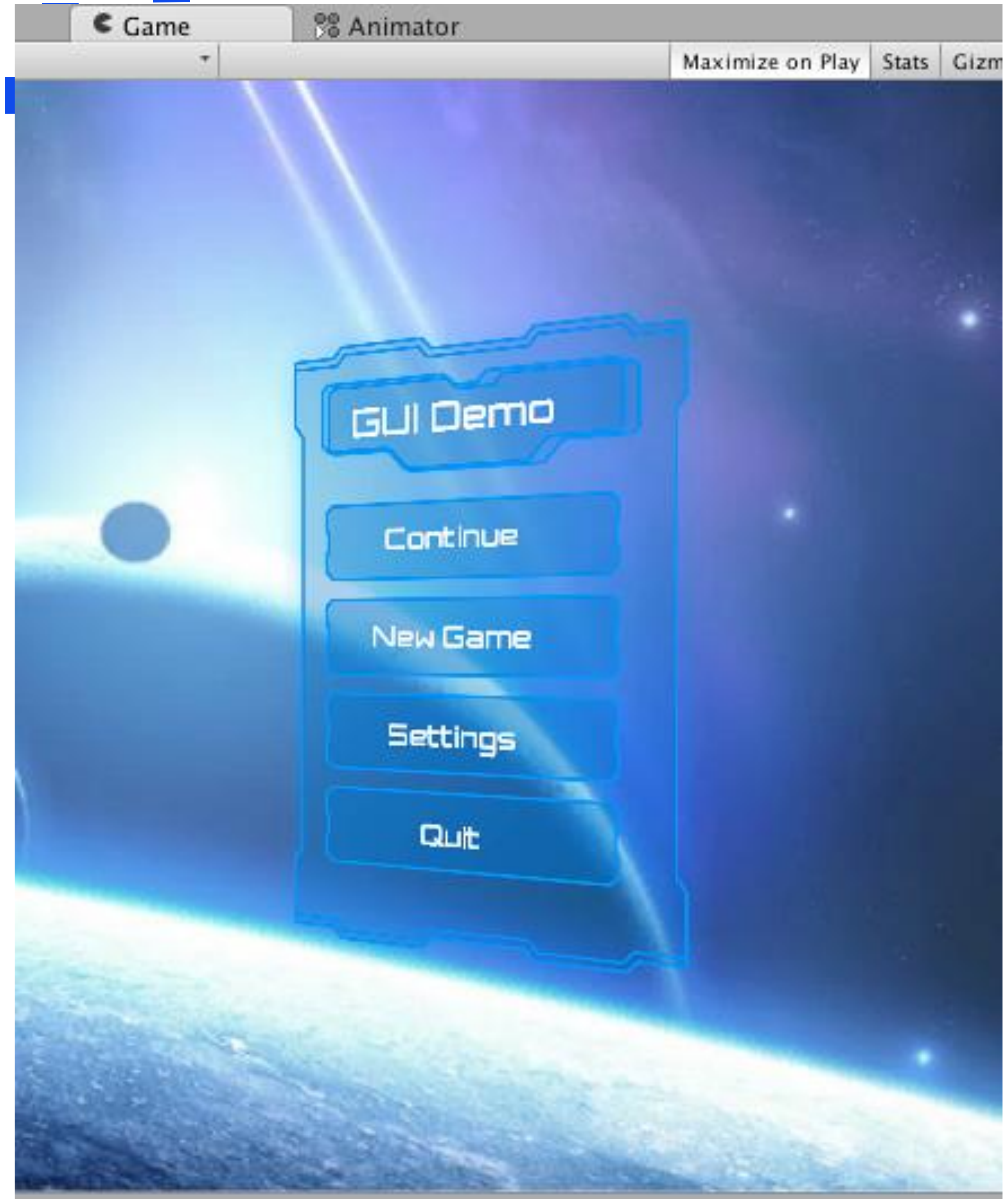


Canvas – Screen Space - Camera

- Occupies a specific spot in **world space**.
  - Just like every other game object in the scene

- Allows for embedding text and icons in the game environment.

- Warning: Because it is not an overlay, <u>the canvas won't be aware of different screen dimensions.</u>
  - Can lead to text that is either too small or that is blurry
  - Not recommended to use as replacement to Screen Space - Overlay



Element Ordering

- Which UI element appears in front, which is behind?

- Canvas:
  - Sort Order – Order of rendering for all Canvas gameobjects
  - Lower values **rendered first** (i.e. rendered behind)

- Within a Canvas:
  - Objects higher in the Hierarchy Window are **rendered first** (i.e. behind lower objects)
  - Parent objects are **rendered first** (i.e. behind child objects)

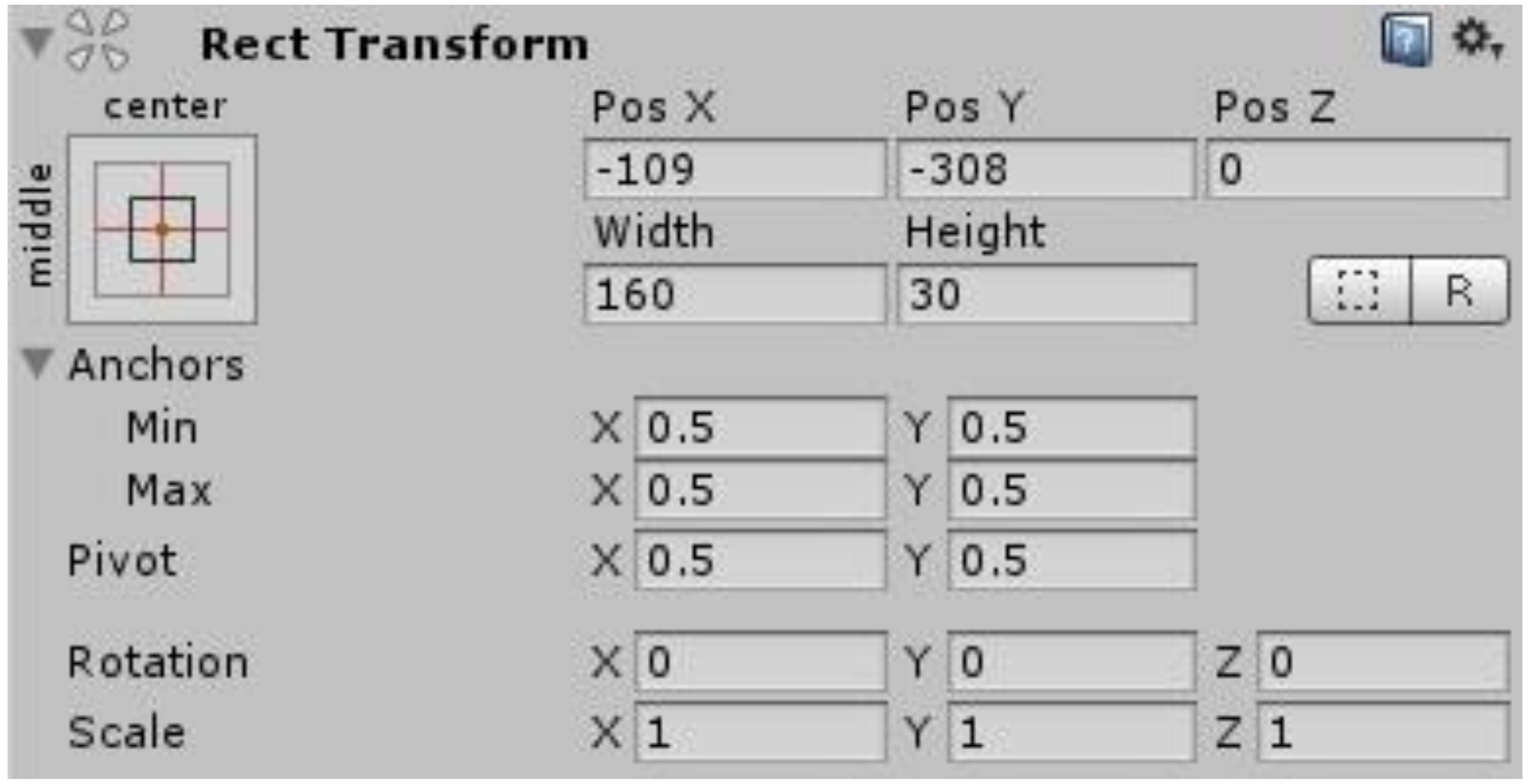
Interaction Ordering

- E.g. If a UI gameobject has a Button component then it can be clicked
- <u>If another UI gameobject is lower in the hierarchy, it will receive the click and block the button.</u>
- EXCEPTION – Child images/text will not stop a parent from being clicked, even though they appear in front of the parent.
  - If both parent and children have interaction (e.g. both are Buttons), the child lowest in the hierarchy will always block the others

Rect Transform

- UI gameobjects do not have a Transform component.
  - They have a **RectTransform** component (which is a sub class of Transform)

- More controls for relative positioning
  - Think HTML/CSS layouts



Rect Transform

center / middle

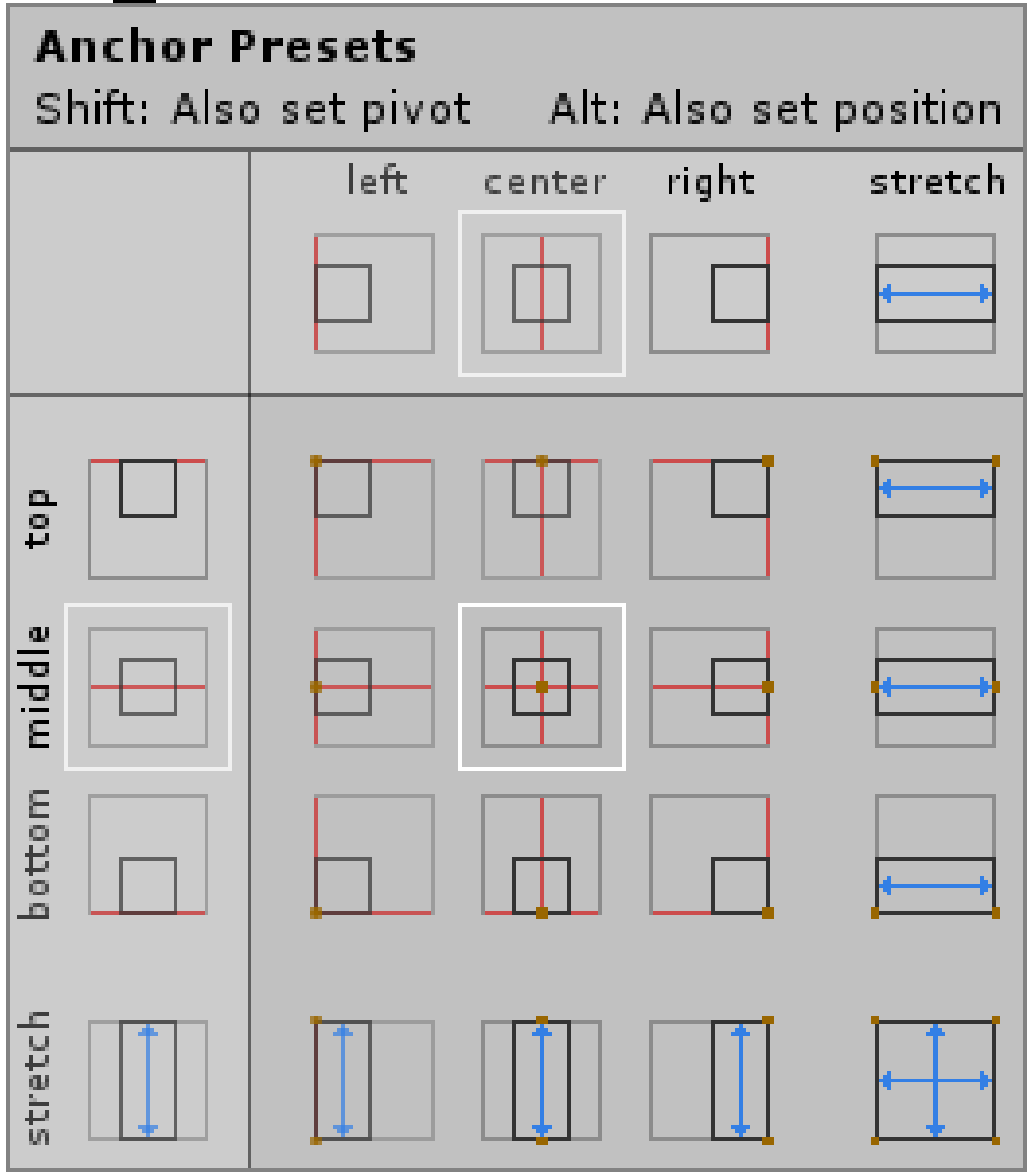
	Pos X	Pos Y	Pos Z
	-109	-308	0
	Width	Height	
	160	30	R

Anchors

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

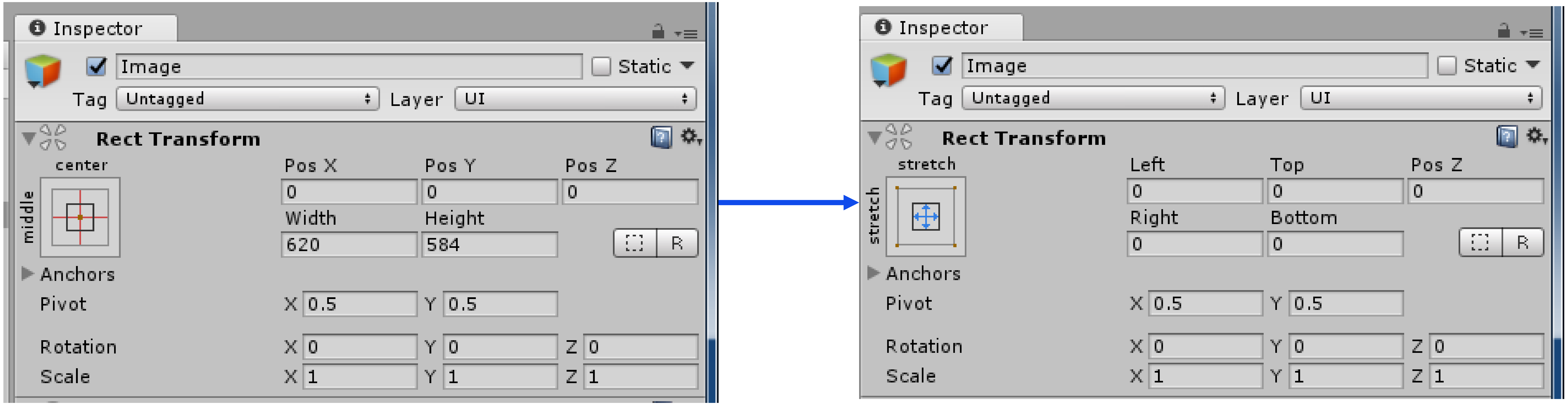
Anchors and Pivots

- Determines relative positioning
- **Anchor** – what part of the screen / parent object to attach to.
  - If that part of the screen / parent moves, the object will moves
- **Pivot** – Where the "centre" of the UI element is
  - Pos X, Y, Z, Width, Height, Rotation values all start at the pivot.



Stretching

- Splits the pivot into 2 or 4 and assigns it to the anchors
  - No more Pos X, Y, Z -> Now [distance from] Left, Right, Top, Down
- If any anchor moves, the RectTransform will **move and scale**



Inspector
Image ☑ — Static ☐
Tag: Untagged — Layer: UI
Rect Transform
center / middle
Pos X: 0 — Pos Y: 0 — Pos Z: 0
Width: 620 — Height: 584
Anchors
Pivot: X 0.5 — Y 0.5
Rotation: X 0 — Y 0 — Z 0
Scale: X 1 — Y 1 — Z 1

Inspector
Image ☑ — Static ☐
Tag: Untagged — Layer: UI
Rect Transform
stretch / stretch
Left: 0 — Top: 0 — Pos Z: 0
Right: 0 — Bottom: 0
Anchors
Pivot: X 0.5 — Y 0.5
Rotation: X 0 — Y 0 — Z 0
Scale: X 1 — Y 1 — Z 1

Screen Resolutions

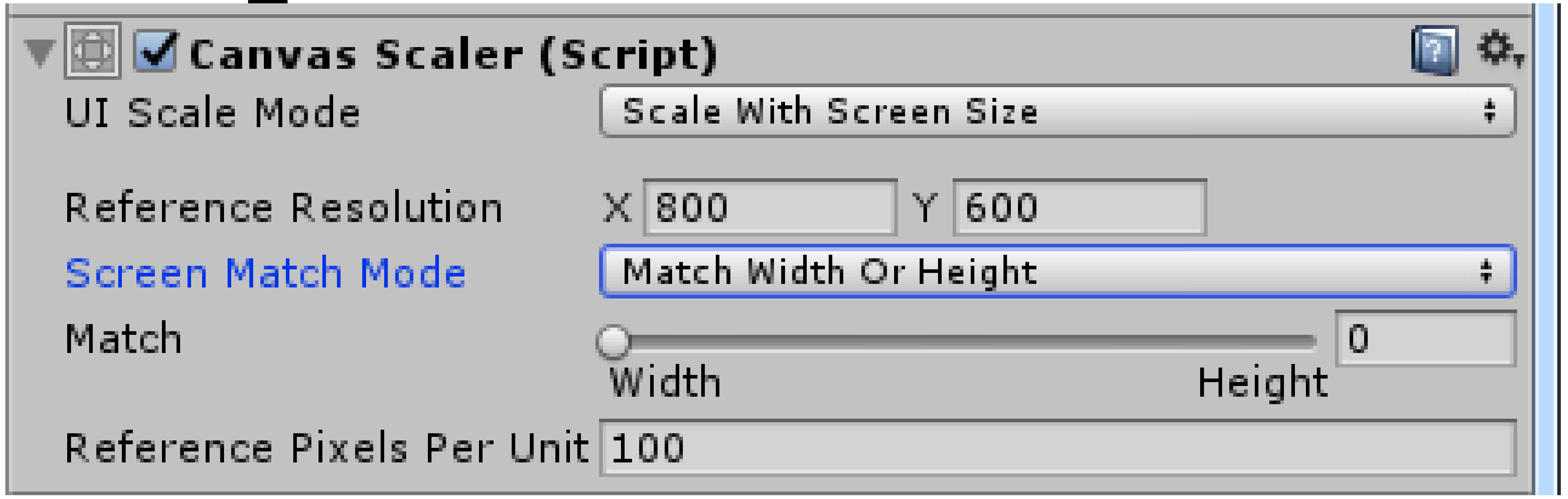
- Designing a UI for one screen doesn't mean it will look good on another screen.

- Wide variety of Aspect Ratios
  - Aspect Ratio = Width / Height = base width : base height

- Monitors
  - 1680x1050 = 16:10, 1920x1200 = 16:9, 1920x1080 = 4:3, 2160x1440 = 3.2, 3840x2160 = 12:5

- Phones (in landscape mode)
  - iPads = 4:3, iPhone 4 = 3:2, Android tablets = 16:10, Galaxy Tab 7 = 17:10, iPhone 5/6 and Galaxy 6/7 = 16:9… and smaller brands with a lot more.

Screen Resolutions - UI


Canvas Scaler (Script)
UI Scale Mode: Scale With Screen Size
Reference Resolution: X 800 Y 600
Screen Match Mode: Match Width Or Height
Match: 0
Reference Pixels Per Unit: 100

- Use anchors intelligently

- Use Canvas Scaler (added to canvas gameobjects automatically)

- Scale with Screen Size
  - Match the width of the screen
  - Match the height of the screen
  - Match both as best as possible

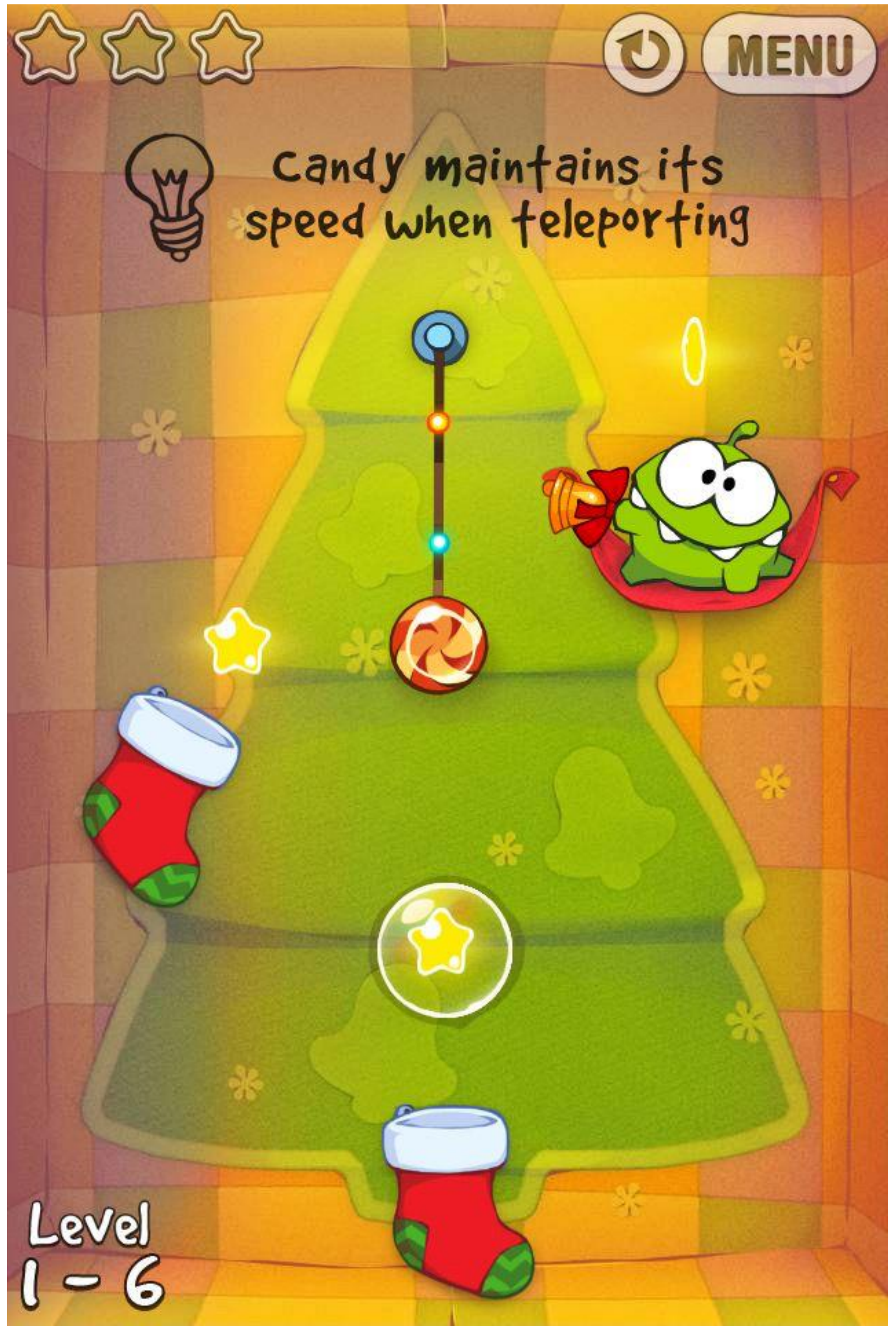
- Not a silver bullet!
  - **Test test test and then test some more on all the aspect ratios** that your customers are likely to use! (and then some hypothetical ones too)

Screen Resolutions – 2D Games

- **This affects 2D phone games a lot!**

- Great article:
- https://v-play.net/doc/vplay-different-screen-sizes/

- “Cut the Rope” does this well with a carboard box boarder that can be cut off in any way due to resolution differences.



Text, Images, Buttons, etc.

- After the previous slides, most actual UI gameobjects are pretty straight forward.

- Most visual elements have a Text component or Image component.

- Most common interaction has its own component (e.g. Button)
  - These pass through the EventSystem component
  - Usually through Event Listeners (similar to Java)
  - On[Event]()
  - E.g. Button has OnClick(), Slider has OnValueChanged()

Accessing UI via Code

- Most standard UI elements under UnityEngine.UI

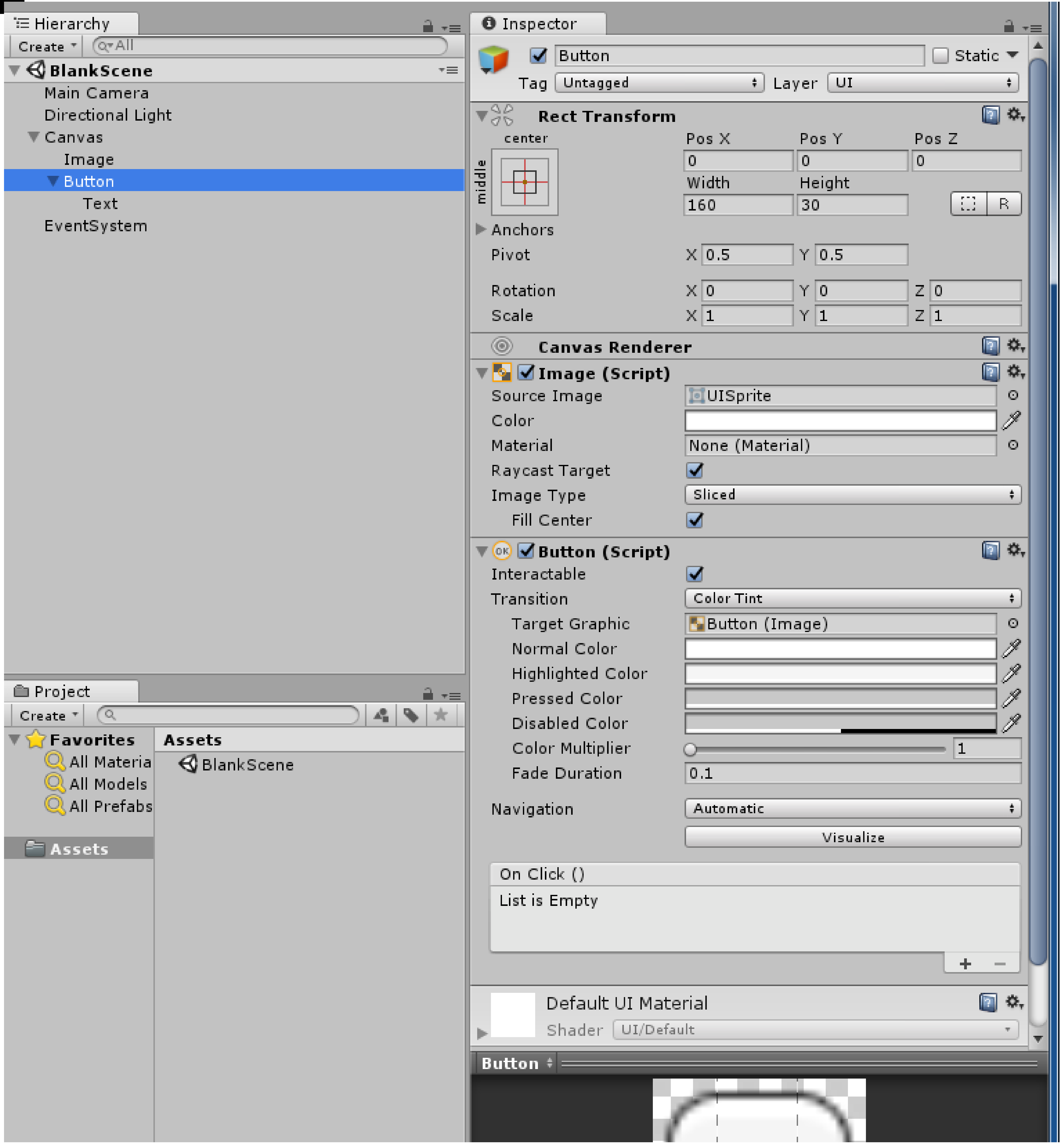
```
using UnityEngine.UI;
```

- Anything that can be seen in the inspector, is likely editable through code.
  - Including RectTransforms
  - And EventListeners (using C# delegates woot!)

```
void Start() {
    myButton.onClick.AddListener(ButtonClickedMethod);
}
void ButtonClickedMethod() {… do stuff when button is clicked…}
```

Button example

- A standard button has:
- An Image component (for the button shape)
- A Button component (for interaction)
- A child with a Text component (for the button text)



```
Hierarchy
Create ▾  Q▾All
▼ BlankScene
    Main Camera
    Directional Light
  ▼ Canvas
      Image
    ▼ Button
        Text
    EventSystem

Project
Create ▾
▼ Favorites          Assets
    All Materia        BlankScene
    All Models
    All Prefabs
  Assets
```

```
Inspector
[✓] Button                                   [ ] Static ▾
Tag  Untagged                 Layer  UI

▼ Rect Transform
  center          Pos X    Pos Y    Pos Z
  middle          0        0        0
                  Width    Height
                  160      30                  R
▶ Anchors
  Pivot           X 0.5    Y 0.5
  Rotation        X 0      Y 0      Z 0
  Scale           X 1      Y 1      Z 1

  Canvas Renderer

▼ [✓] Image (Script)
  Source Image       UISprite
  Color
  Material           None (Material)
  Raycast Target     [✓]
  Image Type         Sliced
    Fill Center      [✓]

▼ [✓] Button (Script)
  Interactable       [✓]
  Transition         Color Tint
    Target Graphic   Button (Image)
    Normal Color
    Highlighted Color
    Pressed Color
    Disabled Color
    Color Multiplier                            1
    Fade Duration    0.1
  Navigation         Automatic
                     Visualize

  On Click ()
  List is Empty
                                                + −

  Default UI Material
  Shader  UI/Default

Button ÷
```