In this lesson we will start adding our Score and we will also create a score manager That will take care of how to increment our score and when to stop counting our score and that will also take care of saving our score to our mobile ,our computer So lets get started doing that First we are goin to create an empty game object and I am going to quickly rename it to Score Manager and create a script called as Score Manager script

And open it in Visual Script

Now first of all we are going to create singleton instance of these ScoreManager

Se we write public static ScoreManager instance

And what we did in awake of UIManager we will do it here.

So we have created singleton instance of our ScoreManager

Now we will create two function

One name is increment Score of return type void

This function will increment our score by one

So first we need to do is declare a variable public int score

So this variable will store our scores

And now in start method we set score to 0

Let desclare another variable highScore which will store our high score

We will set it later

So inside our increment Score function we gonna say

Score+=1

So this will increment our score by 1 it means if there score 2

Now we are going to create another method and name it as startScore method

This start score will repeatedly call our increment score function

So that our score will get incremented repeatedly

So inside it we call the function Invoke Repeating

And as we know we pass it the name of the function that we want to repeat anf after what time we want to call our function after start of the game (I take 0.1f mean 1/10 of the second )

And we pass it the time after which we want to call the function repeatedly (we take 0.5f which means we take ½ seconds)

Now our whole code look like this

public class ScoreManager : MonoBehaviour

{

// Start is called before the first frame update

public static ScoreManager instance;

public int score;

public int highScore;

void Awake()

{

instance = this;

}

void Start()

{

score = 0;

}

void incrementScore()

{

score += 1;

}

// Update is called once per frame

void startScore()

{

InvokeRepeating("incrementScore", 0.1f, 0.5f);

}

void Update()

{

}

}

3:30

So we want to call the incrementScore repeatedly so that it get increased again and again after some interval

Now we create another function called void StopScore()

So this method simply cancel invoking repeatedly our stopScore function so that our score get stopped

So we call CancelInvoke here It will cancel the repeated invoking of the function passed to it

Since we are invoking startScore repeatedly we will pass its name as a string

We need to make these function public so that we can access it from other script

Ok so what we need to do now is that we need to call these function repeatedly from our gameManager so we can control all of these things

Before doing that let us try to save these scores on our machine

If we want to save any score or data on machines then you need to learn about player preps or player preference This player preps help us to store any data in our machine with a key

That key is like that it will give us a value

Let say I store score key with one value

So later on, whenever I give the key score, I will get the value one

Ok so this ia the way we can save our data with the key and later on we can get it with that key

So inside our Start method when our game start

We write code

PlayerPrefs.SetInt method

By this method we are storing an integer since our score is an integer so we need to store it as an integer So first all we want to give the key name that we want to give for the key value for the key value we are going to give score value

So our score will be saved on our computer or device as this score key

So later on if we want ot get the value of the score we are gonna give this score key and we will automatically get these score

So inside the Start method we are storing the score value as zero

Ans we will do is that when we will stop calculating our score and at that time after stopping the score calculation we again store the score like we did in StartScore

public void StopScore()

{

CancelInvoke("incrementScore");

PlayerPrefs.SetInt("score", score);

}

When we stop our score then it means our score is not incrementing anymore at that time we want to simple score to the score key what we need do is save our high score

Now How you do it

Highscore is something that will always be saved in your computer

We need to check if the score is greater than high score saved in our computer, then we want to save our current score as the new high score otherwise we are not going to change the high score because that is already the high score so we used If logic

And at first we check that if the highScore is already stored in the device if no then our correct score is the high score if yes then we check if the current score is greater than the already saved high score then that if yes then then store the current score as high score and if not then does not do anything

So below is the code that do this

public void StopScore()

{

CancelInvoke("incrementScore");

PlayerPrefs.SetInt("score", score);

if (PlayerPrefs.HasKey("highScore"))

{

if (score > PlayerPrefs.GetInt("highScore"))

{

PlayerPrefs.SetInt("highScore", score);

}

}

else

{

PlayerPrefs.SetInt("highScore", score);

}

}

PlayerPrefs.GetInt

GetInt method retriever the value associated with a key

So by this way we are creating this score manager Singleton and setting our score with it

So in the next lesson we will learn about how we will combine our UIManager , scoreManager and Control all of these things from our GameManager and we will fulfil almost all the mechanism of this game.