ICE-3702: Advanced Game Development

Assignment 2: Interaction Showcase

Name: Maximilian Clarke

User ID: 500581367

Player Controller

I used Unity's Character Controller component in order to let the player move around the scene – In the controller, I included methods that allows the player to walk, sprint, crouch and jump. I also created a script that made use of the main camera so that it could be controlled by the mouse.

Miscellaneous Files

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerMovement : MonoBehaviour
{

    public CharacterController controller;

    public float speed = 7.5f;
    public float gravity = -9.81f;
    public float jumpHeight = 3f;

    public Transform groundCheck;
    public float groundDistance = 0.4f;
    public LayerMask groundMask;

    bool isSprinting = false;
    bool isCrouching = false;

    CharacterController characterCollider;

    public Texture2D crossHair;

    public GameObject cam;

    Vector3 velocity;
    bool isGrounded;

    private void Start()
    {
        characterCollider = gameObject.GetComponent<CharacterController>();
    }


    void Update()             //Allows the player to use keyboard input to move and jump
    {
        isGrounded = Physics.CheckSphere(groundCheck.position, groundDistance, groundMask);

        if (isGrounded && velocity.y < 0)
        {
            velocity.y = -2f;
        }

        float x = Input.GetAxis("Horizontal");
        float z = Input.GetAxis("Vertical");

        Vector3 move = transform.right * x + transform.forward * z;

        controller.Move(move * speed * Time.deltaTime);

        if(Input.GetButtonDown("Jump") && isGrounded)
        {
            velocity.y = Mathf.Sqrt(jumpHeight * -2f * gravity);
        }

        velocity.y += gravity * Time.deltaTime;

        controller.Move(velocity * Time.deltaTime);

        if (Input.GetKey(KeyCode.LeftShift).Equals(true) && isCrouching == false)
        {
            isSprinting = true;
            speed = 15f;
        }
        else if(isCrouching == false)
        {
            isSprinting = false;
            speed = 7.5f;
        }

        if (Input.GetKey(KeyCode.LeftControl).Equals(true) && isSprinting == false)
        {
            isCrouching = true;
            characterCollider.height = 1.9f;
            speed = 4f;
        }
        else if (isSprinting == false)
        {
            isCrouching = false;
            characterCollider.height = 3.8f;
            speed = 7.5f;
        }

    }
}
```

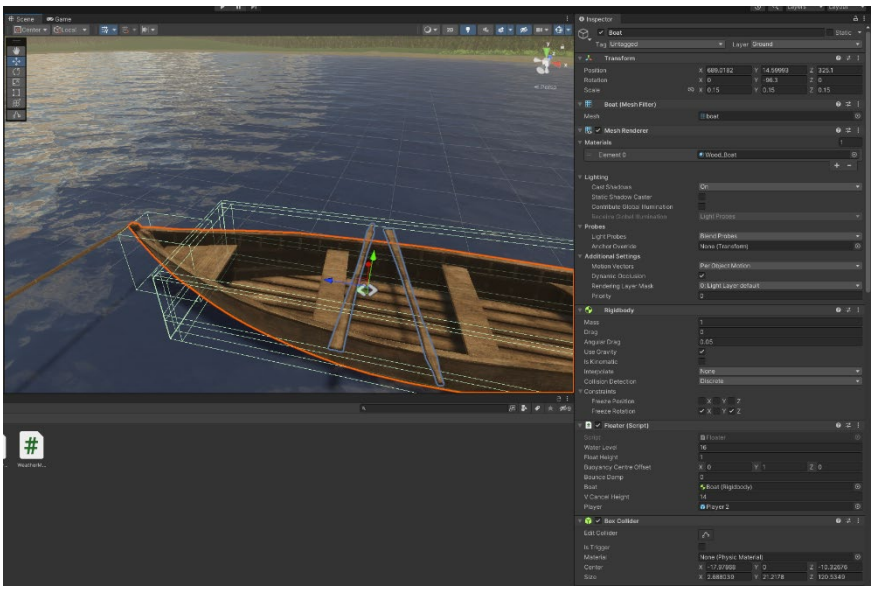
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class MouseLook : MonoBehaviour
{

    public float mouseSensitivity = 100f;

    public Transform playerBody;

    float xRotation = 0f;

    void Start()              //Removes cursor
    {
        Cursor.lockState = CursorLockMode.Locked;
    }

    void Update()             //Allows player to use mouse input to look around
    {
        float mouseX = Input.GetAxis("Mouse X") * mouseSensitivity * Time.deltaTime;
        float mouseY = Input.GetAxis("Mouse Y") * mouseSensitivity * Time.deltaTime;

        xRotation -= mouseY;
        xRotation = Mathf.Clamp(xRotation, -90f, 90f);

        transform.localRotation = Quaternion.Euler(xRotation, 0f, 0f);
        playerBody.Rotate(Vector3.up * mouseX);
    }
}
```

Interaction

For the interactions, I imported some rocks for which I used a rigid body component for their physics. In order to make them interactable for the player, I created a script that allows the player to select the nearest rock with left-click, and this would make the rock a child of the player so that it would move with the player, and when picked up, gravity is disabled for the rock until it is let go by the player.

Rigid Bodies

I used Rigid bodies and colliders for all my physics-based objects, these include the rocks, boat and rope.



Joints / Springs

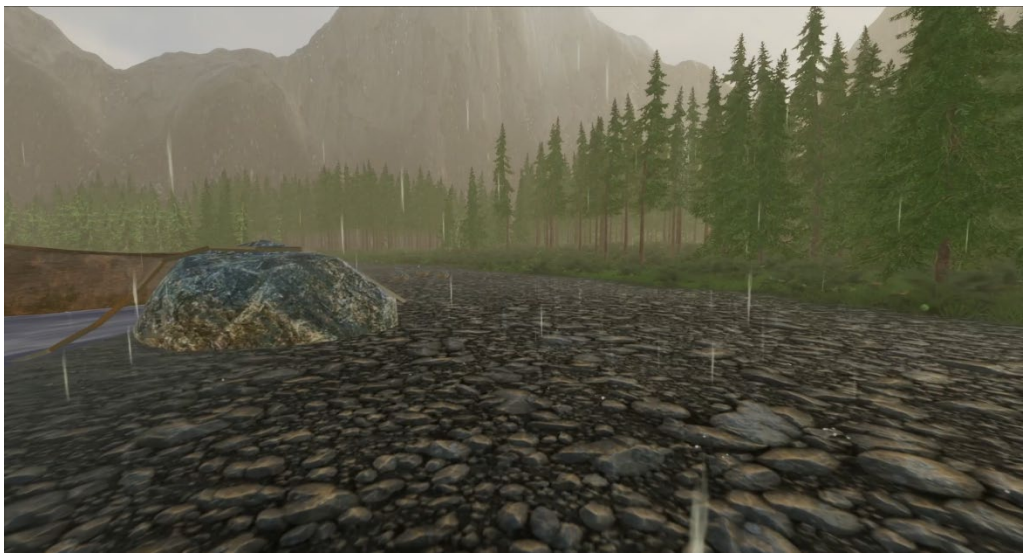
I used joints to make a simple rope using cubes, which I attached to my boat object, for which I used an asset that I imported. I scripted the boat to experience an upward force when in water so that it can seemingly float – this works in conjunction with the rope since the rope can be picked up by the player and they are able to pull the boat across the water using the same method as the rocks – only that when the rope is picked up, the part held by the player becomes kinematic so that it stays in front of the player.

I also created a script that allowed the boat's velocity to be added to the Character Controller's movement when the player is inside the boat – I did this using a box collider trigger.

Particle Systems

Finally for the particle system, I imported a Rain Asset which produced a rain particle system that impacts with the ground. I decided to make the rain only happen occasionally at random intervals, so I created a script that turned the asset on and off at random. As well as this, I configured the Volumetric Clouds in the script so that the sky became overcast at the same time the rain would start. I also made it so that there would be at least 20 seconds of sun/rain at one time.

In my first demonstration I created fireflies which used Unity's particle system, I also made each particle its own point light that could shine on the ground, I also created a script so that they would only appear during night hours.



References

https://assetstore.unity.com/packages/3d/environments/landscapes/rock-pack-210536

https://assetstore.unity.com/packages/3d/vehicles/sea/wood-boat-124043

https://assetstore.unity.com/packages/vfx/particles/environment/rain-maker-2d-and-3d-rain-particle-system-for-unity-34938

https://cubebrush.co/hyadis-mandra/products/bczf2g/rope-tileable-pbr-material