2.Add the 3d assets + first few lines of code

1. 1. If you want, drag a different **material** from *Course Library > Materials* onto the Ground object
   2. Drag 1 **Human**, 3 **Animals**, and 1 **Food** object into the Hierarchy
   3. Rename the character “Player”, then **reposition** the animals and food so you can see them
   4. Adjust the XYZ **scale** of the food so you can easily see it from above

- **Attach** the script to the Player and open it
- In your **Assets** folder, create a “Scripts” folder, and a “PlayerController” script inside
- At the top of PlayerController.cs, declare a new
- [SerializeField]
- ***private float horizontalInput***
- In ***Update()***, set ***horizontalInput = Input.GetAxis(“Horizontal”)***, then test to make sure it works in the inspector
-

```
using UnityEngine;

public class Movement : MonoBehaviour
{
    [SerializeField]
    private float horizontalInput;
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        horizontalInput = Input.GetAxis("Horizontal");
    }
}
```

4.Move the player left-to-right

- here we can get our input with this basic code, we use [SerializeField] for sanitization and allow us to see the value in the inspection panel to make sure we didn't fck up lol
- Declare a new ***public float speed = 10.0f;***
- ***here's my code***

```
using UnityEngine;

public class Movement : MonoBehaviour
{
    [SerializeField]
    private float horizontalInput;
    [SerializeField]
    private float speed = 30.0f;
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        horizontalInput = Input.GetAxis("Horizontal");
        transform.Translate(Vector3.right * horizontalInput * Time.deltaTime * speed);
    }
}
```

5+6. Keep the player inbounds

- In ***Update()***, write an **if-statement** checking if the player’s left X position is **less than** a certain value

- In the if-statement, set the player's position to its current position, but with a **fixed X location**
- Repeat this process for the **right side** of the screen
- Declare new ***xRange*** variable, then replace the hardcoded values with them
- Add **comments** to your code or not lol

```
using UnityEngine;

public class Movement : MonoBehaviour
{
    [SerializeField]
    private float horizontalInput;
    [SerializeField]
    private float speed = 10.0f;
    [SerializeField]
    private float xRange = 20.0f;
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        if (transform.position.x < -xRange)
        {
            transform.position = new Vector3(-xRange, transform.position.y ,
transform.position.z);
        }
        if (transform.position.x > xRange)
        {
            transform.position = new Vector3(xRange, transform.position.y,
transform.position.z);
        }
        horizontalInput = Input.GetAxis("Horizontal");
        transform.Translate(Vector3.right * horizontalInput * Time.deltaTime *
speed);
    }
```

```
}
```

New Functionality

The player can move left and right based on the user’s left and right key presses

The player will not be able to leave the play area on either side
New Concepts & Skills
Adjust object scale
If-statements
Greater/Less than operators
Next Lesson, We’ll learn how to create and throw endless amounts of food to feed our animals!

now use chat gpt to teach me thing and fix a bug that can occur when you move fast

float clampedX = Mathf.Clamp(desiredpoz.x, -Xrange, Xrange) become the min/max

```
using UnityEngine;
//needed a bit of editing coz cgpt3.5 is stooooooooopid
public class Movement : MonoBehaviour
{
    [SerializeField]
    private float speed = 10.0f;
    [SerializeField]
    private float xRange = 10.0f;
    [SerializeField]
    private float horizontalInput;
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        horizontalInput = Input.GetAxis("Horizontal");

        // Calculate the desired position based on the input
        Vector3 desiredPosition = transform.position + Vector3.right * -horizontalInput *
Time.deltaTime * speed;
```

```
        // Clamp the desired position within the x range
        float clampedX = Mathf.Clamp(desiredPosition.x, -xRange, xRange);
        desiredPosition = new Vector3(clampedX, desiredPosition.y, desiredPosition.z);
        // Move the object to the clamped position
        transform.position = desiredPosition;
    }
}
```

Revision #2
Created 2023-05-20 23:47:54 UTC by naruzkurai
Updated 2023-06-18 09:54:37 UTC by naruzkurai