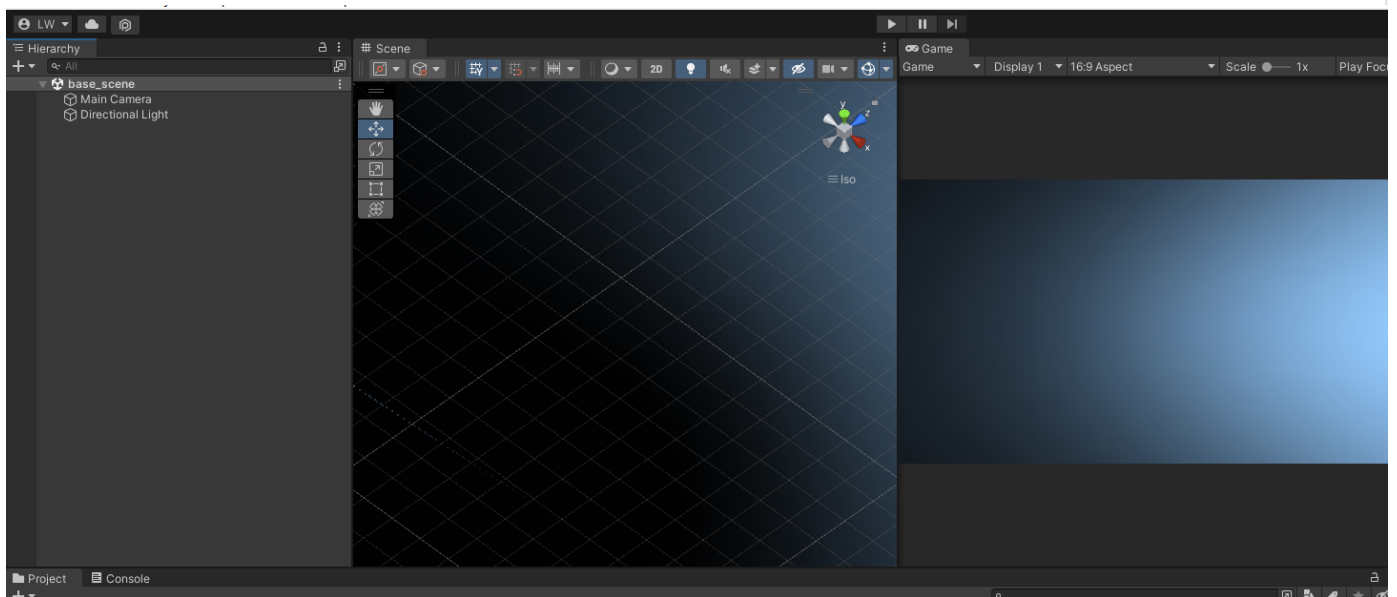


Creature with hinge joints

Step 01 - download unity package

1. Create a new **3D Unity project 2021.3.15f**
2. If you want to follow along, download the base scene [unitypackage](#)
3. Import the unity package
4. open base scene



Step 02 - Creature Script

1. create a new script in the Folder 'Scripts'
2. Call it "RandomSnake"
3. copy the following code into in

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

public class RandomSnake : MonoBehaviour
{
```

```
public GameObject headPrefab;
public GameObject segmentPrefab;
public int segmentCount;
public float speed;
public float damper;
public float spring;
public float changeDirectionTime = 1.0f;

private List<GameObject> segments = new List<GameObject>();
private GameObject head;
private Vector3 randomDirection;
private float timeSinceLastDirectionChange;

void Start()
{
    // Create head and add it to the segments list
    head = Instantiate(headPrefab, transform);
    segments.Add(head);

    // Create the segments and connect them with hinge joints
    for (int i = 0; i < segmentCount; i++)
    {
        GameObject segment = Instantiate(segmentPrefab, transform);
        HingeJoint hingeJoint = segment.AddComponent<HingeJoint>();
        hingeJoint.connectedBody = segments[i].GetComponent<Rigidbody>();
        hingeJoint.autoConfigureConnectedAnchor = false;
        hingeJoint.anchor = new Vector3(0, 0.5f, 0);
        hingeJoint.connectedAnchor = new Vector3(0, -0.5f, 0);
        hingeJoint.axis = new Vector3(1, 0, 0);

        JointSpring jointSpring = hingeJoint.spring;
        jointSpring.damper = damper;
        jointSpring.spring = spring;
        hingeJoint.spring = jointSpring;

        hingeJoint.useSpring = true;
        segments.Add(segment);
    }
}
```

```

// Initialize random direction and timer
randomDirection = Random.insideUnitSphere.normalized;
timeSinceLastDirectionChange = 0;
}

void Update()
{
    // Move the head in a random direction
    head.GetComponent<Rigidbody>().velocity = randomDirection * speed;

    // Update timer and change direction if needed
    timeSinceLastDirectionChange += Time.deltaTime;
    if (timeSinceLastDirectionChange >= changeDirectionTime)
    {
        randomDirection = Random.insideUnitSphere.normalized;
        timeSinceLastDirectionChange = 0;
    }
}
}

```

Step 03 - Prefabs

1. Create a prefab for the head
2. Create a prefab for the body segments
3. add a rigidbody to both the head and the body prefabs

"use gravity" and "use kinematic" must be disabled for the rigidbody componens!

Step 04 - Creature Prefab

1. Add an empty GameObject to the scene at **0,0,0 (Camera should be at 0,0,-10)**
2. Rename it "Creature"
3. Add the **"RandomSnake"** Script to the Creature GameObject in your scene
4. In the Inspector assign the "head" and "segment" prefabs respectively

5. insert a segment count and values for speed, damper, and direction change time
6. Try out which values work best
7. When good values are determined, drag the Creature GameObject into Assets/Prefabs to make it a prefab

explanation for hinge joint settings

In the given script, the `spring` parameter affects the **stiffness of the spring** component within the `HingeJoint` that connects the segments of the creature. The spring force tries to maintain the joint angle at a target angle, which is initially set to zero. When the joint angle deviates from the target angle due to the movement of the connected bodies, the spring force acts to bring the joint angle back to the target angle.

The higher the value of the `spring` parameter, the stiffer the spring will be, and the more force will be exerted to maintain the joint angle at the target angle.

A higher `spring` value will cause the segments to maintain their relative positions more tightly, leading to a more rigid movement. On the other hand, a lower `spring` value will make the segments more loosely connected, allowing them to deviate further from their original positions relative to each other, leading to a more flexible movement.

Step 05 - Spawn Location

1. Create a new empty GameObject and rename it to SpawnLocation
2. Assign the [SimpleSpawn](#) script
3. Adjust the SimpleSpawn script to add lifetime and fade the material at it's end
4. [here](#) is some info on how you can add lifetime to the creatures

TASK:

Create 3D Models for the head and segment prefabs with a 3D software of your choice. Import them, set them up as prefabs and assign them in the Inspector.

Alternatively you can also apply your own design of a creature by setting up and importing a rigged animation, like in the example with the Fish from last week, or create an entirely different script with different behavior.

PROJECT FILES

[base unitypackage \(empty\)](#)

[unitypackage with all scripts and assets](#)

[unity project as *.zip](#)

Revision #8

Created 18 April 2023 02:21:40 by Laura Wagner

Updated 22 January 2024 12:40:47 by Laura Wagner