

ENGLISH VERSION

Add the Timer and Scoring System

Hey everyone!

Super glad to see you all back here for game development using Unity! Today, we are going to focus on working with two new aspects of the game - scoring and timer! We will look at how to include a timer for level two and then we are going to look at adding a scoring system to our game! So, let's get started!

Let's open Unity hub and open the project that we have been working on.

Let's maximize the screen. In the hierarchy section, click on canvas, select level 2 and right click to choose UI and select panel option. Rename it as "level timer panel". Head to the colour section of the panel under the image window and change the alpha to 0 so that it doesn't look like a layer is present on top of it. Also, ensure to disable the ray cast target by unticking this box. We do this because we don't want this panel to be take any touch input. Now, jump back to the hierarchy window, right click on Level Timer Panel, select UI and click on text. Rename it as "timer text".

Now, I'm going to make sure that the timer text is positioned on the top left corner of the screen. To do that, we are going to select timer text and change the position and pivot by holding down the shift key and selecting the box at the first intersection on the top left corner. For selecting, press Shift + Alt key, then click the suitable option. Resize it as shown on the screen. Change the text to "Time Remaining". Also, tick the box that says best fit. Let the alignment to be on the left, but in the centre. We'll move the position of the text box to X as 50 and Y as -25.

Now, let's get started with the script. Go to the project window, assets folder and then inside the scripts. Right click, pick create and select C sharp script. Name it as "level timer". Now double click to open the code editor with Level Timer script.

Once opened, remove the start and update function as we don't need them. Begin by including UI function as

Using UnityEngine.UI;

And let's define the reference of the text box created by writing code as

```
{  
Public text levelTimerText;  
Public int totalTime;  
Public void StartTimer()  
{  
}  
IEnumerator Timer()  
{  
Yield return new WaitForSeconds(2);  
Int _counter = totalTime;  
While (_counter > 0)  
    {  
        _counter --;  
        levelTimerText.text = "Time Remaining: " + _counter.ToString();  
        yield return new WaitForSeconds(1);  
    }  
Debug.Log("Game over show result screen");  
}  
}
```

Inside the start timer function write the code as

StartCoroutine(Timer());

Now, jumping right back to unity editor. We are going to wait until it compiles all the scripts. Go to level timer panel, click on to the inspector window, type level timer and press enter. Drag timer text to the Level timer text field and change the Total time to 5 seconds for now, which basically means that the level 2 has to run for 5 seconds only. This is done only for testing purpose.

Now that we have written the value, that doesn't mean that it is going to start on its own, we are going to have to call the start timer function. To do that, get

to game controller and then double click on level two gameplay to open the Level2Gameplay script.

As you can see that whenever the level two starts, we were calling the function “Init”, so that all the animals are arranged in the grid. So, in the same function. We are going to write

//Start game timer

And then we are going to take reference of level timer script that we have written and then call the function start timer. Let's switch to Level2Gameplay and declare

Public LevelTimer levelTimer;

And to start the timer write

levelTimer.StartTimer();

Let's get back to unity editor and wait for the scripts to get compiled. Select game controller. Now we have to add level timer, before which I'm going to freeze it by clicking on this lock in the inspector window, this is just to ensure that it doesn't move. Then, I'm going to drag and drop level timer on to the box that says level timer. After which, I'm going to unfreeze the lock and run the project, to see what happens.

I'm going to now click the console window, come back to the game and select all the animals. Now that we are done with selecting all the animals, the “congratulations” screen pops up and we have cleared level 1. As you can see the timer for Level 2 starts after 2 seconds of screen switch only. Once the timer hits 0, we are shown the message “game over show result screen”. Our level timer is done, but the screen that's going to have to pop up after that, we are yet to create.

Go to level 2 in hierarchy window. Right click on it, select UI and chose panel option. Now, change the colour of the panel and make it darker. I'm choosing to stick with the shown shade. Then right click on panel, choose UI and select image. Let's increase the size of the image that has been added here. Rename it to “level clear panel”. We will put an image onto this. First, we will go to the

inspector window to click on colour option. I'm picking orange. Increase the size of image and reposition it as shown.

Now, right click on the image again, select UI and click on text. Align the text box to the top center as shown. Expand it so that it is stretched from one end to the other, increase the size to whichever amount you want, align it to the centre and tick the little box that says best fit. Write "well done" in the text box. Increase the size to 128 and bold it. Change the placement of the text by pushing it a bit down, entering a value of -25 into the Y box in the Rect Transform window. Rename it to title text.

Select image, right click, select UI and then choose text. Increase the size of this text box as well and align it to the centre. Change the text to Score:, make the alignment to center, font size to 81 and ensure to choose the best fit option again. Rename it to "Score text". Now, right click on the image, select UI and chose button. Reposition it as shown and increase the size of the button. Expand the button, click on the text, choose best fit and then type in "Play again" in the text field. Rename the button to "Play again button". Change its colour to green. As we have already discussed in previous videos that you can create a prefab out of level create panel. To do that, go to the asset folder, select prefabs and drag level clear panel and drop it into the prefab. Delete the "level clear panel" present in the hierarchy window.

Now, let's instantiate the level clear panel by writing the code in the LevelTimer script by defining it as

```
Public GameObject levelClearPanel;
```

Disable the debug log by using // and writing

```
GameObject _lvlclear = Instantiate(levelClearPanel);
```

We'll declare one more function as

```
Public Transform parentCanvas;
```

And make the game object by adding

```
, parentCanvas
```

Delete the debug log

Go back to unity editor and let it compile. As you can see, it is asking for level clear panel's reference. So, go to prefab folder, drag and drop level clear prefab to level clear panel field. Next, drag canvas from the hierarchy window and drop it to Parent canvas field. Let's run the game and see what happens.

As we can see the game is working fine so far. But we haven't written any functionality on the play again panel as of now. Basically, there are two functionalities that we have to write in here, one is play again. Play again is going to restart the whole scene again or the whole game again, and score is going to display the score which we will do once we implement the scoring system. So, let's implement the functionality of play again button.

To do that, go to the asset folder in your project and double click on level clear panel. If you remember, we had already written this functionality for level 1 time over. Here, we had the button called replay which was starting the whole game again. We are going to use the same functionality in play again button as well. So, go back to the asset folder and double click on Level Clear panel. Click on add Component in the inspector window, type in "replay game" and press enter. Now, go to play again button, go down in the button section. Click on + to add on click event. Drag and drop level clear panel in the object field and from the drop down, select replay game and click on play again function. Click on the small arrow in the hierarchy window to go back to the main window.

So, now that we have implemented the timer, so, we will implement the scoring system. To do that, go to hierarchy window. Just like we have created level timer panel, we can also directly right click on it, duplicate it and rename this to "score panel". Expand it and rename the "timer text" to "score text". We want it to be aligned to the right side, so, select score text, hold on to shift and change the position and pivot to right top side. Change the value to -50 for X and - 25 for Y. Change the alignment to right and change the text from "time remaining" to "score". Next, to add and update the score, we have to go to project folder, assets, scripts, right click on it and create a new script called "GameScore". Go to score panel, remove the level timer script that is attached to the score panel, because this is not a level timer. So, select "games score" in the scripts folder and drag and drop it on the score panel. This is another way to attach scripts. Now, double click on score panel to open the code editor.

Remove the update function as we don't need it. Let's begin coding by defining a function

```
Public UnityEngine.UI;
```

And declare the variables as

```
Public text scoreText;
```

```
Int score;
```

As soon as the scene is started, we would like to set the score by writing

```
{  
Score = 0;  
scoreText.text = "Score:" + score.ToString();  
}
```

Whenever the score is updated or reduced, we will call the function

```
Public void UpdateScore(bool _increaseScore)  
{  
    If (_increaseScore)  
    {  
        score = score + 2;  
    }  
    Else  
    {  
        score = score - 1;  
    }  
    If (score < 0)  
    Score = 0;  
    ScoreText.text = "Score: " + score.ToString();  
}
```

Go back to Unity editor, select score panel. Drag Score text from hierarchy window to Score text field. To increase or decrease the score, select level 2 and double click on LineDrawing script to open the script. This is the script where all logic is happening with respect to start or end animal or the line is drawn or not.

Let's create 2 variables after parentcanvas by writing

Defines.AnimalsList startAnimal;

Defines.AnimalsList endAnimal;

Next go to start line drawing function and define

startAnimal = AnimalScript.startAnimal;

Change the defines to game object by rewriting it as

GameObject startAnimal;

GameObject endAnimal;

Next, go to end line drawing function and write after the if loop

endAnimal = AnimalScript.endAnimal;

So, if you see, there's one condition that we have been working on if animal script is not equals to null, then destroy the line drawing. So, let me just go to unity and run the project and see how the lines are getting drawn. I just want to test whether the lines are getting created or not. As you can see that the lines are not getting created properly. So, click on play button again and switch to code editor to the linedrawing script.

In the end line drawing function you see that there is a small bug that we left initially. That is, we have said that if the animal script dot end animal is not equal to null, then destroy current line object, but rather we should say if the endpoint is not on any of the animals, then we should destroy this line. Otherwise, we should continue using this line. So, change the exclamatory mark to equal to. Also, add an else condition by writing

Else

{

endAnimal = AnimalScript.endAnimal;

}

Here, we would like to create a function which decides whether we are selecting the correct animal and dropping it on the right animal. For example, if I draw a line from lion to leopard, then it will be wrong answer. But if I draw a line from a lion to probably a giraffe, or a rhinoceros, then it will be correct answer because a lion can eat a giraffe and rhinoceros.

So, write a function as

```
Bool UpdateScore()
{
If(AnimalScript.endAnimal == null)
Return false;
Defines.AnimalsList          _startAnimal          =
startAnimal.GetComponent<AnimalType>().animalType;
Defines.AnimalsList          _endAnimal            =
endAnimal.GetComponent<AnimalType>().animalType;
If(!_startAnimal == Defines.AnimalsList.Lion)
    {
        If(_endAnimal == Defines.AnimalsList.Giraffe || _endAnimal ==
        Defines.AnimalsList.Rhino)
            {
                Return true;
            }
        Else
            {
                Return false;
            }
    }
Else if(_startAnimal == Defines.AnimalsList.Sparrow)
    {
        If(_endAnimal == Defines.AnimalsList.Grasshopper)
            {
                Return true;
            }
        Else
            {
                Return false;
            }
    }
Else if (_startAnimal == Defines.AnimalsList.Snake)
    {
```



```
    If(_endAnimal == Defines.AnimalsList.Sparrow || _endAnimal ==
    Defines.AnimalsList.Fox)
        {
            Return true;
        }
        Else
        {
            Return false;
        }
    }
Else if (_startAnimal == Defines.AnimalsList.Fox)
    {
        If(_endAnimal == Defines.AnimalsList.Rat)
            {
                Return true;
            }
        Else
            {
                Return false;
            }
    }
Else if (_startAnimal == Defines.AnimalsList.Eagle)
    {
        If(_endAnimal == Defines.AnimalsList.Snake || _endAnimal ==
        Defines.AnimalsList.Leopard)
            {
                Return true;
            }
        Else
            {
                Return false;
            }
    }
Else if (_startAnimal == Defines.AnimalsList.Leopard)
    {
```

```
    If(_endAnimal == Defines.AnimalsList.Deer)
    {
        Return true;
    }
    Else
    {
        Return false;
    }
}
Return false;
}
```

You can keep on modifying the combinations in food chain as per your requirement and complexity that you would like to bring in to the game.

Now, go to end line drawing and when there is an end animal available what we want to do is to update the score. For that, we have written the function in games score scripts. So, go to the parent canvas function line and write

```
Public GameScore gameScore;
```

And in the end line drawing function, write the code for it. Before that, delete the if condition in the update score function as shown on the screen as it is not needed now. Come back to else condition and write

```
Bool _increaseScore = UpdateScore();
```

```
gameScore.UpdateScore(_increaseScore);
```

One more thing that we can add to this functionality is whenever there is a correct answer, we would like to make the line that we are drawing between both the animals to be green. And whenever it's a wrong answer then the line colour to show red. To do that, continue writing the code as

```
If(_increaseScore)
```

```
currentLineRenderer.material.color = Color.green;
```

```
else
```

```
currentLineRenderer.material.color = Color.red;
```

Now, let us test all the code that we have written! We have written good amount of code so far, so it is important to check if everything is working fine as we intend for it to! Let's go back to unity editor and let it compile. Before running the project, let us go to level 2 in hierarchy window, drag score panel to the game score field present in the line drawing script. Now that we are done with all this, let us run and see what happens.

As the level 2 gameplay time is too short, let's increase by selecting Level Timer Panel and change the total time from 5 to 60 seconds. Let's launch and see what happens now.

For testing purposes, we are going to connect deer to lion, to see if it's correct or wrong. We can see the line turning red because it is the wrong answer. Now, let's connect the lion to rhinoceros, implying that the lion can eat the rhinoceros. It can be seen that the line has turned green. Now, if I imply that a leopard can eat a deer, it has given me a null pointer exception. Let us see what a null pointer exception turns out to be.

Now, go to project, assets, prefabs and double click on level clear panel. In the level clear panel, we have to display the score. For that, go to assets, scripts, right click in the window, select create and chose C-sharp script. Rename it as LevelClearScript. Select LevelClearPanel and click on Add Component, type Level Clear Script and press enter. Double click on it to open the level clear script.

Delete the update & start function for now and include the component by writing

Using UnityEngine.UI;

The purpose of the script is to update the score on the final screen. So, let's begin by writing

```
{  
Public Text scoreText;  
Public void UpdateScore(int _score)  
    {  
        scoreText.text = "Score: " + _score.ToString();  
    }  
}
```

So, this is the only functionality that we need. For the other functionality like the button functionality we have already written the code. Let unity compile this code. Click on the back arrow in hierarchy window to go to the scenes list.

To update the overall score, we will do so from the level timer panel. Double click on Level Timer to open the script. Let's add a code by writing

```
_lvlClear.GetComponent<levelClearScript>().UpdateScore();
```

Define the public class as

```
Public GameScore gameScore;
```

Switch to GameScore Script and write the function outside start function as

```
Public int GetGameScore()  
{  
    Return score;  
}
```

Switch back again to the Level Timer script and inside update score write

```
gameScore.GetGameScore()
```

This code will return the current score of the game and we are done with this script.

Switch back to unity editor, let's freeze level timer panel, drag and drop score panel from hierarchy window to the game score field. Click on unlock to unfreeze it. We have now completed all the functionalities that we wanted, so, let's run the project and see how everything is working together.

We are able to select all the animals as usual and after that is done, let's connect animals in the second level. Let's draw a line from lion to rhinoceros and then from lion to giraffe. We can say that a fox can eat a rat and that a snake can eat a sparrow. We are able to create a food web and every time we get something right as in every time my line turns green, it can be seen that 2 points gets added to my score. For instance, if we select eagle and connect it to snake, implying that the eagle eats snake, my score has increased by 2 points that is from 12 to 14. Similarly, if I draw a line from eagle to giraffe, which is definitely not the right answer, you can see that the line has turned red and that my score has been

deducted by 1 point. Once the game is finished, we can see that the score is still not visible.

Let's open the console to check why this is happening. The console is showing that a mistake has been committed and we know exactly how to fix this. To do that, go to project, assets, prefabs and double click on level clear panel. As it can be seen that we have set the value of text box but the reference is not taken. So, drag score text and drop it on to score text field to reference it. So now, we have fixed the bug and our pop-up screen should work.

Let's get back to our scene. Select level timer panel and change the timer to 10 seconds for now, just to test the functionality. One more thing we have noticed earlier is that our pop-up screen which was orange also had lines appearing on top or in the front of it.

So, what's happening here is we are just reducing the Z value of the panel. To make it work, switch to level clear script and write a start function as

To do that, go to unity editor and select level clear panel and change the z value to -0.02, so that it always comes on top.

Let's launch the project and see what happens. I'm going to connect the animals and as soon as our timer is up, we can actually see the score come up in our pop-up screen! With regard to the text there, we have no errors! But the lines still continue to come in front of the screen.

So, we'll go to LineDrawing script and delete all the lines that we have created so that it doesn't interfere with the panel. Before doing that, let's create a list by writing

```
List<GameObject> lineOjjectsList = new List<GameObject>();
```

Let's scroll down to end line drawing function and write the code after line renderer as

```
lineObjectsList.Add(currentLineObject);
```

and write a new function after end line drawing as

```
public void DestroyLines()  
{
```

```
While(lineObjectsList.Count > 0)
{
    GameObject _tempObject = lineObjectsList(0);
    lineObjectsList.RemoveAt(0);
    Destroy(-tempObject);
}
}
```

We should call this function in level timer script before we show the score panel. To do that, we will define the function

Public LineDrawing lines;

And inside the timer enumerator, let's write

Lines.DestroyLines();

These lines of code will help us hide the lines and the show the panel.

Let's go back to unity editor and assign the reference to it. To do that, select level timer panel and drag Level 2 to drop it in Lines field. Let's test our whole game to check if all the functionalities are working as expected.

Let's get through with the first level like we always do! Now, on to the second level, connect leopard to the deer, fox to the rat. Our timer is up, however both lines were correct, so the score is 2, which has been updated and the best news is that the lines don't interfere with the pop-up screen anymore! If we click the play again button, it is restarting the whole game as seen!

With that, we have finished the score and level timer. One last thing is to select level timer panel in the hierarchy window and set the duration of the timer to 60 seconds, it was set to 10 seconds earlier as shown.

So, we have come to an end of the video where we have completed the timer and scoring system! In the next video, we will learn to create the welcome screen and publish the game!

I must say, great job for having come this far! It has been an amazing journey, I'm very excited to see you for the last leg of it! Worry not, I will see you very soon, until then, do keep practicing and excelling in being the super game developer that you are!

HINDI VERSION

Add the Timer and Scoring System

Hey everyone!

आप सबको unity का इस्तेमाल कर game develop करने के लिए यहाँ वापस देख कर मुझे बहुत खुशी हुई, आज, हम game के दो नए aspects - scoring और timer के साथ काम करने पर ध्यान देंगे! हम देखेंगे कि level 2 के लिए एक timer को कैसे शामिल किया जाए और फिर हम अपने game में scoring system को जोड़ेंगे, तो चलो शुरू करते हैं!

आइए, unity hub open करें और उस project को open करें, जिस पर हम काम कर रहे हैं।

screen को maximize करें। hierarchy section में, canvas पर click करें, level 2 चुनें और UI चुनने के लिए right click करें और panel option चुनें। इसे "level timer panel" नाम दें। image window में panel के colour section पर जाएं और alpha को 0 में बदल दें ताकि ऐसा न लगे कि इसके ऊपर एक layer मौजूद है। इसके अलावा, इस box को uncheck करके ray cast target को disable जरूर करें। हम ऐसा इसलिए करते हैं क्योंकि हम नहीं चाहते कि यह panel कोई touch input ले। अब, hierarchy window पर वापस जाएं, Level Timer Panel पर right click करें, UI चुनें और text पर click करें। इसे "timer text" नाम दें।

अब, मैं यह पक्का करने जा रही हूँ कि timer text screen के top left corner पर रखा जाए। ऐसा करने के लिए, हम timer text को चुनने जा रहे हैं और shift key को दबाकर और top left corner पर पहले intersection पर box को चुन position और pivot को

बदलने जा रहे हैं। चुनने के लिए, Shift + Alt key दबाएं, फिर suitable option पर click करें। screen पर जैसा दिखाया है उसके according इसका size बदलें । text को “Time Remaining” में बदलें। इसके अलावा, उस box पर tick करें जिसका नाम best fit है।

Alignment को left side लेकिन centre में रहने दें। हम text box की position को X में 50 और Y में -25 move करेंगे।

अब, script के साथ शुरुआत करते हैं। project window, assets folder और फिर script के अंदर जाएं। right click करें, create चुनें और C sharp script चुनें। इसे “level timer” नाम दें। अब Level Timer script के साथ code editor open करने के लिये के लिए double click करें।

एक बार open करने के लिये के बाद, start and update function को हटा दें क्योंकि हमें उनकी जरूरत नहीं है। UI function को जोड़कर शुरू करें

Using UnityEngine.UI;

और आइए code लिखकर बनाए गए text box के refrence को define करें

```
{
```

```
Public text levelTimerText;
```

```
Public int totalTime;
```

```
Public void StartTimer()
```

```
{
```

```
}
```



```
IEnumerator Timer()
```

```
{
```

```
Yield return new WaitForSeconds(2);
```

```
Int _counter = totalTime;
```

```
While (_counter > 0)
```

```
{
```

```
_counter --;
```

```
levelTimerText.text = "Time Remaining: " +  
_counter.ToString(); yield return new WaitForSeconds(1); }
```

```
Debug.Log("Game over show result screen");
```

```
}
```

```
}
```

start timer function के अंदर ये code लिखें

```
StartCoroutine(Timer());
```

अब, unity editor पर वापस चलते हैं। हम तब तक इंतजार करेंगे जब तक यह सभी scripts को compile नहीं करता है। Level Timer panel पर जाएं, inspector window पर click करें, Level Timer type करें और enter दबाएं। timer text को level timer text field में खींचें और अभी total time को 5 second में बदल दें, जिसका मतलब है कि level

2 को केवल 5 सेकंड के लिए run करना है। यह केवल testing purpose के लिए किया जाता है।

अब जब हमने value लिख लिया है, तो इसका मतलब यह नहीं है कि यह अपने आप शुरू होने वाला है, हमें start timer function को call करना होगा। ऐसा करने के लिए, game controller पर जाएं और level 2 gameplay script को open करने के लिए level 2 gameplay पर double click करें।

जैसा कि आप देख सकते हैं कि जब भी level 2 शुरू होता है, तो हम "Init" function को call कर रहे थे, ताकि सभी जानवरों को grid में arrange किया जाए। तो, same function में। हम लिखने जा रहे हैं

```
//Start game timer
```

और फिर हम level timer script का reference लेने जा रहे हैं जो हमने लिखा है और फिर start timer function को call करेंगे। चलो Level2Gameplay पर जाएँ और declare करें

```
Public LevelTimer levelTimer;
```

और timer को start करने के लिए लिखें

```
levelTimer.StartTimer ();
```

आइए unity editor पर वापस जाएं और scripts के compile होने का इंतजार करें। game controller को चुनें। अब हमें Level Timer को जोड़ना होगा, जिसके पहले मैं inspector window में lock पर click करके इसे freeze करने जा रही हूँ, यह सिर्फ यह पक्का करने के लिए है कि यह move नहीं करे। फिर, मैं Level Timer को उस box पर खींच कर drop

करने जा रही हूं जिसे Level Timer कहते हैं। जिसके बाद, मैं lock को unfreeze करने और project को चलाने जा रही हूं, यह देखने के लिए कि क्या होता है।

मैं अब console window पर click करने जा रही हूं, game में वापस जाऊंगी और सभी जानवरों को चुनूंगी। अब जब हम सभी जानवरों को चुन लिया हैं तो “congratulations” screen आ गई है और हमने level 1 clear कर लिया है। जैसा कि आप देख सकते हैं कि level 2 के लिए timer केवल 2 second के screen switch के बाद शुरू होता है। एक बार जब timer 0 हो जाता है, तो हमें “game over show result screen” message दिखाया जाता है। हमारा Level Timer को जोड़ने का काम पूरा हो गया है, लेकिन उसके बाद जो screen दिखाई देगी, वह हम अभी तक नहीं बना पाए हैं।

Hierarchy window में level 2 पर जाएं। उस पर right click करें, UI चुनें और panel option चुनें। अब, panel का colour बदलें और इसे dark करें। मैं दिखाई दे रहे shade को चुन रही हूँ, फिर panel पर right click करें, UI चुनें और image चुनें। आइए यहां जो image जोड़ी गई है उसका size बढ़ाएं। इसे “level clear panel” का नाम दें। हम इस पर एक image डालेंगे। सबसे पहले, हम colour option पर click करने के लिए inspector window पर जाएंगे। मैं orange चुन रही हूँ। image का size बढ़ाएं और जैसा दिखाया है उसके according इसका position बदलें।

अब, फिर से image पर right click करें, UI चुनें और text पर click करें। जैसा दिखाया है उसके according box पर text box को align करें। इसको expand करें ताकि यह एक end से दूसरे end तक stretch हो जाए, जितना आप चाहते हैं उतना उसका size बढ़ाएं, इसे centre में align करें और best fit box पर tick करें। text box में “well done” लिखें। size को 128 तक बढ़ाएं और इसे bold करें। text को थोड़ा नीचे push करके और Rect Transform window में Y box के अन्दर -25 value enter करके इसका place बदलें। इसे title text नाम दें।

image को चुनें, right click करें, UI चुनें और फिर text चुनें। इस text box का size बढ़ाएँ और इसे centre में align करें। text को बदलकर score करें :, alignment को centre में करें , font size को 81 तक करें और फिर से सबसे best fit option चुनना न भूलें। इसे “Score text” का नाम दें। अब, image पर right click करें, UI और choose button चुनें। जैसा दिखाया गया है और button के size को बढ़ाएँ और इसका position बदलें। button को expand करें, text पर click करें, best fit option चुनें और फिर text field में “Play again” type करें। button का नाम बदलकर “Play again button” करें। इसका रंग बदलकर green कर दें। जैसा कि हम last video में पहले ही discuss कर चुके हैं कि आप level create panel से बाहर prefab बना सकते हैं। ऐसा करने के लिए, asset folder पर जाएं, prefab चुनें और level clear panel खींचें और इसे prefab में छोड़ दें। hierarchy window में मौजूद “level clear panel” को हटाएं।

अब, LevelTimer script में code लिखकर level clear panel को instantiate करें

```
Public GameObject levelClearPanel;
```

```
// और लिखकर debug log by using को disable करें
```

```
GameObject _lvlclear = Instantiate(levelClearPanel);
```

हम एक और function को declare करेंगे

```
Public Transform parentCanvas;
```

और ये जोड़कर game object बनाते हैं

```
, parentCanvas
```

debug log को delete करें

unity editor पर वापस जाएं और इसे compile करें। जैसा कि आप देख सकते हैं, यह level clear panel reference के लिए पूछ रहा है। तो, prefab folder में जाएं और level clear prefab को level clear panel field तक खींचें और drop करें।

इसके बाद, canvas को hierarchy window से खींचें और parent canvas field पर छोड़ें। चलो game को run करते हैं और देखते हैं कि क्या होता है।

जैसा कि हम देख सकते हैं कि game अभी तक ठीक चल रहा है। लेकिन हमने play panel पर अब तक की किसी भी functionality को नहीं लिखा है। basically, दो functionality हैं जिन्हें हमें यहां लिखना है, एक है play again, play again पूरे scene को फिर से या पूरे game को फिर से शुरू करेगा, और score, score को display करेगा, जो हम scoring system को apply करने के बाद करेंगे। तो, चलो फिर से play again button की functionality को implement करते हैं।

ऐसा करने के लिए, अपने project में asset folder पर जाएं और level clear panel पर double click करें। अगर आपको याद है, तो हमने इस functionality को पहले ही level 1 के लिए लिख दिया था। यहां, हमारे पास replay नाम का button था जो पूरे game को फिर से शुरू कर रहा था। हम फिर से play button में भी इसी तरह की functionality का इस्तेमाल करने जा रहे हैं। तो, asset folder पर वापस जाएं और Level Clear panel पर double click करें। inspector window में add component पर click करें, “replay game” type करें और enter दबाएं। अब, फिर से play again button पर जाएं, button section में नीचे जाएं। click event में जोड़ने के लिए + पर click करें। level clear panel को object field में खींचें और drop करें, और drop down से, replay game चुनें और

play again function पर click करें। main window पर वापस जाने के लिए hierarchy window में small arrow पर click करें।

इसलिए, अब जब हमने timer implement कर दिया है, तो, हम scoring system को implement करेंगे। ऐसा करने के लिए, hierarchy window पर जाएं। जैसे हमने Level Timer panel बनाया है, हम इस पर सीधे right click भी कर सकते हैं, इसे duplicate कर सकते हैं और इसका नाम बदलकर “score panel” रख सकते हैं। इसे expand करें और “timer text” का नाम बदलकर “score text” करें। हम चाहते हैं कि इसे right side align किया जाए, इसलिए, score text को चुनें, shift को दबा कर रखें और position और pivot को बदलकर top right side करें। X के लिये value -50 y के - 25 करें। alignment को right side करें और text “time remaining” से बदलकर “score” करें। अब, score को जोड़ने और update करने के लिए, हमें project project, asset, script पर जाना होगा, उस पर right click करना होगा और "GameScore" नामक एक नई script बनाना होगा। score panel पर जाएं, score panel से जुड़ी level timer script को हटा दें, क्योंकि यह level timer नहीं है। इसलिए, script folder में “games score” चुनें और इसे score panel पर खींचें और छोड़ दें। यह script attach करने का एक और तरीका है। अब, code editor को open करने के लिए score panel पर double click करें।

update function को remove कर दें क्योंकि हमें इसकी जरूरत नहीं है। चलो एक function को define करके coding शुरू करते हैं

```
Public UnityEngine.UI;
```

And declare the variables as

```
Public text scoreText;
```

```
Int score;
```

जैसे ही scene शुरू होता है, हम ये लिखकर score set करना चाहेंगे

```
{  
  
Score = 0;  
  
scoreText.text = "Score:" + score.toString();  
  
}
```

जब भी score update या reduce किया जाता है, हम इस function को call करेंगे

```
Public void UpdateScore(bool _increaseScore)
```

```
{  
  
    If (_increaseScore)  
  
    {  
  
        score = score + 2;  
  
    }  
  
    Else  
  
    {  
  
        score = score - 1;  
  
    }  
  
    If (score < 0)
```

Score = 0;

```
ScoreText.text = "Score: " + score.ToString();  
}
```

```
}
```

Unity editor पर वापस जाएं score panel चुनें। Score text को hierarchy window से Score text field तक खींचें। score को बढ़ाने या घटाने के लिए, level 2 को चुनें और script open करने के लिए LineDrawing script पर double click करें। यह वह script है जिसमें सभी logic start or end animal या line खींची गई है या नहीं इससे related है।

अब parentcanvas के बाद ये लिखकर 2 variable बनाएं

```
Defines.AnimalsList startAnimal;
```

```
Defines.AnimalsList endAnimal;
```

अब start line drawing function पर जायें और define करें

```
startAnimal = AnimalScript.startAnimal;
```

game object को इसे फिर से लिखकर defines change करें

```
GameObject startAnimal;
```

```
GameObject endAnimal;
```


अब , end line drawing function पर जाएं और if loop के बाद लिखें

```
endAnimal = AnimalScript.endAnimal;
```

इसलिए, अगर आप देखें तो, एक condition जिस पर हम है काम कर रहे हैं if animal script null के equal नहीं है तो, line नष्ट हो जाए इसलिए, मैं unity में जाती हूँ और project को run करके ये देखती हूँ lines कैसे खिंची जा रही है। मैं सिर्फ यह test करना चाहती हूँ कि lines बन रही हैं या नहीं। जैसा कि आप देख सकते हैं कि lines ठीक से नहीं बन रही हैं। तो, play button पर फिर से click करें और code editor से linedrawing script पर switch करें।

end line drawing function में आप देखते हैं कि एक small bug है जिसे हमने initially छोड़ दिया था। यही है, हमने कहा है कि अगर the animal script dot end animal null के equal नहीं है तो current line object को नष्ट कर दे, लेकिन हमें यह कहना चाहिए कि अगर endpoint जानवरों में से किसी पर नहीं है, तो हमें इस line को नष्ट कर देना चाहिए। otherwise, हमें इस line का इस्तेमाल जारी रखना चाहिए। इसलिए, exclamatory mark को equal में बदलें। इसके अलावा, एक else condition ये लिखकर जोड़ें

Else

```
{
```

```
endAnimal = AnimalScript.endAnimal;
```

```
}
```

यहां, हम एक ऐसा function

बनाना चाहेंगे, जो यह तय करे कि क्या हम सही जानवर को चुन रहे हैं और उसे सही जानवर पर drop कर रहे हैं या नहीं। उदाहरण के लिए, अगर मैं lion से leopard तक line खींचता हूं, तो यह गलत answer होगा। लेकिन अगर मैं एक lion से शायद giraffe, or a rhinoceros तक एक line खींचती हूं, तो यह सही answer होगा क्योंकि एक lion giraff और rhinoceros खा सकता है।

इसलिए, एक function लिखते हैं

```
Bool UpdateScore()
```

```
{
```

```
If(AnimalScript.endAnimal == null)
```

```
Return false;
```

```
Defines.AnimalsList _startAnimal =  
startAnimal.GetComponent<AnimalType>().animalType;
```

```
Defines.AnimalsList _endAnimal =  
endAnimal.GetComponent<AnimalType>().animalType; If(-_startAnimal ==  
Defines.AnimalsList.Lion)
```

```
{
```

```
If(_endAnimal == Defines.AnimalsList.Giraffe || _endAnimal ==  
Defines.AnimalsList.Rhino)
```

```
{
```

```
Return true;
```

```
}
```

```
Else
```

```
        {  
            Return false;  
        }  
    }  
  
Else if(_startAnimal == Defines.AnimalsList.Sparrow)  
  
    {  
  
        If(_endAnimal == Defines.AnimalsList.Grasshopper)  
  
            {  
  
                Return true;  
            }  
  
        Else  
  
            {  
  
                Return false;  
            }  
    }  
  
Else if (_startAnimal == Defines.AnimalsList.Snake)  
  
    {  
  
        If(_endAnimal == Defines.AnimalsList.Sparrow || _endAnimal ==  
        Defines.AnimalsList.Fox)  
  
            {
```

```
        Return true;
    }
    Else
    {
        Return false;
    }
}

Else if (_startAnimal == Defines.AnimalsList.Fox)
{
    If(_endAnimal == Defines.AnimalsList.Rat)
    {
        Return true;
    }
    Else
    {
        Return false;
    }
}

Else if (_startAnimal == Defines.AnimalsList.Eagle)
```

```
{
```

```
  If(_endAnimal == Defines.AnimalsList.Snake || _endAnimal ==  
  Defines.AnimalsList.Leopard)
```

```
  {
```

```
    Return true;
```

```
  }
```

```
Else
```

```
  {
```

```
    Return false;
```

```
  }
```

```
}
```

```
Else if (_startAnimal == Defines.AnimalsList.Leopard)
```

```
{
```

```
  If(_endAnimal == Defines.AnimalsList.Deer)
```

```
  {
```

```
    Return true;
```

```
  }
```

```
Else
```

```
  {
```

```
    Return false;
```

```
    }  
  }  
  Return false;  
  }  
}
```

आप अपनी जरूरत और complexity के अनुसार food chain में combinations को modify कर सकते हैं जिसे आप game में लाना चाहते हैं।

अब, end line drawing पर जाएं और जब वहाँ end animal available हो तो हम जो करना चाहते हैं वो है उस score को update करना। उसके लिए, हमने game score script में function लिखा है। तो, parent canvas function line पर जाएं और लिखें

Public GameScore gameScore;

और end line drawing function में, इसके लिए code लिखें। इससे पहले, screen जैसा दिखाया गया है update score function में if condition को हटा दें क्योंकि अब इसकी जरूरत नहीं है। else condition पर वापस आएं और लिखें

Bool _increaseScore = UpdateScore();

gameScore.UpdateScore(_increaseScore);

एक और चीज जिसे हम इस functionality में जोड़ सकते हैं जब भी कोई सही answer होता है, तो हम उस line को green बनाना चाहेंगे जिसे हम दोनों

जानवरों के बीच draw करेंगे। और जब भी यह गलत जवाब होता है तो line का colour red दिखे। ऐसा करने के लिए, ये code लिखें

```
if(_increaseScore)
```

```
currentLineRenderer.material.color = Color.green;
```

```
else
```

```
currentLineRenderer.material.color = Color.red;
```

अब, हमने जितना भी code लिखा है सबको test करते हैं, हमने अब तक काफी अच्छे मात्रा में code लिखा है, इसलिए यह test करना महत्वपूर्ण है कि क्या सब कुछ ठीक काम कर रहा है जैसा हमने सोचा है! आइए unity editor पर वापस जाएं और इसे compile करें। project चलाने से पहले, हम hierarchy window में level 2 पर जाएं, line drawing script में मौजूद game score field में score panel खींचें। अब जब हम यह सब कर लिया है, तो हम इसे run करते हैं और देखें कि क्या होता है।

जैसा कि level 2 gameplay का time बहुत कम है, Level Timer panel को चुन कर इसे increase करें और total time को 5 से बदलकर 60 second कर दें। आइए launch करें और देखें कि अब क्या होता है।

testing purposes के लिए, हम deer को lion से जोड़ने जा रहे हैं, यह देखने के लिए कि क्या यह सही है या गलत है? हम line को red होते हुए देख सकते हैं क्योंकि यह गलत answer है। अब, lion को rhinoceros से जोड़ते हैं, जिसका मतलब है कि lion rhinoceros को खा सकता है। यह देखा जा सकता है कि line green हो गई है। अब, अगर मैं यह समझता हूँ कि एक leopard deer को खा सकता है, इसने मुझे एक null pointer exception दिया है। आइए देखते हैं कि एक null pointer exception क्या है।

अब, project, assets, prefabs पर जाएं और level clear panel पर double click करें। level clear panel में, हमें score दिखाना होगा। उसके लिए, assets, scripts, पर जाएं, window में right click करें, create चुनें C-sharp script चुनें और चुनें। इसे LevelClearScript नाम दें। LevelClearPanel चुनें Add Component पर click करें, Level Clear Script type करें और enter दबाएँ। level clear script open करने के लिए के लिए उस पर double click करें।

update & start function को अभी delete करें और component ये को लिखकर include करें

Using UnityEngine.UI;

script का purpose final screen पर score को update करना है। तो, चलो लिखना शुरू करते हैं

```
{
```

```
Public Text scoreText;
```

```
Public void UpdateScore(int _score)
```

```
{
```

```
    scoreText.text = "Score: " + _score.ToString();
```

```
}
```

```
}
```

तो, यही एक functionality है जो हमें चाहिए। button functionality जैसी अन्य functionality के लिए हमने पहले ही code लिखा है। unity को इस code को compile करने दें। scenes list पर जाने के लिए hierarchy window में back arrow पर click करें।

overall score को update करने के लिए, हम level timer panel से ऐसा करेंगे। script open करने के लिए Level Timer पर double click करें। लिखकर एक code जोड़ते हैं

```
_lvlClear.GetComponent<levelClearScript>().UpdateScore();
```

public class को define करें

```
Public GameScore gameScore;
```

GameScore Script पर switch करें और outside start function को लिखें

```
Public int GetGameScore()
```

```
{
```

```
Return score;
```

```
}
```

Level Timer script के अंदर फिर से switch करें और update score के अन्दर लिखें

```
gameScore.GetGameScore()
```

यह code game के current score को return करेगा और हम script का काम पूरा कर लिया हैं।

unity editor पर वापस जाएं, Level Timer panel को freeze करें, score panel को hierarchy window से game score field तक खींचें और drop करें। इसे unfreeze करने के लिए unlock पर click करें। हमने अब उन सभी functionalities को पूरा कर लिया है जो हम चाहते थे, इसलिए, आइए project को run करते हैं और देखें कि सब कुछ एक साथ कैसे काम कर रहा है।

हम हमेशा की तरह सभी जानवरों को चुन पा रहे हैं और जब ये हो जाए, चलो second level में जानवरों को जोड़ते हैं। चलो lion से लेकर rhinoceros तक और फिर lion से giraffe तक एक line खींचते हैं। हम कह सकते हैं कि एक fox एक rat खा सकती है और एक snake एक sparrow को खा सकता है। हमने एक food web बनाया है, और हर बार हमें कुछ सही मिलता है क्योंकि हर बार जब मेरी line green colour में बदल जाती है, तो यह देखा सकते हैं कि मेरे point में 2 point जुड़ जाते हैं। उदाहरण के लिए, अगर हम eagle को चुनते हैं और इसे snake से जोड़ते हैं, तो इसका मतलब है कि eagle snake को खाता है, मेरा score 2 point बढ़ गया है जो 12 से 14 हो गया है

इसी तरह, अगर मैं eagle से giraff तक एक line खींचता हूं, जो पता है कि सही answer नहीं है, तो आप देख सकते हैं कि line red हो गई है और मेरा score 1 point कम हो गया है। एक बार game end होने के बाद, हम देख सकते हैं कि score अभी भी दिखाई नहीं दे रहा है।

आइए यह देखने के लिए console open करें कि यह क्यों हो रहा है। console दिखा रहा है कि एक mistake हुई है और हम ठीक से जानते हैं कि इसे कैसे fix किया जाए। ऐसा करने के लिए, project, assets, prefabs पर जाएं और level clear panel पर double click करें। जैसा कि देखा जा सकता है कि हमने text box के value को set कर दिया है, लेकिन reference नहीं लिया गया है। तो, score text को खींचें और इसे refrence करने के लिए score text field पर छोड़ दें। तो अब, हमने bug को fix कर दिया है हमारे pop up screen को काम करना चाहिए।

चलिए वापस अपने scene पर आते हैं। level timer panel को चुनें और अभी functionality को test करने के लिए timer को 10 second तक के लिय कर दें। एक और बात जो हमने पहले देखी है, वह यह है कि हमारी pop up screen जो orange थी, उसके ऊपर या उसके सामने दिखने वाली lines भी थीं।

तो, यहाँ क्या हो रहा है हम panel के Z value को कम कर रहे हैं। ये काम करे इसके लिय , level clear script पर switch करें और start function लिखें

ऐसा करने के लिए, unity editor पर जाएं और level clear panel चुनें और Z value को -0.02 में बदल दें, ताकि यह हमेशा top पर आए।

आइए project launch करें और देखें कि क्या होता है। मैं जानवरों को जोड़ने जा रही हूँ और जैसे ही हमारा timer end होता है, हम वास्तव में score को हमारे pop up screen में देख सकते हैं! text से related हमारी कोई error नहीं है! लेकिन screen के सामने अभी भी lines आ रही हैं।

तो, हम LineDrawing script पर जाएँगे और हमारे द्वारा बनाई गई सभी lines को हटा देंगे ताकि यह panel के साथ interfere न करे। ऐसा करने से पहले, आइए लिखकर एक list बनाएं

```
List<GameObject> lineObjectsList = new List<GameObject>();
```

चलिए end line drawing function को नीचे scroll करते हैं और line renderer के बाद code लिखते हैं

```
lineObjectsList.Add(currentLineObject);
```

और end line drawing के बाद एक new function लिखें

```
public void DestroyLines()
```

```
{
```

```
While(lineObjectsList.Count > 0)
```

```
{
```

```
GameObject _tempObject = lineObjectsList(0);  
lineObjectsList.RemoveAt(0); Destroy(-  
tempObject);
```

```
}
```

```
}
```

score panel दिखाने से पहले हमें इस function को level timer script में call करना चाहिए , ऐसा करने के लिय हम function को define करेंगे

```
Public LineDrawing lines;
```

और timer enumerator के अंदर, लिखते हैं

```
Lines.DestroyLines ();
```

code की ये lines हमें lines को hide करने और panel को दिखाने में मदद करेगी।

आइए unity editor पर वापस जाएं और इसे reference assign करें। ऐसा करने के लिए, Level Timer panel चुनें और level 2 को खींच कर Lines field में छोड़ें। आइए यह के

लिए हमारे पूरे game का परीक्षण करें कि क्या सभी functionalities जैसा हम चाहते हैं वैसे ही काम कर रही हैं ये check करने के लिये पूरे game को test करें।

चलो पहले level पर चलते हैं जैसे हम हमेशा करते हैं! अब, दूसरे level पर, leopard को deer से, rat को fox को जोड़ें। हमारा timer end हो गया है, हालांकि दोनों lines सही थीं, इसलिए score 2 है, जिसे update किया गया है और सबसे अच्छी खबर यह है कि lines अब pop up screen के साथ interfere नहीं कर रही हैं! अगर हम फिर से play button पर click करते हैं, तो यह पूरे game को दोबारा start कर रहा जैसा कि दिख रहा है!

इसके साथ, हमने score और level timer का काम पूरा कर लिया है। एक आखिरी चीज hierarchy window में level timer panel को चुनना है और timer के duration 60 second तक set करना है, इसे पहले 10 second set किया गया था।

तो, हम उस video के last में आ गए हैं जहाँ हमने timer और scoring system को पूरा किया है! अगले video में, हम welcome screen बनाना और game publish करना सीखेंगे!

मुझे कहना होगा, इतनी दूर आ जाना बहुत बड़ी बात है ! यह एक अद्भुत यात्रा रही है, मैं आपको इसके last stage के लिए देखकर बहुत उत्साहित हूँ! चिंता न करें, मैं आपको जल्द ही मिलूंगी, तब तक, super game developer बनने के लिए practise करें और बेहतर बनते रहें जो आप हैं!