Countdown Timer Coding Tutorial

Coding

* ***Unity:***
* Right click on the hierarchy (the section on the left hand side of the screen), select UI, and then image (if you are struggling to find the image or anything else in the tutorial, click on what you want to search for in the hierarchy, hover over the scene without clicking anything, and press F).
* Stretch the image out so it looks like a small rectangle, then place the image in the middle of the larger rectangle (you will need to zoom out in order to see this rectangle).
* Right click the hierarchy again and select UI, text.
* Place the text inside the rectangle you have just created, and name the text “Countdown” inside the “Text” section in the Inspector (the section on the right hand side of the screen).
* Go to the hierarchy and select UI, button, and place it below the rectangle with “Countdown” in it (remember to click “Button” in order to drag it around, not “Text,” as it will only drag the text instead of the whole button). This will be used for later.
* ***CountdownTimer:***
* Right click the project panel (below your game screen) and select Create, C# Script, and name it “CountdownScript.” Double click your new script and it will open Visual Studio (or something similar).
* Once you’re inside the script, get rid of the 2 starting functions (“// Start is called before the first frame update” to the second } from the bottom of the code, or more simply, lines 7-17) so there are only 2 {}’s below the public class “CountdownScript” line.
* Below the line “using UnityEngine;”, type “using UnityEngine.UI;” to allow the UI to be used in the code.
* Click back to your first class (in between the two {}’s on line 8), and type “[SerializeField]” so you can make the private variables visible. Then type “private Text uiText;”. We use “Text” because we’re using the UI, and ALWAYS make sure to put a “;” at the end of your line of code.
* Underneath the line of code you have just created (and still inside the {}’s), create another “[SerializeField]” but this time, type “private float mainTimer;”, as this is going to be the variable for the timer.
* Enter a new line and type “private float timer;”, representing the variable to count down the timer (not mainTimer because that represents the maximum amount of seconds that the timer will count down from).
* Enter another new line and type “private bool canCount = true;”
* Enter another new line and type “private bool doOnce = false;”
* Below these private variables, type “void Update()” and {}’s, then press enter.
* Inside the {}’s, type if(timer >= 0.0f && canCount), and then type {}’s (this is asking if the timer is greater than or equal to 0)
* In the new {}’s below the code above, type timer -= Time.deltaTime;
* Below this, type uiText.text = timer.ToString(“F”);” (this means that the code identifies the text as a String (a word)).
* Start a new line, but under the {}’s you have just made, and type else if (timer <= 0.0f && !doOnce) then create another {}’s.
* Inside the new {}’s, type canCount = false; then underneath that type doOnce = true; type underneath that uiText.text = “0.00”; and finally underneath that, type timer = 0.0f. This means that when the timer reaches 0, the timer won’t count down anymore and it will stop the timer.
* ***Back To The Hierarchy:***
* Save your current code, then head back to Unity. Once there, right click in the hierarchy (again, the section on the left hand side of the screen), right click and select Create Empty, which will create an empty object called “GameObject” in the hierarchy.
* Next, right click the Transform area in the Inspector (the right hand side of the screen, similar to the hierarchy) and click Reset to reset the transform values (X,Y and Z). Back in the hierarchy, click on and rename the GameObject to “TimerController.”
* Inside your Project (the bottom of the screen), drag the CountdownScript (the code you have worked on above) into the Inspector, and you will see it has been added to “TimerController.”
* Inside the CountdownScript script in the Inspector, you will see “uiText” and “MainTimer.” Set the “MainTimer” value to 5, and drag “Text” (under Image in the Hierarchy) from the Hierarchy to uiText in the Inspector
* ***Back to CountdownTimer (And Creating the Button):***
* Head back inside the CountdownScript, and create a new line in between “private bool doOnce = false” and “void Update().” Type void Start () followed by {}’s and press enter when in between the {}’s (the same as what you did to create the void Update class earlier).
* Inside the {}’s, type timer = mainTimer; Once you have done this, save your code and head back to Unity and press the Play button. You should see the countdown timer working, counting down from 5 (or the number you have set) to 0, not continuing into minus numbers, and clicking the button doesn’t do anything.
* Head back to the code and above the last } in the code, create a new line and type public void ResetBTN() and a set of {}’s.
* Inside the {}’s, type timer = mainTimer; below this type canCount = true; and underneath that type doOnce = false; Save the code then head back to Unity.
* When inside, click on “Button” in the Hierarchy, and over in the Inspector, click on the small + button where it said “On Click ().” Now click and drag “TimerController” to the Inspector where it says “Runtime Only.” After you have done this, click on No Function (found next to “Runtime Only”), and select CountdownScript, ResetBTN () in the drop down menu. (Keep in mind that this is within the Button. If you struggle to find where everything is, remember to click back on Button back in the Hierarchy).
* Once this is complete, save the code and what should happen is that when the timer reaches 0, you can click the reset button and it will countdown from your set number once again. Congratulations, you’ve created a Countdown Timer in Unity! :D
* Video this tutorial was inspired by: https://www.youtube.com/watch?v=E6qEPJUAZNk