

ENGLISH VERSION

Animating and Selecting Animals

Hello friends!

Welcome back to game development using UNITY!

So, in the last video, we had created our scene by placing animals into the theme. Now, we are going to look at how to animate those animals so that they can move and we can make the scene more interesting. Also, we are going to write the logical input to select different animals so that they can be used in the food web.

So, let's get started! Open your UNITY hub and select the project titled 'food web'.

Now, let's maximize the screen. We are looking to animate our animals, so, let us start by adding animation to our rat. We want this rat to move from the current position to behind the bush position, where he will stay for around five seconds and then move back to the original position. To do that, we must select rat from the hierarchy window, go to the window option in the top bar and select the "animation" option. The animation window is open. Create new animation. Before doing that, do not forget to go to the assets animation folder, create a new folder called "rat" and title your animation as "rat_walk".

In the animation window, start recording the animation in its 1st position. For the 2nd position, I want the rat to move from here to there in 2 seconds. So, I will just set the marker to 2 second indication and then move the rat around from here to this position. Now, I want my rat to remain behind the bushes for at least 5 seconds, so I'm going to add the previous 2 seconds to 5 seconds to make it 7 seconds in total. Now, I will set the marker to 7 seconds. As I want the rat to stay here for a while, hence, I'm just minimally changing its position, from a point 9 to point 8 so that it's position can be reset. I'm going to stop the recording and see how the animation plays out. I can see the rat animation is working out well.

Next, we are going to move the grasshopper from one side of the screen to the other. Select the grasshopper from the hierarchy window and position the grasshopper to the extreme left, outside of the screen. Repeat the same steps for adding animation just like we did for rat. Create a new folder called “grasshopper” in the Animations folder and name the animation as ‘Fly’.

Start the recording. The initial position of grasshopper is (-1083, -375). Keep a note of it for resetting the object. I want grasshopper to move from the current position to this extreme right, outside the screen position. Zoom out the animation timeline as I want the animation to run for around 10 seconds, so that it moves slowly. Simply drag and move this grasshopper from start to end position. Now I want to reset the position. To do that, slightly move the line on the frame again and set (-1083, -375) in position field. Now, stop the recording and play the animation. As you can see, the grasshopper is flying from this one end to the another end and then reappearing again at the starting point.

Next, we are going to work on animating the deer! We are going to do two things as far as the deer is concerned, we ensure that the deer has a walk cycle and then we need to move this deer from one side of the screen to another side of the screen. Let’s select deer from the hierarchy window and disable bushes for a while to see how the deer walk will happen.

As of now, the deer is just standing with no animation attached to it. First go to assets, images and animals. From the last time, you should remember that we broke down the single image of the deer into multiple images in the previous videos. Expand the arrow mark to see the walk cycle animation for the deer. Proceed further by creating a new folder. Go to projects > assets > animations > create a new folder called “Deer”. In the animation, simply right ‘Walk’ and save it. So, now a new animation has been created for the deer. Now select all the images of Deer animation and drag on to the animation panel. Animation for the deer is done. Play and see how it works. As we can see the speed is fast. Reduce the speed by dragging the small button to 1-minute and 30 seconds mark. Play and see the animation again. Now, the walking speed of deer is fine and we are done with deer animation. Click on the ‘sprite editor’ after selecting the deer asset to see the frames used in animating the deer.

But I want the deer to come out and walk all the way through and reset back to its original position.

Let's explore a way to do this through code! So basically, my deer should start from slight left of the center part of screen. Let's make the bushes visible. It should walk all the way to the end of screen and reappear again to its original position. The default position is (-284, 200). This is essentially a loop and to make that happen, go to asset folder, create a new folder called scripts. In the script folder, right click, select create and click on C sharp script. Name it as "DeerWalking". Double click on it to open the Visual studio for writing a program.

So, there are two functions that have already been given to us - one is the start function which calls the object to which this script is attached and the update function which gets called continuously with each frame of the animation. First we need to define two fields. In the next line of open parenthesis, write

public Transform startingPoint;

This is going to be the starting point of deer, that is, the left side from where it starts moving. Define another variable called

public Transform endpoint;

This defines the end point of deer. Let's define one more variable called speed by writing

public float speed;

This is the speed with which the deer will move from start point to end point. As we are not initializing any object so let's delete void Start() function as shown.

In our void Update() function, we are going to create a simple variable called

{

float step = speed X Time.deltaTime;

In this step, what happens is that the deer will take the number of steps based on the speed and time. And the delta time is the difference between two frames that are being drawn between 2 image references. As we are attaching this

script to the deer, we are going to take the transform position of the deer. So, we will write

```
transform.position = Vector3.MoveTowards(transform.position, endpoint.position, step);
```

So, this takes the current position, the end position and how much distance deer want to travel. Now once it reaches the end point, we want to reset the position to the starting point. So, let's check the condition by writing a code

```
if(Vector3.Distance(transform.position, endpoint.position) < 0.01f);
```

What we are checking here is if the deer has reached almost or at the end point, we should reset the position of deer to start point. So, we will write

```
{  
Transform.position = startingPoint;  
}
```

This is going to continuously keep the deer moving from one position to another. Now let's set the values of all the public variables that we have defined.

To do that, go back to Unity editor and save the scene to update the script for DeerWalking.

Now, select Deer from Hierarchy window and click on add component from Animator window. Select the name of the script that we have created by typing DeerWalking. Once entered, it is showing me three public variables i.e. the starting point, End point and speed.

As of now, I have not created any starting point. Right click on Canvas in Hierarchy window and select create an empty object and move it near to deer object. Rename it Deer Starting Point. Right click on it and select duplicate. Drag it near Deer Starting Point and rename it Deer End Point.

To set the starting point, select Deer Starting Point and drag the box to (-1080, -200) position and to set the end point of deer, select Deer End Point and drag the box to (-1244, -200) position. Use the move tool to avoid misbalance on Y-axis.

But I want my deer to be present behind the tree, so I'm going to select Deer, Deer starting point & Deer end point and place them above Tree object. Now, it will look like the deer is behind the tree and bushes.

Select Deer option and under Deer Walking script, select Starting Point. A new window opens. Double click on Deer Starting Point option. Similarly, select Deer End Point for End Point option. Set the speed as 5. The speed is fast. Let's change it to 2 now which is optimum.

Now, let's see how the whole scene is working. Click on Play button to launch the scene. We can see our scene is working perfectly.

Next, what I want in my game is that whenever I select an animal, that animal should get selected and disappear from the whole scene and get added to a list from where we can access it while creating the food web.

To do that, minimize the canvas first. Click on '+' and select create empty object and reset the object by clicking on the 3 dots in Transform section and select Reset. Name it as "GameController". In the same Scripts folder, create a new script called "GameController". Next, select the game controller object from hierarchy window and click on Add Component. Type 'GameController' and double click to open the visual studio. Let's delete the start & update function and understand what are we going to put in game.

So, game controller is going to be the script that will handle all the states of your game. By states I mean, when the game starts there will be a welcome screen. Next, a second screen called level scene that comes before the scene of gameplay level, third screen with our food web scene or level one completion screen. So once that is complete, then you go to the level two screen where you create the food web and the last, game over screen. Let's define these states now.

To do that, we can start by defining an enumeration. This game's enumeration is going to contain all the game states. You can initialize them like

```
{
```

```
Enum GameStates
```

```
{
```

```
WelcomeScreen,  
Level1_Gameplay,  
Level1_GameOverScreen,  
Level2_StartScreen,  
Level2_GamePlay,  
Level2_Gameover,  
GameComplete  
}
```

We can add more states to this as we go along, but these are some of the major states in which we are going to create the game. So, for now, let's not worry too much about the welcome screen because that's something that we are going to cover in another video.

Before proceeding further, let's define these states in a different script all together to avoid any confusion. To do that, go back to Unity editor. Go to script folder in your assets, right click and create a new C sharp script and name it as 'Defines'. Double click on it to open the Visual studio. Delete the existing functions.

Delete the MonoBehaviour and change the public class Defines to

```
public Static class Defines
```

and make it static. Now, go to GameController tab and copy the whole code and paste it under defines.

Now to access the states from outside, the game states class is to made public by changing it to

```
public enum GameStates
```

Come back to the Game Controller tab and begin by creating a function

```
Public void UpdateGameState(Defines.GameStates _gameState)
```

Let's define a global variable as

```
{  
Defines.GameStates currentGameState;
```

Let's focus on Level 1 Gameplay screen for now. So, we are going to define a current status by writing

```
{  
currentGameState = _gameState;  
}
```

Update Game state is the function that we are going to call whenever we want to change the state of the whole game. Once this is called, the statement saves the current game state. Next, let's write a switch function as

```
switch(_gameState)  
{  
case Defines.GameStates.WelcomeScreen:  
{  
}  
break;  
case Defines.GameStates.Level1_Gameplay:  
{  
}  
break;
```

This is all about game controller. We are going to start putting code into this file for all states. Let's go back to unity editor and see what we would like to achieve first in the Level 1. The first thing that we want is to be able to select all the animals by clicking on them.

Hence, we will create another object and name it as Level1GamePlay. Accordingly, create a C sharp script in the Asset folder and name it as Level1GamePlay. Double click to open the new tab for it. Again delete the start & update function.

In the level 1 gameplay, we want the users to select the animals and once they get selected, they should disappear.

To do that, we are going to create a function for animal selection by taking game object for it and assign a unique ID to each of them so that they can be identified

once they are selected. After selection they will disappear from the screen. For that, write the code as

```
{  
Public void SelectAnimal(GameObject _animalObject)  
{  
Destroy(_animalObject);  
}
```

Go back to the unity editor and save to refresh it to update the script. Now, select Level 1 gameplay from hierarchy window and reset to initial position. Click on Add component type 'Level1Gameplay'. Maximize the Canvas and go to the animals one by one.

Let's start with fox from the top. Right now, it is just an image as we haven't added any event or function to it. Click on Add component and type 'Event Trigger'. Now click on 'Add New Event Type' to open the options and select 'PointerClick'. Click on '+' to add from list.

Now it's asking for the function that you want to call. So to add that function, drag Level1GamePlay from hierarchy window and drop it to the pointer click variable. From the dropdown menu, select Level1GamePlay and select the function that we have created which is SelectAnimal(GameObject). Now, you will be asked for the parameter or AnimalD. _animalObject is the parameter in our case which in other terms are the animals. First parameter we can define is Fox. Drag Fox to the Game object field. So, whenever I click on the fox, it gets disappeared.

Similarly, we'll do it for all the other the animals. Follow the same steps for adding Event Trigger > PointerClick > Drag Level1GamePlay and drop it to the pointer click variable > Select Level1GamePlay > Function as SelectAnimal(GameObject) > Drag Giraffe to game object. Repeat the steps for all animals.

Great. So what happened here is that whenever any click event happens on the animal, it is going to select the particular animal function with parameter as that animal in Level1GamePlay script. Let's run and see what happens.

I am clicking on the leopard, but it is not getting selected. It is because there is a Raycast that is blocking the click function. Let's select the Forest Background and you can see that Raycast is enabled which means that the Forest Background image is also interactive at the moment. But in our game we don't want all the objects to be interactive except animals. Hence, let's disable it by unticking that. Let's disable for all tree, hanging leaves, Bushes. Let the Raycast enabled only for the animals.

Let's run the scene and see what happens. As you can see we are able to select all the animals from the scene. So far, we have animated the animals and added selection function to it for Level 1 Gameplay.

So that's it from this video. Keep practicing and we will meet in the next video!

HINDI VERSION

Animating and Selecting Animals

Hello friends!

UNITY का use करके game development program में आपका स्वागत है!

पिछले video में, हमने जानवरों को theme में रखकर हमारा scene बनाया था। अब, हम उन जानवरों को animate करने जा रहे हैं ताकि वे चल सके आगे बढ़ सकें और हम scene को और अधिक interesting बना सकें। साथ ही, हम भिन्न भिन्न जानवरों को चुनने के लिए logical Input लिखने जा रहे हैं ताकि उनका use food web में किया जा सके।

तो चलो शुरू करते हैं! अपना UNITY hub open करें और 'food web'title वाली project चुनें।

अब, screen को maximize करें। हम अपने जानवरों को animate करना चाहते हैं, इसलिए, हम अपने rat में animation add करना शुरू करते हैं। हम चाहते हैं कि यह rat current position से bush की position के पीछे चला जाए, जहां वह लगभग 5 second तक रहेगा और फिर original position में वापस आ

जाएगा। ऐसा करने के लिए, हमें hierarchy window से rat को चुनना होगा, top bar में window option पर जाएं और "animation" option चुनें। animation window open है। new animation बनाएं। ऐसा करने से पहले, asset animation folder में जाना न भूलें, "rat" नाम से एक new folder बनाएं और अपने animation को "rat_walk" title दें।

Animation window में, अपने 1st position पर animation record करना शुरू करें। 2nd position , में चाहती हूं कि rat 2 second position में यहां से वहां चला जाए। इसलिए, मैं marker को 2 second के indication पर set करूंगी और फिर rat को इधर से उधर इस position में ले जाऊंगी। अब, मैं चाहती हूं कि मेरा rat कम से कम 5 seconds के लिए bushes के पीछे रहे, इसलिए मैं previous 2 second को 5 second में add करने जा रही हूं ताकि यह कुल 7 second हो जाए। अब, मैं marker को 7 seconds में set करूंगी। जैसा कि मैं चाहती हूं कि rat यहां कुछ time के लिए रुके, इसलिए, मैं बस point 9 से point 8 तक इसकी position को minimally change कर रही हूँ, ताकि यह position reset हो सके। मैं recording close करने जा रही हूँ और देखूंगी कि animation play हो रहा है कि नहीं। मैं देख सकती हूँ कि rat animation अच्छी तरह से काम कर रहा है।

Next, हम screen के एक side से दूसरे side में grasshopper को move करने जा रहे हैं। hierarchy window से grasshopper को चुनें और grasshopper को screen के बाहर, सबसे left side में position करें। Animation को add करने के लिए उन्ही steps को repeat करें जैसे हमने rat के लिए किया था। animation folder में "grasshopper" नाम को एक new folder बनाएं और animation को 'Fly' नाम दें।

Recording शुरू करें। grasshopper की initial position (-1083, -375) है। object को reset करने के लिए इसे note कर लें। मैं चाहती हूँ कि grasshopper current position से extreme right side के, screen position के बाहर

जायें। animation timeline को zoom out करें क्योंकि मैं चाहती हूँ कि animation लगभग 10 seconds तक चले, ताकि यह slow चले। बस इस grasshopper को start से end position तक खींचें और move करें। अब मैं position को reset करना चाहती हूँ। ऐसा करने के लिए, frame पर line को फिर से थोड़ा आगे बढ़ाएं और (-1083, -375) position field में set करें। अब, recording बंद करो और animation को play करो। जैसा कि आप देख सकते हैं, grasshopper इस एक end से दूसरे end तक उड़ रहा है और फिर starting point पर फिर से दिखाई दे रहा है।

इसके बाद, हम deers को animate करने का काम करने जा रहे हैं! जहां तक deer का संबंध है, हम दो काम करने जा रहे हैं, हम यह sure करेंगे कि deer का एक walk cycle है और फिर हमें इस deer को screen के एक end से screen के दूसरे end ले जाने की जरूरत है। चलो hierarchy window से deer को चुनें और थोड़ी देर के लिए bushes को disable करें यह देखने के लिए कि deer का walk करना कैसा होगा।

अब तक, deer बिना किसी animation के साथ खड़ा है। सबसे पहले asset, images और जानवर s पर जाएं। last time से, आपको याद रखना चाहिए कि हमने पिछले video में deer की single image को कई images में तोड़ दिया था। deer के walk cycle animation देखने के लिए arrow के sign को expand करें।

एक new folder बनाकर आगे बढ़ें। projects > assets > animations > "deer" पर जायें एक "Deer" नाम का एक new folder बनाएं। animation में, simply right 'walk' बनाएं और इसे save करें। तो, अब deer के लिए एक new animation बन गया है। अब deer animation की सभी images को चुनें और animation panel पर खींचें। deer के लिए animation पूरा हो गया। game start करते हैं और देखते कि यह कैसे काम करता है। जैसा कि हम देख सकते हैं कि speed fast है। small button को 1 minute और 30 second के mark तक खींचकर speed

कम करें। फिर game start करते हैं और animation को फिर से देखते हैं। अभी, deer की चलने की speed ठीक है और deer को animate करने में use किए गए frame को देखने के लिए deer की asset को चुनने के बाद 'sprite editor' पर click करें।

लेकिन मैं चाहती हूँ कि deer बाहर आएँ और पूरा रास्ता चलकर अपनी original position में वापस आ जाएँ।

चलो कोड के माध्यम से ऐसा करने का तरीका ढूँढते हैं! basically deer को screen के centre part के left की ओर से शुरू करना चाहिए। चलो bushes को visible करते हैं। यह screen के last तक चलना चाहिए और फिर से अपनी original position पर वापस आना चाहिए। default position (-284, 200) है। यह essentially एक loop है और ऐसा करने के लिए, asset folder में जाएँ, scripts नाम का एक नया folder बनाएँ। script folder में, right click करें, create चुनें और C sharp script पर click करें। इसे "Deer Walking" नाम दें।

Program लिखने के लिए Visual studio open करने के लिए उस पर double click करें।

दो functions ऐसे हैं जो हमें पहले ही दिए जा चुके हैं - एक है start function जो object को call करता है जिससे यह script attached होती है और update function जिसे animation के सभी frame के साथ लगातार call होगा। पहले हमें दो fields को define करने की जरूरत है।

open parenthesis की next line में, ये लिखें **public Transform startingPoint;**

यह deer का starting point होने जा रहा है, मतलब, left side जहाँ से यह move करना शुरू होता है। एक और variable को define करें

public Transform end point;

यह deer के end point को define करता है। speed by writing नाम का एक और variable को define करता हैं

public float speed;

यह वह speed है जिसके साथ deer start point से end point जाएगा। जैसा कि हम किसी भी object को initialize नहीं कर रहे हैं तो आइए जैसा दिखाया गया है Start() function को delete करें।

हमारे void Update() function में, हम एक simple variable बनाने जा रहे हैं

{

float step = speed X Time.deltaTime;

इस step में deer speed और time के base पर कई step उठाएगा। और delta time दो frames के बीच का difference है जो 2 image references के बीच खींचा जा रहा है। जैसा कि हम इस script को deer के साथ add कर रहे हैं, हम deer की transform position को लेने जा रहे हैं। तो, हम लिखेंगे

transform.position = Vector3.MoveTowards(transform.position, endpoint.position, step);

तो, यह current position , last position और deer कितना distance travel करना चाहता है लेगा । अब एक बार यह last point पर पहुंच जाता है, हम position को starting point पर reset करना चाहते हैं। तो, चलो एक code लिखकर position को check करें

if(Vector3.Distance(transform.position, endpoint.position) < 0.01f);

हम यहां check कर रहे हैं कि क्या deer almost या end point पर पहुंच गया है, हमें start point पर deer की position को reset करना चाहिए। इसलिय, हम

{

Transform.position = startingPoint;

}

लिखेंगे

यह deer को continuously एक place से दूसरे place पर तक ले जाने के लिए move करता रहेगा।

अब हम उन सभी public variables की value decide करते हैं जिन्हें हमने define किया है।

ऐसा करने के लिए, Unity editor पर वापस जाएं और scene को DeerWalking के script को update करने के लिए save करें।

अब, hierarchy window से deer को चुनें और add component के animator window से add component पर click करें। DeerWalking type करके उस script का नाम चुनें जिसे हमने बनाया था, एक बार enter करने के बाद, यह मुझे तीनो public variables यानी starting point, End point and speed दिखा रहा है।

अभी तक, मैंने कोई starting point नहीं बनाया है। Hierarchy window में canvas पर right click करें और एक empty object बनाएं और deer object के पास ले जाएं। इसे Deer Starting Point का नाम दें। इस पर right click करें और

duplicate चुनें। इसे Deer Starting Point के पास खींचें और इसे Deer End Point का नाम दें।

starting point set करने के लिए, Deer Starting Point चुनें और box को (-1080, -200) position और deer के end point को set करने के लिए, Deer End Point चुनें और box को (-1244, -200) position में खींचें। Y-axis पर misbalance से बचने के लिए move tool का use करें।

लेकिन मैं चाहती हूँ कि मेरा deer पेड़ के पीछे उपस्थित हो, इसलिए मैं , deer starting point और Deer end point को चुन कर उन्हें tree object के ऊपर रखने जा रही हूँ। अब, ऐसा लगेगा कि deer tree और bushes के पीछे है।

Deer Walking script के under deer option चुनें और एक new window खुलेगी। Deer Starting Point option पर double click करें। इसी तरह, End Point option के लिय Deer End point को चुनें

speed को 5 set करें। speed fast है। आइए इसे 2 में change करें जो अब optimum है।

अब, देखते हैं कि पूरा scene कैसे काम कर रहा है। scene launch करने के लिए play button पर click करें। हम देख सकते हैं कि हमारा scene पूरी तरह से काम कर रहा है।

इसके बाद, मैं अपने game में जो चाहती हूँ वह यह है कि जब भी मैं किसी जानवर को चुनूँ, तो वह जानवर चुना जाना चाहिए और पूरे scene से गायब हो जाना चाहिए और एक list में जुड़ जाना चाहिए, जहाँ से हम food web बनाते time इसे access कर सकते हैं।

ऐसा करने के लिए, पहले canvas को minimize करें। '+' पर click करें और create empty object चुनें और Transform section में 3 dots पर click करके object को reset करें और reset चुनें। इसे "GameController" नाम दें। एक ही script folder में, "GameController" नाम एक new script बनाएं। इसके बाद,

hierarchy windowसे “GameController” object चुनें और Add Component पर click करें। scene STUDIO open करने के लिए 'GameController' type करें और double click करें। आइए start और update function को हटा दें और समझें कि हम game में क्या करने जा रहे हैं।

तो, “GameController” script बनने जा रहा है जो आपके game के सभी states को handle करेगा। states से मेरा मतलब है, जब game शुरू होगा तो एक welcome screen होगी। इसके बाद, एक दूसरी screen जिसे level screen कहा जाता है, game play level के screen से पहले आता है, third screen हमारे food web screen या level one completion screen के साथ है। इसलिए एक बार जब यह complete हो जाता है, तो आप level 2 screen पर जाते हैं, जहां आप food web and the last, game over screen बनाते हैं। आइए अब इन states को define करें।

ऐसा करने के लिए, हम एक enumeration को define करके शुरू कर सकते हैं। इस game के enumeration में सभी game states होंगे। आप उन्हें initialize कर सकते हैं

```
{
```

```
Enum GameStates
```

```
{
```

```
WelcomeScreen,
```

```
Level1_Gameplay,
```

```
Level1_GameOverScreen,
```


Level2_StartScreen,

Level2_GamePlay,

Level2_Gameover,

GameComplete

}

हम इसमें और states को add कर सकते हैं क्योंकि हम साथ चलते हैं, लेकिन ये कुछ major state हैं जिनमें हम game बनाने जा रहे हैं। तो, अब, चलो welcome screen के बारे में बहुत ज्यादा चिंता न करें क्योंकि यह कुछ ऐसा है जिसे हम दूसरे video में cover करेंगे।

आगे बढ़ने से पहले, आइए किसी भी confusion से बचने के लिए इन states को एक अलग script में एक साथ define करें। ऐसा करने के लिए, Unity editor पर वापस जाएं। अपनी asset में script folder पर जाएं, right click करें और new C sharp script बनाएं और इसे 'Defines' नाम दें। Visual studio open करने के लिए उस पर double click करें। existing functions को delete कर दें।

MonoBehaviour delete करें और public class Defines कर दें

public Static class Defines

और इसे static करें। अब, GameController tab पर जाएं और पूरे code को copy करें और इसे defines के under paste करें।

अब states को बाहर से access करने के लिए, game states class को change करके इसे public करना है

public enum GameStates

“GameController” tab पर वापस आएँ और एक function बनाकर शुरुआत करें

Public void UpdateGameState(Defines.GameStates _gameState)

एक global variable को define करते हैं

```
{
```

Define.GameStates currentGameState;

अब के लिए Level 1 Gameplay screen पर ध्यान दें। इसलिए, हम लिखकर current position को define करने जा रहे हैं

```
{
```

currentGameState = _gameState;

```
}
```

update game state वह function है जिसे हम जब भी पूरे game की state को बदलना चाहेंगे call करेंगे। एक बार जब यह call होता है, तो current game state को save करता है। अब, एक switch function लिखें

switch(_gameState)

```
{
```

case Defines.GameStates.WelcomeScreen:

```
{
```

```
}
```

break;

case Defines.GameStates.Level1_Gameplay:

```
{  
}
```

break;

यह सब “GameController”के बारे में है। हम सभी states के लिए इस file में code करना शुरू करने जा रहे हैं। आइए unity editor पर वापस जाएं और देखें कि हम पहले level 1 में क्या हासिल करना चाहते हैं। पहली चीज जो हम चाहते हैं वह है कि उन पर click करके सभी जानवर को चुन सकें।

इसलिए, हम एक और object बनाएंगे और इसे Level1GamePlay नाम देंगे। उसके according , assest folder में एक C sharp script बनाएं और इसे Level1GamePlay नाम दें। इसके लिए नया tab open करने के लिए double click करें। फिर से start और update function को delete कर दें।

Level 1 game play में, हम चाहते हैं कि users जानवर को चुनें और एक बार चुने जाने के बाद, वे गायब हो जाएं।

ऐसा करने के लिए, हम इसके लिए game object लेकर जानवर को चुनने के लिए एक function बनाने जा रहे हैं और उनमें से सभी को एक unique ID assign करें ताकि उन्हें चुने जाने के बाद पहचाना जा सके। चुने जाने के बाद वे screen से गायब हो जाएंगे। उस के लिए, ये कोड लिखें

```
{
```

Public void Selectजानवर (GameObject _जानवर Object)

```
{
```

Destroy(_जानवर Object);

}

}

unity editor पर वापस जाएं और script को इसे refresh करके script update करने के लिए इसे save करें। अब, hierarchy window से level 1 game play चुनें और initial position में reset करें। Add component पर click करें 'Level1Gameplay' type करें। canvas को maximize करें और जानवर पर एक-एक करके जायें।

चलो ऊपर से fox से शुरू करते हैं। अभी, यह केवल एक image है क्योंकि हमने इसमें कोई event या function नहीं add किया है। Add component पर click करें और 'Event Trigger' type करें। अब option open के लिए 'Add New Event Type' पर click करें और 'PointerClick' चुनें। list से add करने के लिए '+' पर click करें।

अब यह उस function के लिए पूछ रहा है जिसे आप call करना चाहते हैं। तो उस function को add करने के लिए Level1GamePlay को hierarchy window से खींचें और pointer click variable पर drop कर दें। drop down menu से, Level1GamePlay को चुनें और उस function को चुनें जो हमने बनाया है जो Select animal (GameObject) है। अब, आपको parameter या animal D के लिए कहा जाएगा। _animal Object हमारे case में parameter है जो other words में animal है। पहला parameter जिसे हम define कर सकते हैं वह है fox, fox को game object field में खींचें। इसलिए, जब भी मैं fox पर click करूँ, वह गायब हो जाए।

इसी तरह, हम इसे बाकी सभी जानवर के लिए करेंगे। add करने के लिय same steps follow करें Event Trigger > PointerClick > Drag Level1GamePlay and drop it to the pointer click variable > Select Level1GamePlay > Function as Select animal (GameObject) > Drag Giraffe to game object.

Great ! तो यहाँ ये हुआ है कि जब भी animal पर कोई click event होगी, तो वह perticular जानवर function को parameter के साथ Select करेगा जैसे Level1GamePlay script में। चलिए देखते हैं और क्या होता है।

में leopard पर click कर रही हूं, लेकिन यह चुना नहीं जा रहा है। ऐसा इसलिए है क्योंकि एक Raycast enable है जो click function को रोक रहा है। चलो Forest Background को चुनते हैं और आप देख सकते हैं कि Raycast enable है जिसका मतलब है कि Forest Background image भी इस moment interactive है। लेकिन हमारे game में हम यह नहीं चाहते कि सभी objects animals को छोड़कर interactive हों। इसलिए, आइए इसे unticking करके इसे disable करें। चलो सभी all tree, hanging leaves, Bushes के लिए disable करें। raycast को केवल animals के लिए enable करें।

चलिए scene run करते हैं और देखते हैं क्या होता है। जैसा कि आप देख सकते हैं कि हम scene से सभी जानवर को चुन पा रहे हैं। अब तक, हमने सभी जानवरों को animate किया है और Level 1 Gameplay के लिए इसमें selection function add किया है।

तो इस video में इतना ही है। practise करते रहें और हम अगले video में मिलेंगे!