Extended Reality in Industry 4.0 (ERI)

Lecture 8: User Interface Elements in Unity

Dr. Kalpana Shankhwar,
kalpana@iiitd.ac.in
Assistant Professor
Department of Human Centered Design,
IIIT Delhi

Content

- Canvas
- Basic layout
- Visual components
- Interaction components
- Demo

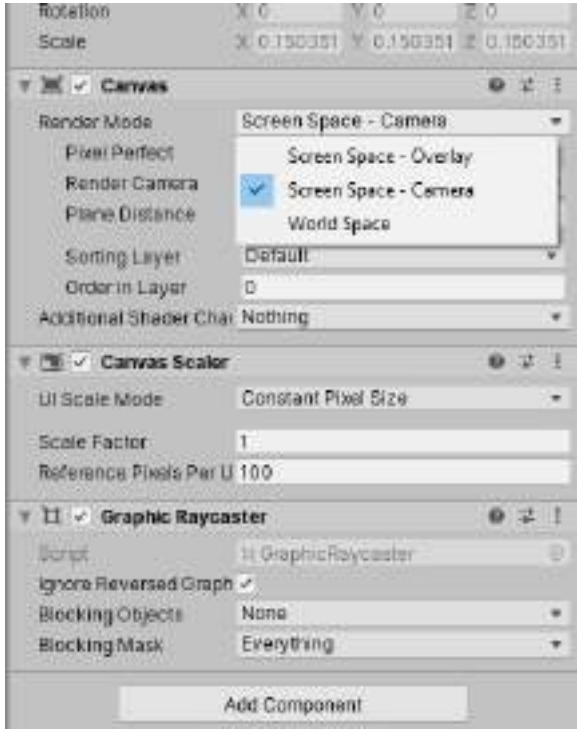
Introduction

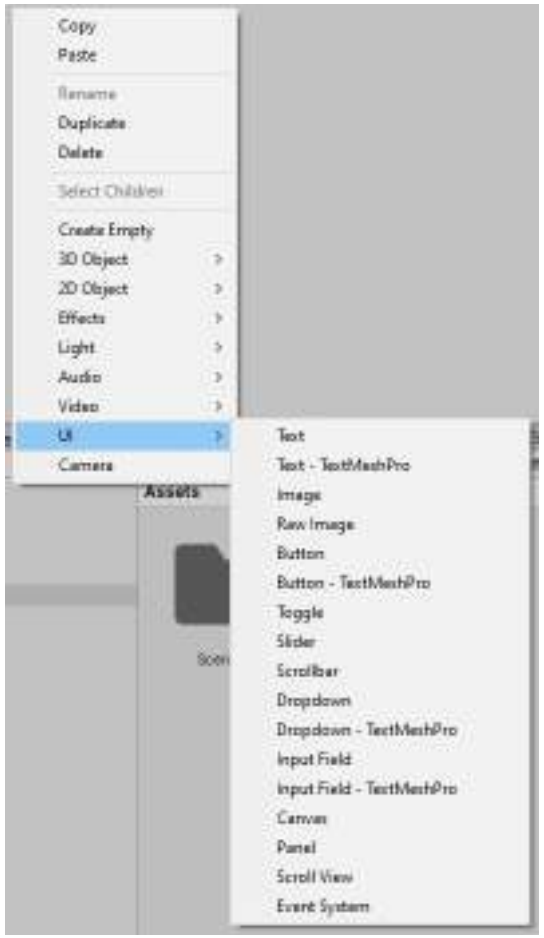
Unity user interface is most commonly used toolkit to create the user interface for games or applications.

- **Canvas** is a rectangular area which contains UI elements.

When a UI element such as an Image is created using menu GameObject>UI>Image, a Canvas is created automatically

If there is not a Canvas in the scene.



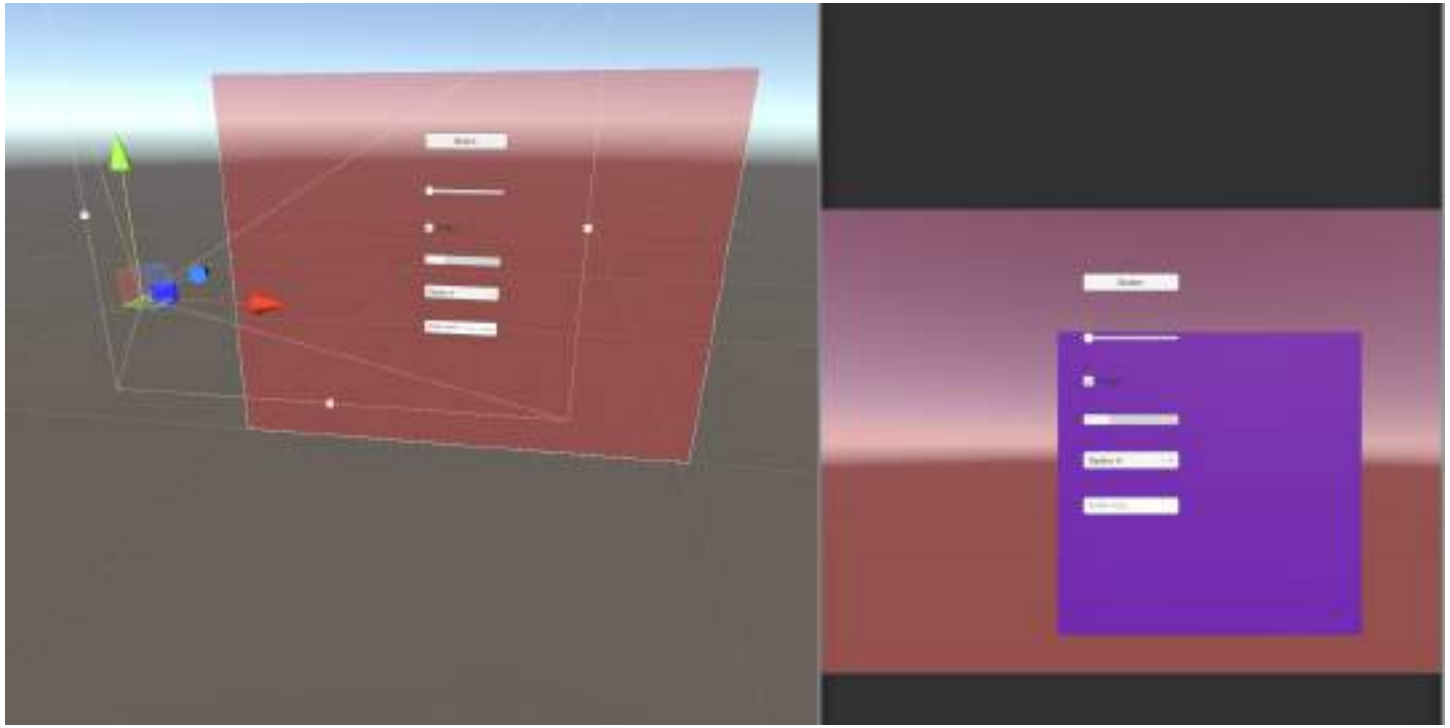


- **Render mode**
  1. Screen space – Overlay
  2. Screen space - Camera
  3. World space

Introduction

1. Screen space – Overlay

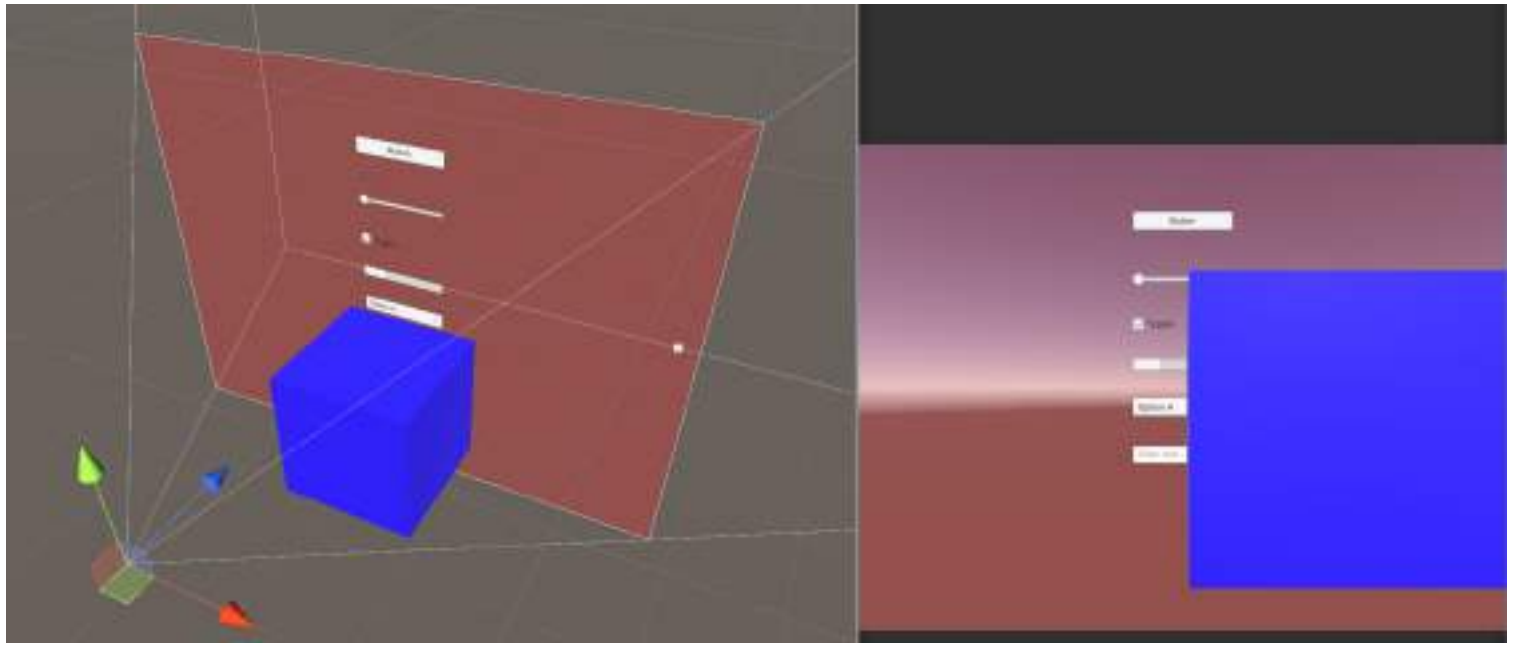
This render mode places UI elements on the screen rendered on top of the scene. If the screen is resized or changes resolution, the Canvas will automatically change size to match this.



Introduction

2. Screen space - Camera

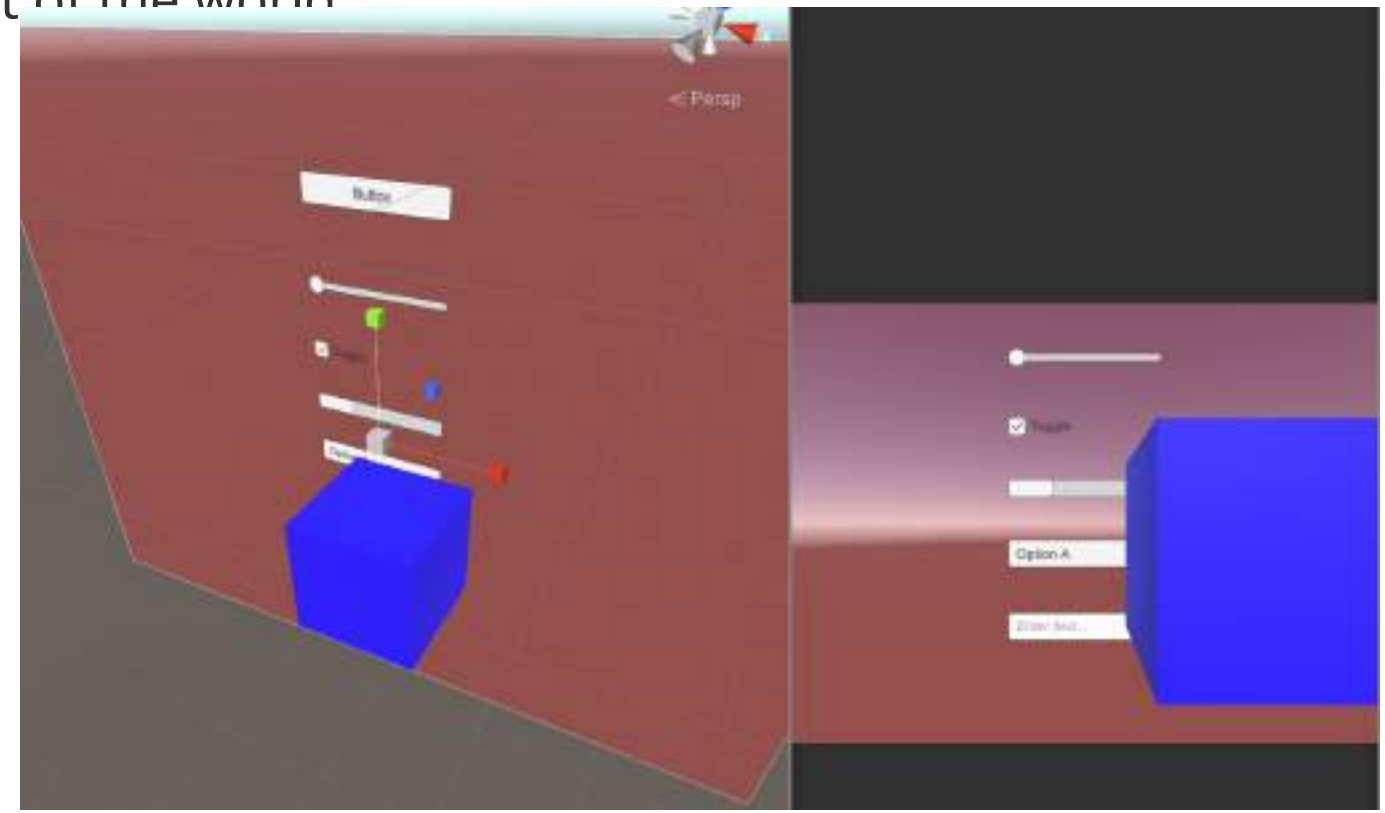
This is similar to Screen Space - Overlay, but in this render mode the Canvas is placed a given distance in front of a specified **Camera**. The UI elements are rendered by this camera, which means that the Camera settings affect the appearance of the UI. The gameojects can also be rendered in front of the UI.



Introduction

3. World space

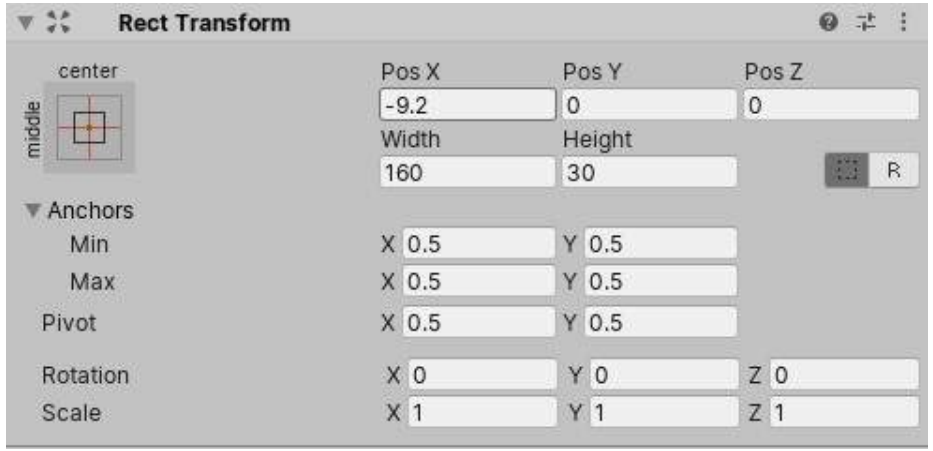
In this render mode, the Canvas will behave as any other object in the scene. The size of the Canvas can be set manually using its Rect Transform, and UI elements will render in front of or behind other objects in the scene based on 3D placement. This is useful for UIs that are meant to be a part of the world.



Basic layout

Rect transform

The Rect Transform is a new transform component that is used for all UI elements instead of the regular Transform component.

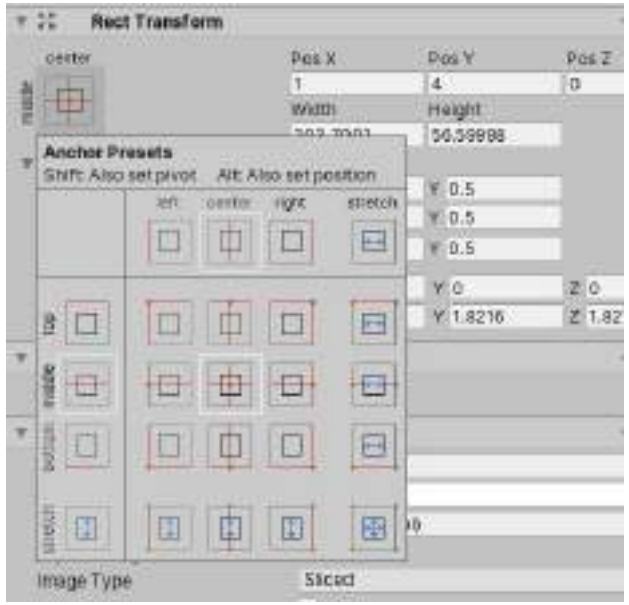


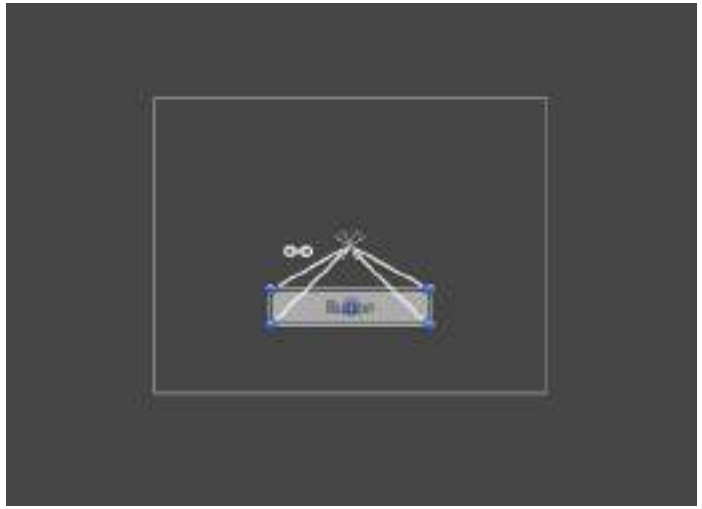
Basic layout

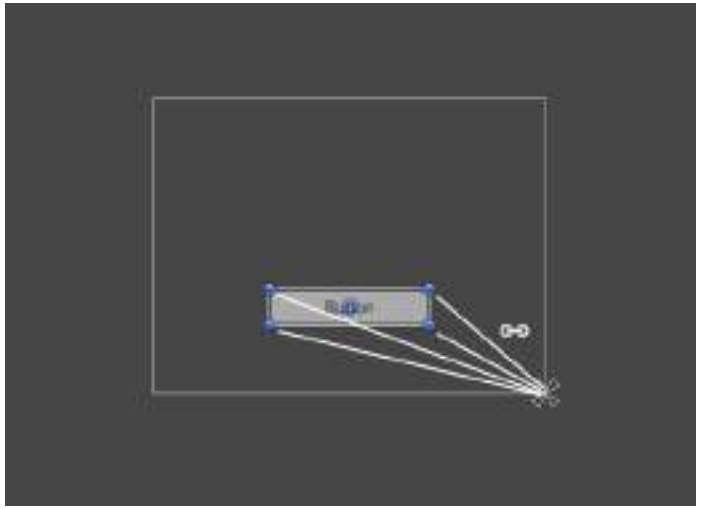
Anchors

Rect Transforms include a layout concept called **anchors**. Anchors are shown as four small triangular handles in the Scene View.

If the parent of a Rect Transform is also a Rect Transform, the child Rect Transform can be anchored to the parent Rect Transform in various ways. For example, the child can be anchored to the center of the parent, or to one of the corners.

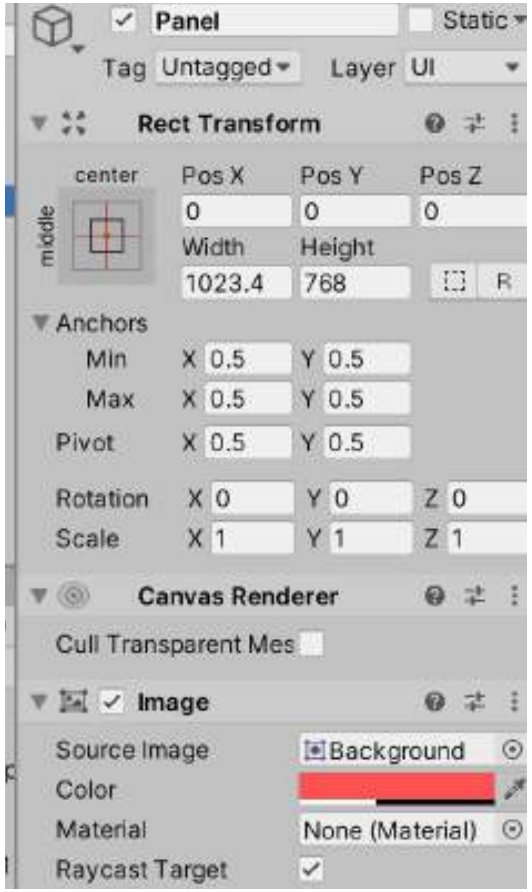


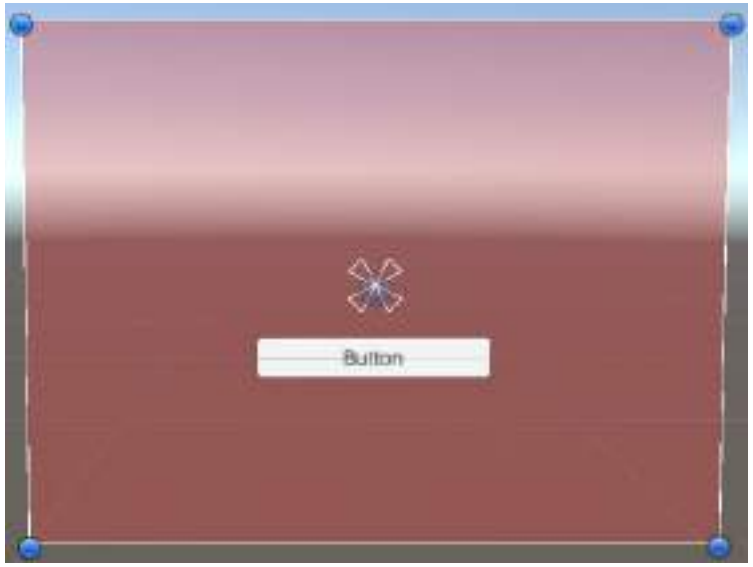




Visual components

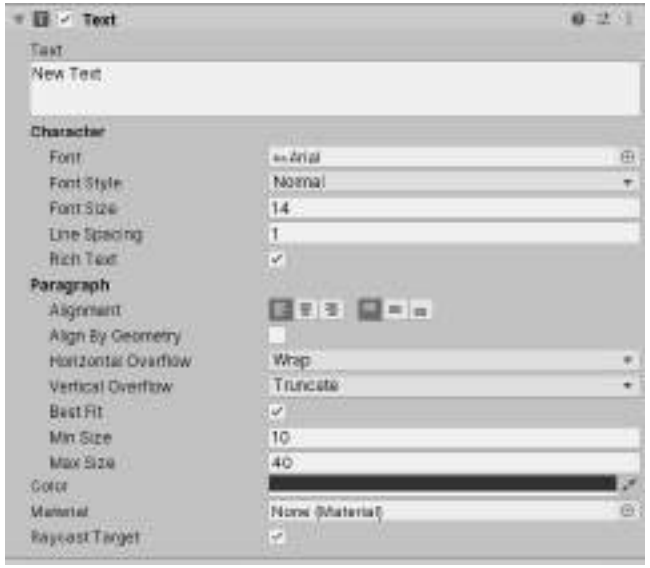
- Panel

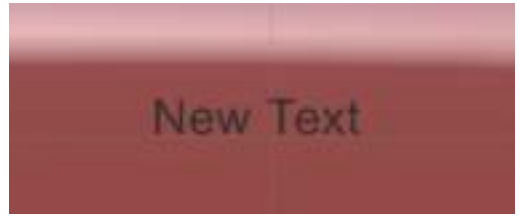




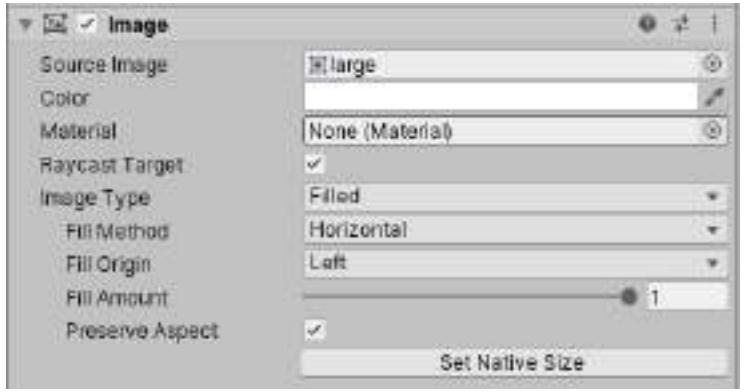
Visual components

- Text





- Image



Interaction components

Interaction components in the UI system handle interaction such as mouse click or touch events or interaction through keyboard or controller.

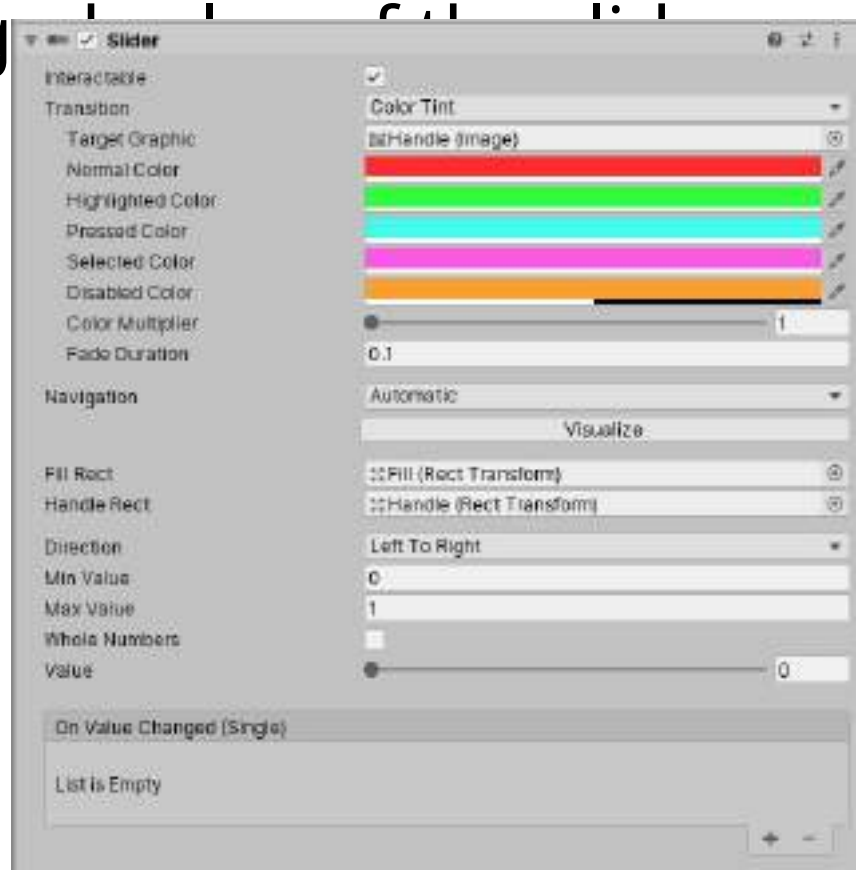
- Button
- Slider
- Toggle
- Scrollbar
- Dropdown
- Input field

Button

The button controls the response to the click from the user and is used to initiate or confirm an action.

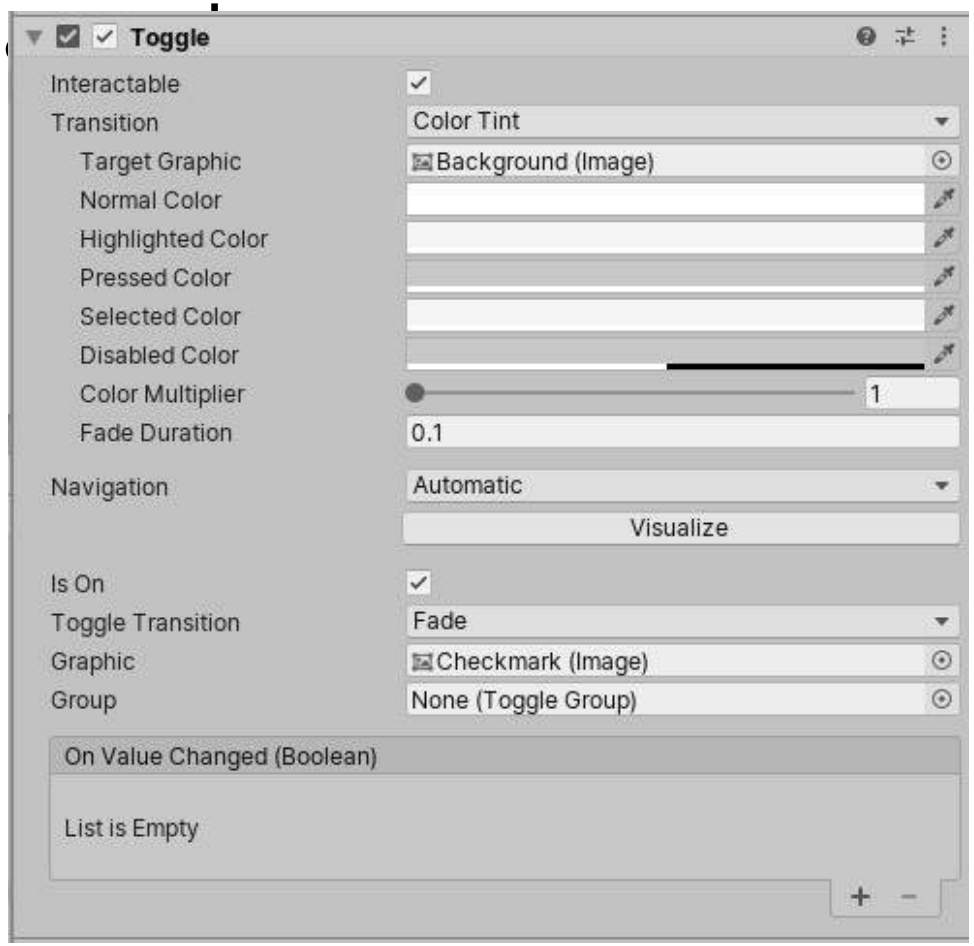
Slider

A slider has a decimal number Value that the user can vary between a minimum to maximum value by dragging the slider. It has a OnValueChanged UnityEvent to define the action on the chang



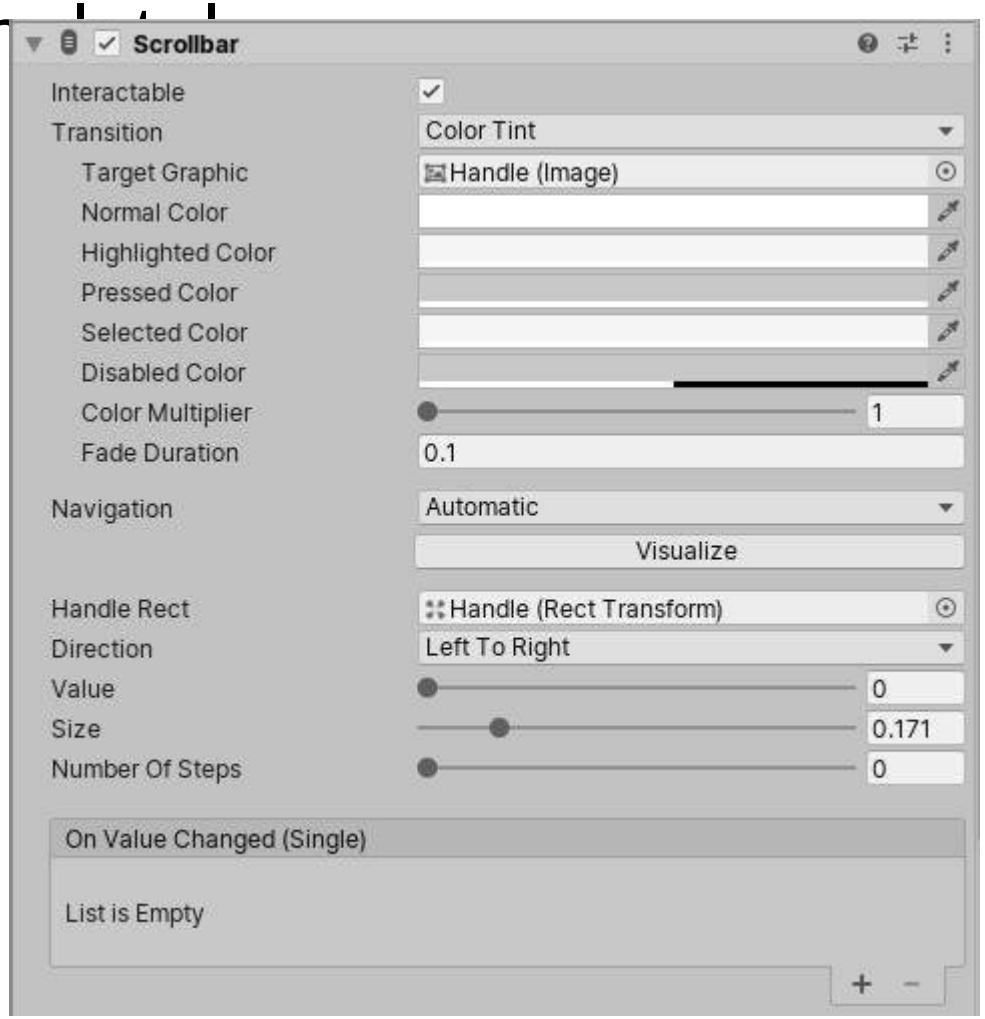
Toggle

Toggle control is a checkbox that allows user to switch an option on and off. The action can be defined by OnValueChanged UnityEvent when the value is ch



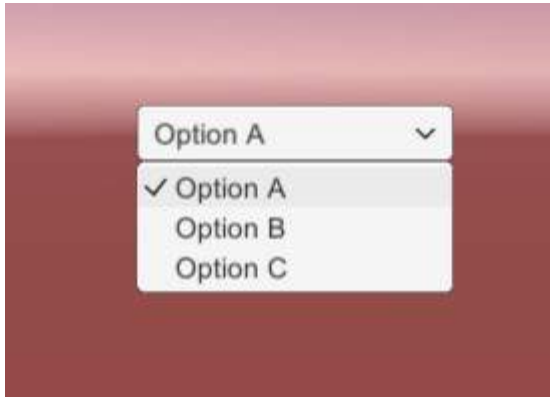
Scrollbar

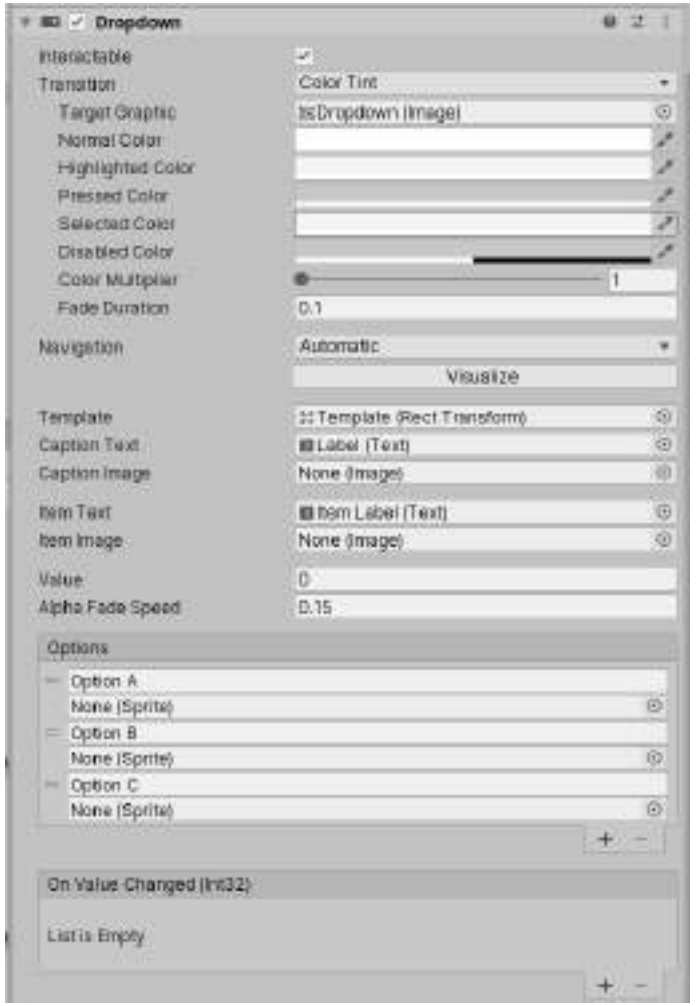
The Scrollbar control allows user to scroll an image or text that is too large to see com



Dropdown

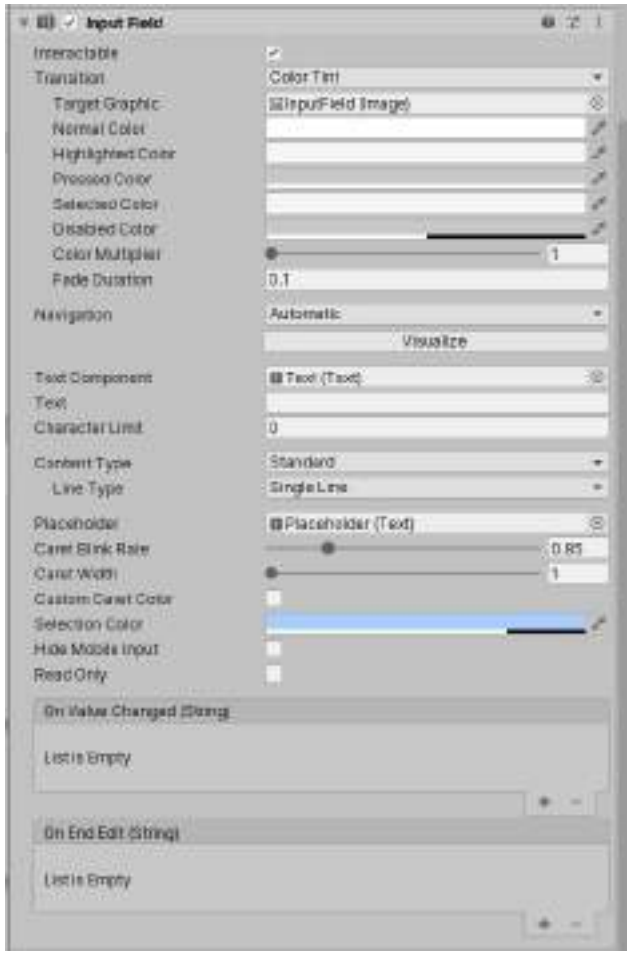
A dropdown has a list of options to choose from. It allows user to choose a single option from the list.





Input field

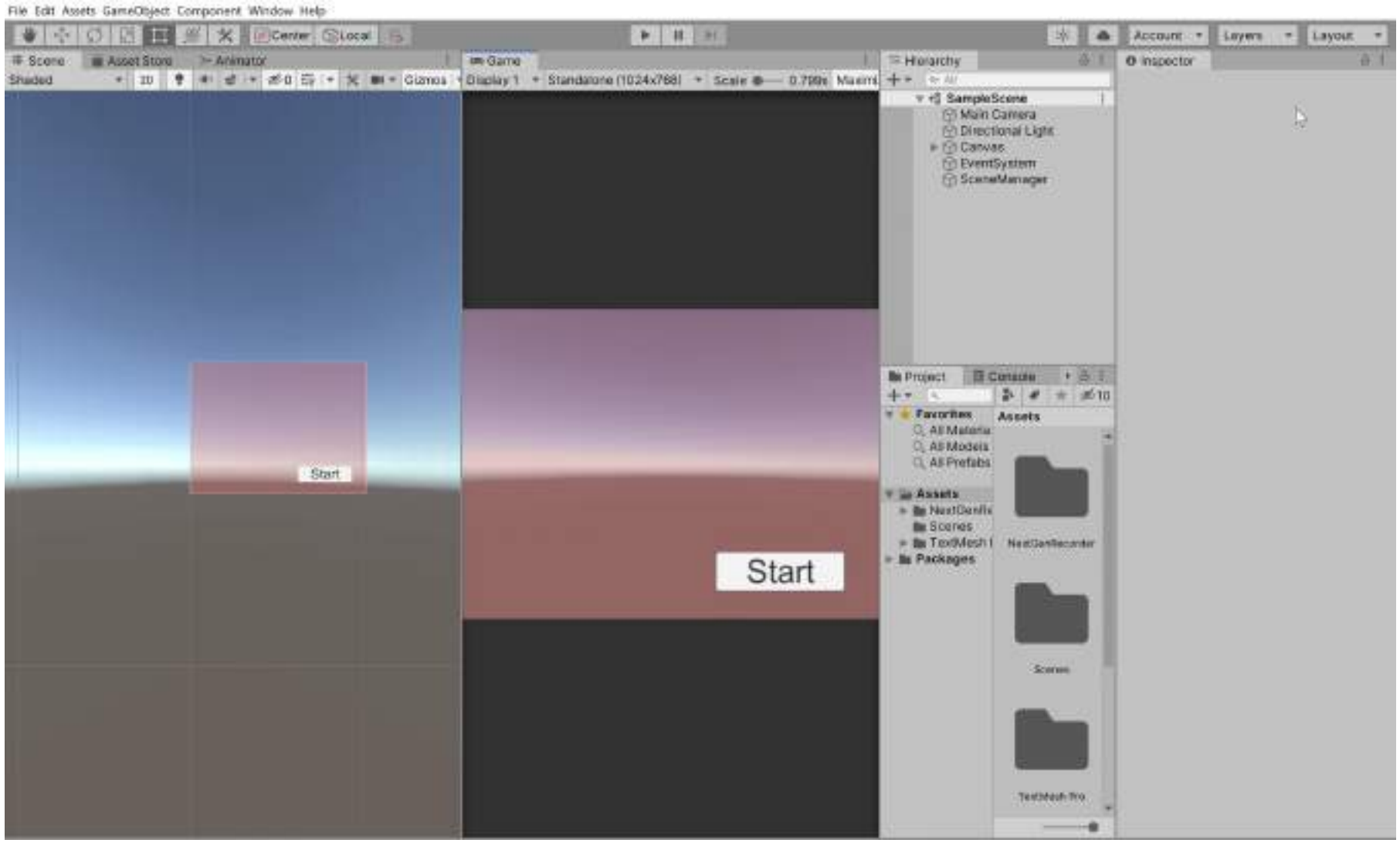
An input field let the user enter the text to initiate or complete an action. It has a UnityEvent to define what it will do when user has finished editing it.



Class activity 5.

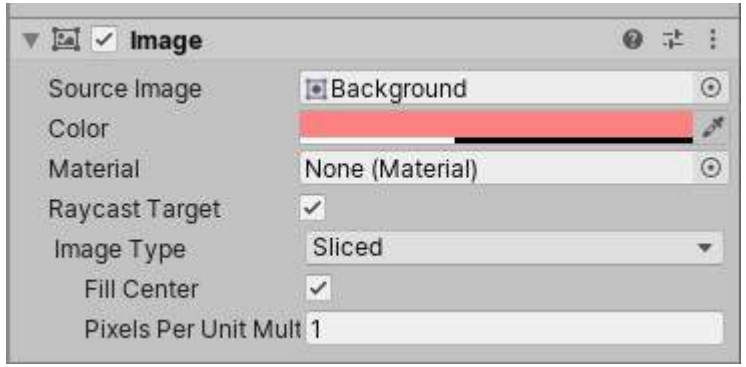
UI element: **Button**

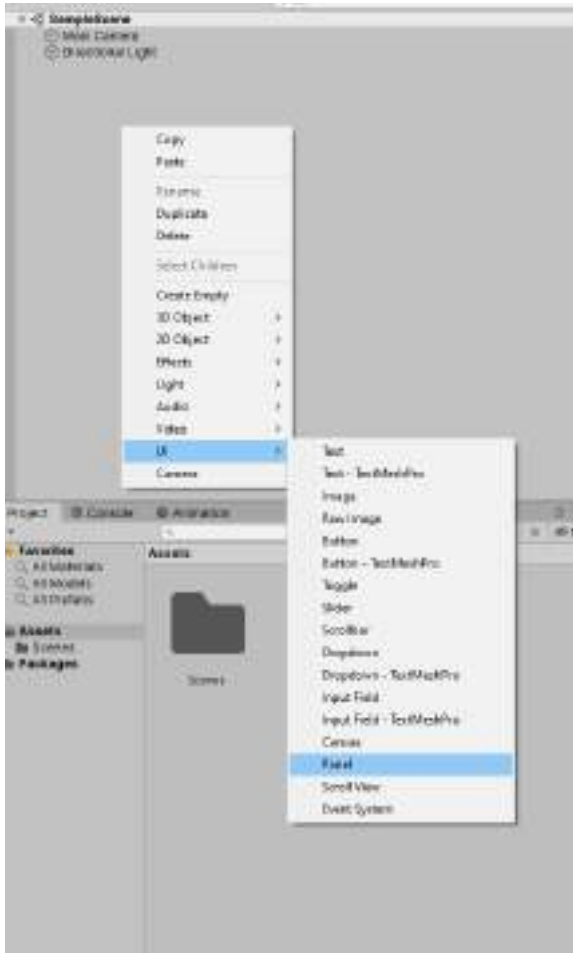
This example uses buttons to switch the scenes by creating 4 scenes with names: Sample Scene, Scene 1, Scene 2 and Scene 3 with the button name: Start, Next, End.



Steps:

1. Create the panel of red colour with transparent background.

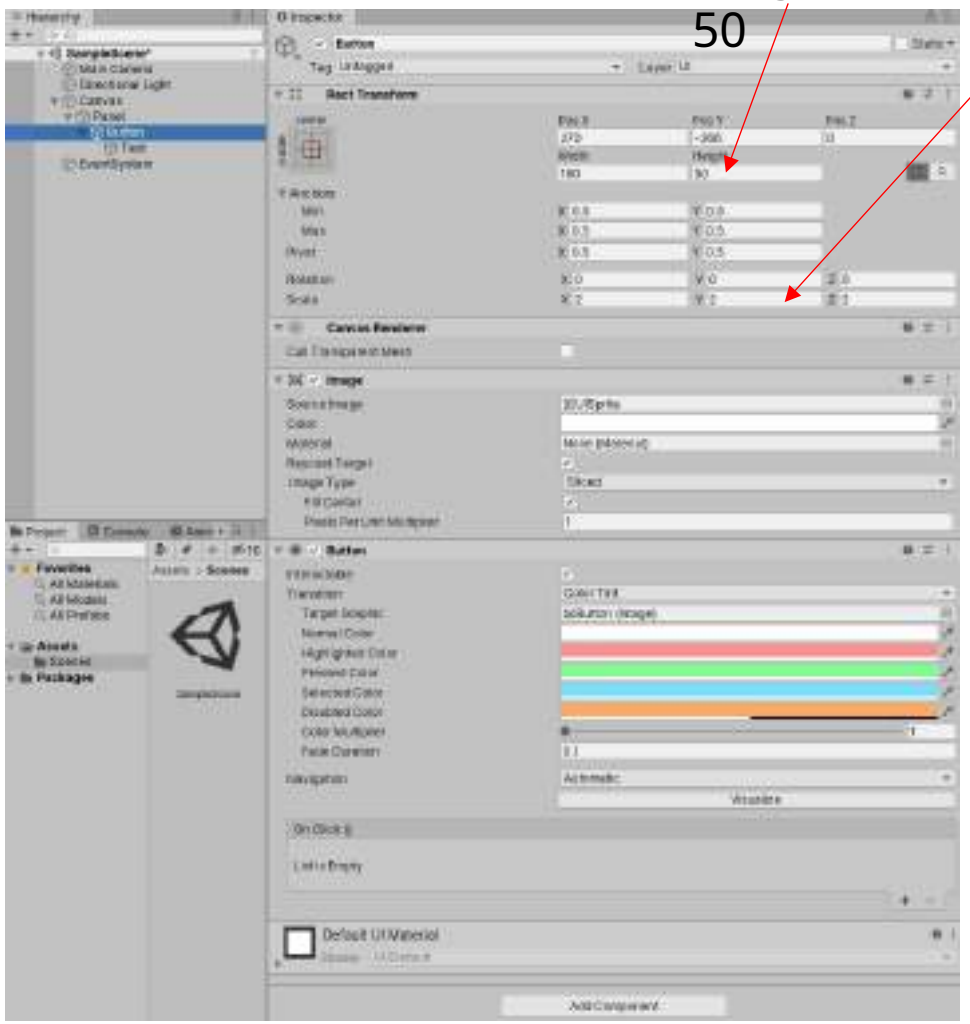


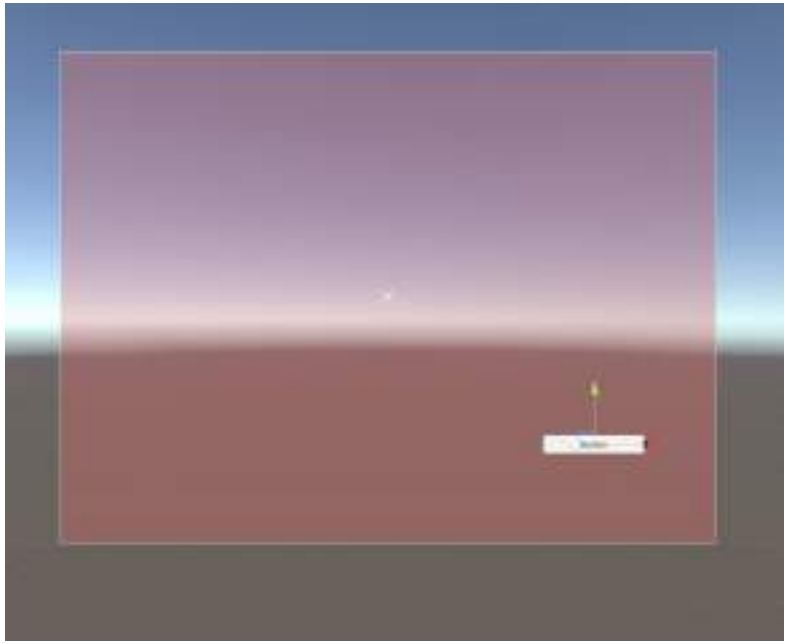


2. Create a Button as a child of panel.

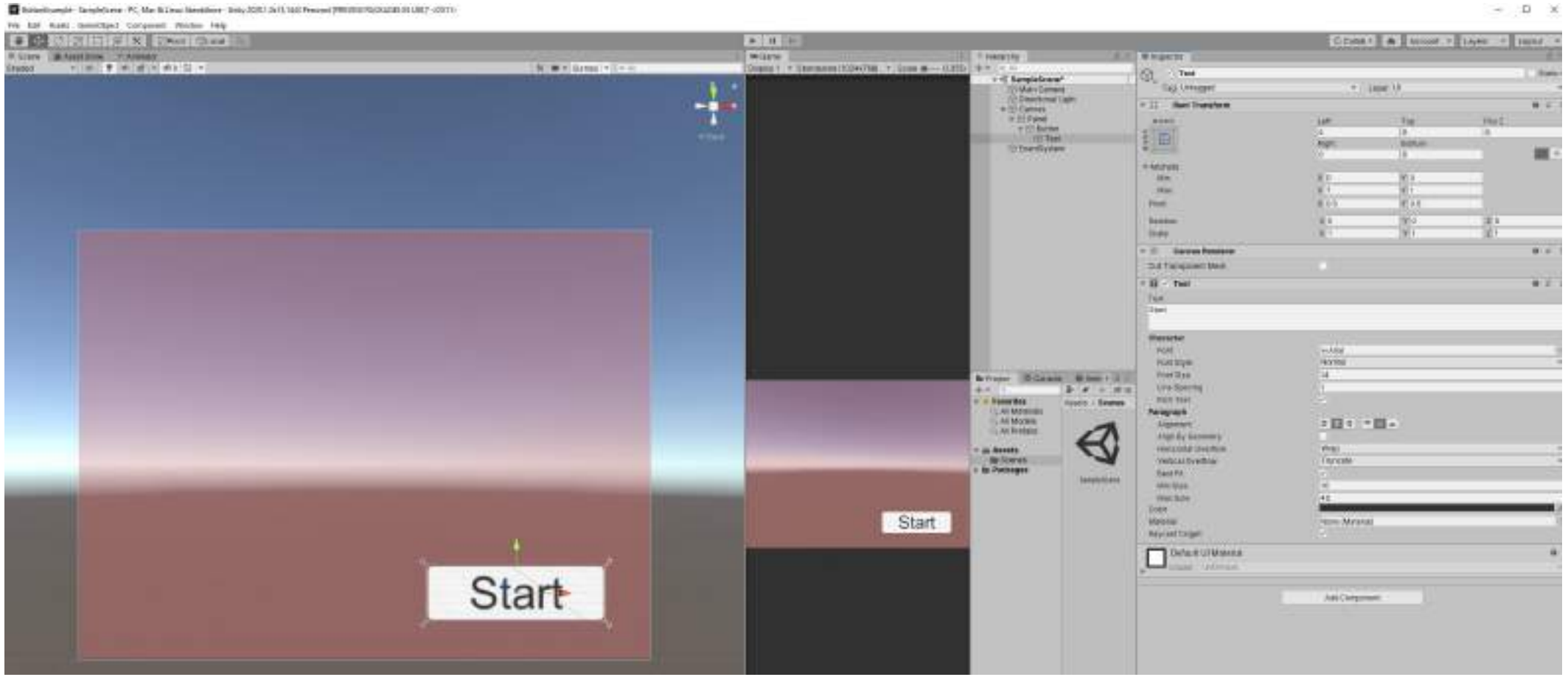
Change the Height to 50

Set the scale 2

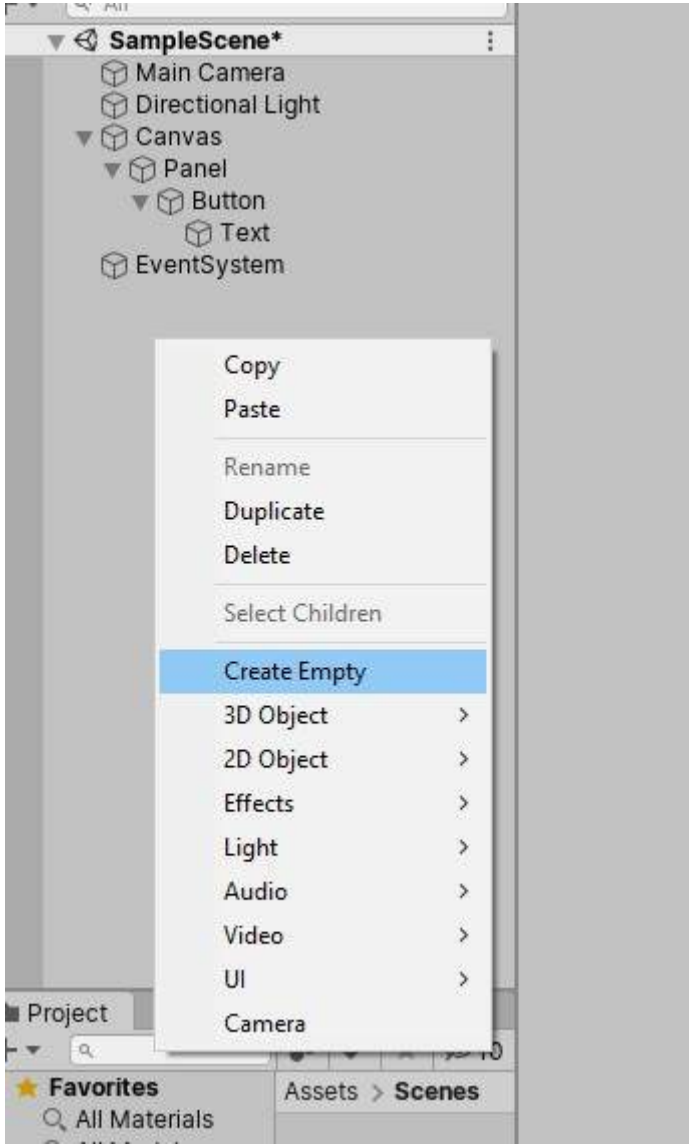


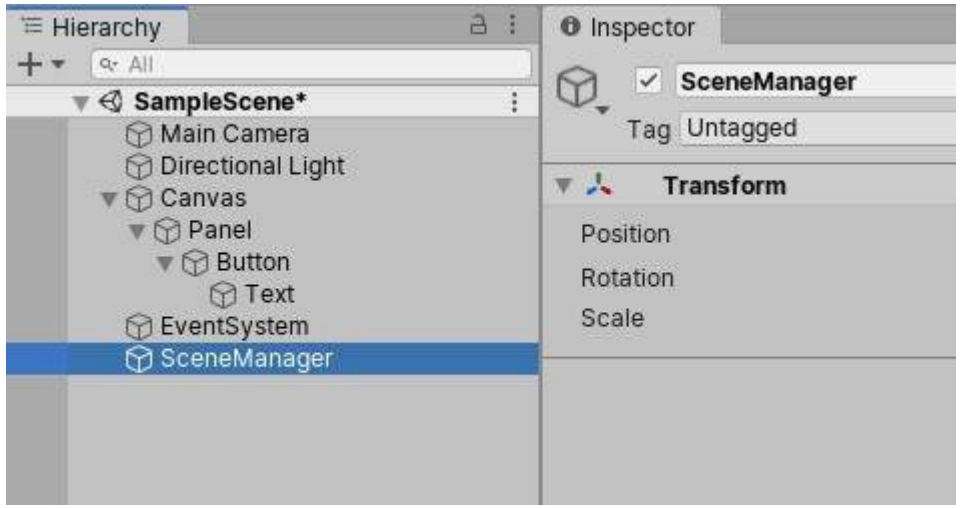


3. Change the name of the button “**Start**” from the **text** component.

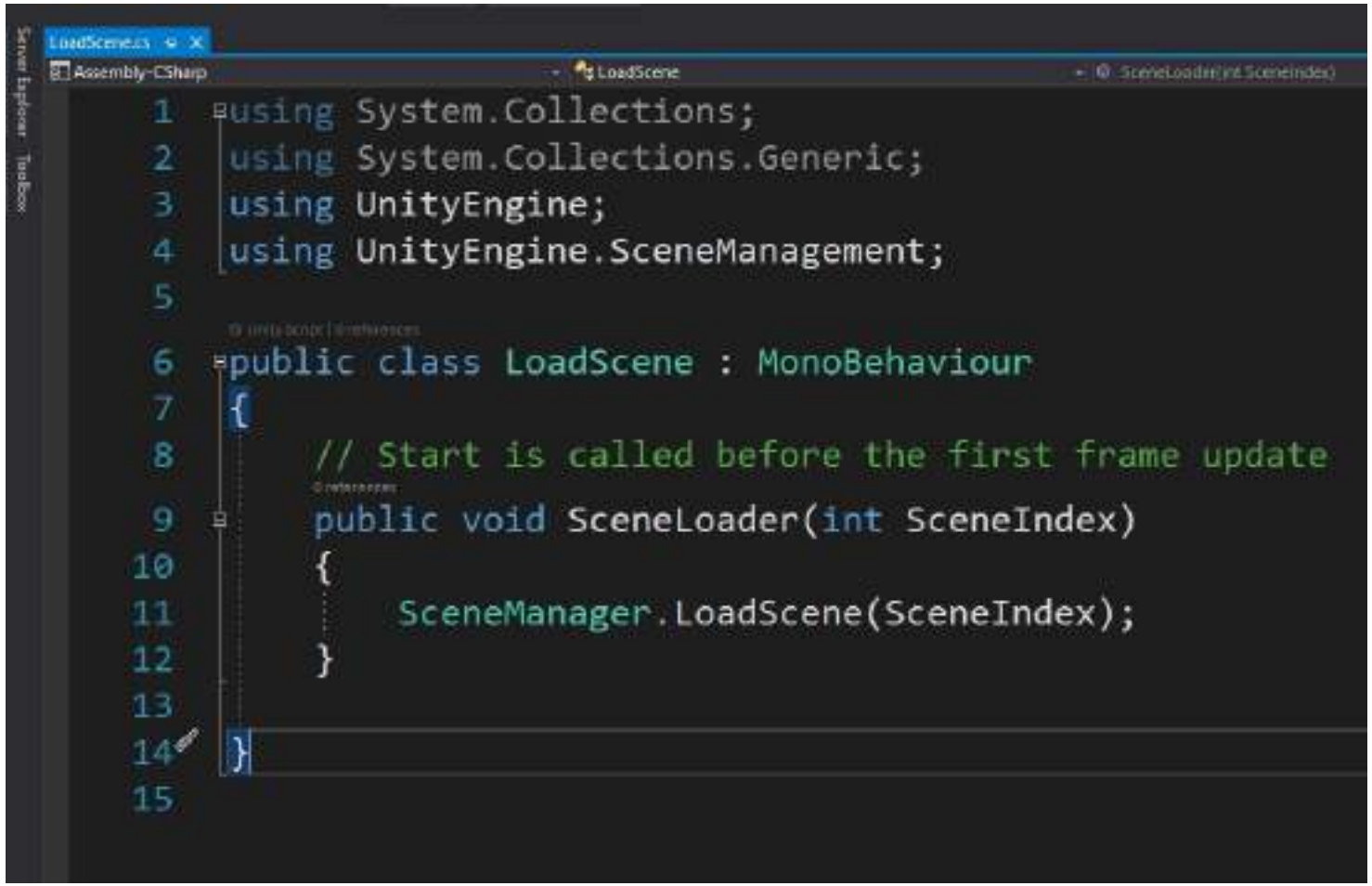


4. Create an empty object and name it “SceneManager”.

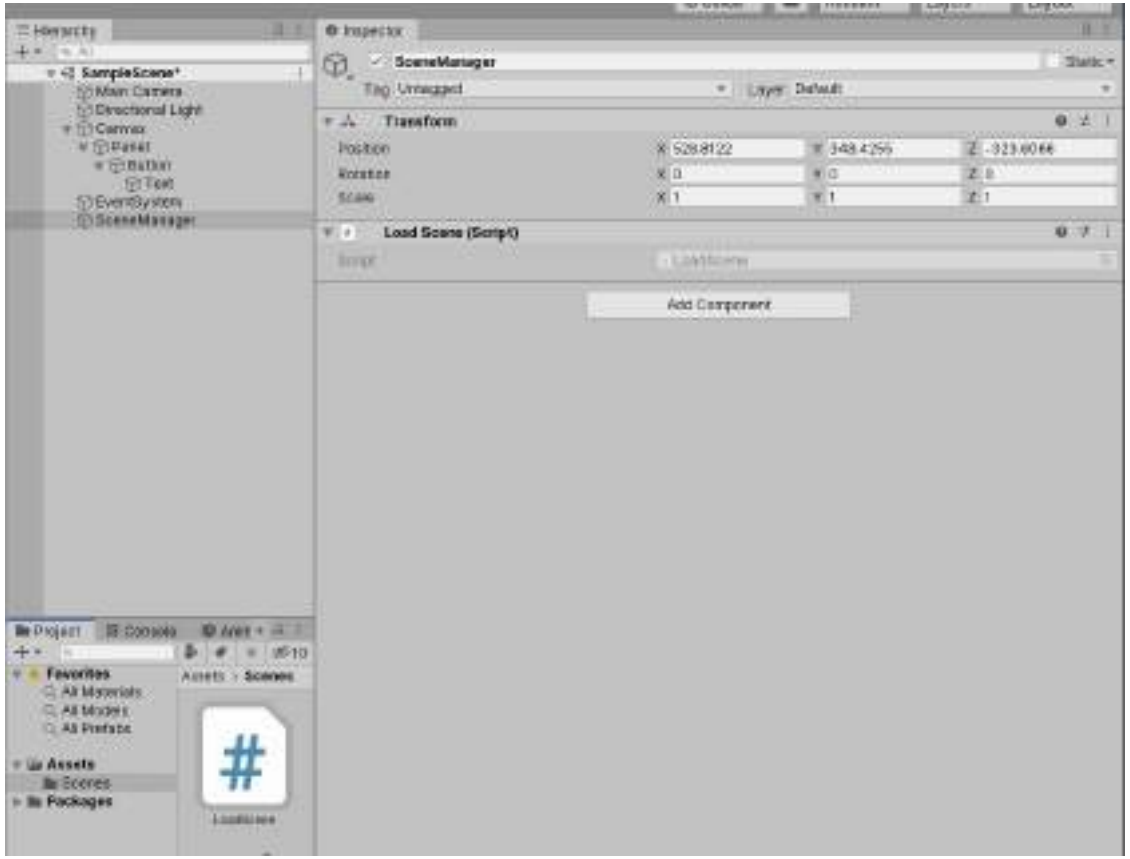




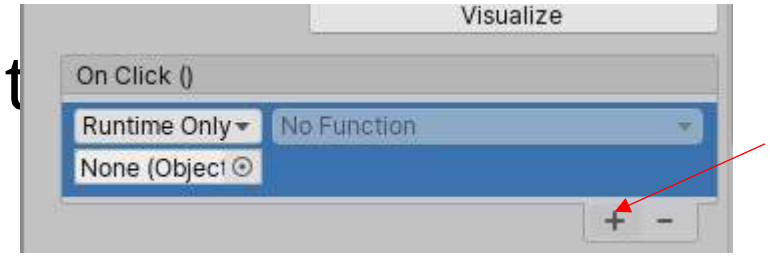
5. Create a script and name it “LoadScene”.



6. Attach the “LoadScene” script to the “SceneManager”.



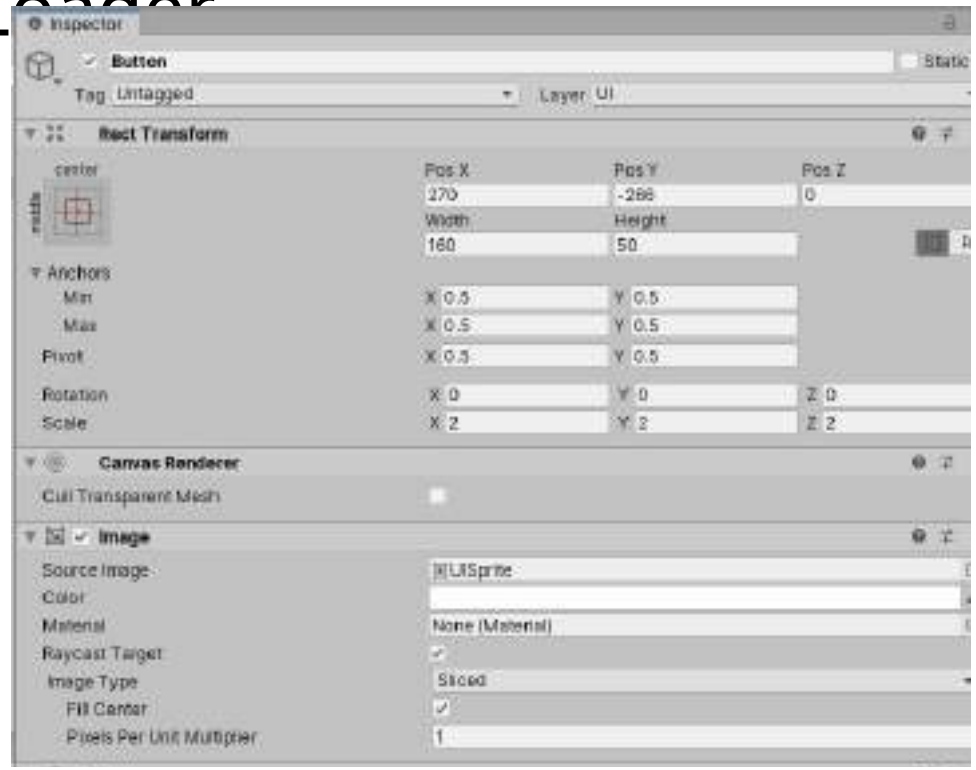
7. Add the “On Click” event in the But

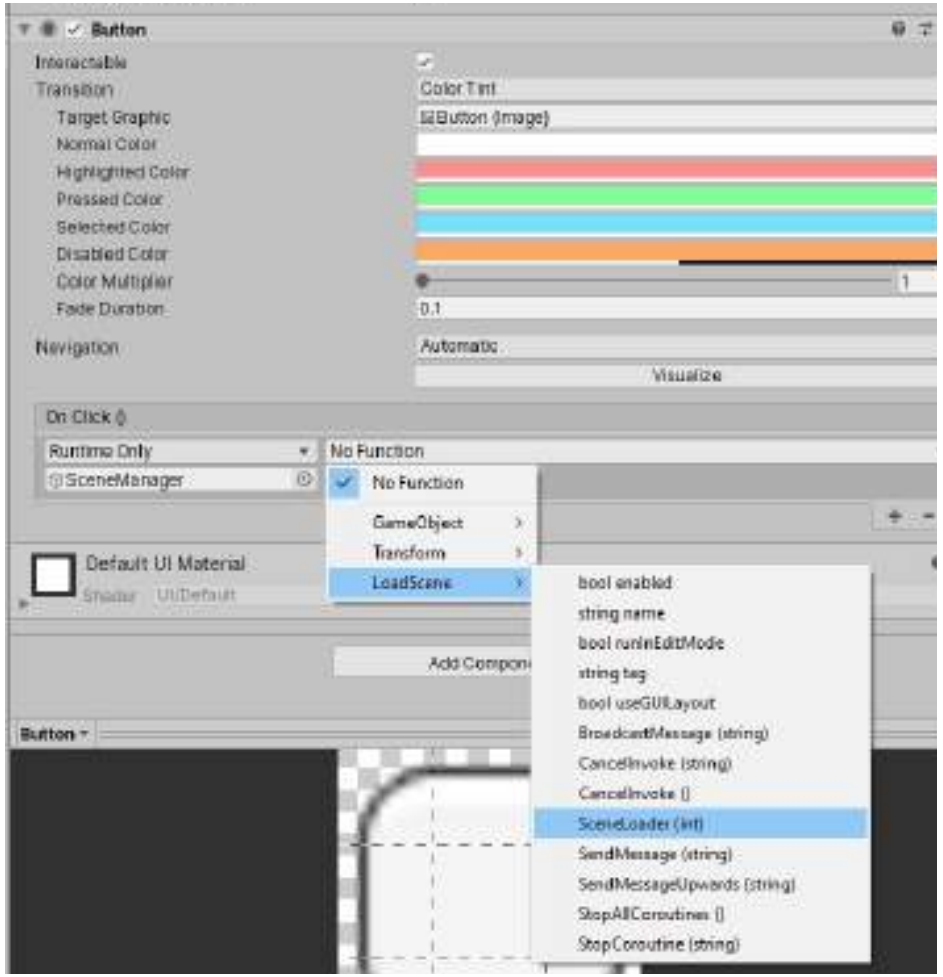


Click the “+” symbol

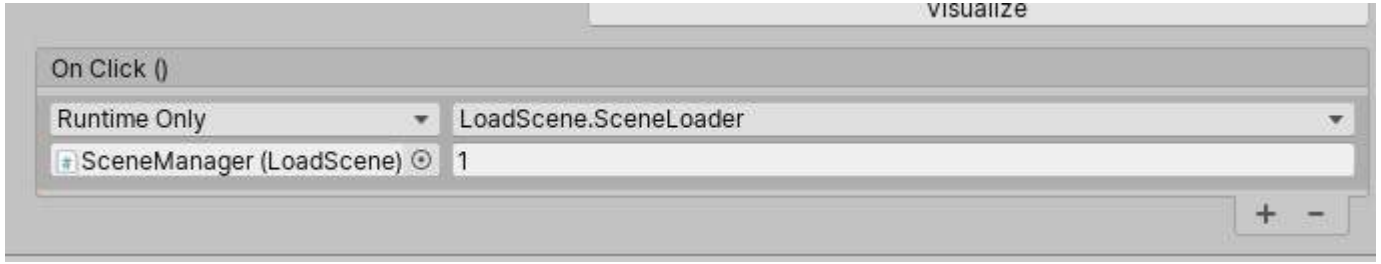
8. Drag the “SceneManager” gameObject and drop it in the “On Click” event list.

9. Find LoadScene option from the dropdown menu and select SceneLoader





10. Enter the Scene Index “1” to switch to respective scene.

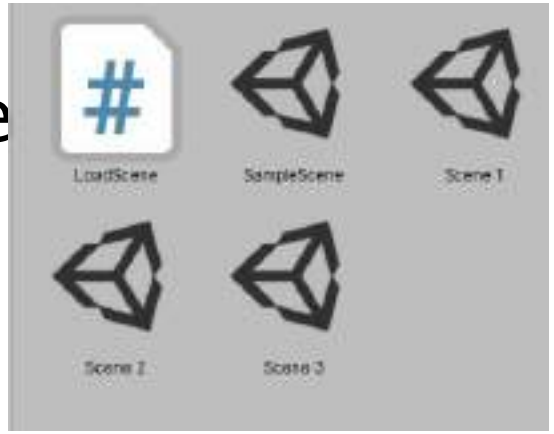


11. Create another scene with name “Scene 1”.

12. Repeat the steps from 1 to 10 except keeping the panel colour green, Button name “Next” and the scene index “2”.

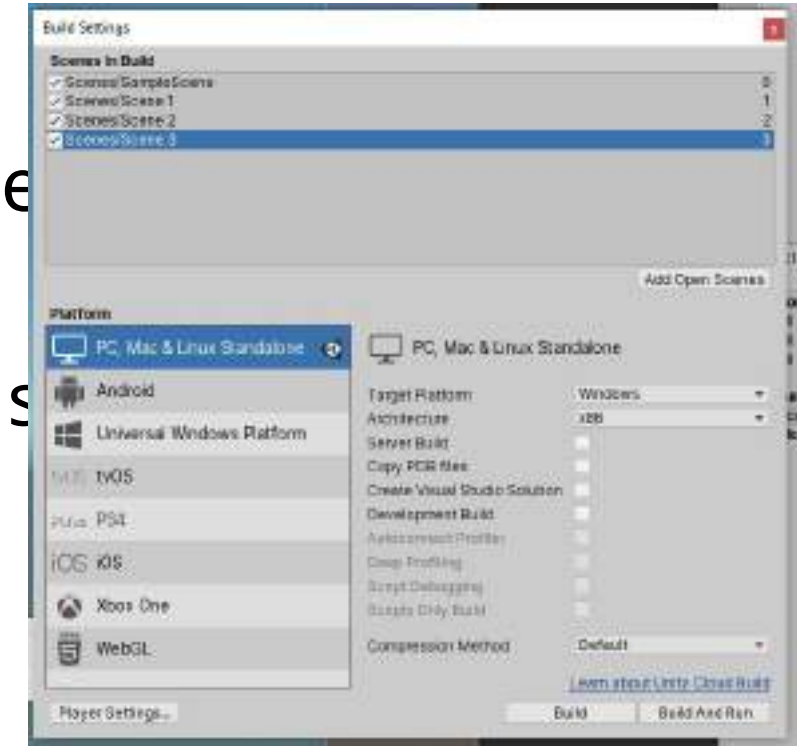
13. Similarly create “Scene 2” by creating the panel with blue colour, Button name “End” and the scene index “3”.

14. Create a blank Scene with name “Scene



15. Select **Build** option from the **Find** dropdown menu.

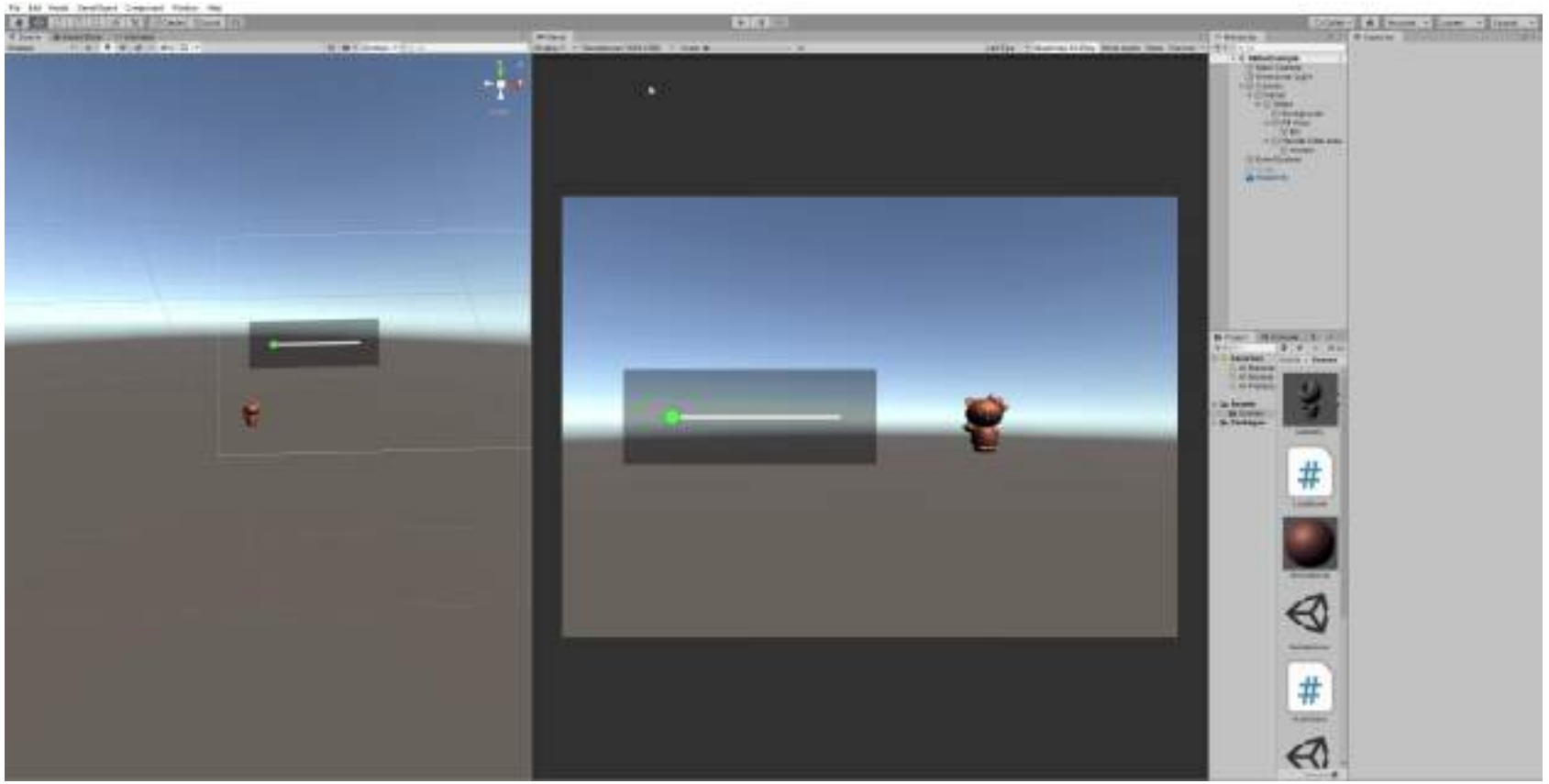
16. Drag the scenes to the Scenes in Build space define the scene index.
17. Hit the play Button and click the buttons to s the scene.



Class activity 6

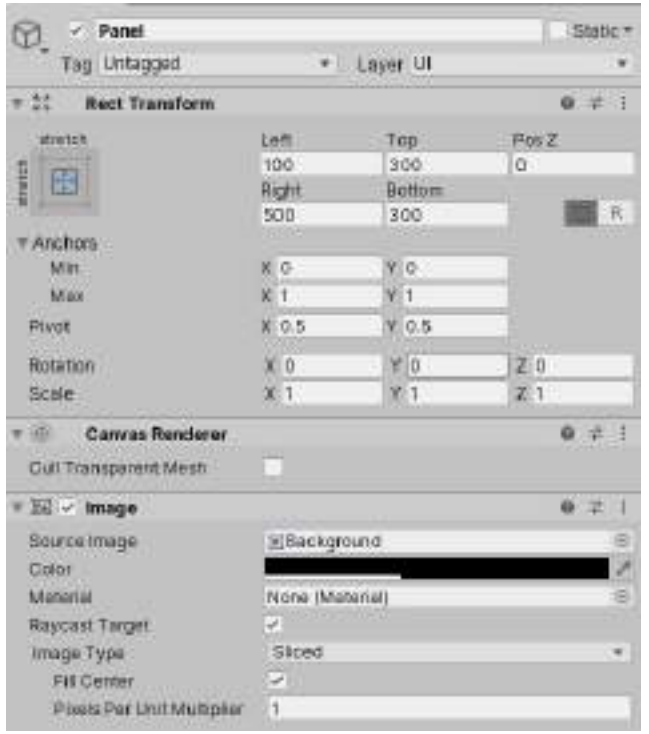
UI element: **Slider**

This example represents the application of Slider to vary the size of an object.

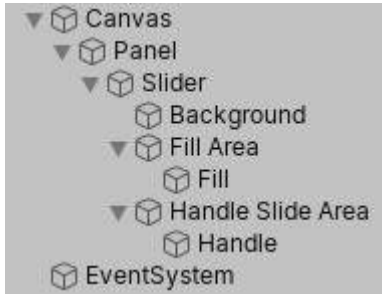


Steps:

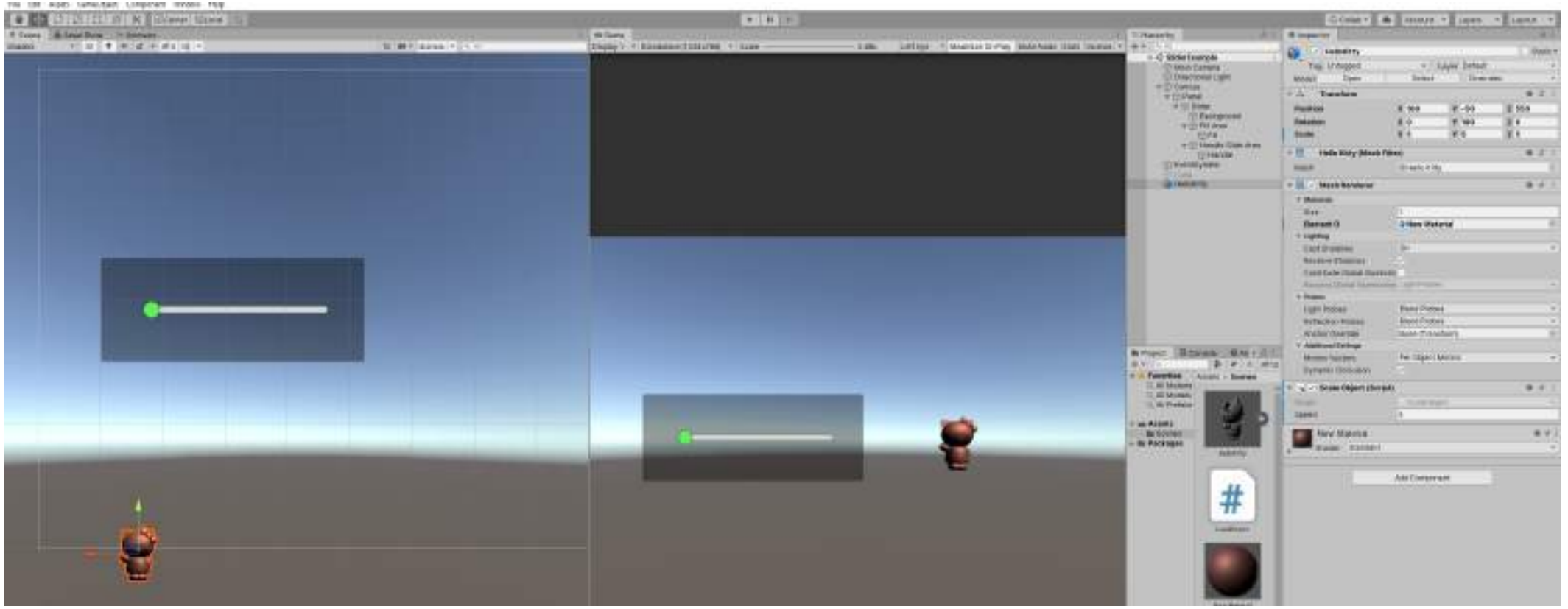
1. Create a Unity Scene named “SliderExample”.
2. Add a “Panel” with transparent black colour in the hierarchy.



3. Add a “Slider” under the panel with the fill of yellow colour and handle of green colour.



4. Import the “HelloKitty” gameobject in the assets section and drag it on the unity scene.



5. Create a script called “ScaleObject”.

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

public class ScaleObject : MonoBehaviour
{
    public float InitSize = 5f;
    Vector3 temp;

    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        temp = transform.localScale;
        temp.x = InitSize;
        temp.y = InitSize;
        temp.z = InitSize;
        transform.localScale = temp;
    }
    public void AdjustScale(float size)
    {
        InitSize = size ;
    }

}
```

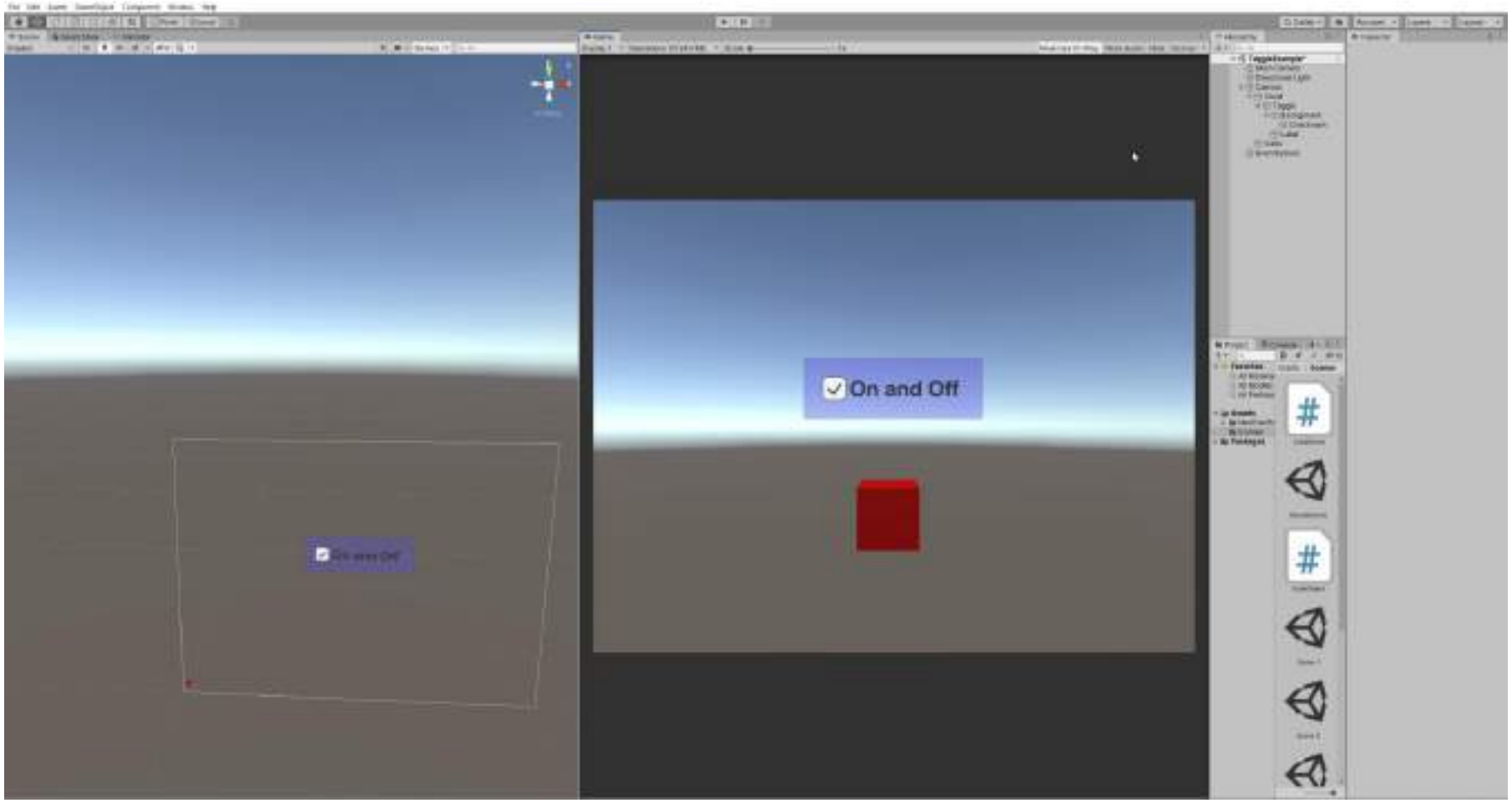
6. Drag the script in the inspector of “HelloKitty” gameobject.
7. Drop the “HelloKitty” gameobject to the OnValueChanged event and se
   ScaleObject.AdjustScale.
8. Set the Min value 5 and Max value 15 to vary the scale between these nu

“HelloKitty”
gameobejct

8. Hit the play button and drag the slider from left to right to increase the size of the gameobject.

Class activity 7. This example illustrates the application of toggle to enable and disable an object.

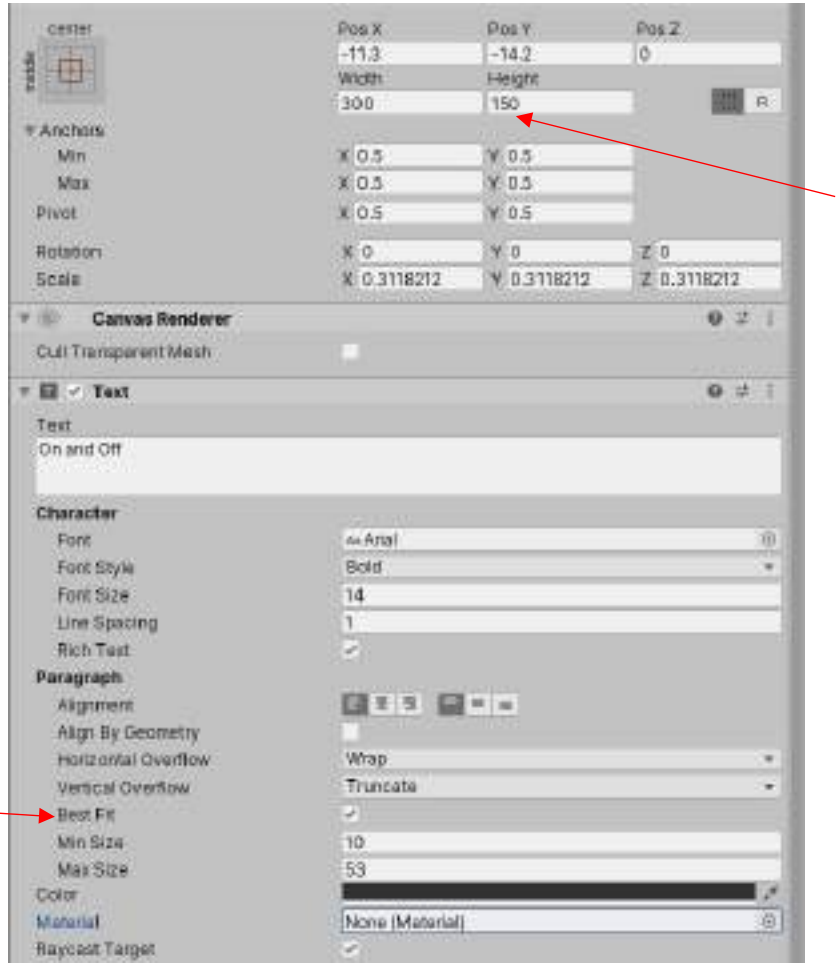
UI element: **Toggle**



Steps:

1. Create a toggle from UI menu as a Child object of Panel.

2. To obtain good quality of the text, change the Text setting of the Label by keeping the Width = 300 and Height = 150 and check the best fit option.

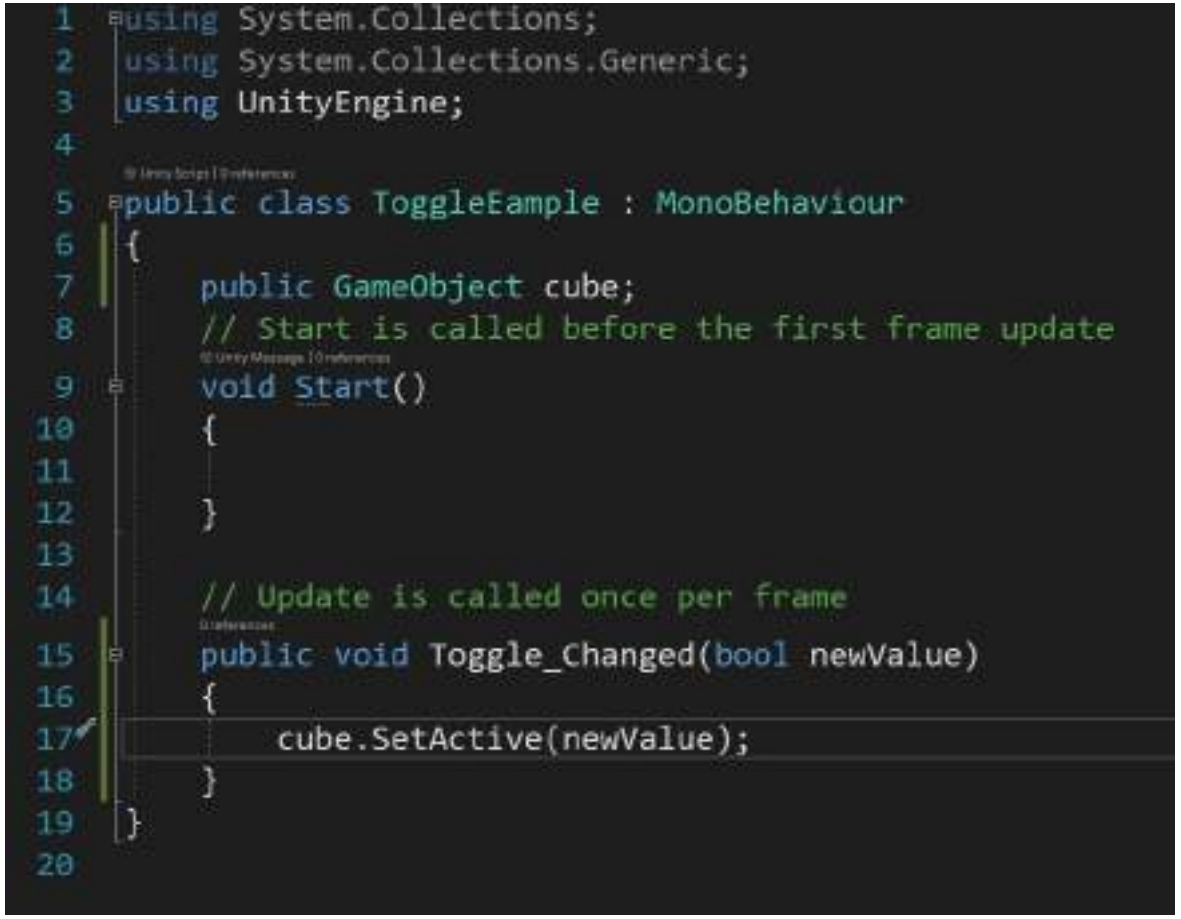


Rect transform settings

Check best fit option

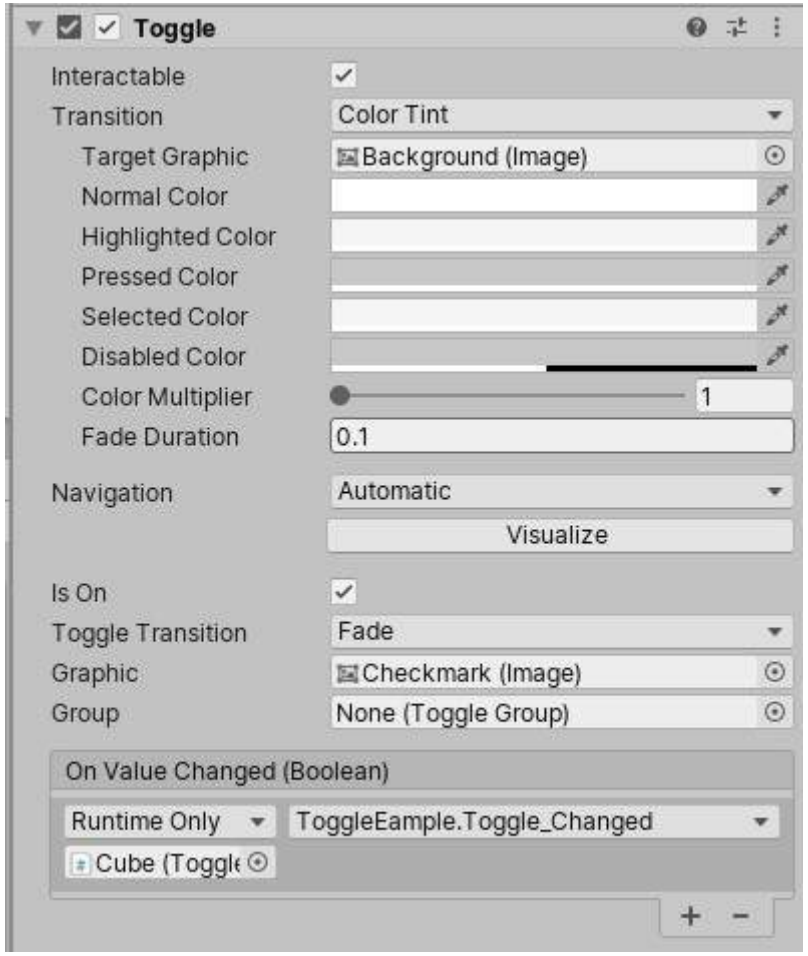
3. Create a gameobject “Cube” from 3D object menu and attach a material with red colour.

4. Create a script ToggleExample and drag on the gameobject “Cube”.



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

public class ToggleEample : MonoBehaviour
{
    public GameObject cube;
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    public void Toggle_Changed(bool newValue)
    {
        cube.SetActive(newValue);
    }
}
```

5. Drag the “Cube” to the On Value Changed (Boolean) section and select ToggleExample.Toggle_Changed.

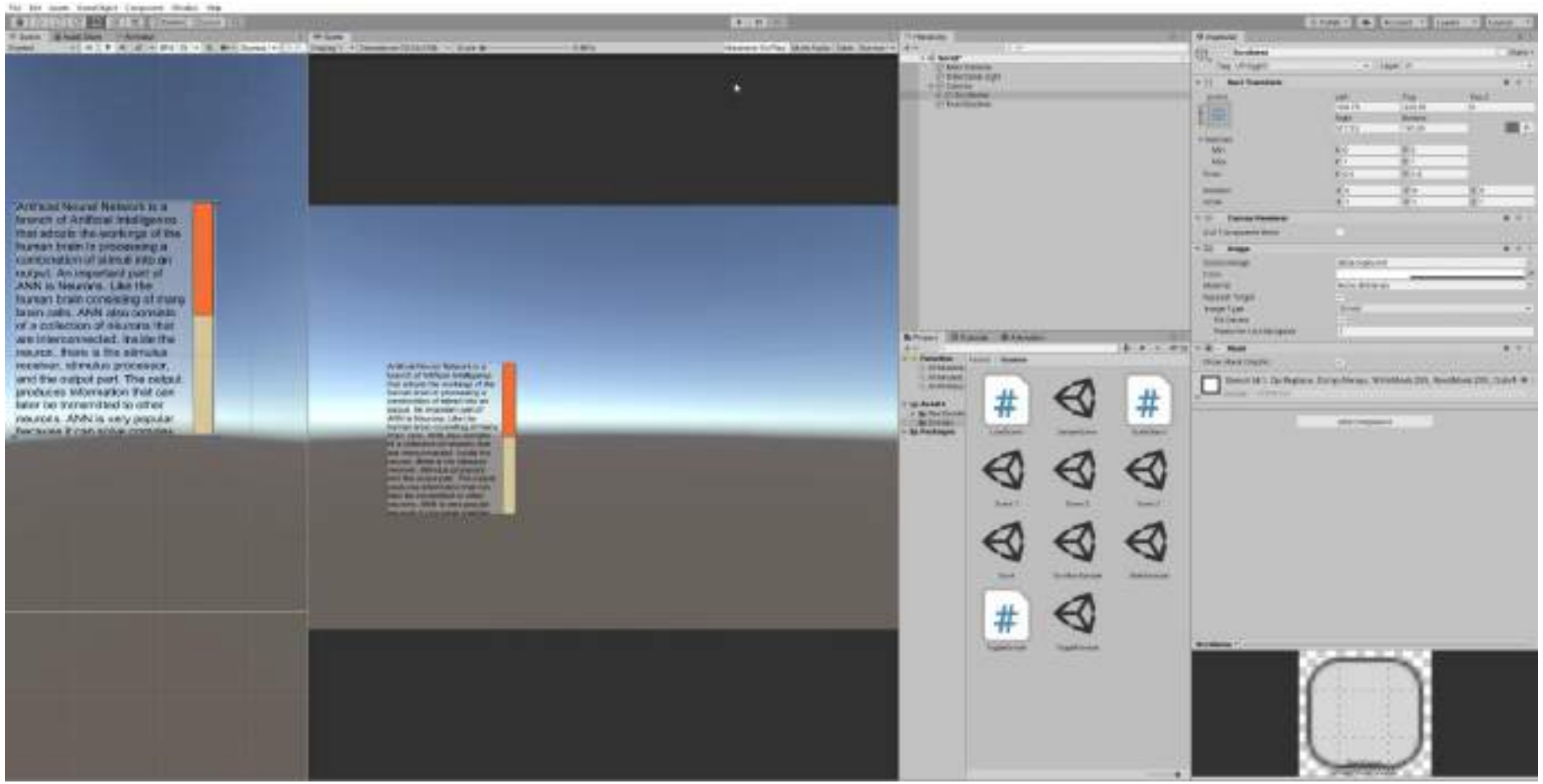


6. Hit the Play button and check and uncheck the toggle to see the resutls.

Class activity 8.

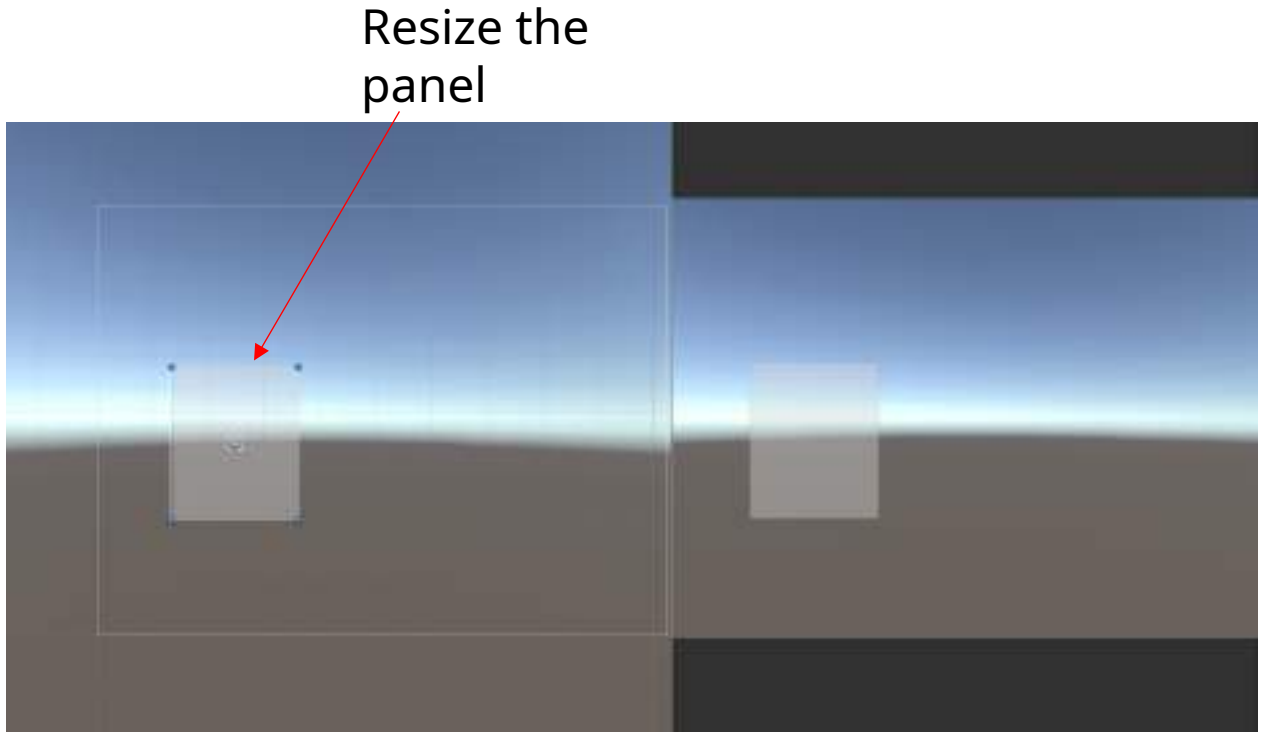
This example demonstrate the application of a scrollbar to scroll a text.

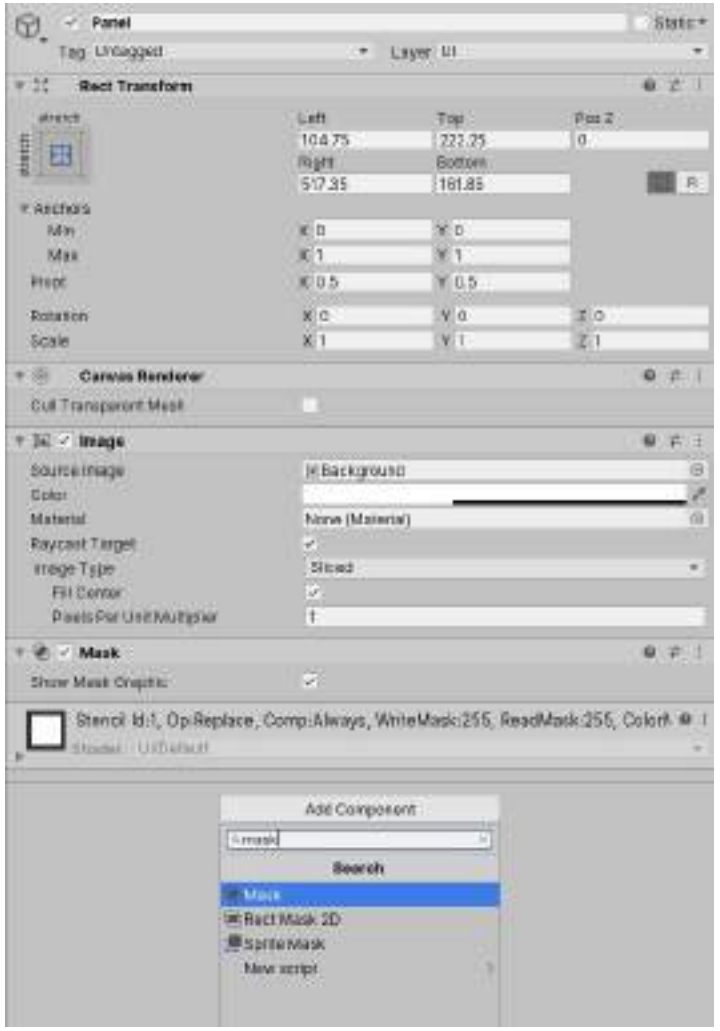
UI element: **Scrollbar**



Steps:

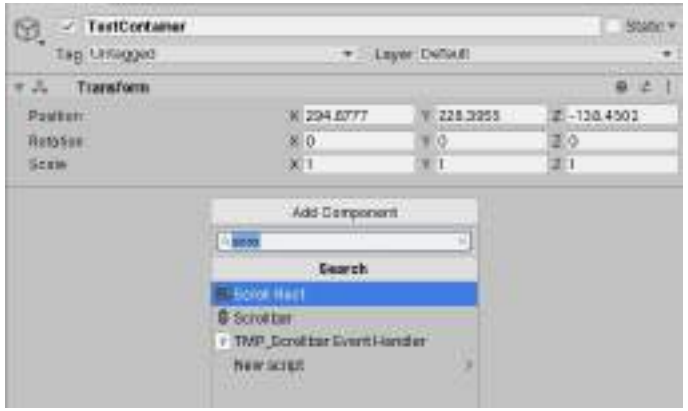
1. Create a panel and resize it to cover the gameview and add “Mask” component from the “Add component” menu.





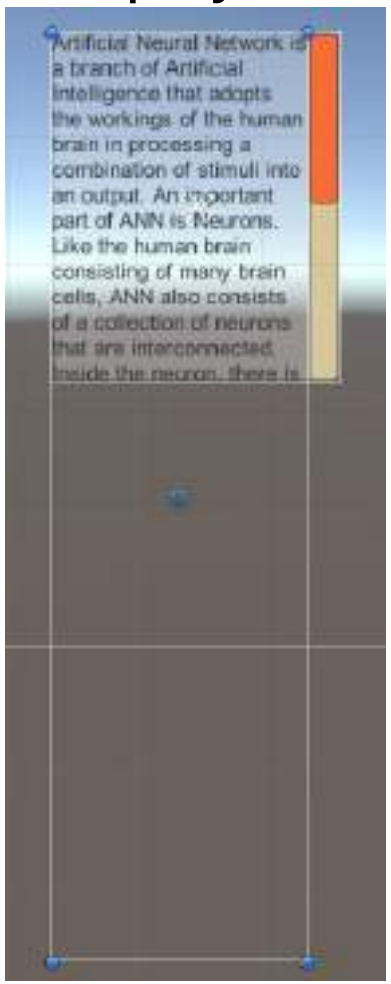
2. Create an empty gameobejct under the “panel” and name it “TextContainer”.

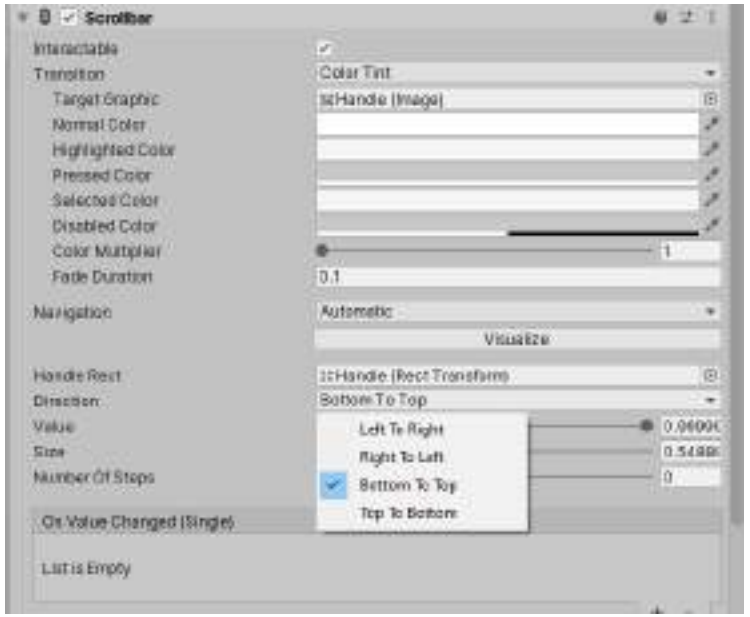
2. Match boarders of “TextContainer” with the boarders of “Panel”.
3. Add “Scroll Rect” compoents in “TextContainer”.



4. Create a “Scrollbar” from the UI menu under “TextContainer”.
5. Create a “Text” from UI menu also under “TextContainer”.

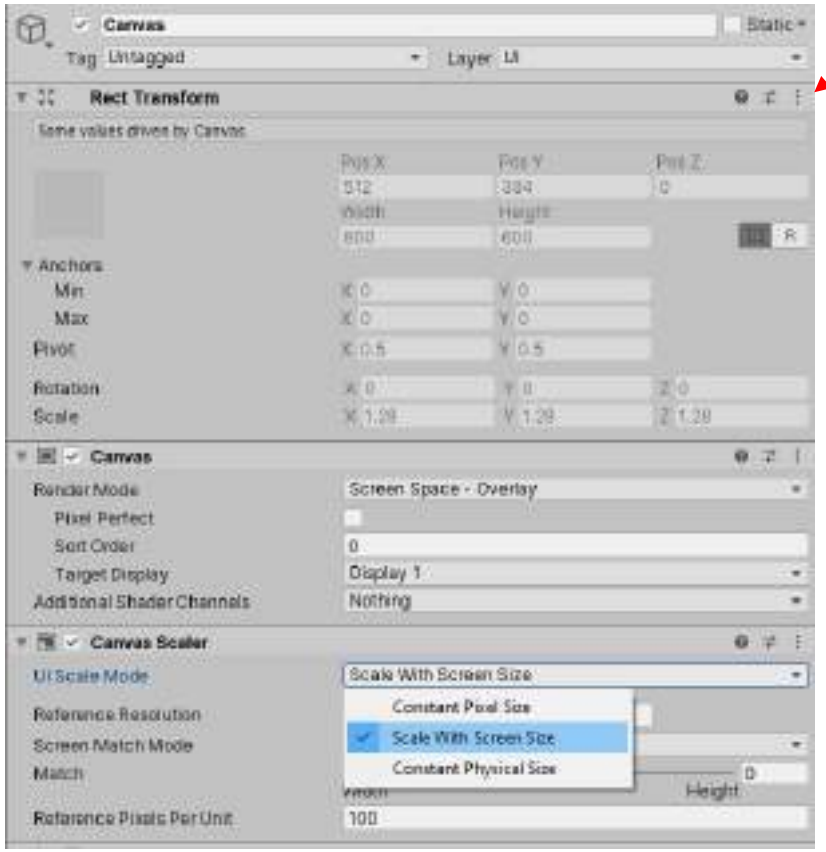
6. Add any random text with at least 12 lines in the Text UI and match its boarders with the panel and drag it down to display the text on scrolling.

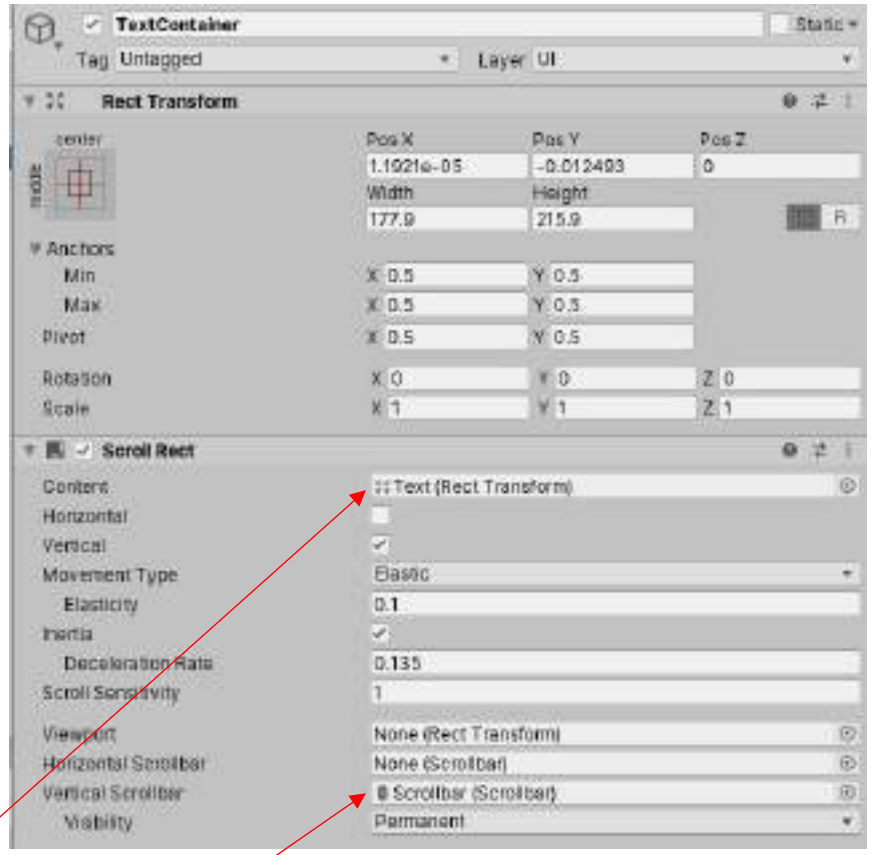




7. Set the direction of scrollbar “Botton to Top”.

8. In the Canvas inspector, set the UI scale mode as “Scale with screen size”





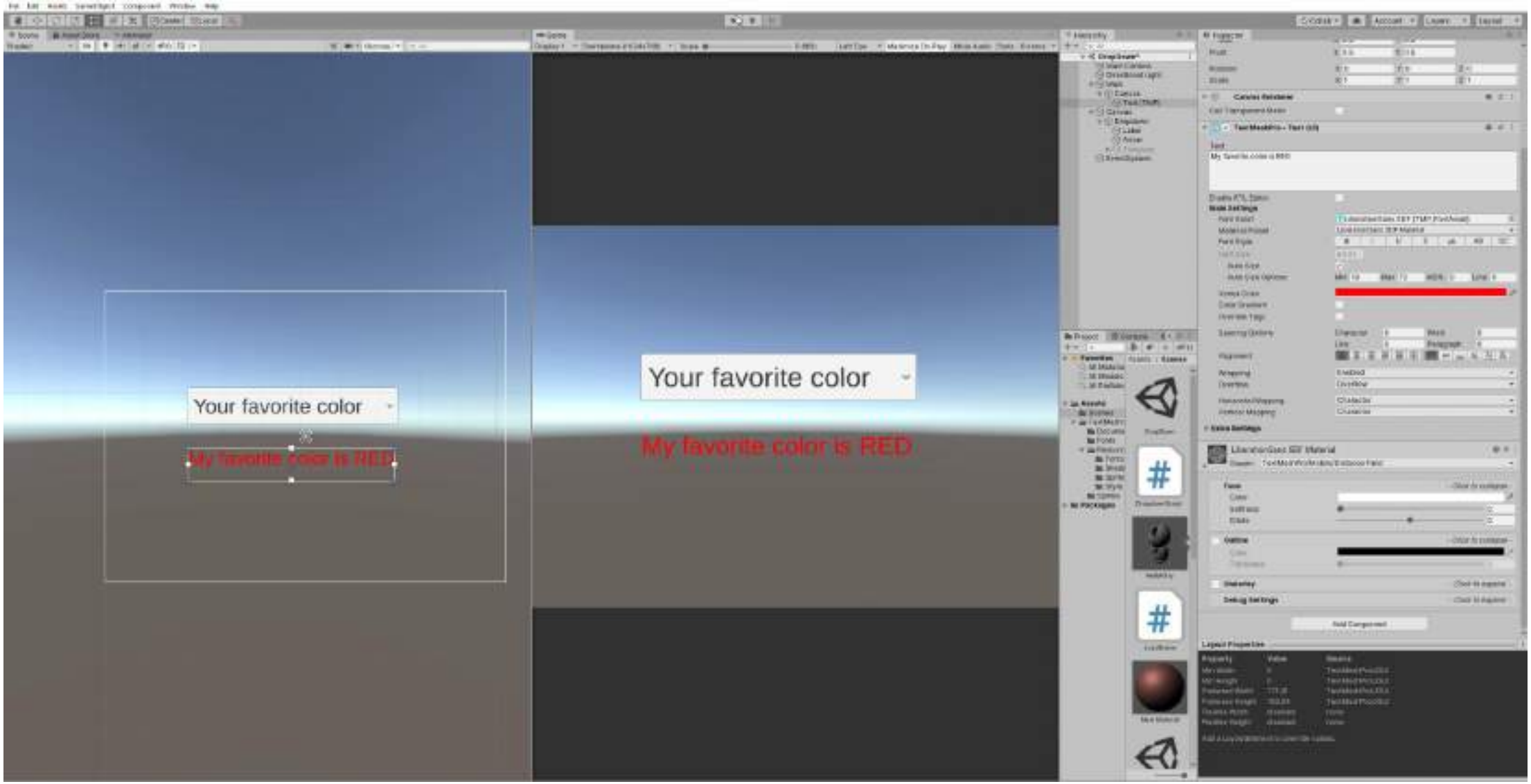
9. Drag the “Text” UI to the “Content” section of the “Scroll Rect” compoents of “TextContainer”.

10. Drag the “Scrollbar” UI to the “Vertical scrollbar” section of the “Scroll Rect” compoents of “TextContainer”.

11. Hit the play button and use the scrollbar to scroll the text.

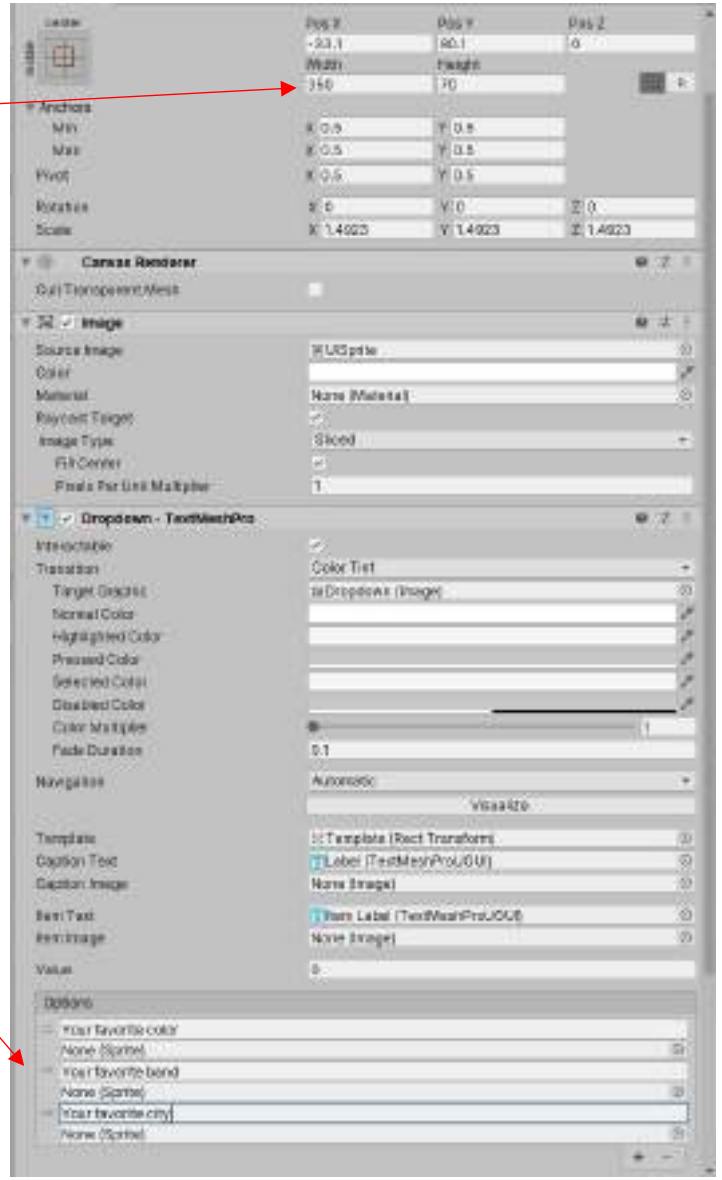
Class activity 9. This example demonstrates the application of Dropdown UI.

UI element: **Dropdown**



Steps:

1. Create a “Dropdown – TextMeshPro” from the UI menu.
2. Resize the width 350 and height 70.
3. Change the text in the three options in “Options” section by replacing option A, B and C with “Your favorite color”, “Your favorite band” and “Your favorite city”.

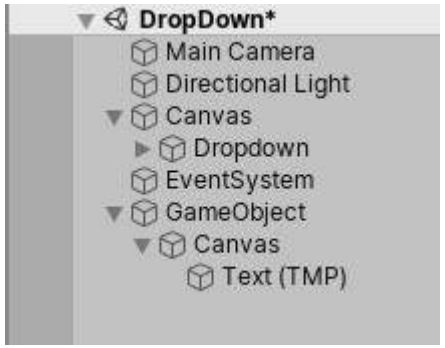


4. Write a C# script.

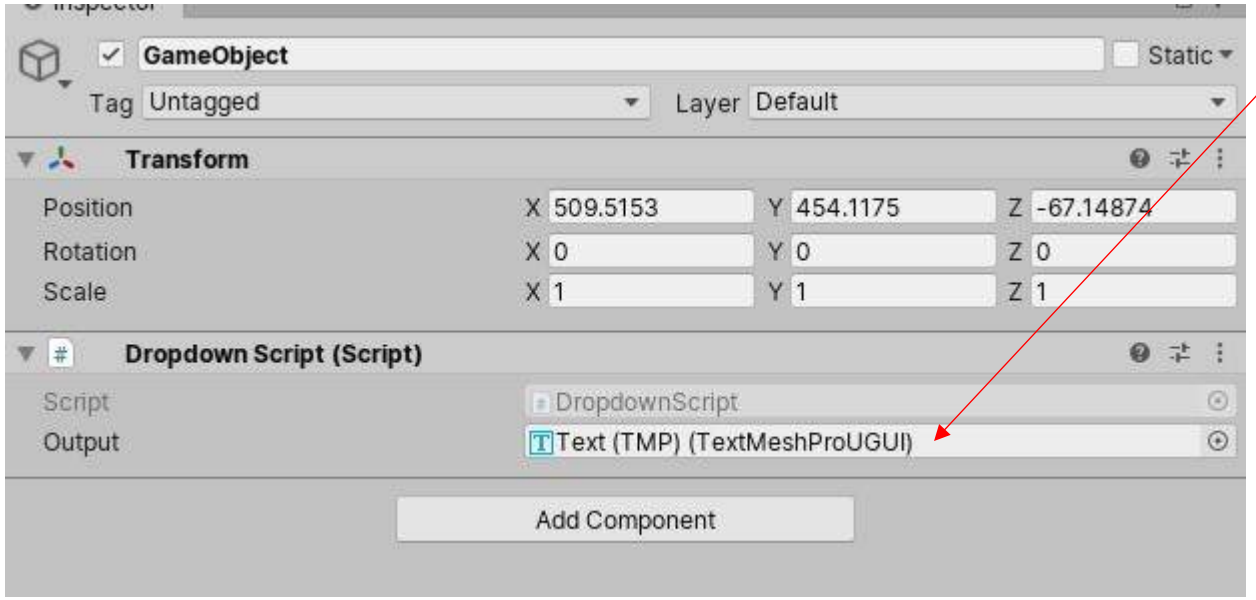
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;

public class DropdownScript : MonoBehaviour
{
    public TextMeshProUGUI output;
    // Start is called before the first frame update
    public void HandleInputData(int val)
    {
        if (val ==0)
        {
            output.text = "My favorite color is RED.";
        }
        if (val == 1)
        {
            output.text = "My favorite band is BTS.";
        }
        if (val == 2)
        {
            output.text = "My favorite city is TAIPEI.";
        }
    }

}
```

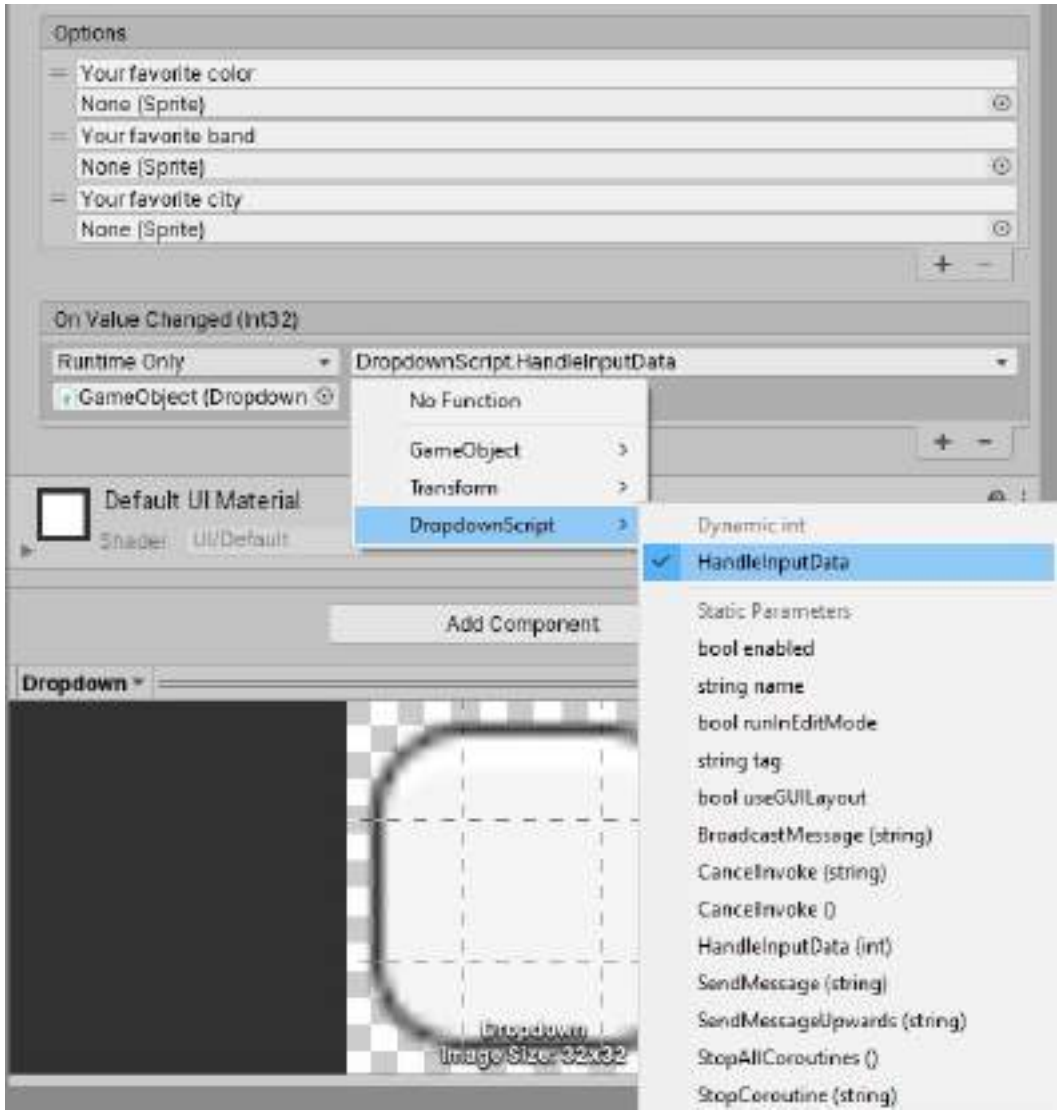
5. Create an empty “GameObject” and attach the script to it.

6. Create a “Text – TextMeshPro” from UI menu as a Child of “GameObject”.



7. Drag the “Text (TMP)” to the output section of the script.



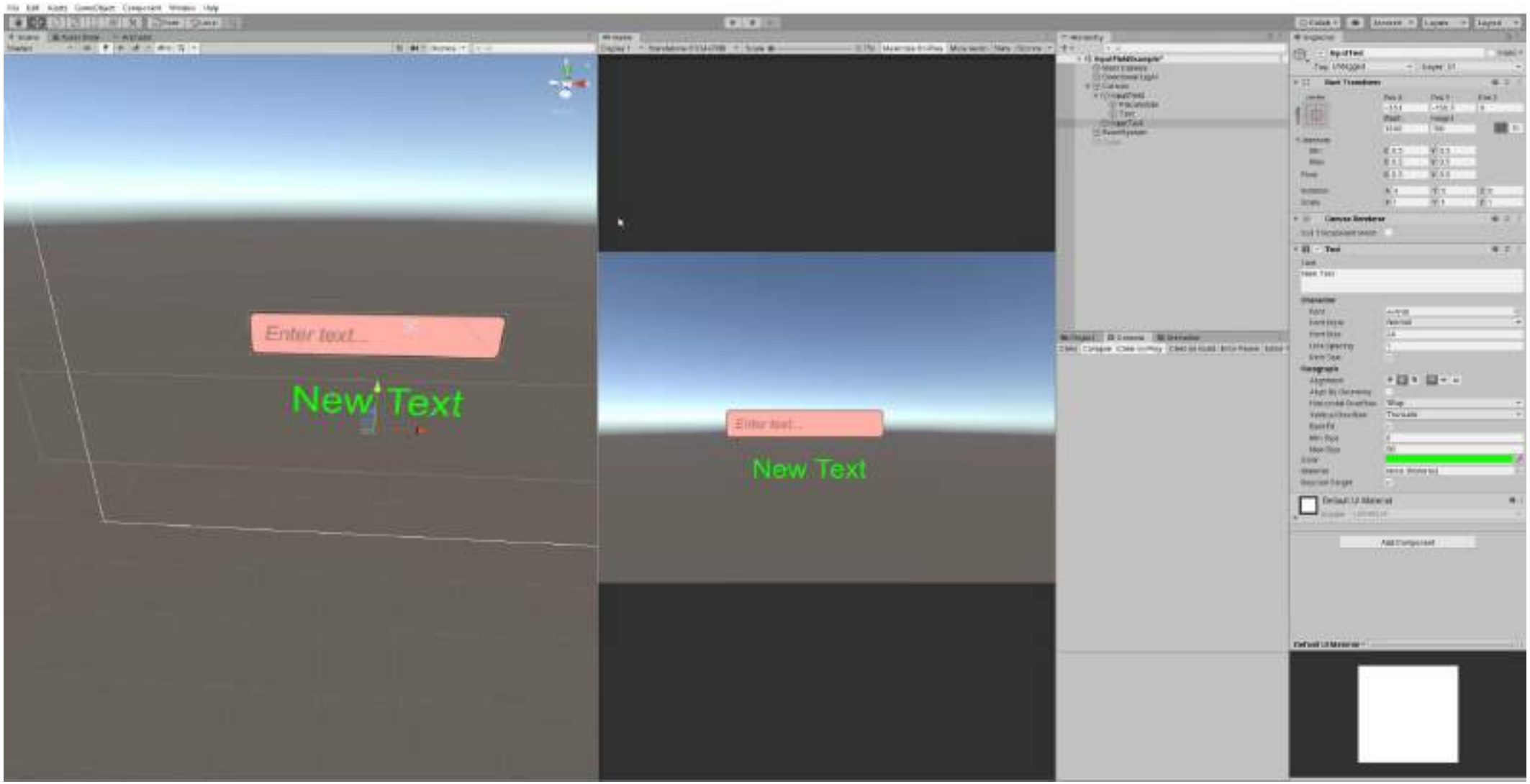


8. Drag the “GameObject” to the “On Value Changed(Int32) option of the inspector of “Dropdown” and select DropdownScript.HandleInputData.

9. Hit the play button and select the options to check the answers.s

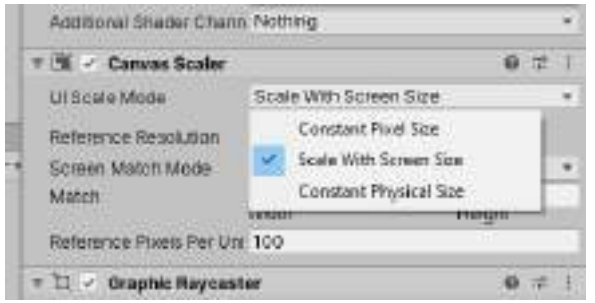
Class activity 10. This example shows the application of InputField UI.

UI element: **InputField**



Steps:

1. Create an Input Field UI and change its background color.
2. Change the Canvas scalar settings to “Scale with screen size”.
3. Write a script to display the text in the Canvas entered in “InputField”.

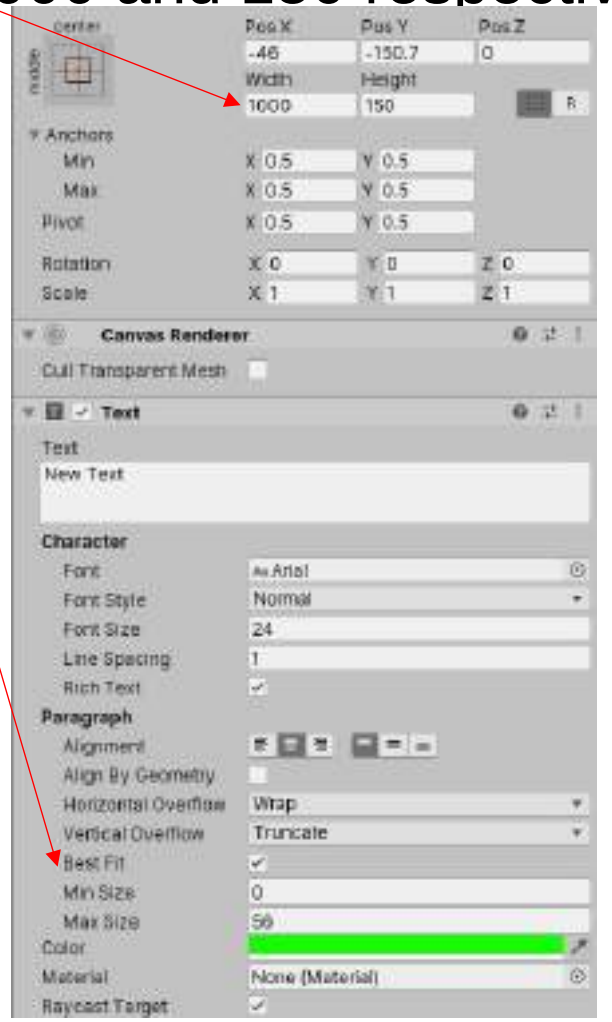



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class InputField : MonoBehaviour
{
    private string input;
    public Text Newtext;
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        Newtext.text = input;
    }
    public void ReadStringInput (string s)
    {
        input = s;

    }
}
```

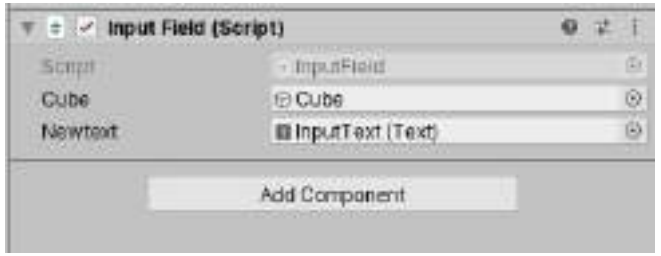
4. Attach the script to the “Main Camera”.

5. Create a “Text” under Canvas and name it “InputText”.

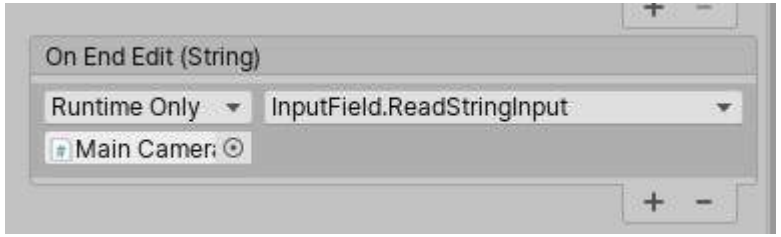
6. Set the Width and Height 1000 and 150 respectively and keep the font size 24 and check the best fit option.



7. Drag the “InputText” to the “NewText” option of the script.



8. Drag the “Main Camera” to the “On End Edit (String)” event of the “Input Field” and select InputField.ReadStringInput.



9. Hit the play button and write any text in the Input Field and enter to check the displayed text on Canvas.

Assignment 1: This example demonstrates the assembly of Nut and Bolt on a plate with a Hole by clicking the buttons to select the respective component and clicking the “Assemble” button to perform the assembly of respective components.