3. Instantiation

The first thing we must do is give the projectile some forward movement so it can zip across the scene when it's launched by the player.

1.Make the projectile fly forwards

- Create a new "MoveForward" script, **attach** it to the food object, then open it
- Declare a new **public float speed** variable**;**
- In **Update()**, add **transform.Translate(Vector3.forward * Time.deltaTime * speed);**, then **save**
- In the **Inspector**, set the projectile's **speed** variable, then test

```
using UnityEngine;


public class Pizza_gun : MonoBehaviour

{

    public GameObject Ammo;


    public float Weapon_Bullet_Projectile_Speed = 60.0f;

    // Start is called before the first frame update

    void Start()

    {

        Ammo = GetComponent<GameObject>();

    }


    // Update is called once per frame

    void Update()

    {

        transform.Translate(Vector3.forward * Time.deltaTime * Weapon_Bullet_Projectile_Speed);

    }

}
```

- Create a new "Prefabs" folder, drag your food into it, and choose **Original Prefab**
- In PlayerController.cs, declare a new **public GameObject projectilePrefab;** variable

- **Select** the Player in the hierarchy, then **drag** the object from your Prefabs folder onto the new **Projectile Prefab box** in the inspector
- Try **dragging** the projectile into the scene at runtime to make sure they fly
- I added the ammo = get component thing for maybe powerups
- In PlayerController.cs, in **Update()**, add an **if-statement** checking for a spacebar press: **if (Input.GetKeyDown(KeyCode.Space)) {}**
- Inside the if-statement, add a comment saying that you should **// Launch a projectile from the player**
- Inside the if-statement, use the **Instantiate** method to spawn a projectile at the player's location with the prefab's rotation
- **Select** all three animals in the hierarchy and *Add Component* > **Move Forward**Rotate all animals on the Y axis by **180 degrees** to face down
- Edit their **speed values** and **test** to see how it looks
- Drag all three animals into the **Prefabs folder**, choosing "Original Prefab"
- **Test** by dragging prefabs into scene view during gameplay
-

```
using UnityEngine;

public class Movement : MonoBehaviour
{
    [SerializeField]
    private float speed = 10.0f;
    [SerializeField]
    private float xRange = 10.0f;
    [SerializeField]
    private float horizontalInput;
    [SerializeField]
    public GameObject ProjectilePrefab;
    // 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;

    if (Input.GetKeyDown(KeyCode.Space))

    {

        // shoot pizza

        Instantiate(ProjectilePrefab, transform.position, ProjectilePrefab.transform.rotation);

    }

  }

}
```

5.Make animals into prefabs

- **Rotate** all animals on the Y axis by **180 degrees** to face down/ towards the player
- **Select** all three animals in the hierarchy and *Add Component > **Move Forward***
- Edit their **speed values** and **test** to see how it looks
- Drag all three animals into the **Prefabs folder**, choosing "Original Prefab"
- **Test** by dragging prefabs into scene view during gameplay

6.Destroy projectiles offscreen

- Create "DestroyOutOfBounds" script and apply it to the **projectile**
- Add a new **_private float topBound_** variable and initialize it **_= 30;_**
- Write code to destroy if out of top bounds **_if (transform.position.z > topBound) { Destroy(gameObject); }_**
- In the Inspector **Overrides** drop-down, click **Apply all** to apply it to prefab

7. Destroy animals offscreen

- Create **else-if statement** to check if objects are beneath **_lowerBound_**: **_else if (transform.position.z < lowerBound)_**
- **Apply** the script to all of the animals, then **Override** the prefabs
-

8.Lesson Recap

New Functionality

- The player can press the Spacebar to launch a projectile prefab,

Projectile and Animals are removed from the scene if they leave the screen

New Concepts & Skills

- Create Prefabs
- Override Prefabs
- Test for Key presses
- Instantiate objects
- Destroy objects
- Else-if statements

Next Lesson

- Instead of dropping all these animal prefabs onto the scene, we'll create a herd of animals roaming the plain!

(

Lesson 2.2 - Food Flight
Skills practiced:
Absolute Beginner Code Comprehension
Interpret simple code
Improve simple code using the features of an IDE
Absolute Beginner Application Scripting
Use common logic structures to control the execution of code.
Write code that utilizes the various Unity APIs
Implement appropriate data types
Write code that integrates into an existing system
Implement a code style that is efficient and easy to read
Prototype new concepts
)

Revision #2
Created 21 May 2023 00:44:25 by naruzkurai
Updated 18 June 2023 09:54:37 by naruzkurai