

ENGLISH VERSION

Creating Animal Food Web

Hey everyone! Welcome back to game development tutorial using UNITY!

Today, we are going to create a food web of all the animals that we had selected in level 1. So, let's go ahead and open Unity hub. I'm going to select the food web project. Let's maximize the screen now.

So, if you remember in our last video, we had created a prefab called animal image which we had used to create a grid. So, if we go to hierarchy then canvas and open level 2, all the animals are added in to the grid.

If we go to game controller, we will see that in level 2, we had taken animal image as one of the prefabs. Now, we will select animal image from the project window and click on 'Add component'. Type 'Animal Type' and press enter to assign animal type of component. This is done to identify which image corresponds to which animal.

To add animal type image, select game controller from hierarchy window and double click on Level 2 game play script option in inspector window. It will open the visual studio with the script of level 2 gameplay. Now, we need to assign the animal type to each and every image that we are creating. So, let's write a code for the same as shown on the screen

```
_tempAnimal.GetComponent<AnimalType>().animalType = _animal;
```

What it does it that the temp animal image that we have created get accessed by component and we assign the animal that we are using to generate the image here. So, now the animal type is assigned to every animal.

Next, go back to unity editor and click on play. As you can see, the grid of all animals is created. Let's select Level 2 from hierarchy window and expand animal grid. You can see that clone image of all animals are created and it is showing under animal type.

Press the play button again to stop. Go to the asset window, right click and select create. Click on C sharp script and rename it to 'AnimalScript'. Double click on it to open the visual studio. We have created this script to detect certain events like selection of animal through mouse click that happens on the animal images. Along with that, we're going to use this script to create a logic around selecting first animal, drawing a line from one animal to another animal.

So, let's get started writing a code as shown on the screen. We'll write

Using UnityEngine.EventSystems;

Extend the public class with

, IPointerDownHandler, IPointerUpHandler, IPointerEnterHandler, IPointerExitHandler

So, I've extended all these interfaces. Next, we'll quickly delete the start & update function.

As of now, the interfaces are showing an error because I have to extend and write their function. To do that the easy way, right click on the DownHandler and select Quick Fix and the first option in it. You'll see that it automatically creates the public function for the particular interface. Delete the internal function as it is not required.

Similarly, we can define the function for UpHandler as

Public void OnPointerUp(PointerEventData eventData)

```
{  
}
```

Similarly, write the function for enter & exit handler too. You can do that through quick fix or you can choose to write it manually.

So, we have finished writing all the functions for interfaces of pointer down, enter, exit and up. Now, we will define 2 public variables as

Public static GameObject startAnimal;

Public static GameObject endAnimal;

We have made them static as we want to use them from any other script as well as we want them to contain only one value at the time.

In the pointerdown function, we will define the condition as

```
{  
startAnimal = gameObject;  
}
```

And in the pointerup function, we will define it as

```
{  
startAnimal = null;  
}
```

We'll go back again to pointerdown function and write

```
endAnimal = null;
```

as we do not want to keep the traces of other animals once overwritten

Also, whenever a pointer enters into the image of any of the animal, I want to write the condition in the pointerenter function as

```
If(eventData.pointerEnter != startAnimal)
```

```
{  
endAnimal = gameObject;  
}
```

What does this function do? So, whenever you point or click on any image, the function gives the data associated to that particular image. In our case, we want to detect the particular animals whenever we are on it or click on it then that animal should not be equal to the start animal, which is the animal on which we have already clicked. Don't worry about it for now as I'll explain it while playing the game.

For pointerexit, once the animal is selected, I want to end it and exit that particular animals. So, I will write the code as

endAnimal = null;

So, let's just quickly understand it again in unity editor. Go to Assets, prefabs and select Animal image. Click on Add component in the inspector window and type Animal Script. Press enter to add it. Run the game and select the animals in level 1 as usual. You will notice the changes happening in the hierarchy window automatically.

Once the grid page loads, select the event system and maximize the folder in the inspector window by dragging it upwards from the bottom & click on lock for a moment. As you can see in the log that the position value is constantly changing depending on the movement of pointer on the screen.

So, to click & select the animals, OnPointerDown function gets called where it selects it as a game object. Once the initial animal is selected and then the mouse pointer clicks on any other animal image, then OnPointerEnter function is called. OnPointerExit gets called once you are finished selecting the end animal. Lastly, OnPointerUp is called whenever the left mouse button is pressed and it is released post selection of initial animal.

Now, go back to unity hub. In the asset folder, select scripts. Right click, select create and C-sharp to create a new script. Rename it as 'LineDrawing'. Line drawing basically ensures that when we click on any animal, drag it from there and release that on any of the other animals, then a line is drawn between the first and the last animal.

So, double click on line drawing script to open the code editor. To begin with, we will define some variables. We will start by writing

Vector3 startPosition;

GameObject currentLineObject;

LineRenderer currentLineRenderer;

Each line renderer requires material for which we will write it as

Public Material lineMaterial;

Next, define the thickness of the line by writing

Public float lineThickness;

Also, define the canvas which will ensure to draw the line on a particular canvas.
Write the code as

Public Canvas parentCanvas;

Let's go to the start function and begin writing the code for parent canvas as

```
parentCanvas =  
GameObject.FindObjectOfType<Canvas>().GetComponent<Canvas>();
```

What this code does is, it finds the object of canvas type and then we will take the canvas component of it.

Next, we will go to the update function and write the code for mouse press functions as

```
If(Input.GetMouseButtonDown(0))  
{  
}  
}
```

This first condition will tell us if the mouse button is pressed or not. Next, we will write the other condition

```
Else if (Input.GetMouseButton(0))  
{  
}  
}
```

```
Else if (Input.GetMouseButtonUp(0))  
{  
}  
}
```

The second condition will tell us if the mouse button is still pressed and it is moved on the screen and the third condition will tell us as soon as we release the click on mouse button.

Now, we want to start drawing a line as soon as the mouse button is pressed down. To do that we will define the function in public class as

```
Void StartLineDrawing()
```

```
{  
}
```

Another function would be to preview the line as we are dragging it anywhere on the screen. To do that, we will write

Void PreviewLine()

```
{  
}
```

And the last function would be to stop drawing the line as soon as the mouse button is release. So, let's write it as

Void EndLineDrawing()

```
{  
}
```

So far, we are able to get the coordinates from mouse function in general coordinate system but it is not understood by canvas coordinate system. Hence, we would require to convert the positions received from mouse pointer to canvas coordinates. So, we will write one more function for it above the line drawing function as

Vector3 GetMousePosition()

```
{
```

```
Vector 2 movePos;
```

```
RectTransformUtility.ScreenPointToLocalPointInRectangle(parentCanvas.trans  
form as RectTransform, Input.mousePosition, parentCanvas.worldCamera,  
out movePos);
```

```
Vector3 positionToReturn =  
parentCanvas.transform.TransformPoint(movePos);
```

```
positionToReturn.z = parentCanvas.transform.position.z - 0.01f;
```

```
return positionToReturn;
```

```
}
```

Now, let's write the code for the start line drawing function as

```
startPosition = GetMousePosition();  
currentLineObject = new GameObject();  
currentLineObject.transform.position = startPosition;  
currentLineRenderer = currentLineObject.AddComponent<LineRenderer>();  
currentLineRenderer.material = lineMaterial;  
currentLineRenderer.startWidth = lineThickness;  
currentLineRenderer.endWidth = lineThickness;
```

So, this code gives me the start position and set it as a marking point and current line renderer will have the line material with a particular thickness at both the starting and ending point.

Let's write the code for preview line function which will show the line drawing from the starting to ending point as

```
currentLineRenderer.SetPositions(new Vector3[] { startPosition,  
GetMousePosition() });
```

As soon as we release the mouse button press, we need to reset it to the start position. Let's start writing the code for End Line drawing function as

```
startPosition = Vector3.zero;  
currentLineObject = null;  
currentLineRenderer = null;
```

Next, let's call these functions in the update function as

```
StartLineDrawing();
```

in first condition

```
PreviewLine();
```

in second condition

EndLineDrawing();

In third condition respectively.

So, let's go back to unity hub and let it compile all the scripts we have just written. Select Level 2 from hierarchy and unlock the inspector window. Drag the level 2 preview log window to the bottom. Now, click on Add component and type 'Line Drawing' and enter. Line drawing script gets added.

Now click on asset and create a new folder called Material. Open the material folder and click on + button. Select material option and rename it Line Material. In the inspector window, and select shader option. From the drop down, select UI and default option. Click on tint and select the black colour and close the window.

Select Level 2 again from hierarchy window and drag line material from asset section to Line material field as shown. Change the line thickness to 0.05. Click on play to test the progress made so far.

So, as you can see that our lines are getting created however, there is no fixed state. We need to ensure that the lines start from one animal and end with another. We will get to fixing that. But for now, we can draw lines! Let's switch back to the edit mode and go to the script.

As we remember to create 2 static variables i.e. start and end animal in the animal script. So, go to the line drawing script and in the mouse pointer down function, we will define

If(AnimalScript.startAnimal != null)

It means that we would like to start drawing the line if the start animal is selected and not from any blank area.

In the second condition, we will define

If(AnimalScript.startAnimal != null)

It means that we would like to preview the line only if the start animal is selected.

In the endline drawing function, we will write the code as

If(AnimalScript.endAnimal != null)


```
{
```

```
Destroy(currentLineObject);
```

```
}
```

It means that line should destroy itself if the end line is not present on the end animal. For example, if the deer is the start animal and line is dragged to empty space, then the line should get destroyed.

Let's go back to the unity hub and test our code. Click on the play button and maximize the game window. Select all the animals from the screen to go to the next level.

On the animal grid, start drawing the line from leopard to deer. If you properly draw a line from one image to another, you will be able to draw it just like you saw on the screen.

With that, this video comes to an end! As for the next video, we will work with creating a scoring system for our game! This means that if and when I draw lines connecting the right animals in the right order to create a food web, I will get points, however if I go wrong by connecting incorrect animals to each other, my points will reduce accordingly! This is what we are going to look at in our next video and I hope you're as excited for it as I'm!

Hope to see you very soon! Bye Bye!

HINDI VERSION

Creating Animal Food Web

Hey everyone! UNITY का उपयोग करके game development tutorial में आपका स्वागत है!

आज, हम उन सभी जानवरों का एक food web बनाने जा रहे हैं जिन्हें हमने level 1 में चुना था। तो, चलिए आगे बढ़ते हैं और unity hub open करते हैं। मैं food web project को चुनने जा रही हूँ। अब screen को maximize करें।

अगर आप को हमारे पिछले video याद हैं, तो हमने एक prefab बनाया था जिसका नाम animal image था, जिसे हमने grid बनाने के लिए use किया था।, अगर हम hierarchy में जाते हैं फिर canvas में जाते हैं और level 2 open करते हैं वहाँ सभी जानवर को grid में add हो गए हैं।

अगर हम game controller के पास जाते हैं, तो हम देखेंगे कि level 2 में, हमने animal image को अपना एक prefab बनाया था। अब, हम project window से जानवरों की image को चुनेंगे और 'add component' पर click करेंगे। 'animal type' लिखें और animal type के component assign करने के लिए enter दबाएं। यह पहचानने के लिए किया जाता है कि कौन सी image किस जानवर से मेल खाती है।

animal type image को add करने के लिए, hierarchy window से game controller को चुनें और inspector window में level 2 game play script option पर double click करें। यह level 2 gameplay की script के साथ visual studio खोलेगा। अब, हमें अपने द्वारा बनाई जा रही प्रत्येक image को animal type assign करना होगा। तो, आइए उसके लिए code लिखें जैसा screen पर दिखाया गया है

```
_tempAnimal.GetComponent<AnimalType>().animalType = _animal;
```

यह क्या करता है कि हमने जो temp animal image बनाई है वह component द्वारा access की जाती है और हम उस जानवर को assign करते हैं जिसका उपयोग हम यहां image बनाने के लिए कर रहे हैं। तो, अब animal type हर जानवर को assign हो गया है।

Next, unity editor पर वापस जाएं और play पर click करें। जैसा कि आप देख सकते हैं, सभी जानवरों का grid बनाया गया है। आइए hierarchy window से level 2 को चुनें और animal grid को expand करें। आप देख सकते हैं कि सभी जानवरों की clone image बनाई गई है और यह animal type के under दिखाई दे रही है।

बंद करने के लिए फिर से play बटन दबाएँ। asset window पर जाएं, right click करें और create चुनें। C sharp script पर click करें और इसे 'AnimalScript' का नाम दें। visual studio open करने के लिए उस पर double click करें। हमने इस script को certain

events का पता लगाने के लिए किया है, जैसे कि animal images पर mouse click से animal को चुनना।

इसके साथ ही, हम इस script का उपयोग करने जा रहे हैं ताकि पहले जानवर को चुनने के लिए logic तैयार किया जा सके, एक जानवर से दूसरे जानवर की एक रेखा खींची जा सके।

तो, चलिए एक code लिखना शुरू करते हैं जैसा कि screen पर दिखाया गया है। हम लिखेंगे

Using UnityEngine.EventSystems;

Extend the public class with

, IPointerDownHandler, IPointerUpHandler, IPointerEnterHandler, IPointerExitHandler

इसलिए, मैंने इन सभी interface को बढ़ा दिया है। अब, हम start & update function को जल्दी से delete कर देंगे।

अब तक, interface एक error दिखा रहा है क्योंकि मुझे उनके function को extend करना और लिखना है। ऐसा करने का आसान तरीका है DownHandler पर right click करें और Quick Fix चुनें और उसमें पहला option चुनें। आप देखेंगे कि यह particular interface के लिए public function को automatically से बनाता है। internal function को हटा दें क्योंकि इसकी जरूरत नहीं है।

इसी तरह, हम UpHandler के लिए function को define कर सकते हैं

Public void OnPointerUp(PointerEventData eventData)

```
{  
}
```

इसी तरह, enter & exit handler के लिए भी function लिखें। आप इसे quick fix से कर सकते हैं या आप इसे manually लिखना चुन सकते हैं।

इसलिए, हमने pointer के interface down, enter, exit and up के सभी functions लिखना पूरा कर दिया है। अब, हम 2 public variables को define करेंगे

Public static Game Object start Animal;

Public static Game Object end Animal;

हमने उन्हें static बना दिया है क्योंकि हम उन्हें किसी भी script से उपयोग करना चाहते हैं और साथ ही हम चाहते हैं कि उन्हें उस time केवल एक value हो।

pointerdown function में, हम condition को इस प्रकार define करेंगे

```
{  
startAnimal = gameObject;  
}
```

और pointer function में, हम इसे ऐसे define करेंगे

```
{  
Start Animal = null;  
}
```

हम दोबारा pointer down function पर वापस जाएंगे और लिखेंगे

End Animal = null;

जैसा कि हम एक बार over written होने के बाद बाकी जानवरों का trace नहीं चाहते हैं

इसके अलावा, जब भी कोई pointer किसी भी जानवर की image में enter करता है, तो में pointer function में condition लिखना चाहती हूँ

If(eventData.pointerEnter != startAnimal)

```
{
```

```
endAnimal = gameObject;
```

```
}
```

यह function क्या करता है? , जब भी आप किसी image को point या click करते हैं, तो function उस particular image से related data देता है। हमारे मामले में, हम particular animals का पता लगाना चाहते हैं जब भी हम उस पर होते हैं या उस पर click करते हैं तो उस animal को start animal के बराबर नहीं होना चाहिए, जो कि वह animal है जिस पर हमने पहले ही click किया है। अभी इसके बारे में चिंता न करें क्योंकि मैं game खेलते समय इसे समझाऊंगी।

Pointer exit के लिए, एक बार जानवर चुने जाने के बाद, मैं इसे end करना चाहती हूँ और उस particular animals से बाहर निकलना चाहती हूँ। तो, मैं code लिखूंगी

```
End Animal = null;
```

तो, चलिए unity editor में इसे फिर से समझते हैं। Assets, prefabs पर जाएं और Animal image चुनें। inspector window में Add component पर click करें और Animal Script type करें। इसे add करने के लिए enter दबाएं। game को run करें और हमेशा की तरह level 1 में जानवरों को चुनें

आप hierarchy window में होने वाले changes को automatically देखेंगे।

grid page load होने के बाद, event system को चुनें और inspector window में folder को नीचे से ऊपर की ओर खींचकर और एक पल के लिए lock पर click करें। जैसा कि आप log में देख सकते हैं कि screen पर pointer के movement के base पर position value लगातार बदल रही है।

इसलिए, animals को click करने और चुनने के लिए, On Pointer Down function को call किया जाता है, जहां वह इसे game object के रूप में चुनता है। एक बार initial animal को चुन लिया जाता है और फिर mouse pointer किसी और जानवर की image पर click करता है, तो On Pointer Enter function call होता है। एक बार जब आप end animal

को चुन लेते हैं तो On Pointer Exit call होता है। अंत में, जब भी left mouse button दबाया जाता है तो On Pointer Up call होता है और ये initial animal के post selection को release कर देता है।

अब, unity hub पर वापस जाएं। asset folder में, script को चुनें। नई script बनाने के लिए right click करें, create और C-sharp को चुनें। इसे 'LineDrawing' नाम दें। Line drawing यह ensure करती है कि जब हम किसी भी animal पर click करते हैं, तो उसे वहां से खींचकर किसी और जानवर पर छोड़ देते हैं, फिर पहले और आखिरी जानवर के बीच एक line खींच जाती है।

इसलिए, code editor open करने के लिए line drawing script पर double click करें। शुरू करने के लिए, हम कुछ variable को define करेंगे। हम लिखकर शुरू करेंगे

Vector3 startPosition;

GameObject currentLineObject;

LineRenderer currentLineRenderer

हर line renderer को material की जरूरत होती है जिसके लिए हम इसे लिखेंगे

Public Material line Material;

अगला, line की thickness को लिखकर define करें

Public float lineThickness;

इसके अलावा, canvas को define करें जो एक particular canvas पर line खींचना ensure करेगा। code लिखें

Public Canvas parent Canvas;

चलिए start function में जाते हैं और parent canvas के लिए code लिखना शुरू करते हैं

```
parentCanvas =  
GameObject.FindObjectOfType<Canvas>().GetComponent<Canvas>();
```

यह code क्या करता है, यह canvas type की object को खोजता है और फिर हम इसके canvas component को ले लेंगे।

अब, हम update function पर जाएंगे और mouse press functions के लिए code लिखेंगे

If(Input.GetMouseButtonDown(0))

```
{  
}
```

यह पहली condition हमें बताएगी कि mouse button दबाया गया है या नहीं। आगे, हम दूसरी condition लिखेंगे

Else if (Input.GetMouseButton(0))

```
{  
  
}
```

Else if (Input.GetMouseButtonUp(0))

```
{  
  
}
```

दूसरी condition हमें बताएगी कि क्या mouse button अभी भी दबाया गया है और इसे screen पर move किया गया है और third condition जैसे ही हम mouse button पर click को हटाते उसके बारे में बतायगा।

अब, हम mouse button को दबाते ही एक line खींचना शुरू करना चाहते हैं। ऐसा करने के लिए, हम function को public class में define करेंगे

Void StartLineDrawing()

```
{  
  
}
```

एक और function line का preview देगा क्योंकि हम इसे screen पर कहीं भी खींच रहे हैं। ऐसा करने के लिए, हम लिखेंगे

Void PreviewLine()

```
{  
}
```

और last function mouse button को release करते ही line को खींचना बंद करना होगा। तो, चलो इसे लिखें

Void EndLineDrawing()

```
{  
}
```

अब तक, हम general coordinate system में mouse function से coordinates ले रहे हैं लेकिन यह canvas coordinate system द्वारा समझ में नहीं आता है। इसलिए, हम mouse pointer से मिले pointer को canvas coordinates में बदलेंगे। तो, हम line drawing function के ऊपर इसके लिए एक और function लिखेंगे

Vector3 GetMousePosition()

```
{
```

Vector 2 movePos;

RectTransformUtility.ScreenPointToLocalPointInRectangle(parentCanvas.transform as RectTransform, Input.mousePosition, parentCanvas.worldCamera, out movePos);

Vector3 positionToReturn = parentCanvas.transform.TransformPoint(movePos);

positionToReturn.z = parentCanvas.transform.position.z - 0.01f;

return positionToReturn;

अब, start line drawing function के लिए code लिखें

Start Position = Get Mouse Position();

Current Line Object = new Game Object();


```
Current Line Object.transform.position = startPosition;
```

```
Current Line Renderer = current Line Object.AddComponent<LineRenderer>();
```

```
currentLineRenderer.material = lineMaterial;
```

```
currentLineRenderer.startWidth = lineThickness;
```

```
currentLineRenderer.endWidth = lineThickness;
```

तो, यह code मुझे start position देता है और इसे marking point के रूप में set करता है और current line renderer में starting और ending point दोनों पर particular thickness के साथ line material होगी।

चलिए preview line function के लिए code लिखते हैं जो line drawing को starting से ending point तक दिखाएगा

```
currentLineRenderer.SetPositions(new Vector3[] { startPosition, GetMousePosition() });
```

जैसे ही हम mouse button press को release करते हैं, हमें इसे start position में reset करना होगा। आइए End Line drawing function के लिए code लिखना शुरू करें

```
startPosition = Vector3.zero;
```

```
currentLineObject = null;
```

```
currentLineRenderer = null;
```

इसके बाद, इन function को update function के रूप में call करें

```
StartLineDrawing();
```

पहली condition में

```
PreviewLine ();
```

दूसरी condition में

```
EndLineDrawing ();
```

third condition में।

तो, चलिए unity hub पर लौटते हैं और इसे उन सभी script को compile करते हैं जिन्हें हमने अभी लिखा है। hierarchy से level 2 चुनें और inspector window unlock करें। level 2 preview log window को नीचे तक खींचें। अब, Add component पर click करें और 'Line Drawing' type करें और enter दबाएँ। Line drawing script add हो जाती है।

अब asset पर click करें और Material नाम का एक new folder बनाएं। material folder खोलें और + बटन पर click करें। material option चुनें और इसे Line Material का नाम दें। inspector window में, और shader option चुनें। drop down से, UI और default option चुनें। tint option पर click करें और black colour चुनें और window close करें।

जैसा दिखाया गया है hierarchy window से level 2 को फिर से चुनें और asset section से Line material field तक Line material खींचें। line की thicknes को 0.05 में बदलें। अब तक हुई progress का test करने के लिए play पर click करें।

इसलिए, जैसा कि आप देख सकते हैं कि हमारी लाइनें बन रही हैं, लेकिन कोई fix state नहीं है। हमें यह ensure करने की जरूरत है कि lines एक जानवर से शुरू होती हैं और दूसरे के साथ समाप्त होती हैं। हम इसे fix करेंगे। लेकिन अभी के लिए, हम lines खींच सकते हैं! आइए edit mode पर वापस जाएं और script पर जाएं।

जैसा कि हमें 2 static variables बनाने है जो कि start और end animal है animal script में, line drawing script पर जाएं और mouse pointer down function में, हम define करेंगे

If(AnimalScript.startAnimal != null)

इसका मतलब है कि हम line को draw करना शुरू करना चाहते हैं अगर start animal चुना हुआ है और किसी खाली area से नहीं।

दूसरी condition में, हम define करेंगे

If(AnimalScript.startAnimal != null)

इसका मतलब है कि हम line का preview तभी करना चाहेंगे जब start animal चुना हुआ हो।

endline drawing function में, हम code को इस तरह लिखेंगे

If(AnimalScript.endAnimal != null)

```
{
```

```
Destroy(currentLineObject);
```

```
}
```

इसका मतलब है कि अगर end line end animal पर present नहीं है तो line को खुद को खत्म कर देना चाहिए। उदाहरण के लिए, अगर deer starting animal है और line को empty space पर घसीटा जाता है, तो line नष्ट हो जानी चाहिए। आइए unity hub पर वापस जाएं और हमारे code को test करें। play button पर click करें और game window को maximize करें। next level तक जाने के लिए screen से सभी animal को चुनें।

animal grid पर, leopard से deer तक की line खींचना शुरू करें। अगर आप ठीक से एक image से दूसरी image तक एक रेखा खींचते हैं, तो आप इसे वैसे ही खींच पाएंगे जैसे आपने screen पर देखा था।

इसके साथ, यह video समाप्त हो जाता है! अगले video के लिए, हम अपने game के लिए scoring system बनाने के साथ काम करेंगे! इसका मतलब यह है कि अगर जब मैं एक food web बनाने के लिए सही order में सही जानवरों को जोड़ने वाली lines खींचती हूं, तो मुझे points मिलेंगे, हालांकि अगर मैं गलत जानवरों को एक-दूसरे से जोड़कर गलत हो जाती हूं, तो मेरे point उसके अनुसार कम हो जाएंगे! यह हम अपने अगले video में देखने जा रहे हैं और मुझे आशा है कि आप इसके लिए उतने ही excited होंगे जितना कि मैं!

उम्मीद है बहुत जल्द ही आपसे मिलूंगी ! bye bye!