

中南大学

《动画与游戏程序设计》

课程实验报告



实验名称 基于 Unity 的 Roll A Ball 游戏项目

学生姓名 _____

学 号 _____

专业班级

指导教师

学 院

目 录

一、实验概述	3
1.1 实验名称.....	3
1.2 实验目的.....	3
1.3 实验要求.....	3
1.4 开发步骤指导.....	3
二、实验步骤	3
三、最终游戏效果	14
四、游戏制作遇到的问题及解决方法	15
五、总结与体会	错误!未定义书签。

一、实验概述

1.1 实验名称

基于 Unity 的 Roll A Ball 游戏项目。

1.2 实验目的

本次实验的总的目的是通过具体的程序的编写与 unity 软件相结合，将所学的知识内化，即将在课堂上学到的知识集成在一起，并实现相关功能，从而锻炼自己的程序编写、程序调试能力以及对 unity 的掌握能力。

1.3 实验要求

3.1 基本要求

1. 构建一个小球滚动的游戏场景；
2. 创建一个小球，按键盘上的上下左右键，小球会朝相应的方向移动，小球移动的时候相机也要相应移动；
3. 在场景中创建多个立方体，每个立方体都在旋转；小球与立方体发生碰撞的时候，立方体消失，计分板上得分加“1”；
4. 当得分达到“5”分时，在屏幕上显示“XXX 同学，你赢了！”，如果不能输出中文，可以用英文代替。

3.2 加分项目

1. 添加小球和立方体发生碰撞的特效，添加立方体随机生成，添加小球撞击障碍物的物理效果。
2. 你能想到的可以实现的其他效果。

1.4 开发步骤指导

开发步骤指导见附件教学视频。

Siki 学院：<http://www.sikiedu.com/my/course/862>

二、实验步骤

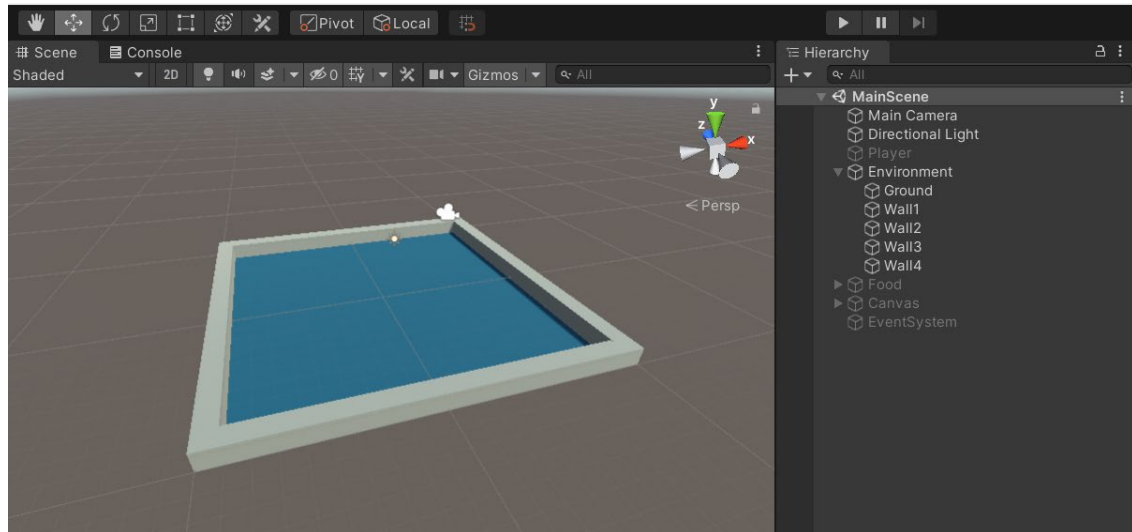
1、创建游戏环境

先创建一个空游戏对象（Create Empty），并重命名为 Environment。游戏环境主要由地面及墙壁组成，使用 3D 对象 Plane 作为地面，3D 对象 Cube 作为墙

壁。

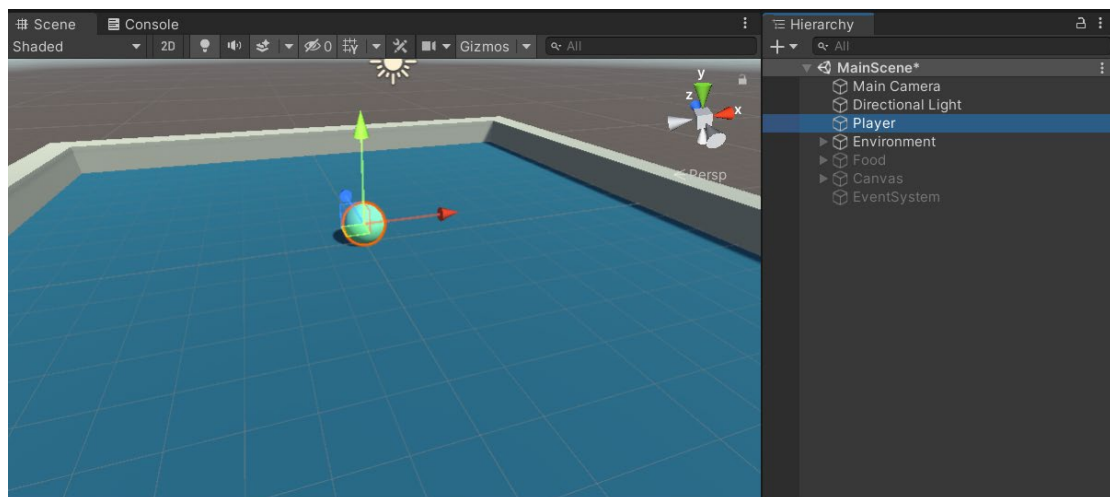
- ① 在 Environment 对象下创建 Plane 对象重命名为 Ground, 相对位置(0, 0, 0);
- ② 在 Environment 对象下创建 4 个 Cube, 作为 4 面墙壁, 调整相对位置及大小, 使其刚好包围 Ground 对象。

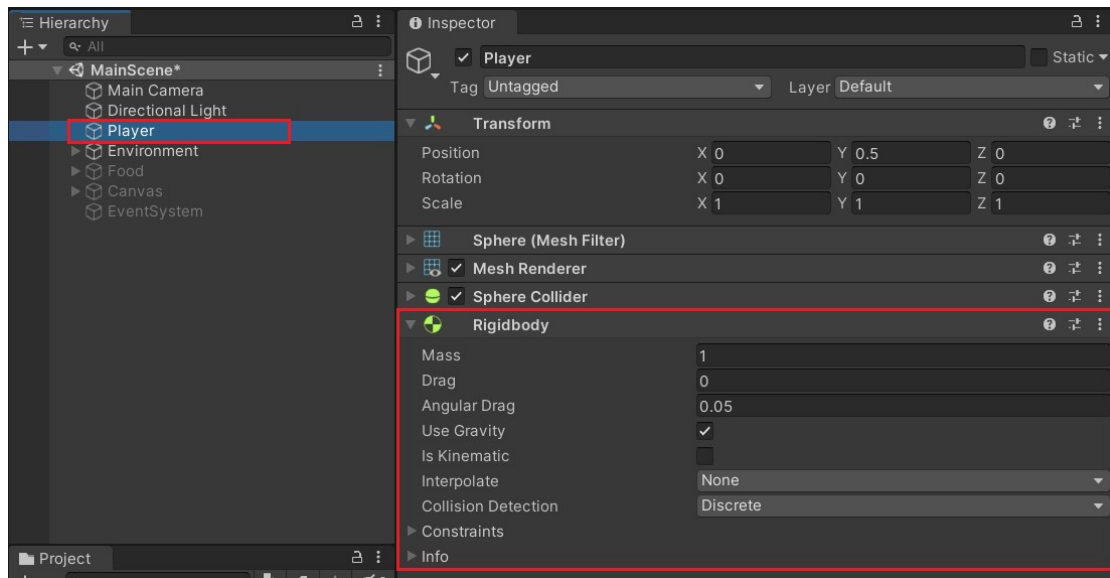
创建后的界面如下:



2、创建小球对象

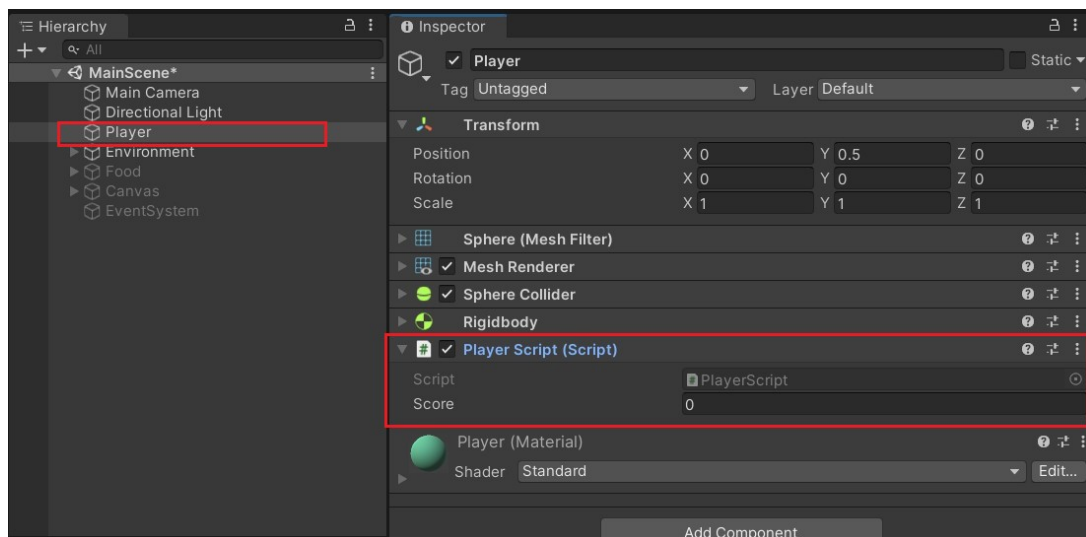
使用 3D 对象 Sphere 作为小球。在 MainScene 下创建 Sphere 对象并重命名为 Player。并且为其添加 Rigidbody 组件, 设置为刚体, 以便后面进行触发检测。





3、创建脚本控制小球移动

创建 PlayerScript 脚本并将其添加到 Player 对象上控制小球 Player 的移动。



双击脚本用 Visual Studio 打开，脚本代码如下：

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class PlayerScript : MonoBehaviour
6. {
7.     private Rigidbody rigidbody;
8.
9.     // Start is called before the first frame update
10.    void Start()
11.    {
12.        rigidbody = GetComponent<Rigidbody>();
```

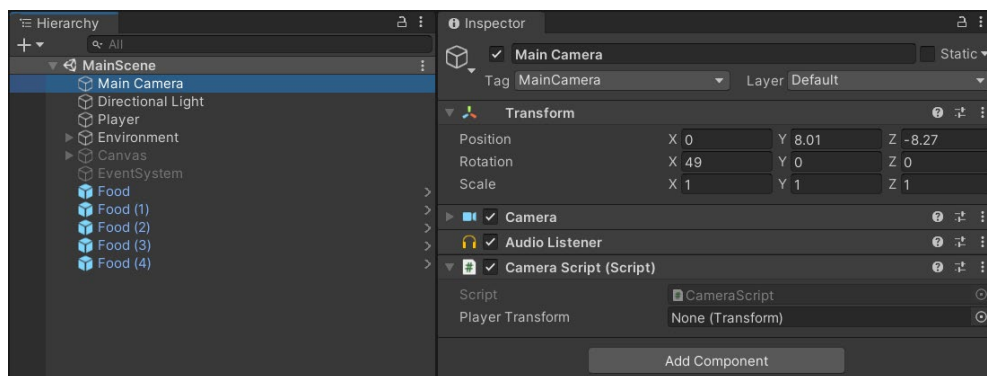
```

13.     }
14.
15.     // Update is called once per frame
16.     void Update()
17.     {
18.         // 监听键盘输入控制物体移动
19.         float h = Input.GetAxis("Horizontal");
20.         float v = Input.GetAxis("Vertical");
21.         rigidbody.AddForce(new Vector3(h, 0, v) * 3);
22.     }
23. }

```

4、相机跟随小球

创建 CameraScript 脚本，并将其添加到 Main Camera 对象上使其跟随小球移动。思路为，先计算相机与小球的初始矢量距离，然后在 Update()函数中实时让相机与小球保持这个矢量距离。



脚本内容如下：

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class CameraScript : MonoBehaviour
6. {
7.     public Transform playerTransform;
8.     private Vector3 offset;
9.
10.    // Start is called before the first frame update
11.    void Start()
12.    {
13.        // 计算相机与待跟随物体之间的初始矢量距离
14.        playerTransform = GameObject.Find("Player").GetComponent<Transform>();
15.        offset = transform.position - playerTransform.position;

```

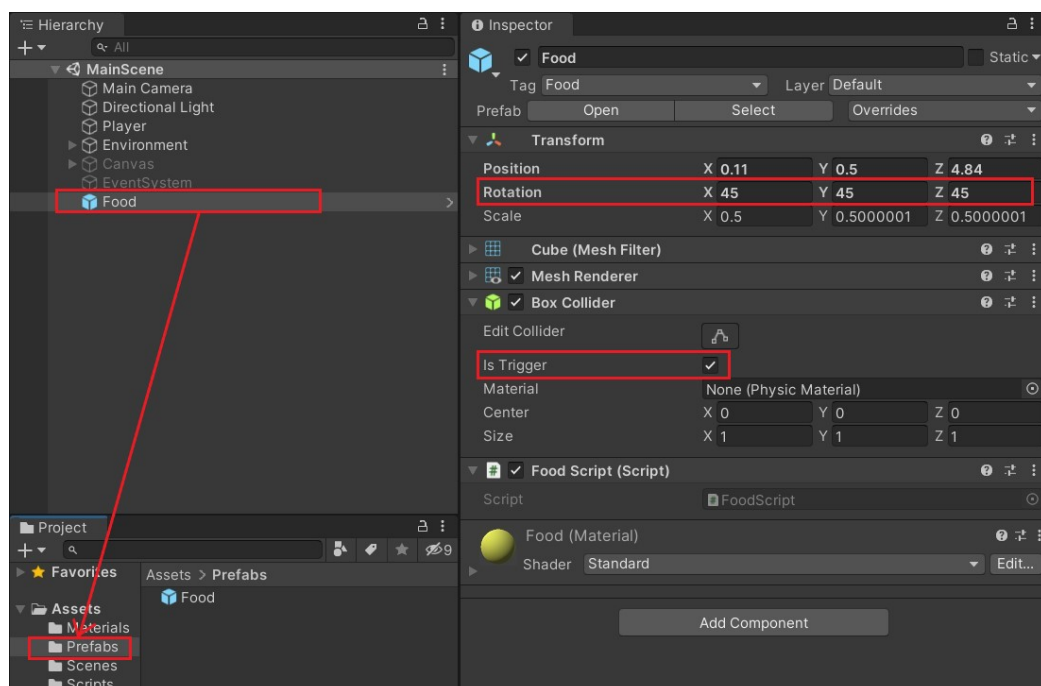
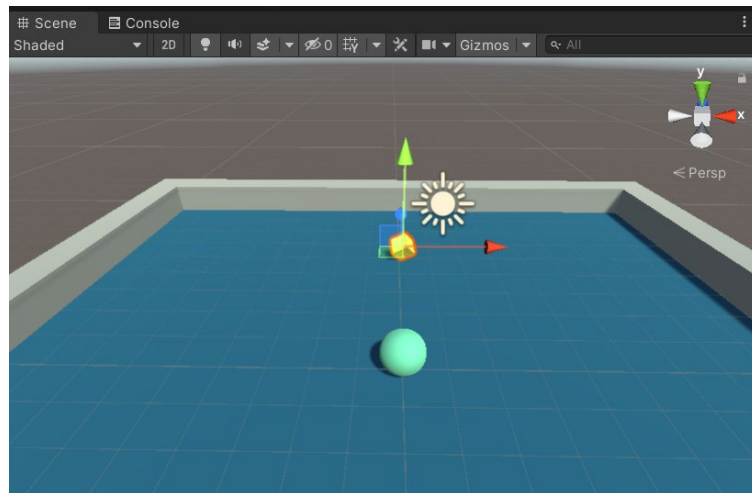
```

16.     }
17.
18.     // Update is called once per frame
19.     void Update()
20.     {
21.         // 根据物体移动后的实时位置更新相机的位置
22.         transform.position = playerTransform.position + offset;
23.     }
24. }

```

5、创建食物预制体

使用 3D 对象 Cube 作为食物，创建 Cube 并重命名为 Food，将其 Rotation 设置为(45, 45, 45)，碰撞器设置为触发器。设置完成后将其置为预制体，便于后续随机生成食物。



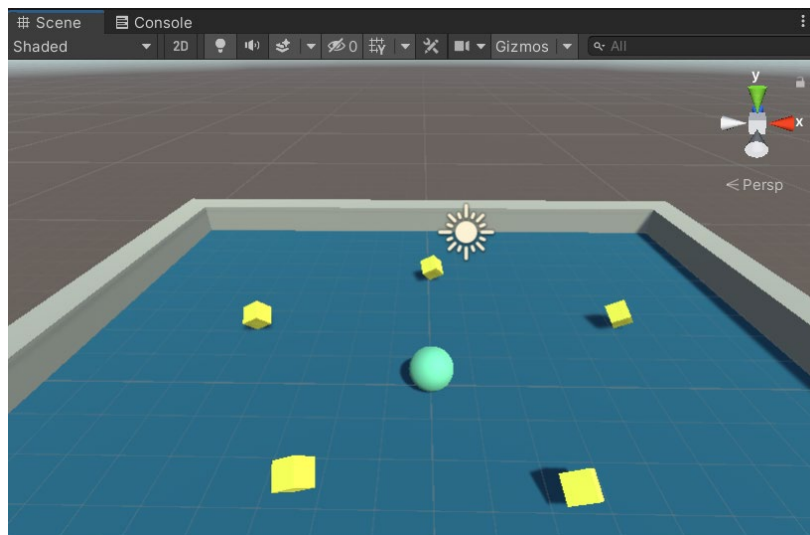
6、控制 Food 对象旋转

创建 FoodScript 脚本并将其添加到 Food 对象上, 该脚本控制 Food 对象的旋转。思路为, 让小球每帧围绕 Y 轴旋转 1° 。

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class FoodScript : MonoBehaviour
6. {
7.     // Start is called before the first frame update
8.     void Start() { }
9.
10.    // Update is called once per frame
11.    void Update()
12.    {
13.        // 围绕 y 轴旋转, 每次旋转 1°
14.        transform.Rotate(Vector3.up);
15.    }
16.}
```

7、创建 5 个初始食物

将 Food 预制体拖拽至 Hierarchy 面包板中, 创建另外 4 个食物对象。



8、在场景中随机生成 Food 对象（加分项）

创建 RandomFood 脚本, 将其添加到 Environment 对象, 该脚本控制立方体的随机生成。思路为, 每隔 3 秒在地面上随机生成一个 Food 预制体。

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

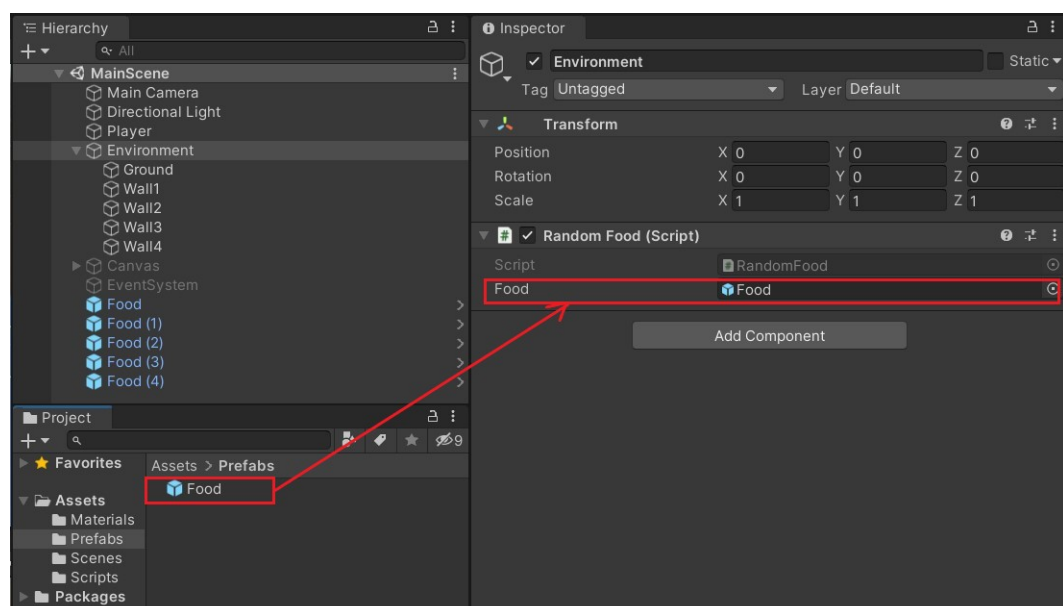


```

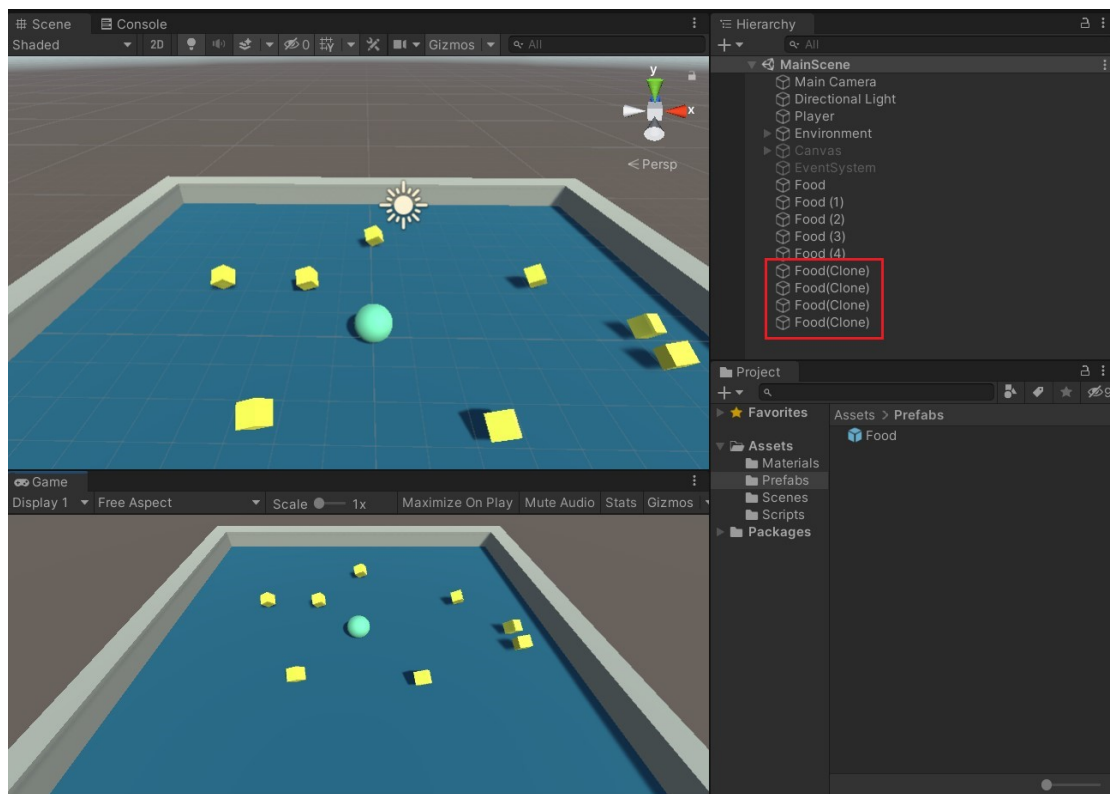
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class RandomFood : MonoBehaviour
6. {
7.     public GameObject m_Food;
8.
9.     private float time = 0;
10.
11.     // Start is called before the first frame update
12.     void Start() { }
13.
14.     // Update is called once per frame
15.     void Update()
16.     {
17.         time += Time.deltaTime;
18.         if (time >= 3)
19.         {
20.             int x = Random.Range(-8, 9);
21.             int z = Random.Range(-8, 9);
22.             Vector3 foodPosition = new Vector3(x, 0.5f, z);
23.             Instantiate(m_Food, foodPosition, m_Food.transform.rotation);
24.             time = 0;
25.         }
26.     }
27. }

```

public 变量 m_Food 在 Unity 中指定，将 Prefabs 文件夹中的 Food 预制体拖拽到 RandomFood 脚本的 Food 变量中。



运行游戏，场景中每过 3 秒将会在随机位置生成立方体。



9、实现球体“吃”立方体

修改 Player 对象的 PlayerScript 脚本，实现当小球碰撞到立方体时，立方体消失，即实现小球“吃”立方体的效果。这里采用触发检测的效果，是为了消除刚体碰撞带来的物理效果。

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class PlayerScript : MonoBehaviour
6. {
7.     private Rigidbody rigidbody;
8.
9.     // Start is called before the first frame update
10.    void Start()
11.    {
12.        rigidbody = GetComponent<Rigidbody>();
13.    }
14.
15.    // Update is called once per frame
16.    void Update()
17.    {
```

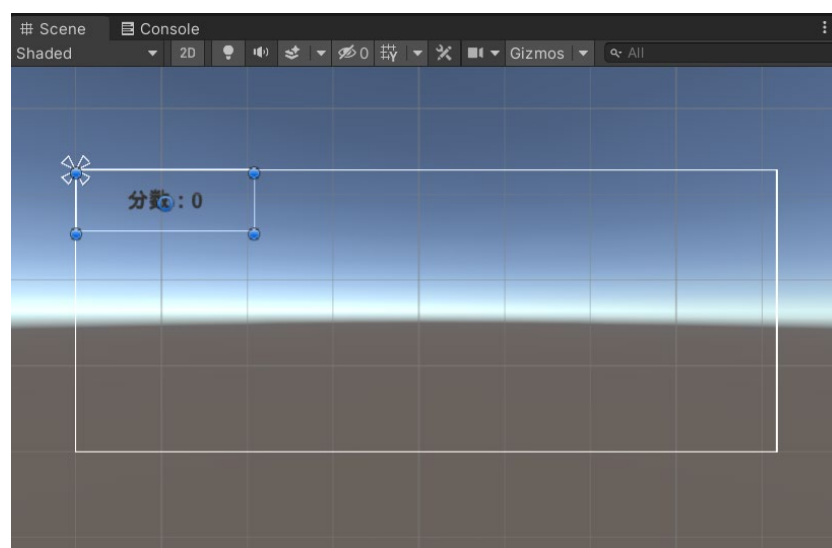
```

18.         // 监听键盘输入控制物体移动
19.         float h = Input.GetAxis("Horizontal");
20.         float v = Input.GetAxis("Vertical");
21.         rigidbody.AddForce(new Vector3(h, 0, v) * 3);
22.     }
23.
24.     // 触发检测, 检测 Player 是否碰到了立方体 Food
25.     private void OnTriggerEnter(Collider other)
26.     {
27.         // 如果碰撞到的物体是 Food, 销毁该游戏物体, 否则不处理
28.         if (other.tag == "Food")
29.         {
30.             Destroy(other.gameObject);
31.         }
32.     }
33. }

```

10、创建计分板

在 Hierarchy 面板右键->UI->Text 创建计分板 Score 对象, 将其放置在 UI 对象的左上角用于分数显示。



11、实现计分功能

修改 Player 对象的 PlayerScript 脚本, 实现小球每碰到一个立方体, 分数就加一, 并更新 UI 显示。

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;

```

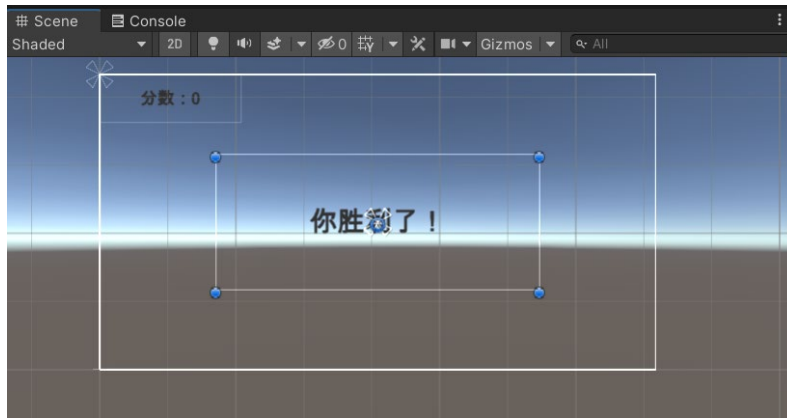
```

5.
6. public class PlayerScript : MonoBehaviour
7. {
8.     private Rigidbody rigidbody;
9.     private Text scoreText;
10.
11.     public int score = 0;
12.
13.     // Start is called before the first frame update
14.     void Start()
15.     {
16.         rigidbody = GetComponent<Rigidbody>();
17.         scoreText = GameObject.Find("Canvas/Score").GetComponent<Text>();
18.     }
19.
20.     // Update is called once per frame
21.     void Update()
22.     {
23.         // 监听键盘输入控制物体移动
24.         float h = Input.GetAxis("Horizontal");
25.         float v = Input.GetAxis("Vertical");
26.         rigidbody.AddForce(new Vector3(h, 0, v) * 3);
27.     }
28.
29.     // 触发检测, 检测 Player 是否碰到了立方体 Food
30.     private void OnTriggerEnter(Collider other)
31.     {
32.         // 如果碰到的物体是 Food, 销毁该游戏物体, 分数加 1
33.         if (other.tag == "Food")
34.         {
35.             Destroy(other.gameObject);
36.             score++;
37.             // 更新 UI 显示
38.             scoreText.text = "分数: " + score;
39.         }
40.     }
41. }

```

12、实现游戏胜利功能

在之前已创建的 Canvas 对象下，右键创建一个 Text 对象，重命名为 Win；将其放置在 Canvas 对象的中间，用以显示胜利提示。



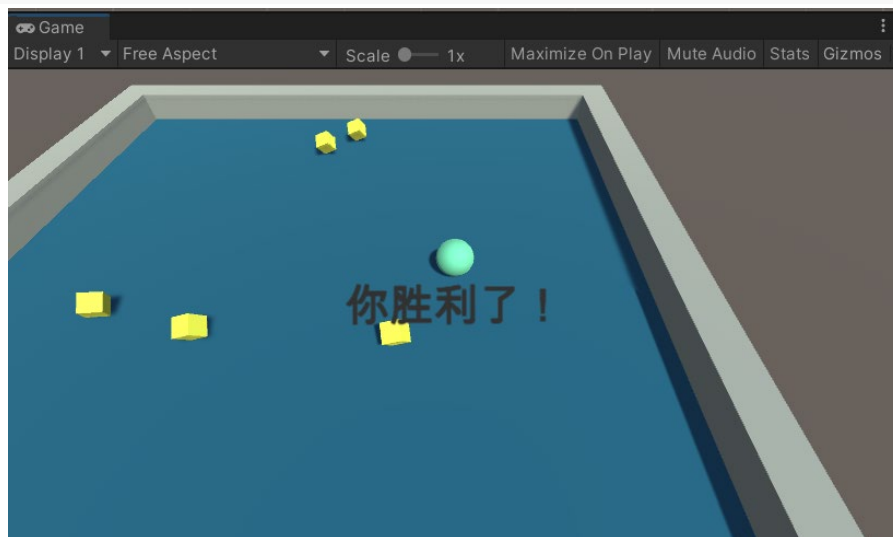
同时，继续修改 PlayerScript 脚本，实现当计分达到 5 分时，提醒用户游戏胜利。

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5.
6. public class PlayerScript : MonoBehaviour
7. {
8.     private Rigidbody rigidbody;
9.     private Text scoreText;
10.    private Text winText;
11.
12.    public int score = 0;
13.
14.    // Start is called before the first frame update
15.    void Start()
16.    {
17.        rigidbody = GetComponent<Rigidbody>();
18.        scoreText = GameObject.Find("Canvas/Score").GetComponent<Text>();
19.        winText = GameObject.Find("Canvas/Win").GetComponent<Text>();
20.        ;
21.        // 初始时设置 Win 对象不显示
22.        winText.enabled = false;
23.    }
24.
25.    // Update is called once per frame
26.    void Update()
27.    {
28.        // 监听键盘输入控制物体移动
29.        float h = Input.GetAxis("Horizontal");
30.        float v = Input.GetAxis("Vertical");
```

```

30.         rigidbody.AddForce(new Vector3(h, 0, v) * 3);
31.
32.         // 当分数达到 5 分时，显示用户胜利，隐藏计分板
33.         if (score == 5)
34.         {
35.             scoreText.enabled = false;
36.             winText.enabled = true;
37.         }
38.     }
39.
40.     // 触发检测，检测 Player 是否碰到了立方体 Food
41.     private void OnTriggerEnter(Collider other)
42.     {
43.         // 如果碰到的物体是 Food，销毁该游戏物体，分数加 1
44.         if (other.tag == "Food")
45.         {
46.             Destroy(other.gameObject);
47.             score++;
48.             // 更新 UI 显示
49.             scoreText.text = "分数: " + score;
50.         }
51.     }
52. }

```



三、最终游戏效果

游戏运行开始时，小球位于平台中心，在小球周围有 5 个初始的旋转的立方体，并且在游戏过程中每隔 3 秒会生成一个立方体。当小球接触到立方体时，立方体会消失，同时在屏幕左上角的分数会加 1，当分数达到 5 时，不再计分，同

时显示“你胜利了!”。

游戏演示视频见附件：[Roll A Ball 演示视频.mp4](#)

四、游戏制作遇到的问题及解决方法

在实验当中遇到的第一个问题是，小球与立方体发生碰撞后会出现卡顿或者反弹的现象，游戏体验性不佳。后面发现是因为使用了碰撞器的缘故，会产生碰撞的物理效果，因此将立方体设置为触发器 `Trigger`，在脚本中给使用触发检测的函数 `OnTriggerEnter()` 替代碰撞检测的函数 `OnCollisionEnter()`。

第二个问题是设置立方体随机生成的时间间隔，最初的思路是每隔 3 秒生成一个立方体。最初时的代码如下：

```
1. time += 1;
2. if (time >= 3)
3. {
4.     int x = Random.Range(-8, 9);
5.     int z = Random.Range(-8, 9);
6.     Vector3 foodPosition = new Vector3(x, 0.5f, z);
7.     Instantiate(m_Food, foodPosition, m_Food.transform.rotation);
8.     time = 0;
9. }
```

测试发现，游戏一运行，小球瞬间就布满了整个地面，仔细检查代码发现，`Update()` 函数是每一帧调用一次，`time += 1` 实际上是每一帧加 1，达到的效果为实际为每帧生成 1 个。因此，想要实现每 1 秒加 1，需要每一帧加 `Time.deltaTime`，更改代码如下：

```
1. time += Time.deltaTime;
2. if (time >= 3)
3. {
4.     int x = Random.Range(-8, 9);
5.     int z = Random.Range(-8, 9);
6.     Vector3 foodPosition = new Vector3(x, 0.5f, z);
7.     Instantiate(m_Food, foodPosition, m_Food.transform.rotation);
8.     time = 0;
9. }
```