

Creating Level 2 and Animal Grid

Hello guys!

Welcome back to the game development tutorial using Unity! In the last video, we have finished designing and coding Level 1.

In this video, we will begin designing and coding Level 2 where we will first change the scene and then display different animals in a grid form. So, let's get started.

Open Unity hub and double click on "Food web" project.

Maximize the screen and set the game scene. Before we proceed further, let's play the game created so far. Select all the animals to see the next level congratulatory message. But in order to start the next level, I need the counter to go from 54321 and then display the next level.

To do that, go back to the edit mode. Double click on the Level1Gameplay script in the project window to open the code editor.

Once the Level1Gameplay script opens, you can see that we have instantiated the levelCompleteScreen but we haven't made it functional. Let's come back to unity editor and see which screen are we talking about. Click on Assets, open Prefabs and double click on Level1 Game Complete to see the text message with count down.

To make this functional, come back to code editor and begin writing a new code as shown on the screen by creating another enumerator as

IEnumerator StartLevelSwitchTimer()

```
{
```

```
yield return null;
```

```
int _timer = 5;
```

```
while (_timer > 0)  
{  
    _timer--;  
    yield return new WaitForSeconds(1);  
}  
}
```

Once the complete loop is executed, I want to write in the next line

```
GetComponent<GameController>().UpdateGameState(Defines.GameStates.L  
evel2_StartScreen);
```

Switch to GameController script and define the case for level 2 inside game states as shown on the screen by writing

```
case Defines.GameStates.Level2_StartScreen;  
{  
    debug.Log("Switch to level2");  
}  
break;
```

Now, switch to Level1Gameplay script and start the timer inside the if loop of animal counts

```
startCoroutine(StartLevelSwitchTimer());
```

This will start the countdown timer as soon as the selection of all the animals is complete.

Go to unity editor and click on play. Select all the animals from the scene and the congratulatory message pops up on the screen. Open the game console by click on window option, selecting general and clicking on console. You can see

in the log that we are able to switch to level 2 successfully after 5 seconds but the counter is not working still.

To make that functional, we'll go to assets and select scripts folder. Click on '+' and select C-Sharp script. Rename it to 'Level1Complete'. Double click to open the script in code editor and switch back to unity editor.

Go to assets, prefabs and double click on Level1 Game Complete. As you notice in the hierarchy window, we have a string 'Next Level Starts in 5' which we will take as a reference while coding. So, let's switch to Level1Complete script and begin coding as shown on the screen by writing

using UnityEngine.UI;

Delete the start & update function as we do not require them. Instead write the code

```
{  
  
public Text timerText;  
  
public void UpdateTimerText(int _time)  
  
    {  
  
        timerText.Text = "Next Level Start in..." + _time.ToString();  
  
    }  
  
}
```

Next, we have to call this function which we call in Level1Gameplay script. But before doing that, let's make the levelcomplete game object as public by copying GameObject _levelComplete and pasting it after the private Boolean function. Also, remove the GameObject initialisation from the instantiation line of code as we have define the same globally at the beginning.

Now, come back to the while loop and write the code as shown

```
_levelComplete.GetComponent<Level1Complete>().UpdateTimerText(_timer);
```

So, what happens in these lines of code is as soon as all the animals get selected, we instantiated level1 complete pop up screen and start the timer. In the timer function, we pass the value of timer to Level1Complete script which in turn updates the timer text.

Let's switch to Unity editor. Now, we need to add Level1 Timer class to the prefab. So, double click on Level1 game complete prefab, select Level 1 game complete from hierarchy window, click on add component in the inspector window, type Level 1 Complete and press enter. Drag Level Complete Text 1 from hierarchy to Timer text field. Click on play to check the progress.

Select all the animals in the scene. You can see that the timer is functioning properly and when the timer stops, we get the message as switch to level 2.

Now, let's switch to level 2 and create the level 2 screen.

To do that, click on the play button to switch to edit mode again. Expand canvas option and right click on canvas and select Create empty option. Rename it as Level 1. Select Forest and Level 1 Text and drag it to Level 1 folder.

Again right click on canvas, go to UI and select panel option. Rename it as Level 2. Right now, the panel is transparent, but we are going to add colour to it. Click on colour option in the image window and increase the alpha to 255. Select the background attached to panel by selecting it and pressing the delete button on the keyboard. This changes the colour of background to white now.

Now select Level 2 and right click on it. Select UI option and click on Text. Align the text by clicking on anchor presets, press shift key & alt and choose the center of the screen. You can do this by dragging too. Also, expand it from end to end.

Change the x value to 0, height to 140, change the alignment to center, make it best fit and make the text alignment to the center too. Now I'm going to type my text- "create food web by connecting animals together. Don't forget to disable the raycast target and save the scene. Rename the text to Title.

Again right click on Level 2 and select create empty object. In this empty object, I want the selected animals to get displayed in grid form and then I am able to

connect them with one another by drawing lines. Rename the game object to Animal Grid.

In this grid, there will be 11 animals in total and 2 plants where one will be a tree and another one will be a plant. So, it makes a total of 13 objects altogether.

In the animal grid, click on add component in inspector window and type grid. Select Grid Layout Group option and press enter. Expand the padding menu. Change the cell size to 200 for X & Y both. Change the child alignment to Middle center. Change the constraint to Fixed column count. Change the count to 4. Now to test this out, right click on animal grid, select UI and click on image. Change the colour to red. Right click to duplicate it and keep duplicating it. As you can see the grid is getting created however, it is not looking nice as there is no spacing between them.

So, select animal grid option and tweak the spacing by trial and error. Enter the final values of 250 X 100 for spacing and for cell size, modify it to 150 X 150. Next, select the move tool and drag the object down as shown on the screen using the green arrow.

Now, the spacing is set. So, select all the images by pressing & holding shift key. Right click on the last image and select delete option. Save the scene. Let us move to our next task of displaying the animals selected in level 1.

Select Level2 option and double click on Scripts. Right click inside the window, select create and chose C sharp script. Rename it to 'Level2Gameplay'. Let's quickly attach this script to the game controller itself. To do that, select GameController, drag Level2Gameplay and drop it in the inspector window.

Now, let's write a script for the same. To do that, double click on GameController to open the code editor.

Begin by defining a function inside the public class gamecontroller as

```
public GameObject level2Screen;
```

After the define.gamestates, start a function as

```
private void start()
```

```
{  
  
level2Screen.SetActive(false);  
  
}
```

This sets the screen to not show as soon as the game begins. Also, instead of debug logs, we will write

```
level2Screen.SetActive(true);
```

This command activates the level 2 screen as soon as level 2 starts.

Go back to unity editor and select GameController object. Drag Level2 from hierarchy window and drop in the Level 2 Screen script field. Click on play to test it.

As you can see that the level 2 screen have started appearing however, there is a bug where the congratulations screen is still appearing. So, to debug this, go to the edit mode again and switch to scripting mode. Go to Level1Gameplay script. Write the script after the while loop as

```
destroy(_levelComplete);
```

This command destroys the level 1 screen as soon as it gets complete.

Let's test this code and see if the error is rectified. As you can see, the bug is resolved now. Let's proceed further with adding animal grid on the screen.

To do that, switch to code editor and select. Level2Gameplay script. Delete the start and update function. First, we want to display the sprites of the all animals that we have. To do that, define the function as shown on the screen by writing

```
using UnityEngine.UI;
```

And inside the public class, we will write

```
public sprite [] animalSprites;
```

Now, go back to unity editor and let the script get refreshed. Expand Animal sprites. Now go to assets, images and animals. Select GameController object and lock it.

Expand deer and drag first image of it and drop it to animal sprites. Similarly, drag all the other animal images to the same place as shown on the screen. Also, I want to reference the animals in the same sequence.

To do that, switch to code editor and select defines script tab. After totalleveltime, write the code as

```
public enum AnimalsList
```

```
{
```

```
Deer,
```

```
Eagle,
```

```
Fox,
```

```
Giraffe,
```

```
Grasshopper,
```

```
Lepoard,
```

```
Lion,
```

```
Rat,
```

```
Rhino,
```

```
Snake,
```

```
Sparrow
```

```
}
```

Next switch the tab to GameController and create a list after define.GameStates as

public List<Defines.AnimalsList> animalsLists;

In the start function, we will define it as

animalsLists = new List <Defines.AnimalsList>();

Outside the start function, we will write a function as

public void UpdateAnimalList(Defines.AnimalsList _animalselected)

```
{  
  
}
```

Update animal list is going to add animals to the list whichever animals are getting selected so that this list can be referenced in level 2 which will be used to create the grid. To do that, continue writing the code inside parenthesis as

animalsLists.Add(_animalsSelected);

To make it functional, go to the Unity editor and click on the lock to unlock it. Select Level2 canvas and uncheck level 2 for some time. Next, expand level1 and forest.

Next, go to assets and then scripts. Right click inside the window, select create and chose C sharp script. Rename it to AnimalType. Double click to open in code editor. Delete the start and update function from the script. This script will contain information about the type of animals we are referencing to and it will be attached to all the animals.

To do that, we will define it as

```
{  
  
public Defines.AnimalsList animalType;  
  
}
```

Go back to the unity editor and select all the animals by pressing and holding control key. Click on Add component in inspector window. Type AnimalType and

press enter. Now, select fox from the hierarchy window and from the drop down in animal type script, select fox. Similarly, repeat the steps for all the animals.

Now, go back to code editor. Select Level1Gameplay script. In the SelectAnimal function, we have to update the selected animals to the animal list which is created in the gamecontroller.

In the selectanimal function, write the code

```
GetComponent<GameController>().UpdateAnimalList(_animalObject.GetComponent<AnimalType>().animalType);
```

Switch to Unity editor. Go to assets and go inside prefab. Select Level 2 and enable it by ticking on it and then minimise the canvases. Select Animal Grid canvas. Right click on it, select UI and click on Image. The size of the image is by default 150 X 150. Rename it to Animal Image. Drag it and place it in the prefab section. Delete it from the hierarchy window.

Go to level2Gameplay script in code editor and define the variable as

```
public GameObject prefabAnimalImage;
```

And after defining that, let's define a function as

```
public void Init()
```

```
{
```

```
//Create the grid and add all the animals in it
```

```
list<Defines.AnimalsList> _tempAnimalList =  
GetComponent<GameController>().animalsLists;
```

```
Foreach(Defines.AnimalsList animal in _tempAnimalList)
```

```
{
```

```
gameObject _tempAnimal = Instantiate(prefabAnimalImage,  
parentAnimalGrid);
```

```
_tempAnimal.GetComponent<Image>().sprite =  
animalSprites[((int)_animal]);  
  
}
```

Go at the top of the function and define

```
public Transform parentAnimalGrid;
```

What we did here is that we create a temporary animal list which is the reference of the animal list in game controller. This animal lists contains the sequence of animals how they are selected. Now, it goes through each and every element of this temporary animal list and against which we had created an image of each and every animal.

Now to apply the image of animal to that component we access the image component of the animal that we have created and we apply the animal sprite that we have taken reference of.

Let's discuss the same scenario in Unity to see how it works! We have taken the animal grid as the parent. Let's go to assets, prefab and animal image. Next, we have added animal image to the grid and we have set the sprites of animal grid based on these sequences of animals that we have selected.

Now, let's go to game controller and drag Animal grid to the Parent Animal Grid. Also, drag Animal image from prefab section to Prefab Animal Image field. Click on play to see how it is working.

As you can see the new screen appears, but the screen is empty, nothing has been added to it! This is because we have written all the code in the Init function of Level2Gameplay script which we haven't referenced anywhere.

So, let's go to GameController tab and as soon as we active the level2screen, we need to call the function here as

```
GetComponent<Level2Gameplay>().Init();
```

Come back to unity editor and hit play to check the progress. As you can see on the screen that all the animals are shown on the grid, like we intended!



So, that's it from this video. The next step for us is to create the food web, not to forget, also to include the two more elements to this list- one being the tree and the other one being the bushes. We will be looking at how to do that in our next video!

It has been a real delight learning with all of you! For now, keep practicing and having fun learning all these exciting parts of game development!

See you all very soon!