

ENGLISH VERSION

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.Level2_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 instantiate 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!

HINDI VERSION

Creating Level 2 and Animal Grid

Hello guys!

Unity का उपयोग कर game development tutorial में वापस स्वागत है! पिछले video में, हमने लेवल 1 की designing और coding पूरी की है।

इस video में, हम level 2 को design करना और code करना शुरू करेंगे जहां हम पहले scene को बदलेंगे और फिर विभिन्न जानवरों को ग्रिड रूप में दिखायेंगे। तो चलो शुरू करते हैं।

Unity hub open करें और "Food web" project पर double click करें।

screen को maximize करें और game scene set करें। इससे पहले कि हम आगे बढ़ें, चलो अब तक बनाए गए game खेलते हैं। next level के congratulatory message को देखने के लिए सभी जानवरों को चुनें। लेकिन next level को शुरू करने के लिए, मुझे 54321 से जाने के लिए counter की जरूरत है और फिर next level को दिखाने के लिए।

ऐसा करने के लिए, edit mode पर वापस जाएं। code editor open करने के लिए project window में Level1Gameplay script पर double click करें।

एक बार Level1Gameplay script open होने के बाद, आप देख सकते हैं कि हमने Level Complete Screen को instantiated कर दिया है लेकिन हमने इसे functional नहीं बनाया है। आइए unity editor पर वापस आते हैं और देखते हैं कि हम किस screen के बारे में बात कर रहे हैं। asset पर click करें, prefab open करें और count down के साथ text message देखने के लिए Level1 game पर double click करें।

इसे functional बनाने के लिए, code editor पर वापस आएं और एक नया code लिखना शुरू करें जैसा कि screen पर दिखाया गया है

IEnumerator StartLevelSwitchTimer()

```
{  
yield return null;  
int _timer = 5;  
while (_timer > 0)  
{  
    _timer--;  
    yield return new WaitForSeconds(1);  
}  
}
```

एक बार पूरा loop execute हो जाने के बाद, मैं next line में लिखना चाहती हूँ

```
GetComponent<GameController>().UpdateGameState(Defines.GameStates.Level2_StartScreen);
```

GameController script पर Switch करें और game के अंदर level 2 के लिए case को define करें जैसा कि screen पर लिखा गया है

```
case Defines.GameStates.Level2_StartScreen;
```

```
{  
  debug.Log("Switch to level2");  
}
```

```
break;
```

अब, Level1Gameplay script पर switch करें और animal counts के loop के अंदर timer शुरू करें

```
Start Coroutine (Start Level Switch Timer());
```

सभी जानवरों को चुनें जाते ही ये countdown timer शुरू कर देगा।

unity editor पर जाएं और play पर click करें। scene से सभी जानवरों को चुनें और screen पर congratulatory message होगा। window option पर click करके game console खोलें, general को चुनें और console पर click करें। आप log में देख सकते हैं

हम 5 second के बाद सफलतापूर्वक level 2 पर switch कर सकते हैं लेकिन counter अभी भी काम नहीं कर रहा है।

उस कार्य को करने के लिए, हम assets पर जाएँगे और scripts folder को चुनेंगे। '+' पर click करें और C-Sharp script चुनें। इसे 'Level1Complete' का नाम दें। code editor में script open करने और unity editor पर वापस जाने के लिए double click करें।

asset, prefab पर जाएं और level 1 Game Complete पर double click करें। जैसा कि आप hierarchy window में देखते हैं, हमारे पास एक string है 'Next Level Starts in 5' जिसे हम coding के reference के रूप में लेंगे। तो, चलिए Level1Complete script पर switch करते हैं और coding शुरू करते हैं जैसा कि screen पर दिखाया गया है

using UnityEngine.UI;

start & update function को delete कर दें क्योंकि हमें उनकी आवश्यकता नहीं है। इसके बजाय code लिखें

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

इसके बाद, हमें इस function को call करना होगा जिसे हम Level1Gameplay script कहते हैं। लेकिन ऐसा करने से पहले, हम Game Object _level Complete की copy करके और इसे private Boolean function के बाद paste करके level के level complete game को public बनाते हैं। साथ ही, Game Object initialisation को instantiation line of code से remove कर दें क्योंकि हमने शुरुआत में ही globally define किया है।

अब, while loop पर वापस आएं और जैसा दिखाया गया है उसके अनुसार code लिखें

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

तो, code की इन lines में क्या होता है जैसे ही सभी जानवर चुने जाते हैं, हम तुरंत level1 complete pop up screen को पूरी तरह से instantiated करते हैं और timer को शुरू करते हैं। timer function में, हम timer के value को Level1Complete script में pass करते हैं जो return में timer text को update करता है।

unity editor पर switch करते हैं। अब, हमें prefab में level 1 timer class को add करना होगा। तो, Level1 game complete prefab पर double click करें, hierarchy window से level 1 game पूरा चुनें, inspector window में add component पर click करें, level 1 complete type करें और enter press करें। Level Complete Text 1 को से hierarchy से Timer text field तक खींचे, progress को test करने के लिए play पर click करें।

scene में सभी जानवरों को चुनें। आप देख सकते हैं कि timer ठीक से काम कर रहा है और जब timer बंद हो जाता है, तो हमें message मिलता है switch to level 2।

अब, level 2 पर जाएँ और level 2 screen बनाएँ।

ऐसा करने के लिए, फिर से editor mode पर switch करने के लिए play button पर click करें। canvas option को expand करें और canvas पर right click करें और Create empty option चुनें। इसे level 1 नाम दें। Forest and Level 1 Text को चुनें और इसे level 1 folder में खींचें।

फिर से canvas पर right click करें, UI पर जाएं और panel option चुनें। इसे level 2 नाम दें।

2. अभी, panel transparent है, लेकिन हम इसमें color add करने जा रहे हैं। image window में color option पर click करें और alpha को 255 तक बढ़ाएं। panel पर attach background को चुनकर और key board पर delete button दबा कर उसे चुनें। यह अब background का colour बदलकर white कर देगा। अब Level 2 चुनें और उस पर right click करें। UI option चुनें और text पर click करें। anchor presets पर click करके text को align करें, shift key और Alt press करें और screen का centre चुनें। आप इसे खींचकर भी कर सकते हैं। इसके अलावा, इसे end से end तक expand करें।

X value को 0 में बदलें, height को 140 करें, alignment को centre में करें, इसे सबसे अच्छे से fit करें और text alignment को centre में भी बनाएं। अब मैं अपना text लिखने जा रही हूँ- “create food web by connecting animals together.” । raycast target को disable करने और scene को save करना न भूलें। text को title में बदलें।

फिर से level 2 पर right click करें और create empty object । , मैं चाहती हूँ कि चुनें जानवरों को grid के form में दिखाया जाए और फिर मैं lines खींचकर उन्हें एक दूसरे से connect करूं। game object का नाम बदल कर animal grid करें।

इस grid में total 11 जानवर होंगे और 2 plants होंगे, जिसमें एक पेड़ होगा और दूसरा एक पौधा होगा। तो, यह total मिलाकर 13 object बनते हैं।

Animal grid में, inspector window में add component पर click करें और grid type करें। Grid Layout Group option को चुनें और enter press करें। padding menu को expand करें। X और Y दोनों के लिए cell size को 200 करें। child alignment को middle centre में बदलें। constraint को Fixed column count में बदलें। count को 4. में बदलें। अब इसको test करने के लिए, animal grid पर right click करें, UI चुनें और image पर click करें। colour को red में बदलें। इसे duplicate करने के लिए right click करें और इसे duplicate करते रहें। जैसा कि आप देख सकते हैं कि grid बन रहा है, हालांकि यह अच्छा नहीं लग रहा है क्योंकि उनके बीच कोई खाली जगह नहीं है।

तो, animal grid option को चुनें और trial and error द्वारा spacing को adjust करें। spacing के लिए और cell size के लिए 250 X 100 final value enter करें, इसे 150 X 150 में modify करें। इसके बाद, move tool को चुनें और green arrow का उपयोग करके screen पर दिखाए गए object को नीचे खींचें।

अब, spacing set है। इसलिए, Shift key press करके और hold करके सभी images को चुनें। last image पर right click करें और delete option चुनें। scene

को save करें। आइए हम level 1 में चुने गए जानवरों को दिखाने के अपने के अपने अगले task पर जाएं।

Level2 option चुनें और script पर double click करें। window के अंदर right click करें, C sharp script चुनें और। इसे 'Level2Gameplay' का नाम दें।

आइए इस script को जल्दी से game controller से attach करें। ऐसा करने के लिए, GameController को चुनें, Level2Gameplay खींचें और इसे inspector window में छोड़ दें।

अब, उसके लिए एक script लिखते हैं। ऐसा करने के लिए, code editor open करने के लिए GameController पर double click करें।

public class gamecontroller के अंदर एक function को define करके शुरू करें

```
public GameObject level2Screen;
```

After the define.gamestates, start a function as

```
private void start()
```

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

```
}
```

यह screen को game शुरू होते ही नहीं दिखाने के लिए set करता है। इसके अलावा, debug logs के जगह, हम लिखेंगे

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

यह command level 2 शुरू होते ही level 2 screen को activate कर देता है।

Unity editor पर वापस जाएं और GameController object चुनें। Level2 को hierarchy window से खींचें और Level 2 screen script field में छोड़ें। इसे test करने के लिए play पर click करें।

जैसा कि आप देख सकते हैं कि level 2 screen दिखाई देना शुरू हो गया है, फिर भी एक bug है जहां congratulations screen अभी भी दिखाई दे रही है। तो, इसे debug करने के लिए, फिर से edit mode पर जाएं और scripting mode पर switch करें। Level1Gameplay script पर जाएं। while loop के बाद script लिखें

destroy(_levelComplete);

यह command पूरी होते ही level 1 screen को खत्म कर देगा।

आइए इस code का test करें और देखें कि क्या bug ठीक हो गया है। जैसा कि आप देख सकते हैं, bug अब ठीक हो गया है। आइए screen पर animal grid on the screen के साथ आगे बढ़ें।

ऐसा करने के लिए, code editor पर जाएं और चुनें। Level2Gameplay script। start and update function को हटाएँ। सबसे पहले, हम उन सभी जानवरों के sprites को दिखाना चाहते हैं जो हमारे पास हैं। ऐसा करने के लिए, function को screen पर दिखाए अनुसार define करें

using UnityEngine.UI;

और public class के अन्दर, हम लिखेंगे

public sprite [] animalSprites;

अब, unity editor पर वापस जाएं और script को refresh होने दें। Animal sprites का विस्तार करें। अब assets, images and animals पर जाएं। GameController object को चुनें और इसे lock करें।

Deer को expand करें और इसकी पहली image को खींचें और इसे animal sprites पर छोड़ दें। इसी तरह, अन्य सभी जानवरों के images को उसी स्थान पर खींचें, जैसा कि screen पर दिखाया गया है। साथ ही, मैं animal को उसी sequence में reference करना चाहती हूँ।

ऐसा करने के लिए, code editor पर जाएं और defines script tab select करें।
एसे करने के बाद, code लिखें

```
public enum AnimalsList
```

```
{
```

```
Deer,
```

```
Eagle,
```

```
Fox,
```

```
Giraffe,
```

```
Grasshopper,
```

```
Lepoard,
```

```
Lion,
```

```
Rat,
```

```
Rhino,
```

```
Snake,
```

```
Sparrow
```

```
}
```

इसके बाद tab को GameController पर switch करें और define.GameStates के बाद एक list बनाएं

```
public List<Defines.AnimalsList> animalsLists;
```

start function में, हम इसे define करेंगे

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

start function के बाहर, हम एक function लिखेंगे

```
public void UpdateAnimalList(Defines.AnimalsList _animalselected)  
{  
}
```

Update animal list को list में उन जानवरों को जोड़ेगी, जिन जानवरों को चुना जायगा ताकि इस list को level 2 में reference किया जा सके जो grid बनाने के लिए उपयोग किया जाएगा। ऐसा करने के लिए, parenthesis के अंदर code लिखना continue करें

```
animalsLists.Add(_animalsSelected);
```

इसे functional बनाने के लिए, unity editor पर जाएं और इसे unlock करने के लिए lock पर click करें। कुछ समय के लिए Level2 canvas को चुने और level 2 को uncheck करें। अगला, Level1 और forest को expand करें।

इसके बाद, assest पर जाएं और फिर script| window के अंदर right click करें, C चुनें और C sharp script चुनें, इसका नाम AnimalType पर रखें। code editor में open करने के लिए double click करें। script से start and update function को हटा दें। इस script में उन जानवरों के बारे में जानकारी होगी जिन्हें हम reference कर रहे हैं और यह सभी जानवरों से जुड़ा होगा।

ऐसा करने के लिए, हम इसे define करेंगे

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

Unity editor पर वापस जाएं और control key दबाकर और सभी जानवरों को चुनें। inspector window में Add component पर click करें। AnimalType type करें और enter दबाए। अब, hierarchy window और animal type script में drop down से fox चुनें , इसी तरह, सभी जानवरों के लिए steps को दोहराएं।

अब, code editor पर वापस जाएं। Level1Gameplay script को चुनें करें। SelectAnimal function में, हमें selected animals को animal list में update करना होगा जो gamecontroller में बनाई गई है।

Selectanimal function में, code लिखें

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

unity editor पर switch करें। assets पर जाएं और prefab के अंदर जाएं। level 2 को चुनें और उस पर tick करके इसे enable करें और फिर canvas को minimize करें। Animal Grid canvas को चुनें। उस पर right click करें, UI चुनें और image पर click करें। image size default 150 X 150 है। इसे animal image का नाम दें। इसे खींचें और इसे prefab section में रखें। इसे hierarchy window से delete करें।

code editor में Level2Gameplay script पर जाएं और variable को define करें

```
public GameObject prefabAnimalImage;  
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)];
}
```

function के top पर जाएं और define करें

```
public Transform parentAnimalGrid;
```

हमने यहां क्या किया है कि हम एक temporary animal list बनाएं जो game controller में animal list का reference है। इस animal lists में जानवरों के sequence शामिल हैं कि उन्हें कैसे चुना जाता है। अब, यह इस temporary animal list के हर element से जाता है और जिसके against हमने प्रत्येक जानवर की एक image बनाई थी।

अब animal की image को उस component पर apply करने के लिए हम उस जानवर के image component तक पहुंचते हैं जिसे हमने बनाया है और हम जानवर के sprite को apply करते हैं जिसका हमने reference लिया है।

आइए unity में same scenario पर discuss करें कि यह कैसे काम करता है! हमने animal grid को parent के रूप में लिया है। आइए assets, prefab and animal image पर जाएं। next, हमने जानवरों की image को grid में add कर दिया है और हमने जानवरों के grid के sprites को जानवरों के इन scene के base पर set किया है जिन्हें हमने select किया है।

अब, game controller के पास जाएं और Animal grid को Parent Animal Grid पर खींचें। साथ ही, animal image को prefab section से Prefab Animal Image field में खींचें। यह कैसे काम कर रहा है यह देखने के लिए play पर click करें।

जैसा कि आप देख सकते हैं कि new screen दिखाई देती है, लेकिन screen empty है, इसमें कुछ भी नहीं जोड़ा गया है! ऐसा इसलिए है क्योंकि हमने Level2Gameplay script के Init function में सभी code लिखे हैं, जिसे हमने कहीं भी refrence नहीं किया है।

तो, चलिए GameController tab पर जाएं और जैसे ही हम Level2screen active करते हैं, हमें यहां function को call करने की जरूरत है

Get Component <Level2Gameplay> () Init ()|;

Unity editor पर लौटें और progress को test करने के लिए play को hit करें। जैसा कि आप screen पर देख सकते हैं कि सभी जानवरों को grid पर दिखाया गया है

हमारे लिए अगला step food web बनाना है, , इस list में दो और elements को - एक पेड़ और दूसरा झाड़ियों को शामिल करना न भूलें। हम अपने अगले video में ऐसा करने का तरीका देखेंगे!

यह आप सभी के साथ काम करने और सिखने में बहुत मजा आया ! अभी के लिए, game development के इन सभी रोमांचक हिस्सों को सीखने का अभ्यास करें और मजे करें!

आप सभी को बहुत जल्द मिलेंगे!