

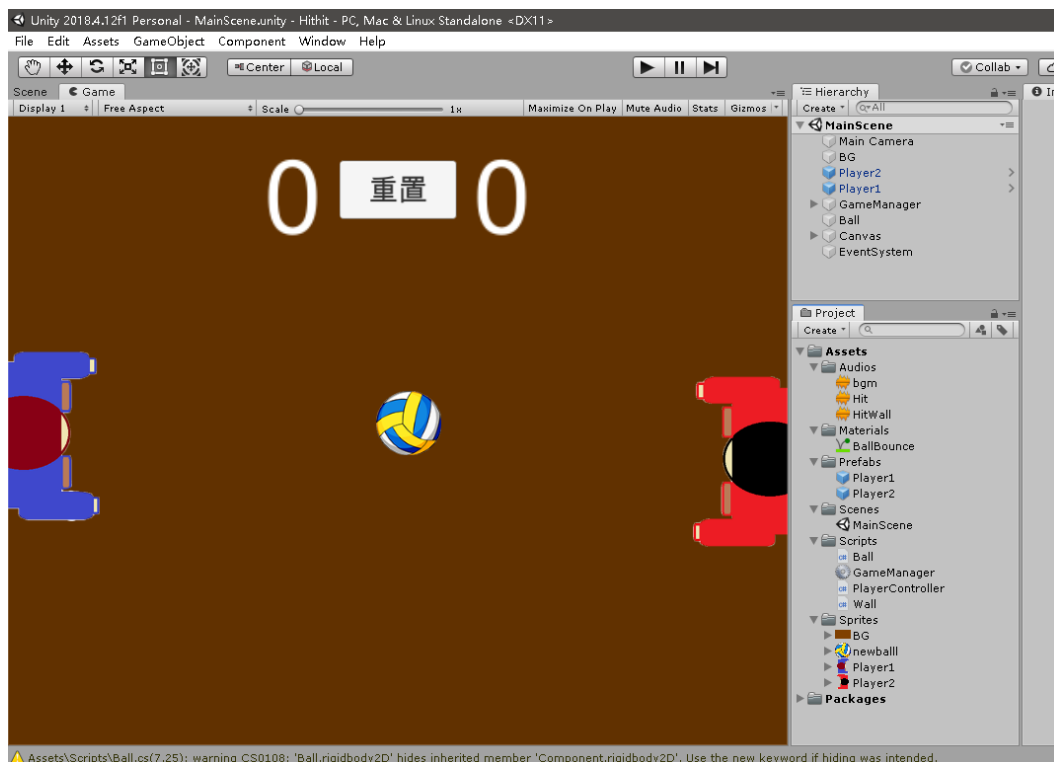
# 代码说明

## 一、环境说明

本项目的运行环境是 windows10 系统下的 Unity 2018.4.12f1 个人版和 vs 2017，代码运行需要安装 Microsoft .NET Framework 4.7.1 Developer Pack。

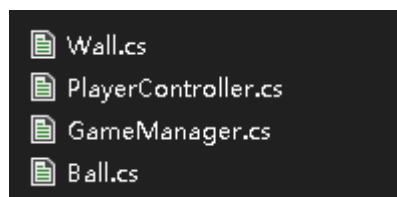
以下为 Unity 界面。本次开发构建的 Assets 文件目录结构如下：

Audios	存放三个音效文件
Materials	设定排球体的弹跳特性
Prefabs	包括两个玩家角色的预设、控制
Scenes	游戏场景
Scripts	游戏代码文件
Sprites	游戏图片素材

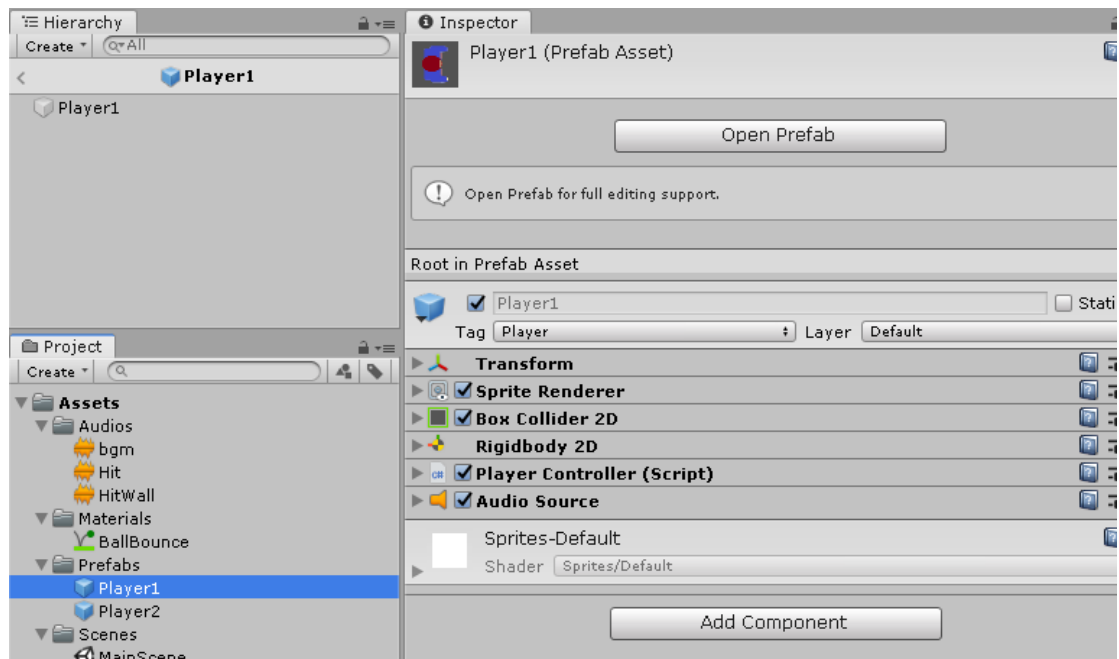


## 二、代码说明

本项目一共包含 4 个代码文件：



### 1. PlayerController



此代码文件是 Player 角色的一个 component，定义了每一帧角色的移动行为和碰撞音效。

内容说明：

```
public class PlayerController : MonoBehaviour
{
    public KeyCode upKey; //定义角色移动用到的上下键
    public KeyCode downKey;
    public float speed = 10; //定义键盘按下时角色移动速度

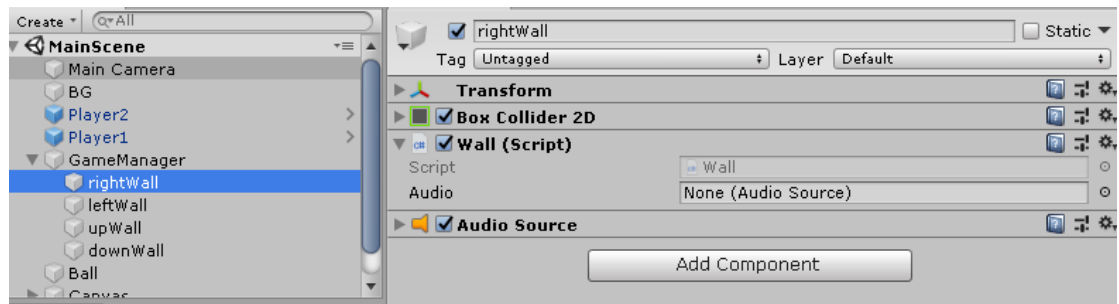
    private Rigidbody2D rigidbody2D;
    private AudioSource audio;

    void Start() //定义启动函数
    {
        audio = GetComponent<AudioSource>(); //引入 Unity 的声控组件
        rigidbody2D = GetComponent<Rigidbody2D>(); //引入 Unity 的 2D 物体控制组件
    }

    // Update is called once per frame
    void Update() //定义每一帧都执行的函数
    {
        if (Input.GetKey(upKey)) //检测键盘的按下情况
        { //按住移动键的期间，角色速度为 10
            rigidbody2D.velocity = new Vector2(0, speed);
        }
        else if (Input.GetKey(downKey))
        {
            rigidbody2D.velocity = new Vector2(0, -speed);
        }
        else //上下移动键均未按下，角色速度为 0
        {
            rigidbody2D.velocity = new Vector2(0, 0);
        }
    }

    void OnCollisionEnter2D() //定义碰撞球体与角色碰撞时的函数
    {
        audio.pitch = Random.Range(0.8f, 1.2f); //设置碰撞音效音量在 0.8-1.2 变动
        audio.Play(); //设置碰撞音效发生
    }
}
```

## 2. Walls



此代码文件是游戏界面四周墙体的一个 component，定义了墙体碰撞时的音效。

内容如下：

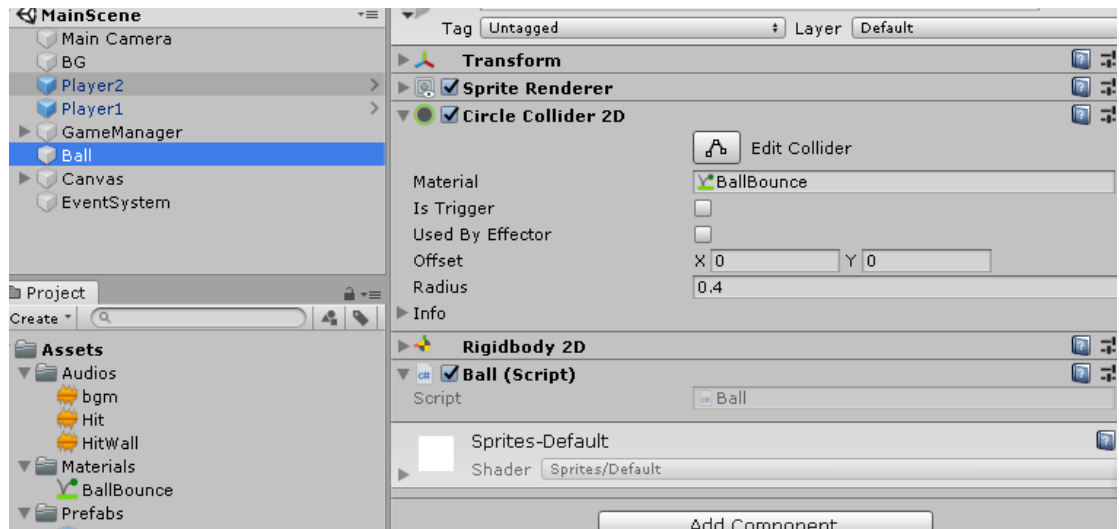
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Wall : MonoBehaviour
{
    public AudioSource audio;
    // Start is called before the first frame update
    void Start() //定义球体启动函数
    {
        audio = GetComponent<AudioSource>(); //引入 Unity 声控组件
    }

    void OnCollisionEnter2D() //定义墙发生碰撞时的函数
    {
        audio.Play(); //播放墙体绑定的碰撞音效
    }

    // Update is called once per frame
    void Update()
    {
    }
}
```

### 3. Ball



此代码为球体的组件，定义了球体的启动位置、速度变化，在碰撞时调用计分函数。

```
public class Ball : MonoBehaviour
{
    private Rigidbody2D rigidbody2D;

    // Start is called before the first frame update
    void Start() //球体启动函数
    {
        rigidbody2D = GetComponent<Rigidbody2D>(); //引入 Unity 的控制组件
        StartBall(); //调用函数刷新球位置、速度
    }

    void OnCollisionEnter2D(Collision2D col) //定义球体碰撞函数
    {
        if(col.collider.tag == "Player") //如果是玩家接到球
        { //如果是，则球体的 Y 坐标速度
            Vector2 velocity = rigidbody2D.velocity; //变为原速度的 1/2 加上玩家
            velocity.y = velocity.y / 2f + col.rigidbody.velocity.y/2; //在 y 坐标速度的 1/2

            if(velocity.x < 7 && velocity.x > -7) //如果球体在 x 轴的速度(由于
            { //摩擦等)下降到 7 以下，则
                velocity.x = velocity.x > 0 ? 10 : -10; //增加到 10
            }
            rigidbody2D.velocity = velocity;
        }

        if(col.gameObject.name == "rightWall" || col.gameObject.name //如果碰到了墙（玩家未接到
        == "leftWall") 球）
        {
            GameManager.Instance.ChangeScore(col.gameObject.name); //引起游戏控制相关的函数改变
            // 玩家分数，并将墙体名字传递过去
        }
    }

    // Update is called once per frame
    void Update(){}
    public void Reset() //定义重置时球体行为
    {
        transform.position = Vector3.zero; //球体位置归原点
        StartBall(); //调用函数刷新球位置、速度
    }

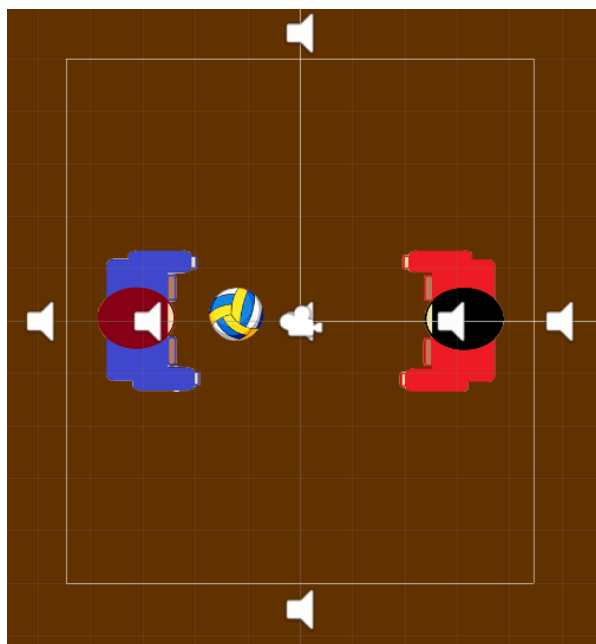
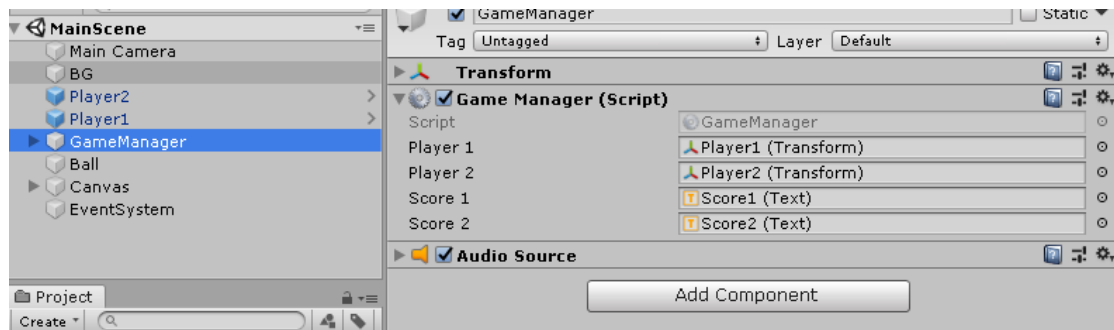
    void StartBall() //设定球起始位置和速度的函数
    {
        int num = Random.Range(0, 2); //随机生成方向指数
        if (num == 1) //指数 1 代表向右发球
        {
            rigidbody2D.AddForce(new Vector2(100, 0));
        }
    }
}
```

```

    }
    else
    {
        rigidbody2D.AddForce(new Vector2(-100, 0)); //指数 0 代表向左发球
    }
}
}

```

#### 4. GameManager



此代码文件为游戏整体控制代码，定义了墙体对品屏幕的自适应、玩家分数的变化和重置按钮的功能。

```

public class GameManager : MonoBehaviour
{
    private static GameManager _instance;
    public static GameManager Instance
    {
        get
        {
            return _instance;
        }
    }
    private BoxCollider2D rightWall; //建立 4 面墙的名字
    private BoxCollider2D leftWall;
    private BoxCollider2D upWall;
    private BoxCollider2D downWall;
}

```

```

public Transform player1; //建立在 Unity 可设定的 player 接口
public Transform player2;
private int score1; //建立分数计数值
private int score2;
public Text Score1; //建立在 Unity 可设定的分数 text 接口
public Text Score2;

void Awake()
{
    _instance = this;
}

// Start is called before the first frame update
void Start() //启动游戏时启动墙体设定和角色重置函数
{
    ResetWall();
    ResetPlayer();
}

// Update is called once per frame
void Update(){ }

void ResetWall() //设定墙体启动函数
{
    //4 堵墙分别连接 Unity 中对应的墙体
    rightWall = transform.Find("rightWall").GetComponent<BoxCollider2D>();
    leftWall = transform.Find("leftWall").GetComponent<BoxCollider2D>();
    upWall = transform.Find("upWall").GetComponent<BoxCollider2D>();
    downWall = transform.Find("downWall").GetComponent<BoxCollider2D>();

    //将屏幕右上方的点坐标转换成游戏界面的坐标，设为参考点 upRightPosition
    float width = Screen.width;
    float height = Screen.height;
    Vector3 upRightPosition = Camera.main.ScreenToWorldPoint(new Vector2(Screen.width, Screen.height));

    //参考右上角的坐标，分别设定四堵墙的中心位置和长宽，使得能够完全堵住游戏界面，不留空隙
    upWall.transform.position = new Vector3(0, upRightPosition.y + 0.5f, 0);
    upWall.size = new Vector2(upRightPosition.x * 2, 1);
    downWall.transform.position = new Vector3(0, -upRightPosition.y - 0.5f, 0);
    downWall.size = new Vector2(upRightPosition.x * 2, 1);
    rightWall.transform.position = new Vector3(upRightPosition.x + 0.5f, 0, 0);
    rightWall.size = new Vector2(1, upRightPosition.y * 2);
    leftWall.transform.position = new Vector3(-upRightPosition.x - 0.5f, 0, 0);
    leftWall.size = new Vector2(1, upRightPosition.y * 2);

}

//定义角色重置后的位置信息
void ResetPlayer()
{
    Vector3 p1Position = Camera.main.ScreenToWorldPoint(new Vector2(100, Screen.height / 2));
    player1.position = new Vector3(p1Position.x, p1Position.y, 0);
    Vector3 p2Position = Camera.main.ScreenToWorldPoint(new Vector2(Screen.width - 100, Screen.height / 2));
    player2.position = new Vector3(p2Position.x, p2Position.y, 0);
}

public void ChangeScore(string wallName) //定义分数变化函数
{
    if(wallName == "rightWall") //如果球撞到对方的墙，则己方分数增加 1 分
    {
        score1++;
    }
    else if(wallName == "leftWall")
    {
        score2++;
    }
    Score1.text = score1.ToString();
    Score2.text = score2.ToString();
}

public void Reset() //定义重置按钮调用的函数
{
    score1 = 0; //分数清零
    score2 = 0;
    Score1.text = score1.ToString();
}

```

```
Score2.text = score2.ToString();
ResetPlayer(); //角色位置重置
GameObject.Find("Ball").SendMessage("Reset"); //球体位置、速度重置
}
}
```