Unity2D 青蛙过河FrogJump

编辑器版本2022.1.8f1c1

2d核心模板

场景搭建

导入资源



放置基础资产



设置俯视角渲染方式



设置各个资产的锚点sprite editor







设置各个资产的图层





使用新的输入方式安装input system





创建Input Action



创建用户控制跳长跳获取点击位置



给青蛙添加控制



查找脚本



选择启动Unity事件



新建c#脚本（脚本名和C#中类名相同）



添加到Frog上



打开脚本添加inputsystem



最简单的调试



绑定到frog上



运行执行了相应代码



添加动画





添加碰撞箱





跳跃代码

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.InputSystem;

public class PlayerController : MonoBehaviour

{

    //获得自身的组件，一般写在开头

    private Rigidbody2D rb;

    public float jumpDistance;//跳跃的距离

    private float moveDistance;//实际移动的距离

    private Vector2 destination;

    private bool buttonheld;//按键是否被长按

    private bool isjump;//跳跃途中的状态

    private void Update()

    {

        //FIXME：临时操作 跳跃到目标位置后重置跳跃状态

        if(destination.y - transform.position.y <= 0.1 )

        isjump = false;

    }

    private void Awake()

    {

        rb = GetComponent<Rigidbody2D>();

    }

    //FixedUpdate是一个稳定的函数每0.2秒执行一次

    private void FixedUpdate()

    {

        //position 是一个二维向量,希望使用lerp线性差值移动到一个地方

        //必须在跳跃结束后才能跳跃

        if(isjump)

        rb.position = Vector2.Lerp(transform.position,destination,0.134f);

    }

    public void Jump(InputAction.CallbackContext context)

    {

        if(context.phase == InputActionPhase.Performed&&isjump == false)

        {

            moveDistance = jumpDistance;

            //Debug.Log("jump!");

            destination = new Vector2(transform.position.x,transform.position.y+moveDistance);

            isjump = true;

        }

    }

    public void Longjump(InputAction.CallbackContext context)

    {

        if(context.performed&&isjump == false)

        {

            moveDistance = 2\*jumpDistance;

            buttonheld = true;

        }

        if(context.canceled && buttonheld == true && isjump == false)

        {

            //Debug.Log("Longjump!");

            buttonheld = false;

            destination = new Vector2(transform.position.x,transform.position.y+moveDistance);

            isjump = true;

        }

    }

    public void Gettouchposition(InputAction.CallbackContext context)

    {

        if(context.performed)

        {

            Debug.Log("Longjump!");

        }

    }

}