

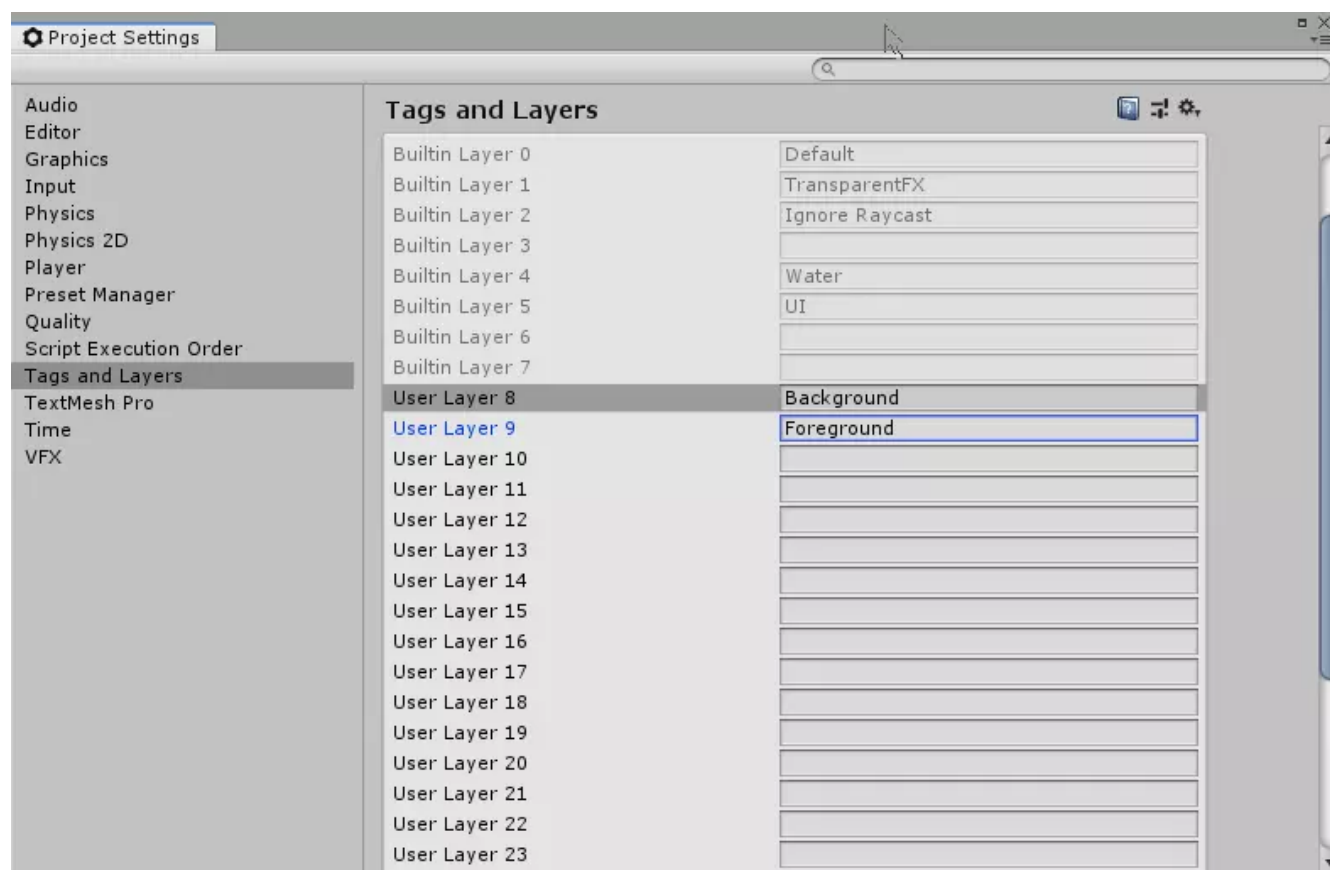
基本 2D 游戏场景

张俊华 16030199025

创建2D项目

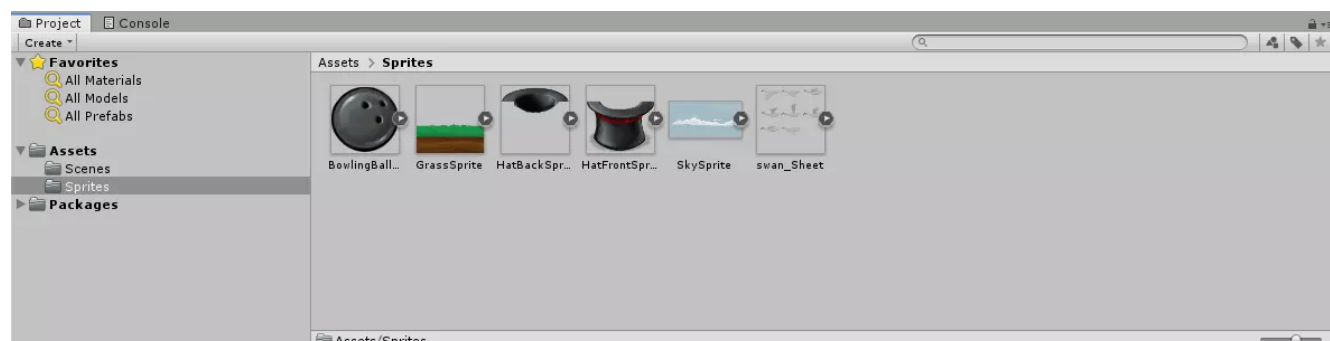
创建工作层

创建工作层来保证游戏在场景中的层次不会错乱



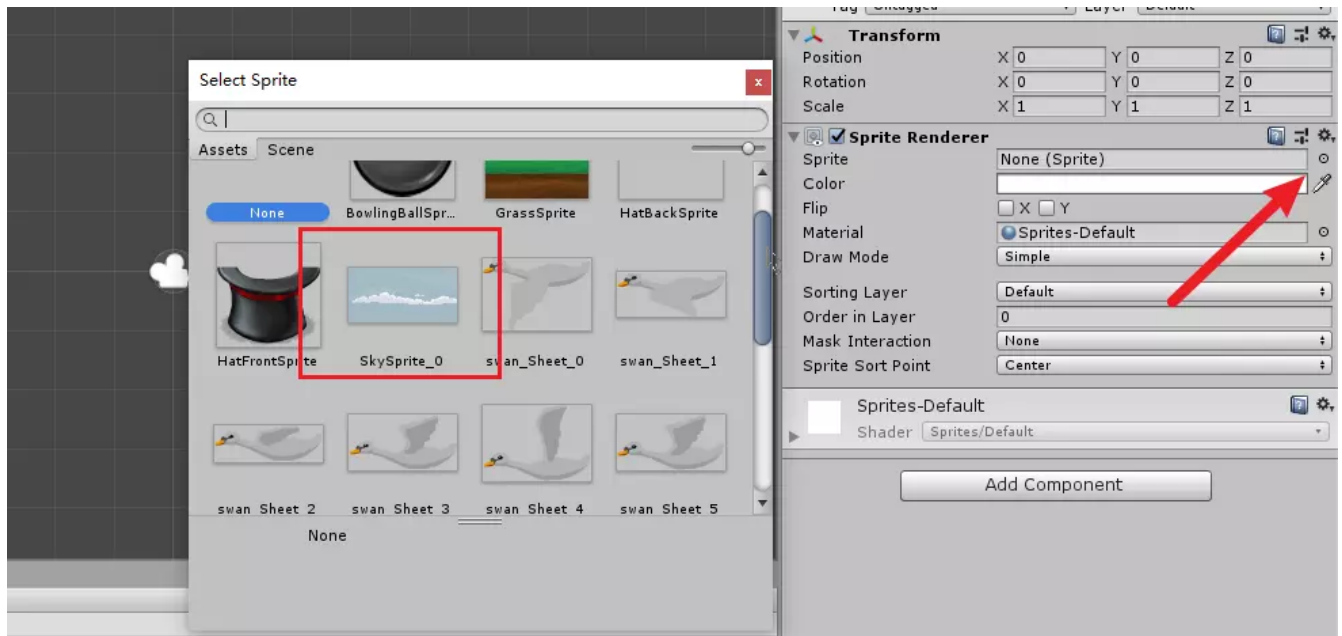
添加静态景物

导入静态图片

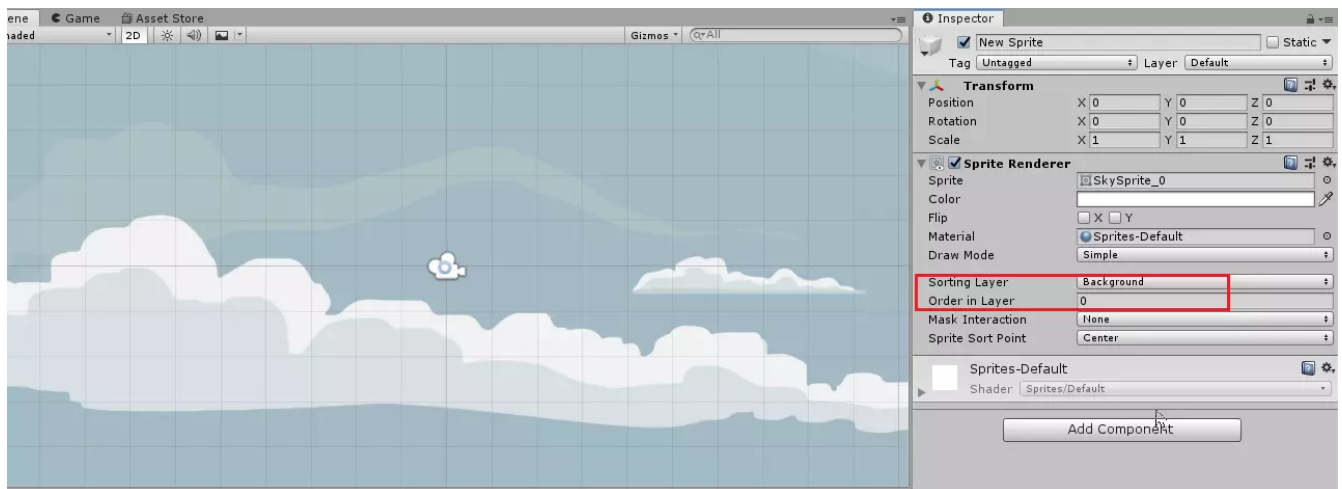


创建静态景物

创建天空

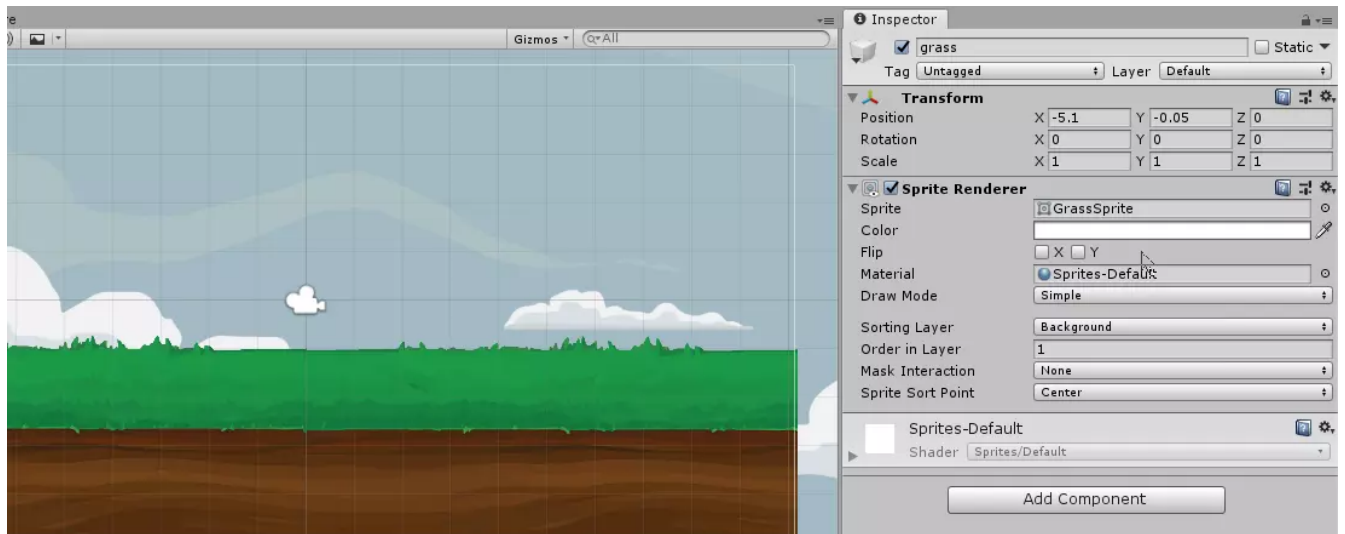


排序图层Sorting Layer设置为Background 图层顺序Order in Layer设置为0



创建草地

设置 grass 的 Sprite Renderer 组件属性:



复制多个 grass，调整摄像机。保证 grass 出现在整个屏幕底部



创建一个空对象用于管理游戏背景

重命名为background

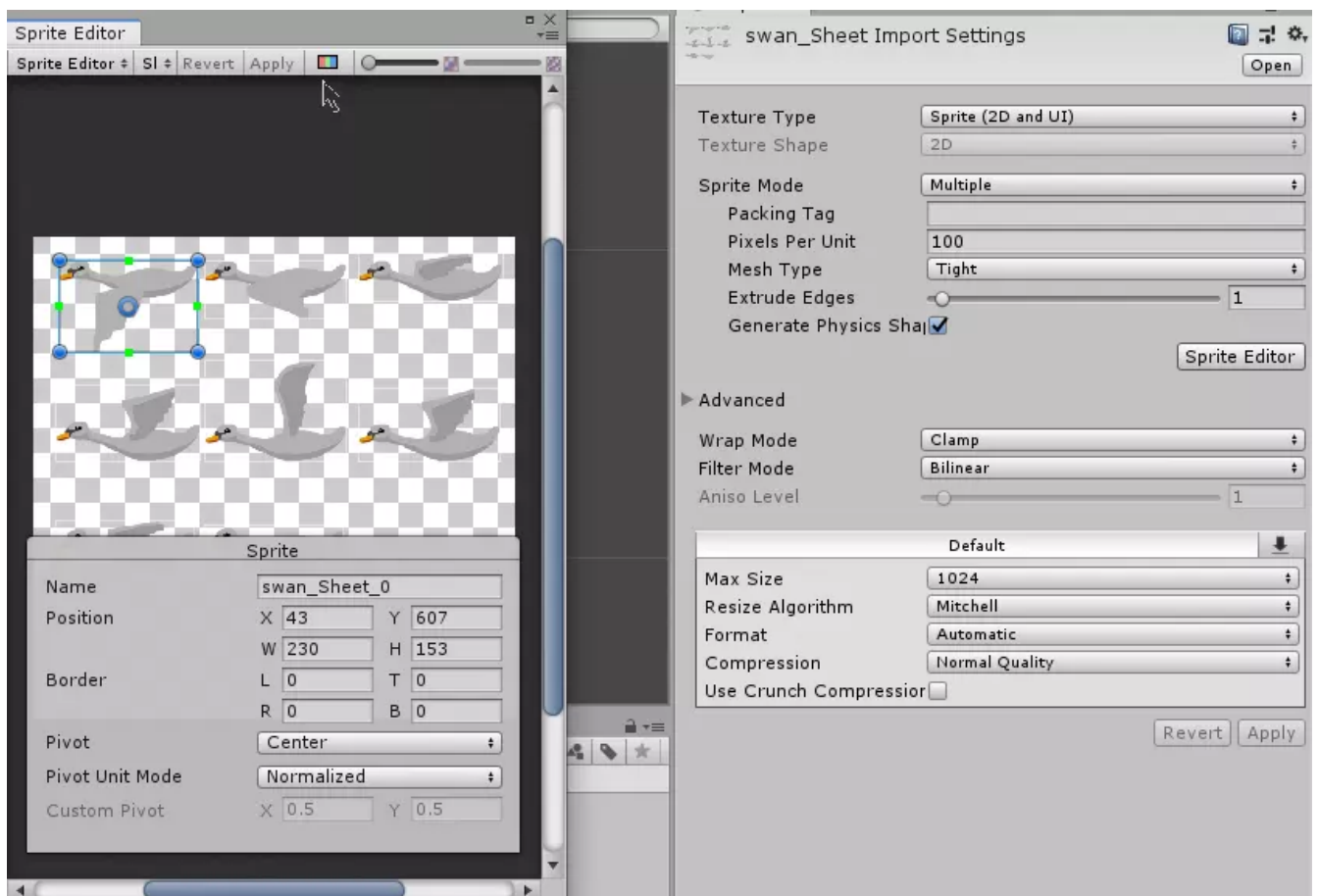
将多个grass对象和sky对象拖到background中



添加角色和控制

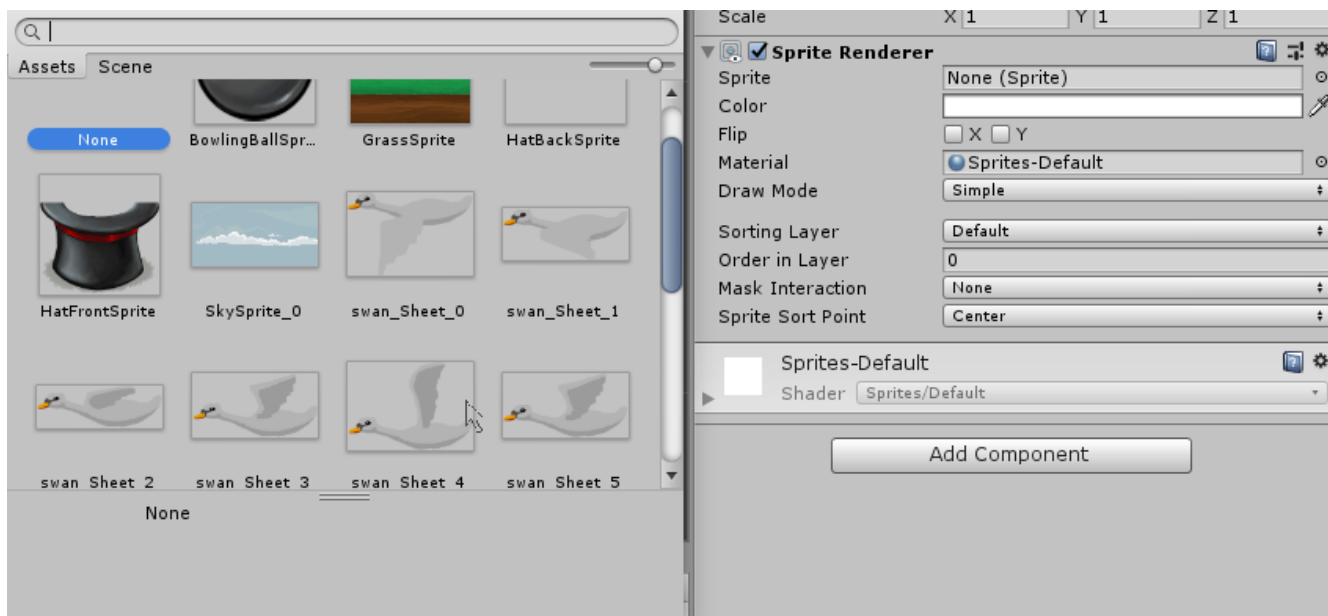
创建 Sprite 动画

切割图片

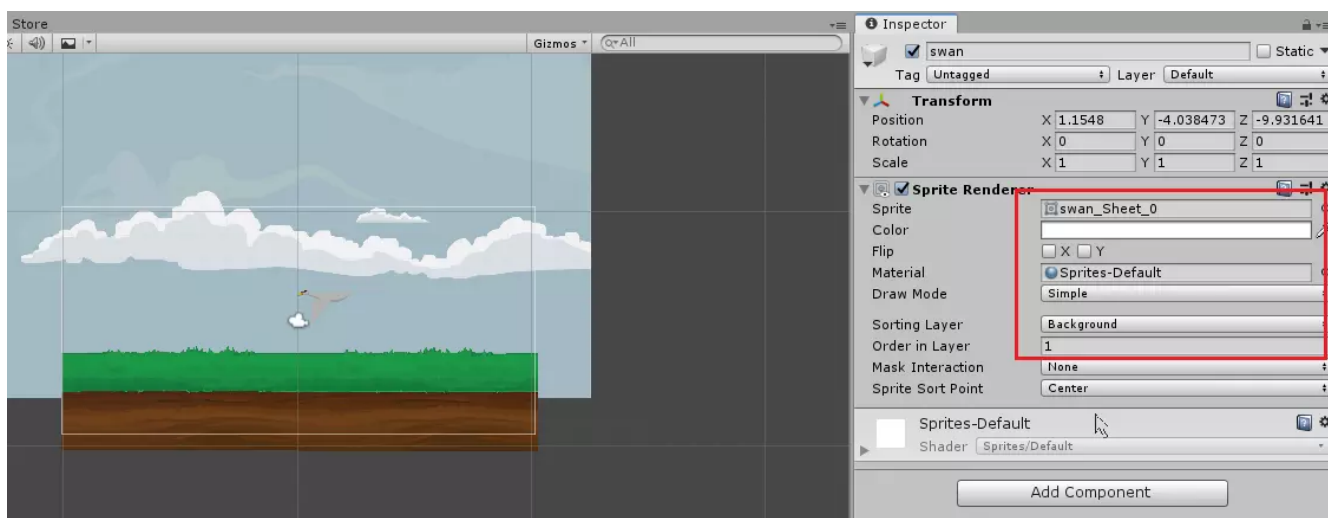


制作天鹅飞行动画

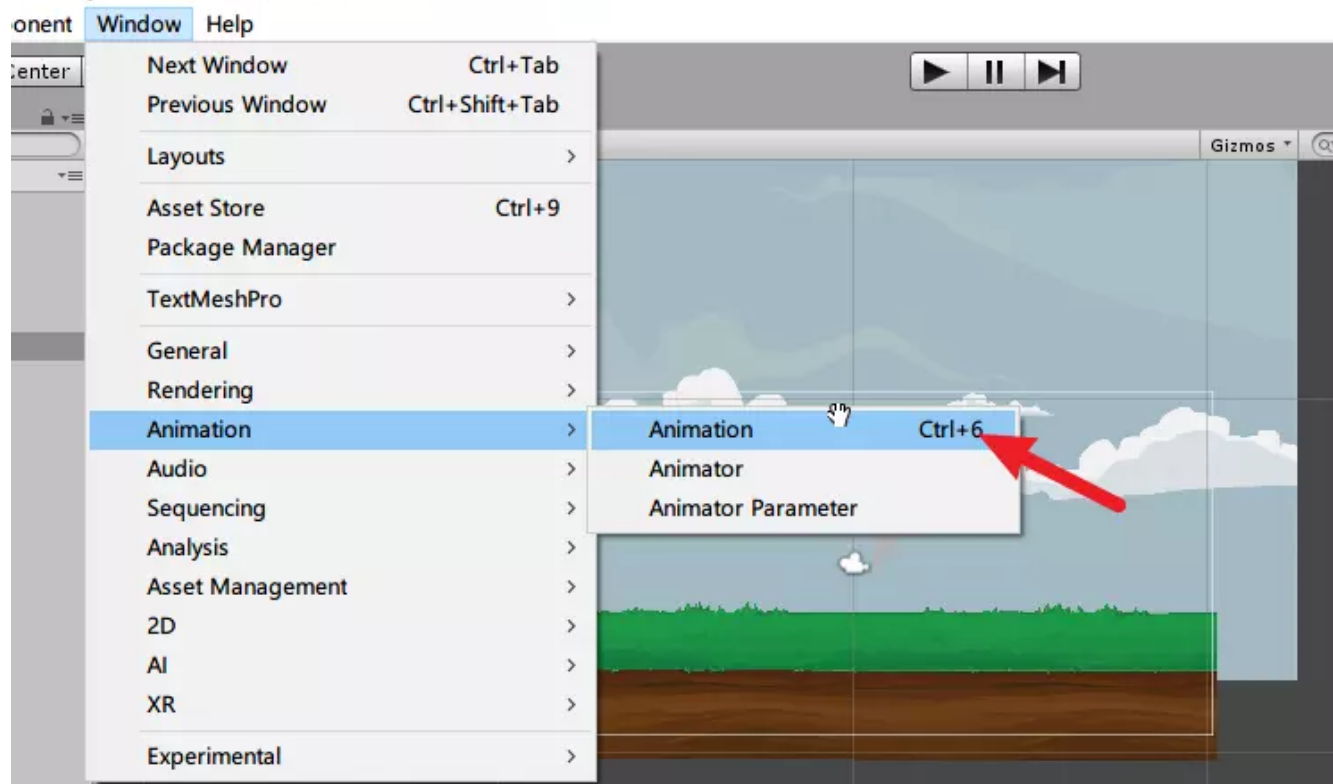
创建 swan Sprite



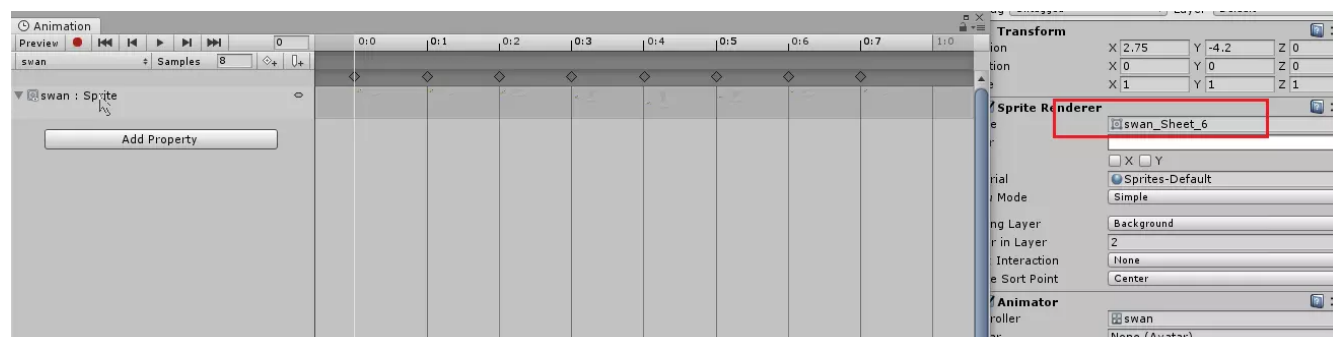
设置 Sprite 图片和 Sorting Layer



创建煽动翅膀的动画

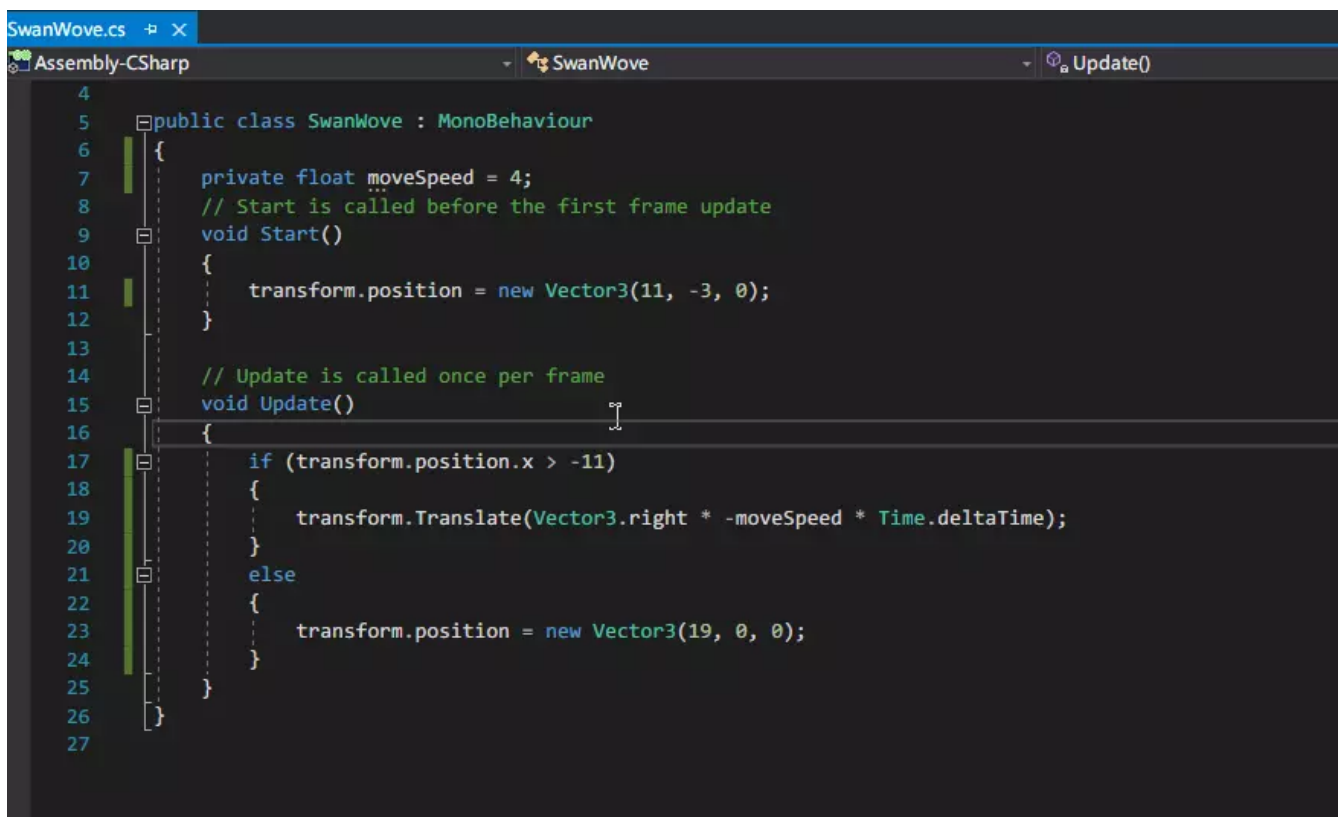


在检视视图中逐帧设置 Sprite，（0到7），并按右键 add key





创建脚本实现飞行效果

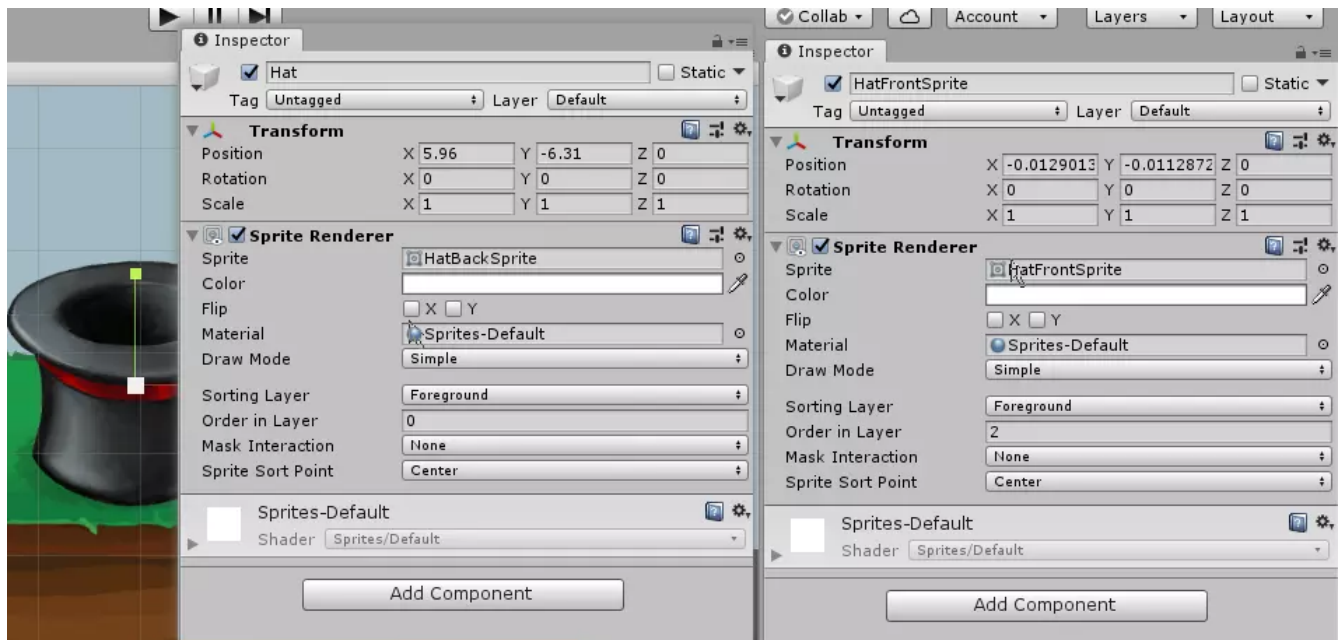


添加脚本到游戏对象 swan 上：

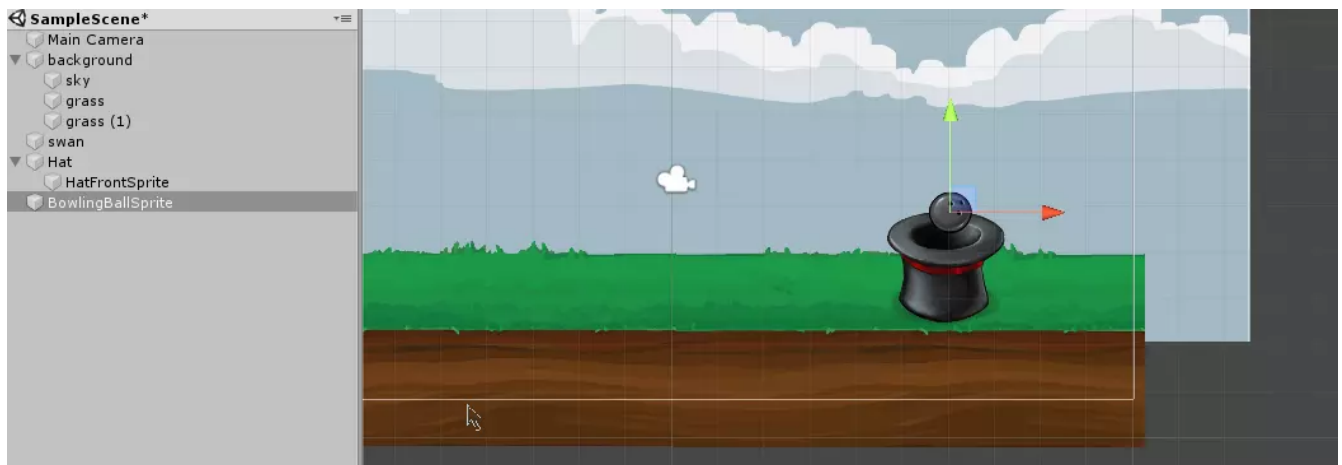
创建主要游戏对象

帽子

调整图层和图层顺序：

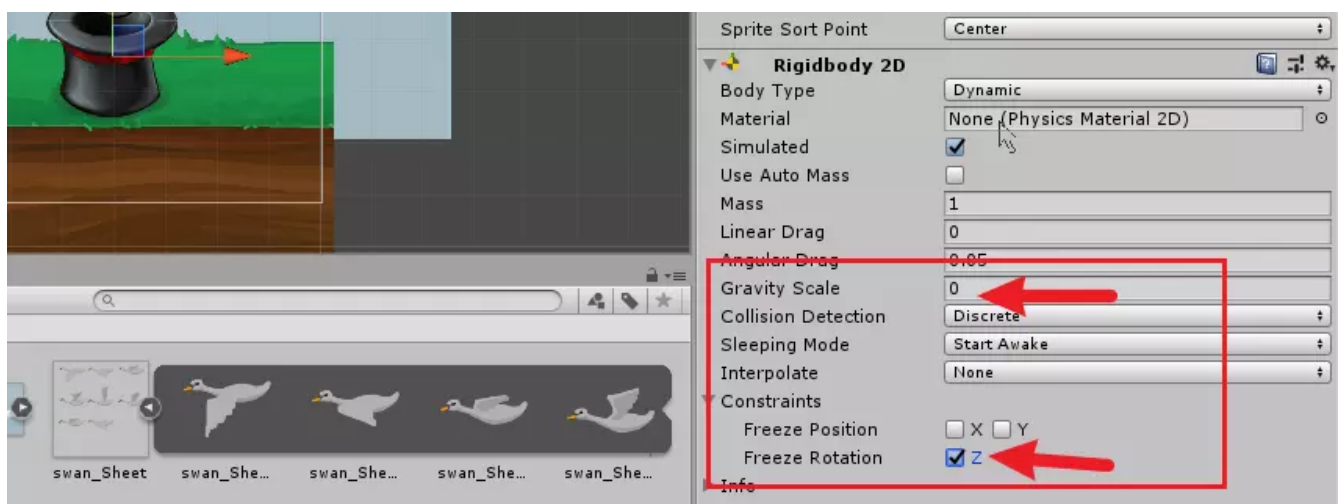


拼凑并调整帽子大小

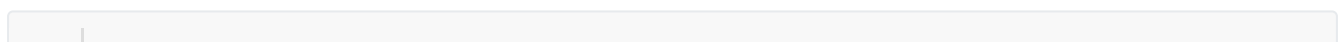


创建2D刚体组件

重力值gravity scale设为0（防止帽子下落）并且冻结旋转（固定帽子的呈现角度）



创建控制帽子移动脚本



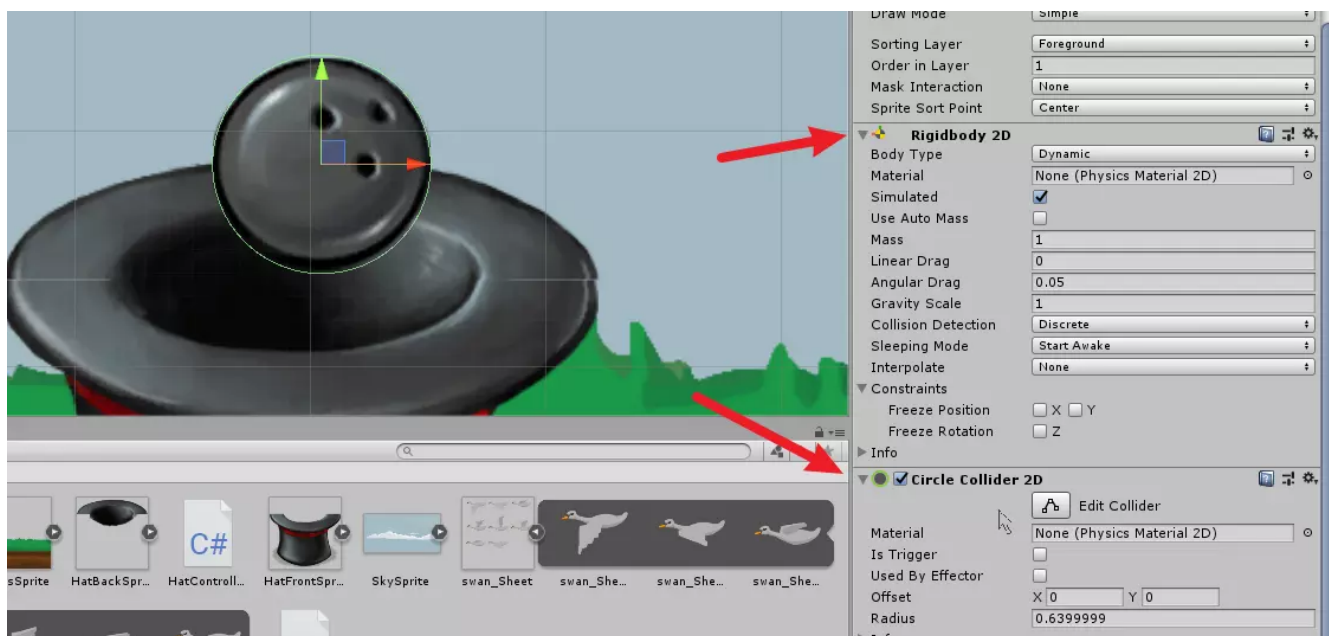

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class HatController : MonoBehaviour
6  {
7      private Vector3 rawPosition;
8      private Vector3 hatPosition;
9      private float maxWidth;
10
11     // Start is called before the first frame update
12     void Start()
13     {
14         Vector3 screenPos = new Vector3(Screen.width, 0, 0);
15         Vector3 movewidth = Camera.main.ScreenToWorldPoint(screenPos);
16         //计算帽子的宽度
17         float hatwidth = GetComponent<Renderer>().bounds.extents.x;
18         // 获得帽子的初始位置
19         hatPosition = transform.position;
20         //计算帽子的移动宽度
21         maxWidth = movewidth.x - hatwidth;
22     }
23
24     // Update is called once per frame
25     void FixedUpdate()
26     {
27         //将鼠标的屏幕位置转化为世界坐标
28         rawPosition = Camera.main.ScreenToWorldPoint(Input.mousePosition);
29         hatPosition = new Vector3(rawPosition.x, hatPosition.y, 0);
30         //设置帽子将要移动的位置,帽子移动范围:hatPosition.x在maxwidth与-maxwidth之间
31         hatPosition.x = Mathf.Clamp(hatPosition.x, -maxwidth, maxwidth);
32         //帽子移动
33         GetComponent<Rigidbody2D>().MovePosition(hatPosition);
34     }
35 }

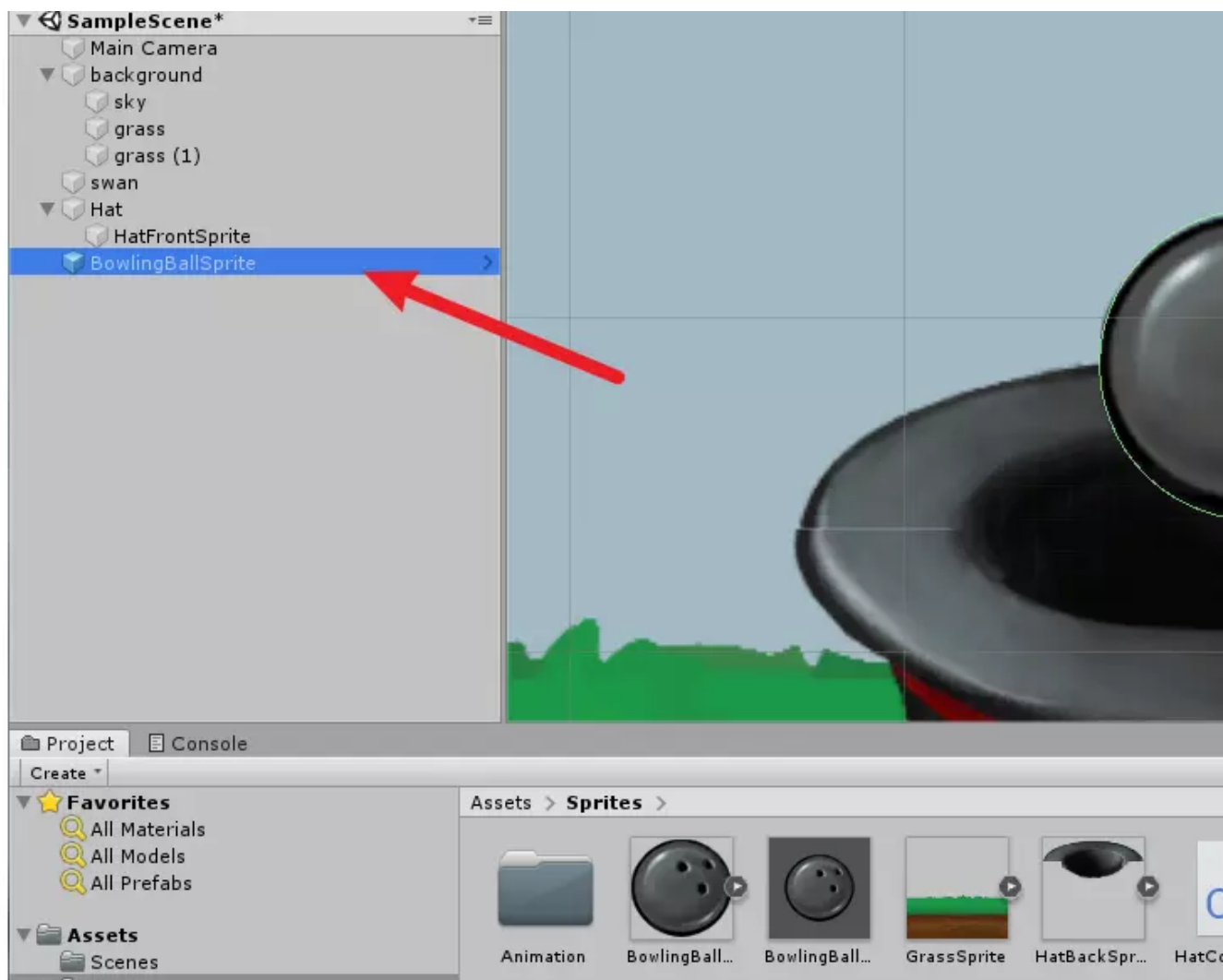
```

保龄球

添加刚体和 2D 圆形碰撞体组件：



拖动到项目视图中，使得保龄球成为预制体



创建新空对象来控制实例化的保龄球高度，创建保龄球实例化脚本

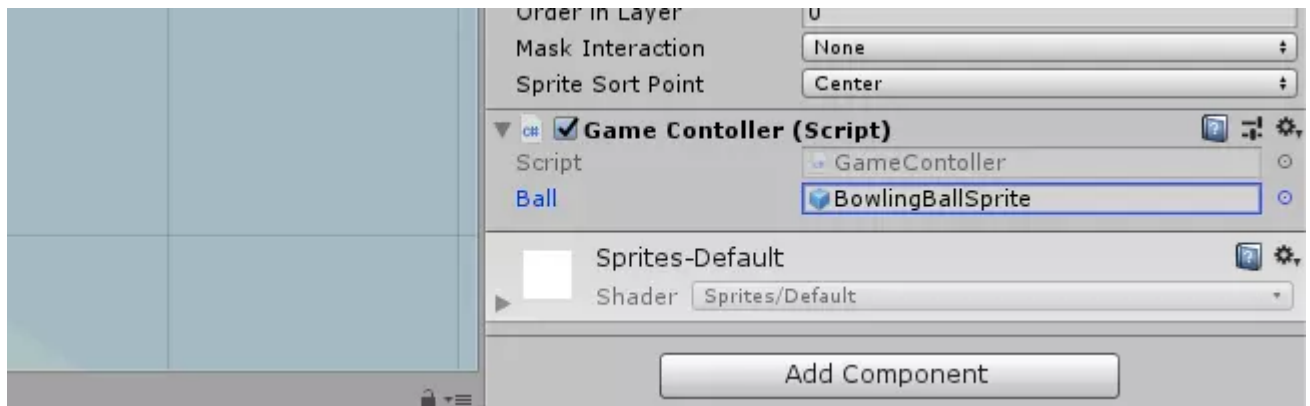
```
1 using System.Collections;
```

```

2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class GameController : MonoBehaviour
6  {
7      //Use this for initialization
8      public GameObject ball; //公共属性可以直接在unity的检视视图中设置
9      private float maxWidth;
10     private float time = 2;
11     private GameObject newball;
12     void Start()
13     {
14         //将屏幕的宽度转化为世界坐标
15         Vector3 screenPos = new Vector3(Screen.width, 0, 0); Vector3 movewidth =
Camera.main.ScreenToWorldPoint(screenPos);
16         //获取保龄球自身的宽度
17         float ballwidth = ball.GetComponent<Renderer>().bounds.extents.x;
18         //计算保龄球实例化位置的最大宽度
19         maxWidth = movewidth.x - ballwidth;
20     }
21
22
23     // Update is called once per frame
24     void FixedUpdate()
25     {
26         time = time - Time.deltaTime;
27         if (time < 0)
28         {
29             // 产生一个随机数,代表实例化下一个保龄球所需的时间
30             time = Random.Range(1.5f, 2.0f);
31             //在保龄球实例化位置的宽度内产生一个随机数,来控制实例化的保龄球的位置
32             float posX = Random.Range(-maxWidth, maxWidth);
33             Vector3 spawnPosition = new Vector3(posX, transform.position.y, 0);
34             //实例化保龄球
35             newball = (GameObject)Instantiate(ball, spawnPosition,
Quaternion.identity);
36             //10秒后销毁
37             Destroy(newball, 10);
38         }
39     }
40 }
41 }
42

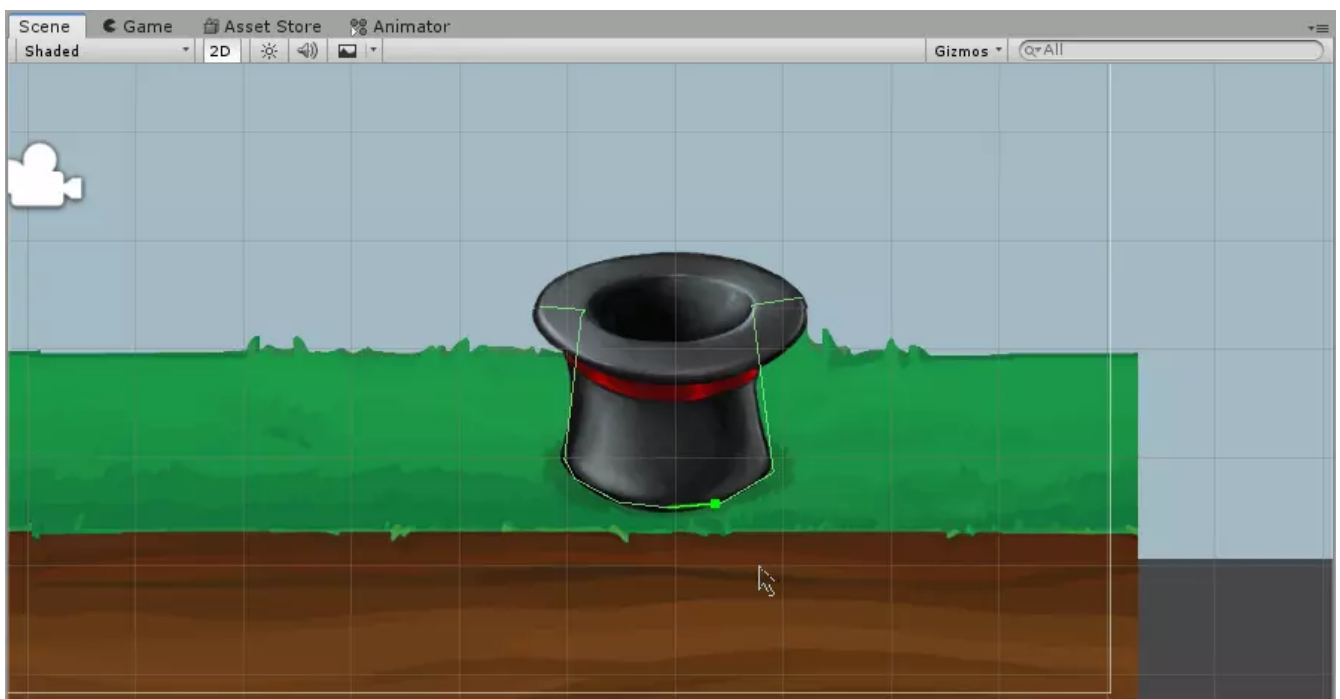
```

添加脚本到 GameObject 中, 并选择 GameObject 组件

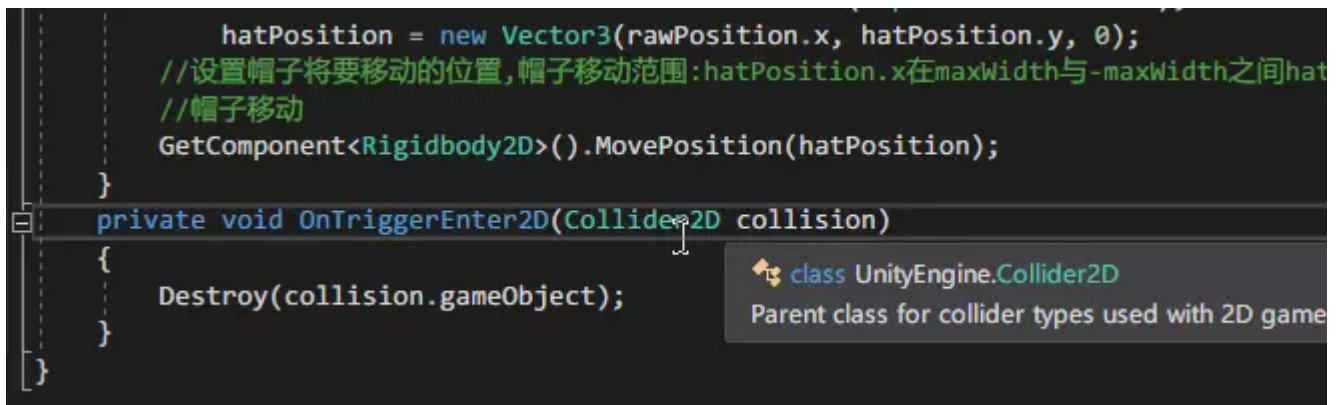
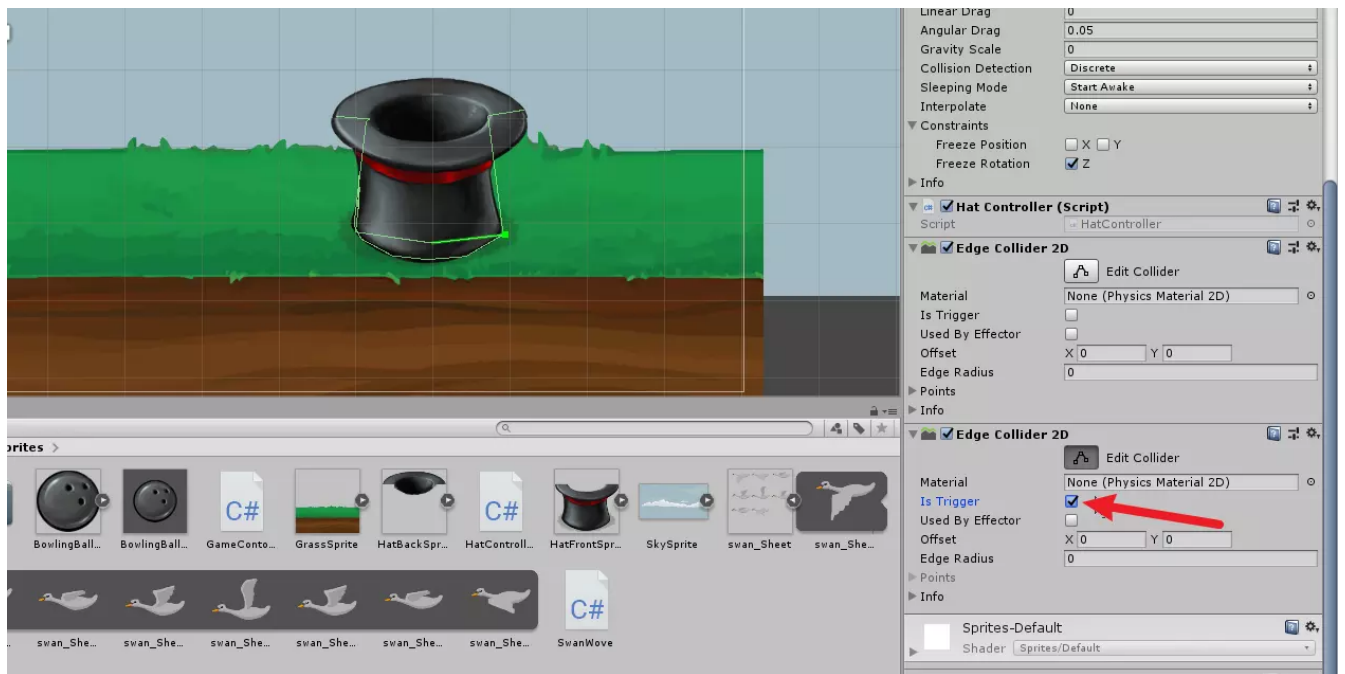


创建 2D 物理阻挡

为帽子创建 2D 边缘碰撞体



创建触发器



创建 2D 盒状碰撞体



最终结果

