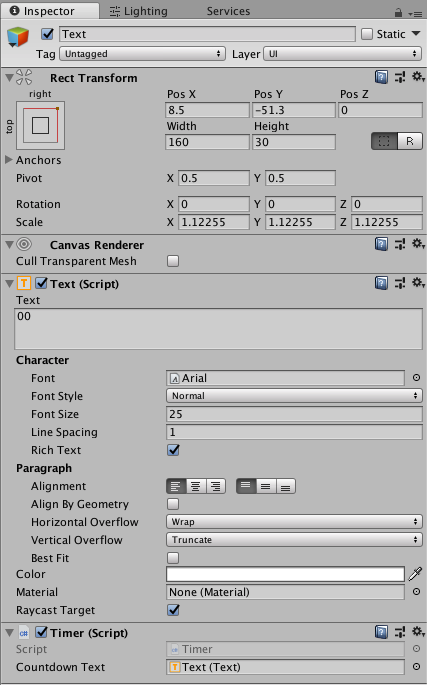
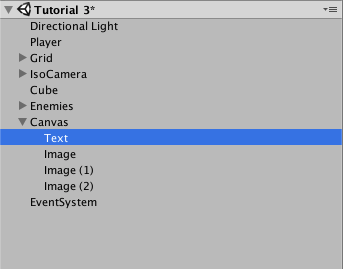
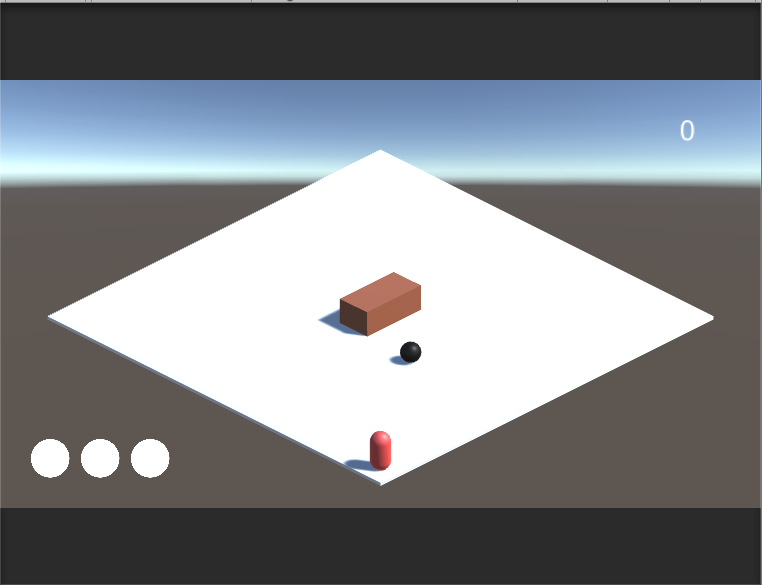
**Countdown, changing text colour and game over**

To set up our timer, first open the canvas, right click, select UI and click Text. We’ll set the text to display 00 as a placeholder but this will be overridden.

****

By selecting the square in the top right of the the rect transform, we can anchor our text to a corner of the screen and use the pos X and Y values to adjust it.

****

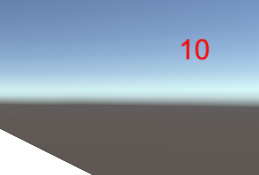
Create a new C# script and assign this to your text object. To start, we’ll add using UnityEngine.UI; to the systems list so we can access the UI. We’ll then create a float value for our currentTime (set to 0) and a startingTime (set to 60). The current time will use starting time and decrease it in value. We’ll then add [SerializeField] Text countdownText – this is used to access the text that we’ve placed on our canvas.

****

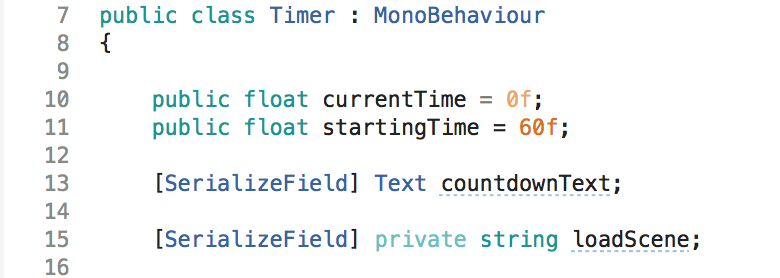
To ensure that our current time takes the value of our starting time, we’ll add currentTime = startingTime to the start function. In update, we’ll decrease the current time by 1 every second by using currentTime -= 1 \* Time.deltaTime. We’ll then access the countdown text and equate it to the current time using a string. To do this, we’ll type countdownText.text = currentTime.ToString(“0”). The timer should now function but will continues in to the minus values.

****

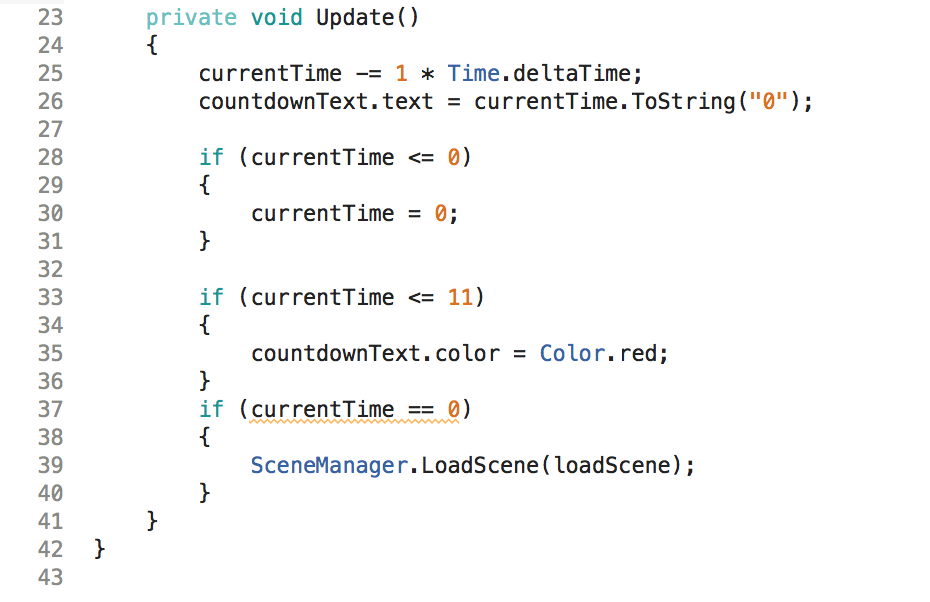
To ensure that the timer stops at 0, we can add the if statement if (currentTime <= 0){ currentTime = 0;). In addition, we can also set the text to change colour upon reaching a specific number by adding the if statement if(currentTime <= 11){ countdownText.color = Color.red;}.

****

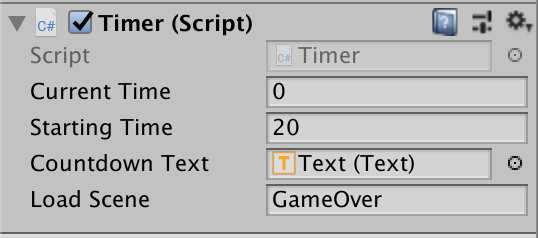
Next, we’ll set the timer to send the player to the game over screen if they run out of time. To do this, add using UnityEngine.SceneManagement; to the start of the script. Next set up a private seriliazed string to allow us to input the scene. This can be done by adding [SerializeField] private string loadScene.



Finally, we’ll add an if statement to send the player to the gameover screen on reaching 0 seconds. To do this, we’ll type if (currentTime ==0){ SceneManager.LoadScene (loadScene);.



In the timer script attached to the text, drag and drop the text from the canvas in to Countdown text and set the Load Scene to GameOver.



Now, we should have a timer that counts down from 60, flashes red upon reaching 10 seconds and will send the player to the game over screen if they run out of time.