Adding Animal Counter and Level Timer

Hello everyone! Welcome back to the game development tutorial using UNITY!

In the last video, we have completed our jungle scene where some animals are animated and some are stationary to make it look interesting. Also, we have coded for animal selection which is a requirement for our gameplay.

In this video, we will add the counter which counts the number of animals once selected. Also, we will add a timer in Level 1 to ensure that the level is completed within a stipulated time. If the level didn't get completed in a stipulated time, then we are going to learn how to put in a pop up message that says, "Game over! Please try again."

Let's start by opening UNITY HUB! Select the saved project named 'Food Web'. First, we are going to merge the Level1Controller with GameController to make it easier for functioning. To do that, go to GameController, click on Add Component and type Level 1 Gameplay and press enter. Level1Controller is added successfully and delete the Level1Controller from the hierarchy window.

Now, open the canvas and select the "Fox" tab. You will see the object script is missing as we have moved the script previously attached to the GameController. So, let's update it. We need to update the object for all animals separately. To do that, select 'Fox' and drag GameController to the Object section of Event Trigger component. From the drop down, select Level1Gameplay and click on 'SelectAnimal' option. Similarly, repeat the same steps for all the other animals like deer, lion, leopard, Snake, Eagle, Rhino, Rat, Sparrow and Grasshopper.

Once finished, press Control + S to save the scene directly. Now that we have updated the scripts, let us run and see if it is working! Let's zoom in a bit by dragging the slider. Select the animals and I can see everything is working fine.

Let's go back to the edit mode again. Scale it to minimum by using the slider so as to see the complete screen.

Before proceeding further with the timer, let's organize the canvas a little more. For that, right-click on canvas and click on 'Create Empty' object. Rename it as

'Forest'. Select all the objects in the list from Forest background to Grasshopper and drag them to the forest object. Now, it looks more arranged.

Let's create another UI element called 'Text'. For that, right-click on canvas and select 'UI' and from the list select 'Text' object. Select color option from Hierarchy window and change it to white. Double click on the Text to highlight it. Increase the size of the box, change the font size to 64 and align it centrally. Also, change the height to 72 and change the text to 'Animal Remaining:' Resize the box, make it left aligned and drag it to the top left corner of the screen. Add the number of animals i.e. 14 in the text. Rename it to 'Animals Remaining Text'.

Next, we need to update the text every time an animal is selected. To do that, let's jump to the project window and select Level1Gameplay and double click to open it in Visual Studio.

To update the text every time, first we will access the text label that we have created there. So, to access it, we're going to write the code:

using UnityEngine.UI;

And inside the public class, we will define

{

public Text animalsRemainingText;

But what is the maximum number of animals that we have?

Let's go to defines and write

Public static int totalAnimals = 14;

Now, go to the Level1Gameplay tab. Next we want to update the number of animals in the text as soon as the animal is selected. For that, we'll define another variable called private integer animals count.

Private int animalsCount;

To set the value of animals count, we are going to define the start function in the public class

Private void Start()

And in the start function we are going to set the animal count by writing

{

AnimalsCount = Defines.totalAnimals;

}

This function is to be applied to the animals remaining text. So, we can do that by writing

animalsRemainingText.text = "Animals Remaining: " + animalsCount.ToString();

But this is something that we are going to continuously keep on updating. So, what we are going to do, we can create a variable to update animal by writing

Private void updateAnimals()

And copy the animals remaining text function to paste it in the update animals function and reduce the count by writing

animalsCount--;

What animals count minus minus does is that every time an animal is selected, it reduces the count so as to display it on the screen.

Next, when do we call this animal update function? So, we are going to call it as soon as an animal is selected i.e. in the select animal function. For that, write a code

UpdateAnimals();

So, what we have done here is that first we have declared 2 variables i.e. calling animals remaining text here which is the text box that we have created in the scene and other is animals count which is keeping track of how many total animals are currently present in the game. In this start of the game, we have set animals count to total animals which is the default value of 14 and then we have set the value of animals count to animals remaining text. Then we have created one function called update animals which reduces the number of animals and displays it on the text every time an animal is selected. Then the update animal function updates the text every time the animal is selected.

Let's go back to the Unity editor and update this function into the scene.

To update it, select the GameController object and drag Animals Remaining Text to the script field in the Inspector window. Now, click on the play button to test it.

You can see that as soon as the animal is selected, the counter keeps decreasing.

After selecting all animals, still the count is showing 3 remaining. So, let's check on the number of animals we have in our game. I can see that we have 11 animals in total so let's update the code accordingly. Switch back to the Defines sheet and change the total animals count from 14 to 11.

Next, I want to show a pop-up screen that says 'Great job! You have completed this level!' as soon as I have finished selecting all the animals. To do that, right-click on Canvas, select UI and click on Panel option. Rename it to 'Game Complete'. Again right-click on Game complete, select UI and click on Image option. Resize & position it as shown on the screen. Change the colour to your choice. Rename it 'Popup Image'.

Right click on the image, select UI and click on Text option. Resize & reposition it appropriately. Change the text to Congratulations!. Tick the option Align by Geometry & Best fit. Change the font size to 128, font size to bold, center the text. Resize to 94 and position it to set appropriately on the screen. Rename it to 'Congratulations Text'.

Again right-click on Congratulations text and select duplicate. Drag it below on the screen. Rename it 'Level Complete Text'. Change the text to 'You have selected all the animals' and change the size to 64. Again duplicate the text and drag it below. Change the text to 'Next Level Starts in….5'

Now, go to the asset in project folder and create a new folder called 'Prefab'. In the Hierarchy window, rename the 'Game Complete' object to 'Level 1 Game Complete'. Drag it to the Prefab folder to make it as a prefab. Now delete it from the hierarchy because we want to instantiate that at the runtime. We do not want it to be present in the scene and only appears when the level is complete.

Next, let's go to the Level1Gameplay script. As soon as the Level 1 is complete, let's define a variable as

Public GameObject levelCompleteScreen;

And we are going to instantiate it in the UpdateAnimal function. We write it as

If(animalCount == 0)

then

{

GameObject _levelComplete = Instantiate(levelCompleteScreen, parentCanvas);

}

Let's define the parent canvas as

Public Transform parentCanvas;

Now, this code will help the message pop up as soon as the level is complete.

Go back to the editor and let unity refresh the script.

Select the GameController object. Once selected, you will see that 2 more public variables that we defined starts appearing. Parent Canvas is the field which will help in showing the new screen. So, drag Canvas and place it in Parent canvas. Then drag Level 1 Complete prefab and place it in the Level Complete Screen. Save your scene and click play to see the result of it. So, as you can see the game is working perfectly so far.

Now, we'll start working on the 'Next level starts in' sentence. Go back to the edit mode and create a timer to play this level which indicates how much time is left to complete the game.

To do that, right-click on Animal Remaining Text and select duplicate. To move it to the right side, get on to the rect transform window, click the fourth box from the left, which should be right at the intersection of right and top. Now rename this as "Time Remaining Text". Also, change the text to 'Time Remaining: 30' and change the alignment to left side. Reposition time remaining & animals remaining text to make it more visible. Disable the Raycast for Animal Remaining Text so as to not interfere with the animal selection. Similarly, disable it for Time remaining text. Don't forget to save the scene.

Let's code for Time Remaining Text. To do that, go to the Level1Gameplay script in visual studio.

We'll define the variable as

Public Text timeRemainingText;

Next go to defines script and create a new variable as

Public static int totalLevel1Time = 30;

Switch back to Level1Gameplay script tab. Now, we want the timer to reduce time by 1 second until it reaches 0. To do that, we will create an enumerator by writing

IEnumerator Level1Timer();

{

Yield return null;

}

And declare a temporary variable & code for timer count in reverse order as

Int _tempCount = Defines.totalLevel1Time;

While (_tempCount > 0)

{

_tempCount--; ***(Read it as tempCount minus minus)***

timeRemainingText.text = "Time Remaining: " + _tempCount.ToString();

yield return new WaitFor Seconds(1);

}

Debug.Log ("The timer is complete");

So, what does this code do is that it starts the timer in reverse order and start reducing the count by 1 second.

Now, we want the timer to start as soon as the Level 1 begins. For that, let's create a new function after start function by writing

Public void Init()

{

StartCoroutine(Level1Timer());

This will only start if the function is called.

Go to the start function and define it as

Init();

Go to the defines time as reduce the time from 30 to 10 for testing purpose. Switch to editor and click on play. As I can see the time is not reducing, there's an error found.

Go to the console section next to project section and see the issue. The error tells us that though we have coded the time remaining module but we didn't link it to our game scene. So, let do so.

To do that, select GameController. Drag Time Remaining Text object to the blank field of Level 1 Gameplay. Save your scene and click on play to check again.

Let's leave the screen running and from the Windows section in the top bar, select general and click on console. You will see that the message appears on the screen as soon as 10 seconds are over. So, our timer function is complete for the game.

Go back to the Edit mode. Right-click on Canvas and select 'Create Empty'. Rename it to 'Level1 Texts'. Select the 2 texts and drop it in the Level1 Texts object. This will make the text appear always on the screen.

The next screen that we want is the one which indicates that the time is over when someone is not done finding all the animals. For that, right click on the canvas, select UI and click on 'Panel'. Rename it as "Level1 Time Over".

Right click on it, select UI and click on 'Image' object. Resize & reposition it accordingly in the center. Rename it as 'Popup image'. Change the colour as per your choice. Right click on popup image, select UI and click on Text object. Resize it appropriately & reposition at the top. Change the font size to 64 and change the text to 'Game Over' in the white box.

Right-click on the text and select duplicate. Drag it to the center and change the text to 'Better Luck Next Time!'. Select the Game over text object and change the font style to bold.

We'll add button to this popup image. To add button, right-click on Popup image, select UI and click on Button object. Resize and reposition it appropriately. Expand the button, select Text option and tick on Best fit option. Change the text to 'Replay'. Rename the button object to 'Replay Button'.

Let's go to the project window and drag Level1 Time Over and make it as a prefab. Delete it from the hierarchy window.

Now, go to the Level1Gameplay script and below time remaining text function, define a public function as

Public GameObject timeOverScreen;

What we want now is as soon as the time is up for Level 1, all the touch functions gets disabled and the game over message should pop up on the screen.

For that, go to the timer function, delete the debug line code and write the new code in place of it as

GameObject _tempScreen = Instantiate(timeOverScreen, parentCanvas);

Go back to the editor and select GameController. Drag Level1 Time Over object from prefab and drop it in the Time Over Screen field. Click on Play and see what happens.

As you can see, the game is working fine. Let's code for the replay button now to make it functional.

For that, go to the assests, go inside scripts, right-click and create new C-sharp script. Rename it as ReplayGame. We're creating this script separately so that we can use the same script in the future for our other games. Double-click on it to open the new Replay game script.

Let's code this now.

We'll delete the start & update function and define a function for replay. To do that, write

{

Public void PlayAgain()

And in the beginning define the function as

Using UnityEngine.SceneManagement;

And then come back to play again function and write

{

SceneManager.LoadScene("Forest Scene");

}

Let the script gets refreshed. Now, go to assets again. Select Prefab and double click on Level1 Time Over object. It will open the Prefab editor. Select Level1 Time Over object. Click on Add component in the inspector window and type 'Replay Game' and press enter.

Select Replay button object where we have on-click event added. Click on + to show the click function. Drag Level1 Time Over object from hierarchy window and drop it in the object field of on click function. From the drop down, select Replay game option and click on Play Again function.

Click on the small arrow to come back again to scene. Click on play and see what happens. As you can see, once I click on replay, the game starts from the beginning with timer getting reset to the maximum.

Go to Level1Gameplay script.

One thing we noticed here is that one more condition will come into focus now. If the player is able to select all the animals in time and congratulations screen appears, then the time timer should get stopped too as we do not want it to appear on the screen in this condition.

There are 2 ways to do that. One way is that we let the timer completely run and say whenever the game gets over, the screen shouldn't get pop up on the screen. The other way is to define the variable and calling the function to stop the pop-up.

So, let's define a variable after levelCompleteScreen as

Private bool isLevelComplete;

And in the start function, define as

isLevelComplete = false;

In the Level1Timer function, we will modify the while condition as

While (_tempCount > 0 && !isLevelComplete)

It means that until the timer count is greater than 0 or the level is not complete, the timer will keep running otherwise it will break the loop and the code that will run is

If(!isLevelComplete)

{

}

And move the instantiate object code inside the if loop and it will show the time over screen.

So, we can say that with animal remaining count function, we are checking the count for animals selected and as soon as the animal remaining count become 0 then we say

isLevelComplete = true;

Just to recap, we have created a variable called isLevelComplete which is set to false in the beginning and as soon as the animal remaining count reaches to zero then the condition isLevelComplete become true which indicates that the level is complete and in the timer, we have put the condition in while that, if the level is complete, skip the code mentioned in the loop and if the level is completed, don't show the time over screen.

Also, go to defines tab and change the total level time from 10 to 30 seconds.

Go back to the editor and click on play. Let's select all the animals within the limited time and see what message pops up.

So, as you can see the congratulatory message pops up on selecting all the animals and time remaining is also stopped. It means that the code written is working properly.

So far, we have completed all the parts of Level 1 game play which is selection of animals in stipulated time and if not we can replay the game.

From the next video onwards, we will jump into the level 2 of this game where we are going to display our selected animals and create a food web out of them through connections.

So, keep practicing & exploring unity.

See you in the next video! Bye bye.