

第八次实验： 动画

学号：22920212204392 姓名：黄勛

一、 实验目的

- 掌握动画制作方法

二、 实验条件

- 系统环境：Windows 10 21H2
- 软件环境：Unity 3D 2021.3.14f1c1

三、 实验内容

- 1.选择并加入合适的动画
- 2.编写脚本
- 3.实现用键盘控制角色移动和动画切换
 - WSAD 分别控制前后左右移动
 - 空格让角色跳跃
 - U 键攻击

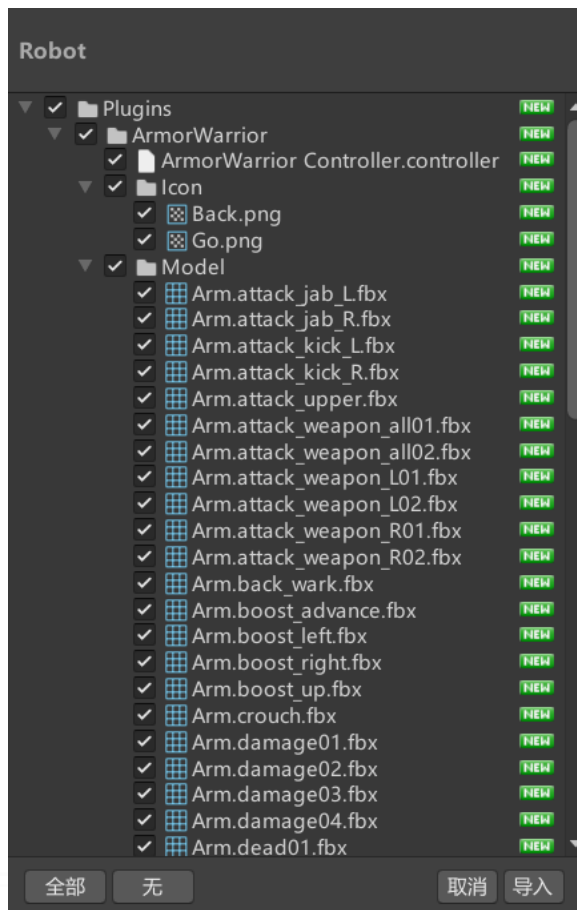
四、 实验项目步骤：

为了实现实验内容，主要的工作是创建 3D 模板学习动画制作，主要内容与注释如下：

下：

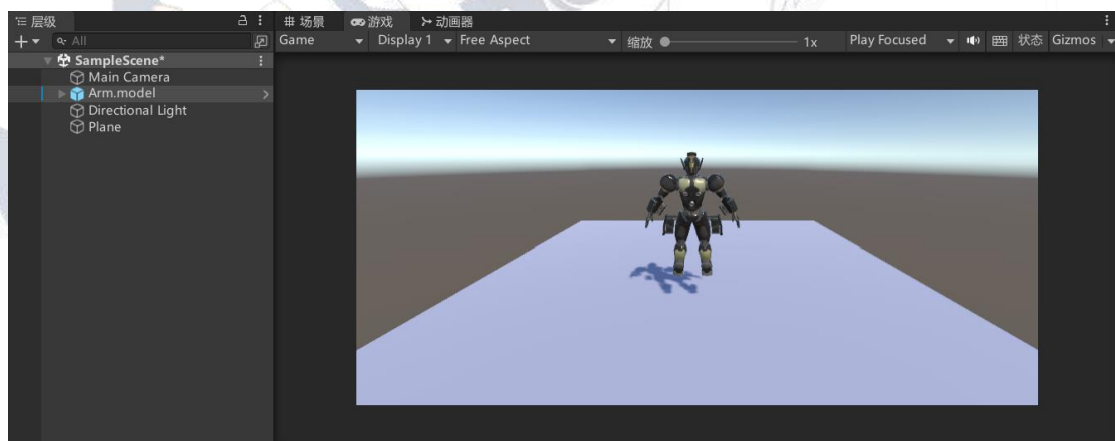
（一）新建一个工程，通过 unity editor 导入实验需要的 Robot 角色资源。

Assets > Import Package > Custom Package > 选中 Robot.unitypackage



在路径 Assets/Plugins/ArmorWarrior/Prefabs 下找到 Arm.model 的预设体

鼠标左键按住预设体，拖动到场景中，并添加一个平面



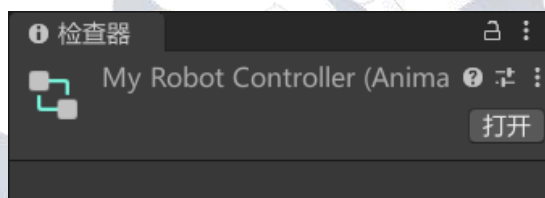
给机器人添加刚体和碰撞体，使得机器人可以站在平面上并且会正常跳跃和掉落



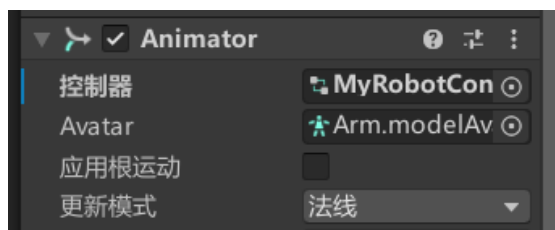
(二) 设置机器人的 Animation Controller

创建 Robot 动画控制器

Project->Create->Animator Controller



替换机器人 Animator 组件中原有的 Controller

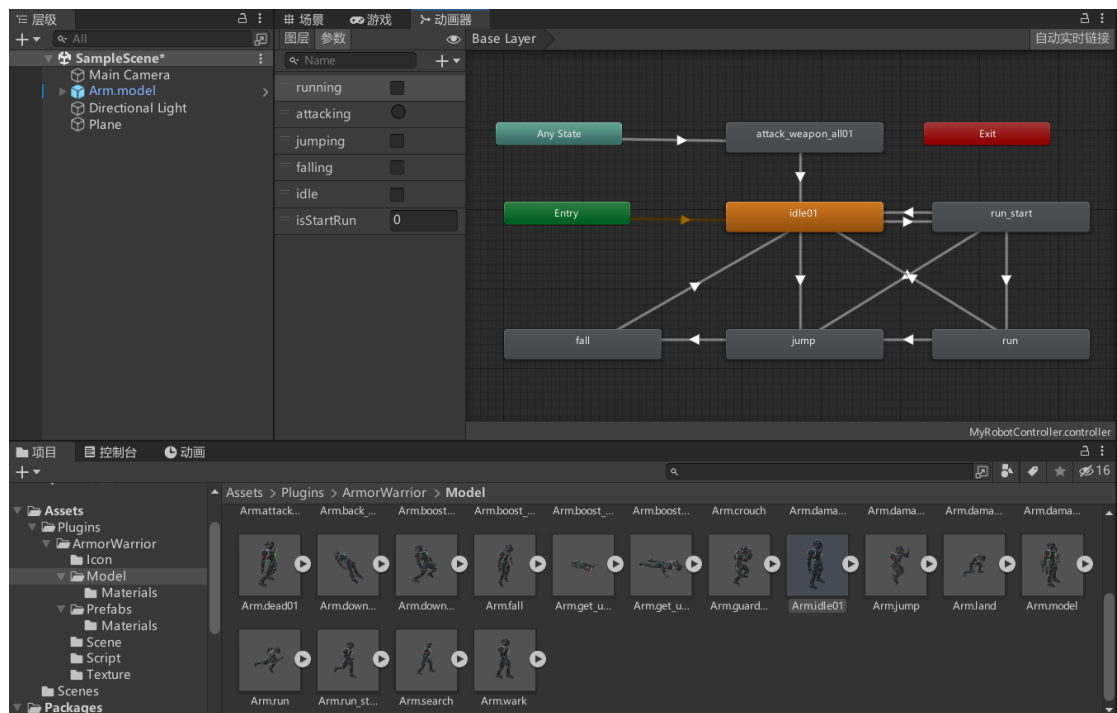


编辑 Robot 动画控制器

在路径 Assets/Plugins/ArmorWarrior/Model 下找到 Animation Clip:

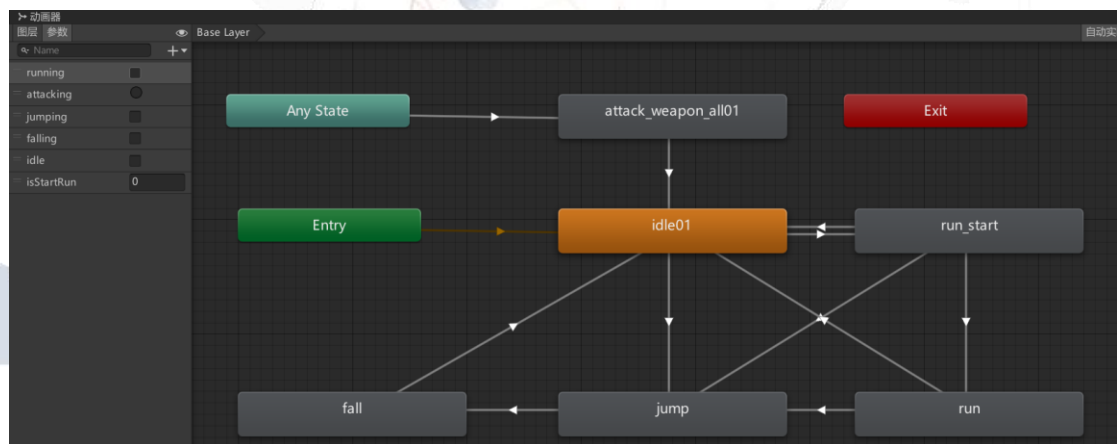
Idle01, 将 Idle 动画用鼠标拖到控制器中, Entry 会自动连接一条橙线到动画, 其他

动画操作以此类推。



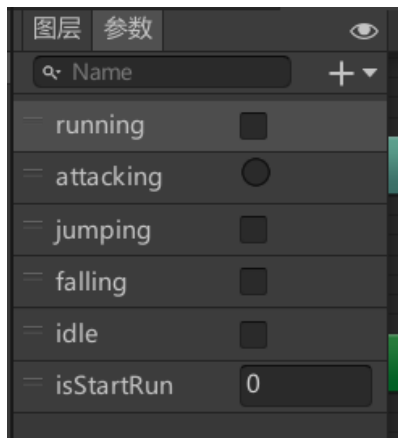
(三) 设置机器人基础状态机

添加剩余动画片段，将它们彼此进行合理的链接，以方便跳转

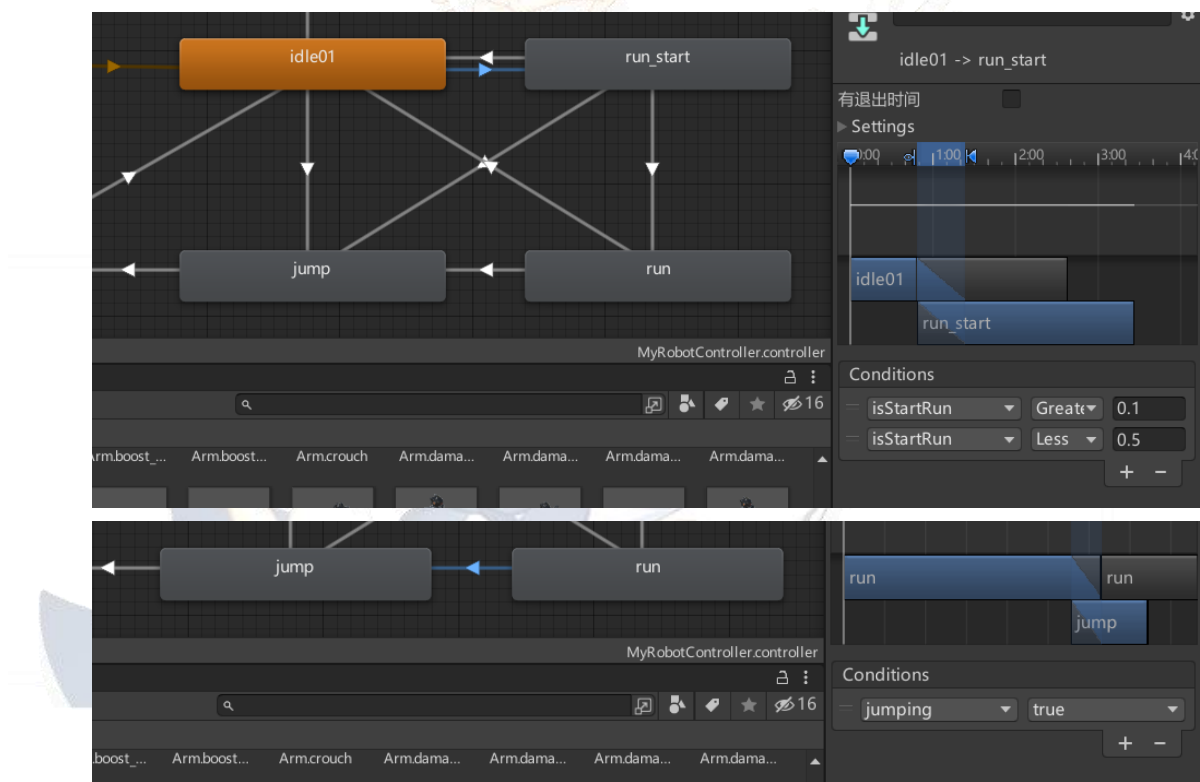


(四) 添加状态机之间的切换

打开 RobotController 的 Paramters 视图，点击右上角 “+”，选择 Trigger



选中两个动画状态之间的白线，在右侧视图中可以看到 Conditions 属性为空，
 点击右下角“+”用 Parameters 添加的状态转换的条件



其他动画跳转以此类推

(五) 编写控制角色动画的脚本

初始变量

```
0 references
5 public class KeyController : MonoBehaviour
6 {
7     4 references
8     private Rigidbody rb;
9     14 references
10    private Animator anim;
11    1 reference
12    public float forwardSpeed;//机器人前进速度
13    1 reference
14    public float backwardSpeed;//机器人后退速度
15    4 references
16    public float rotateSpeed;//机器人旋转速度
17    7 references
18    private Vector3 velocity;//机器人移动速度
19    1 reference
20    public float jumpforce;//机器人跳跃力度
```

获取动画控制器

```
void Start()
{
    rb = GetComponent<Rigidbody>();
    anim = GetComponent<Animator>();
}
```

通过动画控制器触发动画切换条件

```
1 reference
90 void SwitchAni()//切换动画
91 {
92     if (Input.GetKeyDown(KeyCode.A) || Input.GetKeyDown(KeyCode.D) || Input.GetKeyDown(KeyCode.W) || Input.GetKeyDown(KeyCode.S))
93     {
94         anim.SetFloat("isStartRun", 0.3f);
95     }
96     if (Input.GetKeyUp(KeyCode.A) || Input.GetKeyUp(KeyCode.D) || Input.GetKeyUp(KeyCode.W) || Input.GetKeyUp(KeyCode.S))
97     {
98         anim.SetFloat("isStartRun", -0.3f);
99         anim.SetBool("running", false);
100     }
101     if (anim.GetFloat("isStartRun") > 0.2)
102     {
103         anim.SetBool("running", true);
104     }
105
106     //fall
107     if (anim.GetBool("jumping") && rb.velocity.y < 0)
108     {
109         anim.SetBool("falling", true);
110         anim.SetBool("jumping", false);
111     }
112
113     if (anim.GetBool("falling") && rb.velocity.y == 0)
114     {
115         anim.SetBool("falling", false);
116     }
117     //attacking
118     if (Input.GetKeyDown(KeyCode.U))
119     {
120         anim.SetTrigger("attacking");
121     }
```

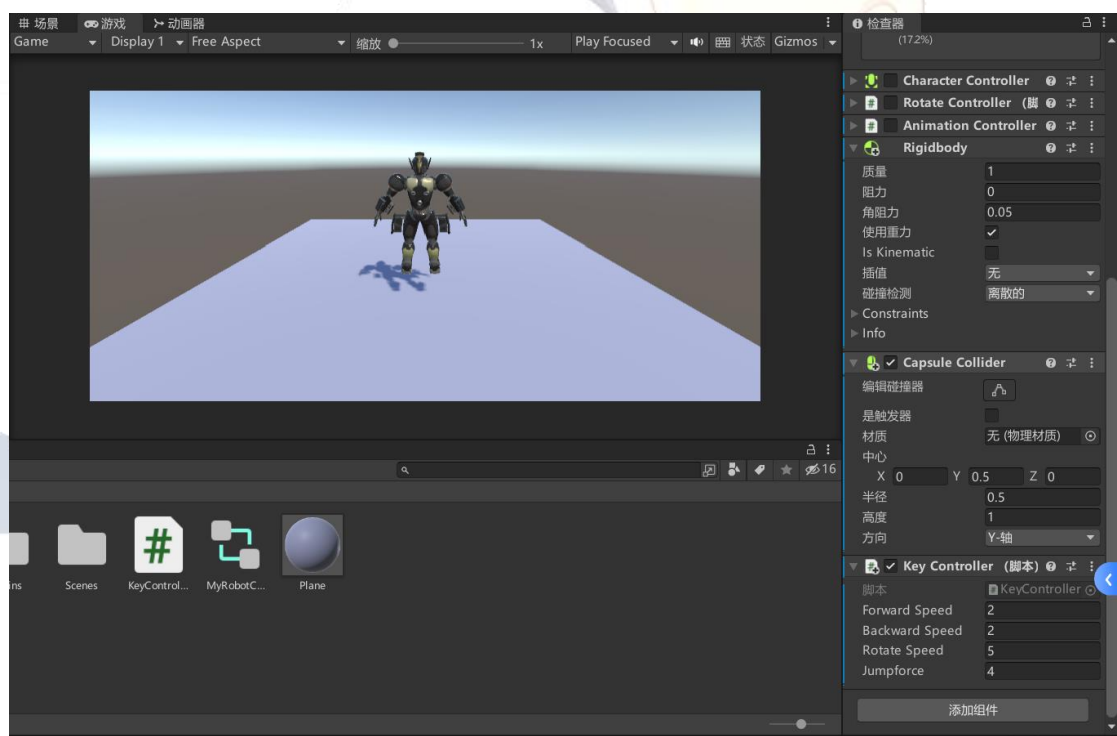
监控 WASD、空格、U 键来实现机器人的移动

```
0 references
30 void FixedUpdate()
31 {
32     float h = Input.GetAxis("Horizontal");
33     float v = Input.GetAxis("Vertical");
34     //控制机器人上下左右移动
35     if (h > 0){
36         velocity = new Vector3(0, 0, v+h);//左右移动也会带动前后移动
37     } else{
38         velocity = new Vector3(0, 0, v-h);
39     }
40     velocity = transform.TransformDirection(velocity);//将机器人的移动方向转换为世界坐标系的方向
41     if (v > 0)
42     {
43         velocity *= forwardSpeed;
44     }
45
46     else
47     {
48         velocity *= backwardSpeed;
49     }
50     transform.localPosition += velocity * Time.fixedDeltaTime;
```

实现机器人面向方向的旋转

```
52 //控制机器人旋转 只移动到左右，达到90度旋转后，再按下左右键，机器人不会再旋转
53 if (h > 0 && transform.rotation.y < 0.7)//右转
54 {
55     transform.Rotate(0, h * rotateSpeed, 0);
56     //输出机器人的旋转角度
57     Debug.Log(transform.rotation.y);
58 }
59 else if (h < 0 && transform.rotation.y > -0.7)//左转
60 {
61     transform.Rotate(0, h * rotateSpeed, 0);
62     Debug.Log(transform.rotation.y);
63 }
64 else if (h == 0 && transform.rotation.y > 0.05)//逆时针转到0度
65 {
66     transform.Rotate(0, -v * rotateSpeed, 0);
67     Debug.Log(transform.rotation.y);
68 }
69 else if (h == 0 && transform.rotation.y < -0.05)//顺时针转到0度
70 {
71     transform.Rotate(0, v * rotateSpeed, 0);
72     Debug.Log(transform.rotation.y);
73 }
```

将脚本挂载到 NPC 角色上



此时即可实现实验所要求的全部内容！

最终效果详见视频演示

五、 实验心得总结：

这次实验我学习了如何使用 Unity 的动画器进行动画的编辑，也理解了动画器本身是一个状态机。在实验项目的实现中，我学会了给动画添加事件，并且添加脚本实现各种事件。本次实验收获颇丰，希望下次可以取得更大进步！

