Netcode Boat Game

Christopher Noland

Goal

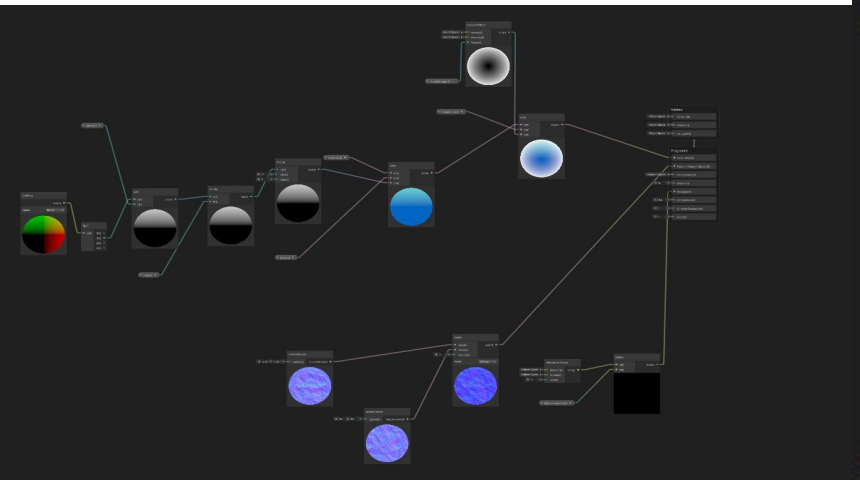
- Create an interactive water mesh
  - Using MeshRenderer
- Allow for player boat to be buoyant
- Point Collection for player with a synchronized timer
- Respawning buoy for point collection

Overview

- Wave Transform stays in the scene permanently
- Host and Client Oriented; 2 Player game
- Track Player scores with Network Variables
- Game Ends once timer reaches 0

Wave Mesh

- Use Methods to Generate Triangles and Vertices
- Use Shaders to add graphics to project



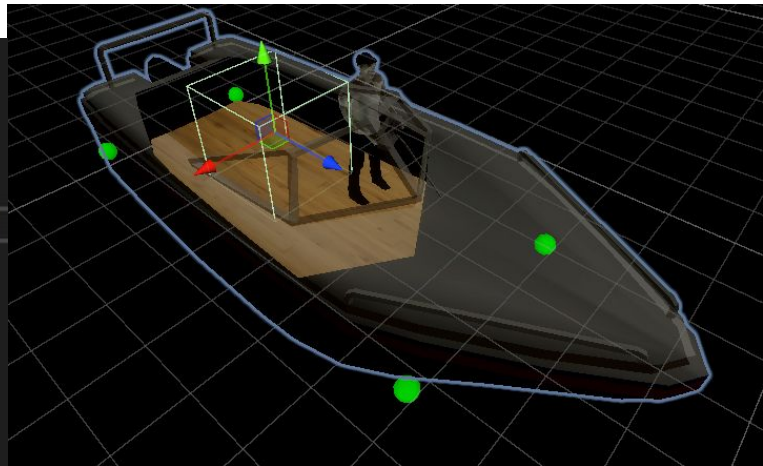
```
private int[] GenerateTries()
{
    var tries = new int[Mesh.vertices.Length * 6];

    for (int x = 0; x < Dimensions; x++)
    {
        for (int z = 0; z < Dimensions; z++)
        {
            tries[index(x, z) * 6 + 0] = index(x, z);
            tries[index(x, z) * 6 + 1] = index(x + 1, z + 1);
            tries[index(x, z) * 6 + 2] = index(x + 1, z);
            tries[index(x, z) * 6 + 3] = index(x, z);
            tries[index(x, z) * 6 + 4] = index(x, z + 1);
            tries[index(x, z) * 6 + 5] = index(x + 1, z + 1);
        }
    }

    return tries;
}
```

```
private Vector3[] GenerateVerts()
{
    var verts = new Vector3[(Dimensions + 1) * (Dimensions + 1)];

    for (int x = 0; x <= Dimensions; x++)
    {
        for (int z = 0; z <= Dimensions; z++)
        {
            verts[index(x, z)] = new Vector3(x, 0, z);
        }
    }

    return verts;
}
```

Buoyancy

- Use a WaterFloat Script to Calculate the floatation
- Uses points attached to the player

```
for (int i = 0; i < FloatPoints.Length; i++)
{

    WaterLinePoints[i] = FloatPoints[i].position;
    WaterLinePoints[i].y = Waves.GetHeight(FloatPoints[i].position);
    newWaterLine += WaterLinePoints[i].y / FloatPoints.Length;
    if (WaterLinePoints[i].y > FloatPoints[i].position.y)
    {
        pointUnderWater = true;
    }
}
```



Player

- On Host or Client, player is spawned along with the player prefab
- Uses Netcode for Gameobjects to assign ownership

```
//Rotational Force
Rigidbody.AddForceAtPosition(steer * transform.right * SteerPower / 100f, Motor.position);

//compute vectors
var forward = Vector3.Scale(new Vector3(1, 0, 1), transform.forward);
var targetVel = Vector3.zero;

//forward/backward poewr
if (Input.GetKey(KeyCode.W))
    PhysicsHelper.ApplyForceToReachVelocity(Rigidbody, forward * MaxSpeed, Power);
if (Input.GetKey(KeyCode.S))
    PhysicsHelper.ApplyForceToReachVelocity(Rigidbody, forward * -MaxSpeed, Power);

//Motor Animation // Particle system
Motor.SetPositionAndRotation(Motor.position, transform.rotation * StartRotation * Quaternion.Euler(0, 30f * steer, 0));
if (ParticleSystem != null)
{
    if (Input.GetKey(KeyCode.W) || Input.GetKey(KeyCode.S))
        ParticleSystem.Play();
    else
        ParticleSystem.Pause();
}

//moving forward
var movingForward = Vector3.Cross(transform.forward, Rigidbody.velocity).y < 0;

//move in direction
Rigidbody.velocity = Quaternion.AngleAxis(Vector3.SignedAngle(Rigidbody.velocity, (movingForward ? 1f : 0f) * transform.forward, Vector3.up) * Drag, Vector3.up) * Rigidbody.velocity;
```

Points & Timer

- Use Network Variables for Scores and Timers
- Timer Begins on Network Spawn

```
Unity Script (2 asset references) | 8 references
public class ScoreManager : NetworkBehaviour
{
    public Text scorePlayer1;
    public Text scorePlayer2;

    public static float scoreCount;

    public static NetworkVariable<float> scoreTest1 = new NetworkVariable<float>(0);
    public static NetworkVariable<int> scoreTest2 = new NetworkVariable<int>(0);


    Unity Message | 0 references
    private void Update()
    {
        scorePlayer1.text = "Player 1 Score: " + Mathf.Round(scoreTest1.Value);
        scorePlayer2.text = "Player 2 Score: " + Mathf.Round(scoreTest2.Value);
    }
}
```

```
public static NetworkVariable<float> remainTime = new NetworkVariable<float>(30);

public static bool isTimer = false;


Unity Message | 0 references
private void Update()
{



    int minutes = Mathf.FloorToInt( remainTime.Value / 60 );
    int seconds = Mathf.FloorToInt( remainTime.Value % 60 );

    timerText.text = remainTime.Value.ToString();//string.Format("{0:00}:{1:00}", minutes, seconds);

    if(remainTime.Value == 0)
    {
        if(ScoreManager.scoreTest1.Value > ScoreManager.scoreTest2.Value )
        {
            GameOver.text = "Player 1 Wins!";
        }
        else if(ScoreManager.scoreTest2.Value > ScoreManager.scoreTest1.Value )
        {
            GameOver.text = "Player 2 Wins";
        }
    }
}
```

End

- When Timer reaches 0 the game “ends”
- Declares winner
- On key press the Server is shutdown

Problems

- Water Mesh Synchronization
- Buoy Spawning
- Score Keeping
- Timer Sync

Demo