using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using Cinemachine;

using Unity.Netcode;

public class PlayerMovementRB : NetworkBehaviour

{

public Transform personalCamera;

public override void OnNetworkSpawn()

{

CinemachineVirtualCamera vcam = personalCamera.gameObject.GetComponent<CinemachineVirtualCamera>();

if (IsOwner)

{

vcam.Priority = 1;

}

else

{

vcam.Priority = 0;

}

}

[Header("Movement")]

public float playerSpeed;

public float jumpHeight;

public float gravity;

[Header("Lanes")]

[SerializeField] Transform personalCamera;

[SerializeField] float laneDistance = 4; //distancia entre duas lanes

private int desiredLane = 1; //0:esquerda 1:meio 2:direita

private float desiredHeight = 1;

Rigidbody rb;

void Start()

{

rb = GetComponent<Rigidbody>();

}

// Update is called once per frame

void Update()

{

ChangeLane();

}

void ChangeLane()

{

if (Input.GetKeyDown(KeyCode.LeftArrow))

{

desiredLane = Mathf.Max(desiredLane - 1, -1);

}

else if (Input.GetKeyDown(KeyCode.RightArrow))

{

desiredLane = Mathf.Min(desiredLane + 1, 3);

}

if (Input.GetKey(KeyCode.UpArrow) && (desiredLane == -1 || desiredLane == 3))

{

desiredHeight += Time.deltaTime \* 2f;

}

if (Input.GetKey(KeyCode.DownArrow) && (desiredLane == -1 || desiredLane == 3))

{

desiredHeight -= Time.deltaTime \* 2f;

}

desiredHeight = Mathf.Clamp(desiredHeight, 2f, 5f);

if (desiredLane >= 0 && desiredLane <= 2)

{

desiredHeight = 2f;

Vector3 targetPosition = new Vector3((desiredLane - 1) \* laneDistance, 1f, transform.position.z + (playerSpeed \* Time.deltaTime));

transform.position = Vector3.Lerp(transform.position, targetPosition, 10 \* Time.deltaTime);

}

else if (desiredLane == -1)

{

Vector3 targetPosition = new Vector3(-1.25f \* laneDistance, desiredHeight, transform.position.z + (playerSpeed \* Time.deltaTime));

transform.position = Vector3.Lerp(transform.position, targetPosition, 10 \* Time.deltaTime);

}

else if (desiredLane == 3)

{

Vector3 targetPosition = new Vector3(1.25f \* laneDistance, desiredHeight, transform.position.z + (playerSpeed \* Time.deltaTime));

transform.position = Vector3.Lerp(transform.position, targetPosition, 10 \* Time.deltaTime);

}

}

}