



中山大學

## ISE309 Virtual Reality Technology Assignment

### 虚拟现实技术考核作业

**Deadline: 2019.12.25 12: 00 pm**

**提交截止日期: 2019.12.25 晚上 12 点**

姓名: 王珺 叶茂青

学号: 17363079 17363092

邮箱: [673817269@qq.com](mailto:673817269@qq.com)

## Assignment Regulations(考核作业规章)

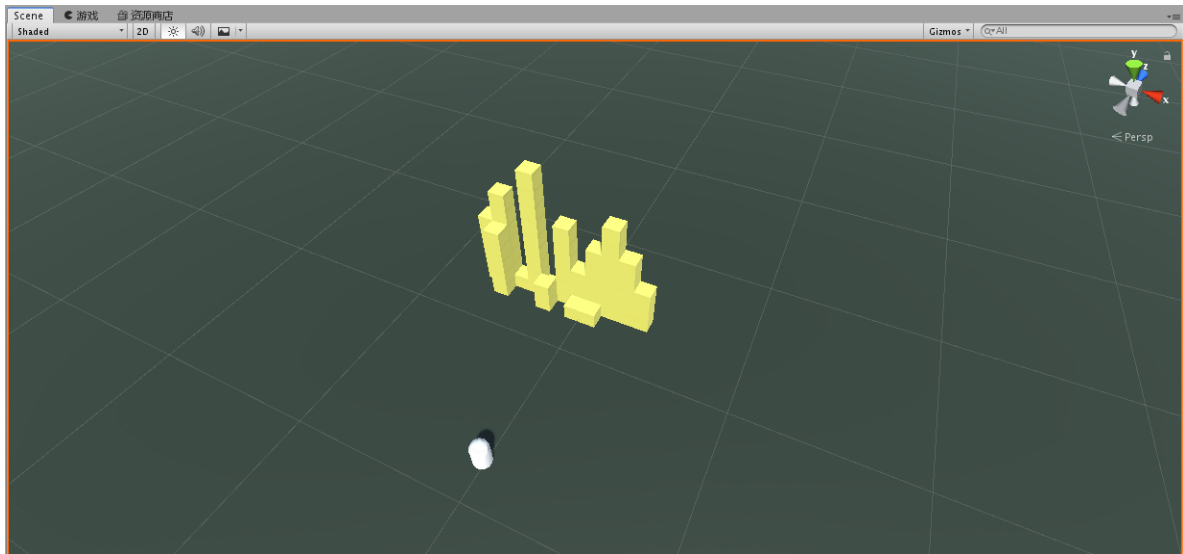
1. This is a personal assignment. Each group MUST submit only the soft copy of the report via sending an email to [sysu\\_ise309@163.com](mailto:sysu_ise309@163.com) and the topic of this email should be “Your name1\_ID1\_name2\_ID2\_Assignment3”, otherwise, your email will be swallowed by hundreds of emails in the mailbox. 这是一次小组独立完成的考核作业，每一组其中一位同学应在提交截止日期前将作业的电子版通过邮件发送至 [sysu\\_ise309@163.com](mailto:sysu_ise309@163.com)。邮件主题应为“**你的姓名\_学号\_你的搭档姓名\_学号\_Assignment3**”，否则主题不明的邮件将会被淹没在邮箱中的其他邮件中。该电子版作业将被直接批改计分，对于没有提交电子版的同学，后果请自负。
2. A coversheet can be created in your own way but the following information must be included: group members' full names, student ID numbers, and email addresses. 作业的封面可以设计，但请包含以下重要信息：小组成员的中文全名、学号和邮箱。
3. You may refer to textbooks, lecture notes, and the Internet to discover approaches to problems, however the assignment should be your own group work. Any plagiarism is NOT ACCEPTABLE. Once we find that, then you will get 0 mark on this assignment. 在作答过程中你也许需要参考课本、课件和网络，这是可以的。但是请确保这份作业是你们小组独立完成的。任何形式的抄袭或作弊都是不被接受的，一旦发现本作业将按 0 分处置。
4. Assignments may be accepted up to 5 days after the deadline has passed; a late penalty of 5% will apply for each day late without an extension being granted. Submissions over 5 days late will not be marked. Emailed submissions will NOT be accepted without exceptional circumstances. 晚于截止日期 5 天内提交的电子版作业仍可受理，但每晚一天总成绩减少 5%（即晚一天提交满分为 95 分，晚两天提交满分为 90 分），以此类推。晚于截止日期 5 天后的提交作业视同无效提交，将不会被批改和打分。
5. Once the group members are settled, for the following assignments, the group members can't be changed. Every time, the assignment only can be submitted by one member. 一旦小组组员（2 位）确定，不允许更改。后面的作业也都要两个人一起完成。并且，每次作业由一人上传一次即可，不需要两个人都上传。请大家开发一个打砖块的小游戏，大致思路及要实现的功能为：
  1. 创建地板，并创建 60 个 cube 堆叠而成的任意造型，其中 cube 要赋予刚体属性。
  2. 鼠标单击创建小球，小球具有刚体属性，并具有初速度。
  3. 键盘的上下左右可以调整小球的创建位置。
  4. 鼠标可以控制射击角度，例如上移代表增大射击仰角。

注意：

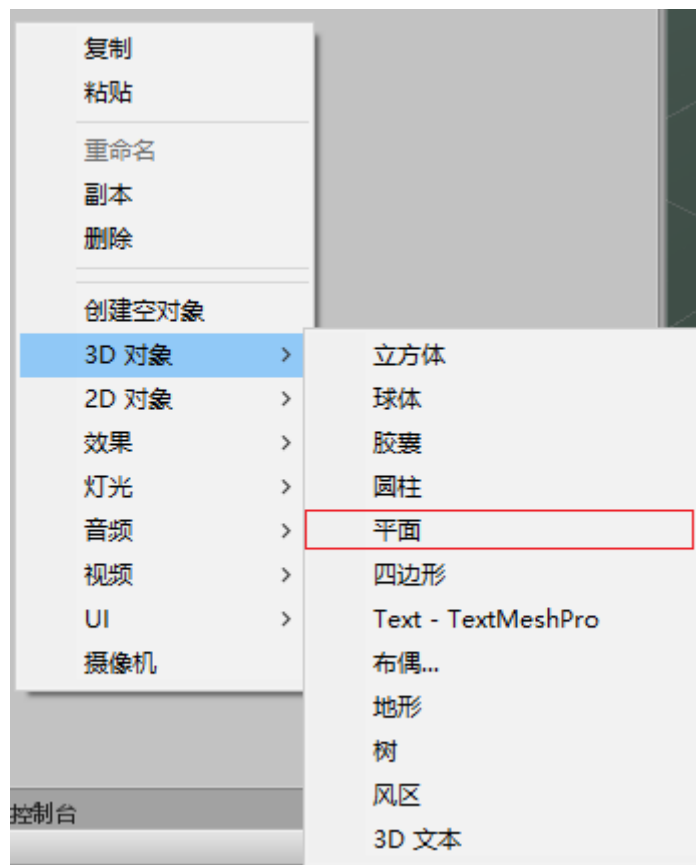
1. 请在作业中附上制作过程中的关键步骤截图，例如为子弹赋予刚体属性，设计属性，移动属性。可参照下述样例。
2. 请将开发好的游戏按照可执行文件(.exe)导出文件并同作业文档一起提交。
3. 在该游戏基本功能的基础上额外设计其他功能，增加游戏的趣味性可以有适当加分。

# 创建地板，砖块，小球

最后的成果如图



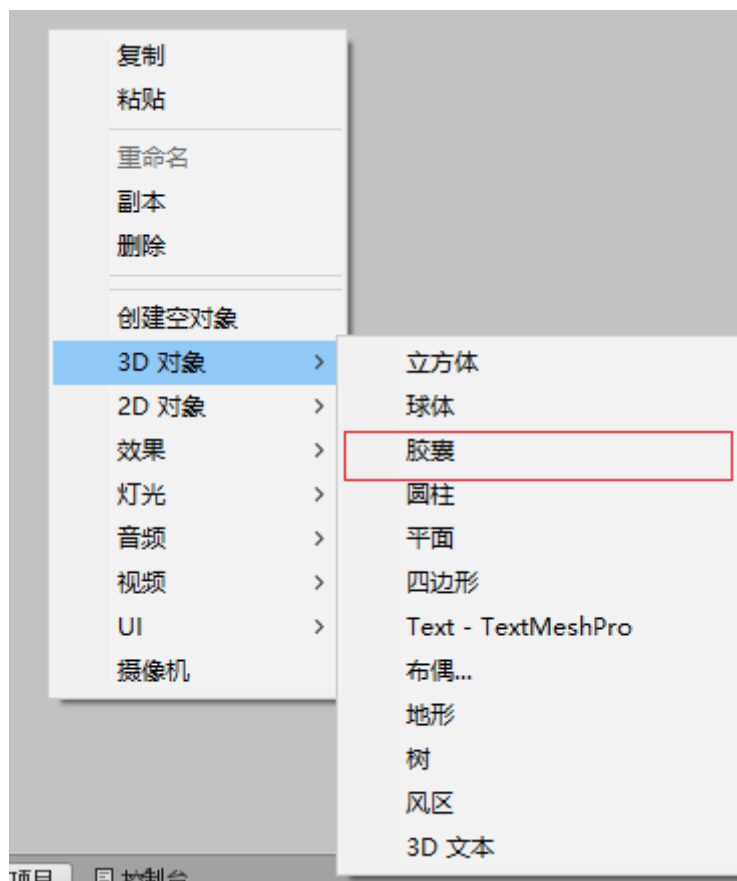
地板用3D对象->平面建出



方块使用3D对象->立方体建出，然后为方块添加刚体属性



胶囊用于表示玩家，采用第一人称的视角，所以摄像头作为胶囊的子对象



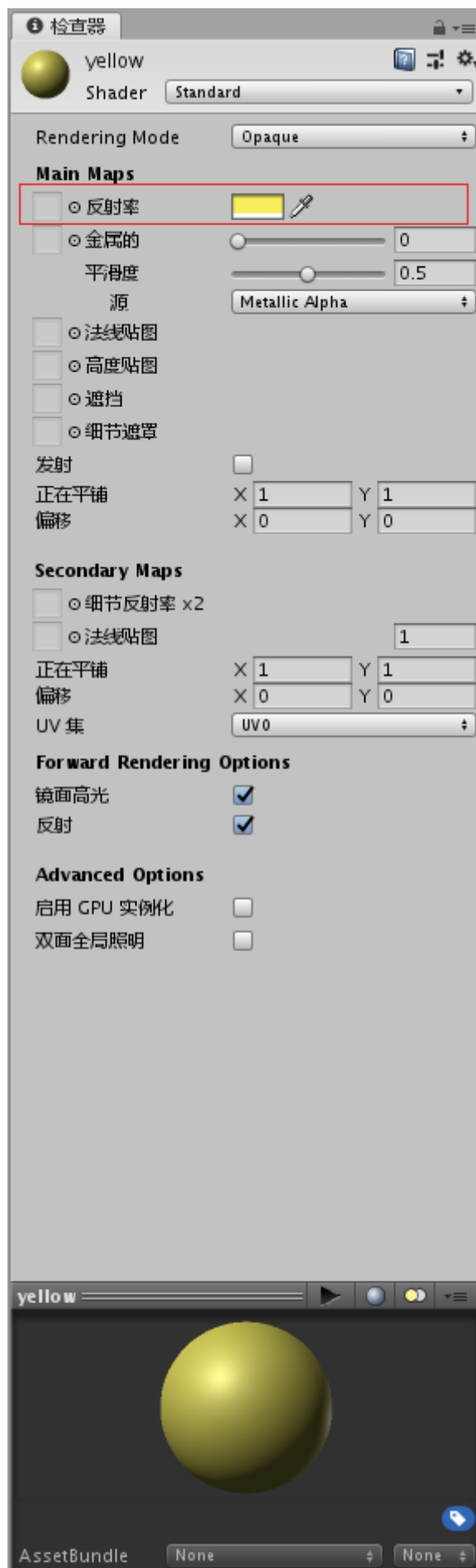
射出的子弹用球体创建，然后保存为预制体



给物体赋予颜色，创建一个材质，然后选择想要的颜色，最后将材质拖动到物体上即可

创建	>	C# 脚本	
在资源管理器中显示		着色器	>
打开		正在测试	>
删除		可播放内容	>
重命名		Assembly Definition	
Copy Path	Alt+Ctrl+C	TextMeshPro	>
打开附加场景		场景	
导入新资源...		Prefab Variant	
导入包	>	音频混合器	
导出包...		材质	
在场景中查找引用		镜头眩光	
选择依赖对象		渲染器纹理	
刷新	Ctrl+R	光照贴图参数	
重新导入		自定义渲染纹理	
重新导入所有		Sprite 图集	
从预制件中提取		Sprites	>
运行 API 更新程序...		平铺	
更新 UIElements 架构		动画器控制器	
打开 C# 项目		动画	
		动画器覆盖控制器	





## 创建射击动作

```

public float fireRate = 0.3f;

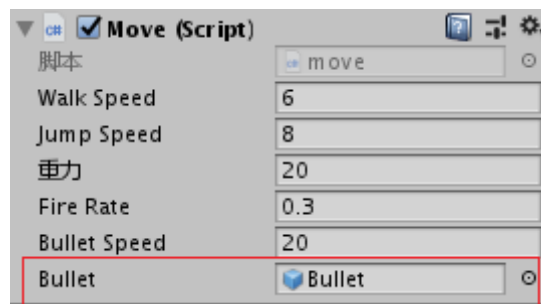
private float nextFire = 0.0f;

void Fire(){
    if(Input.GetButton("Fire1") && Time.time > nextFire){
        GameObject b = GameObject.Instantiate(bullet, transform.position +
transform.TransformDirection(new Vector3(0f,0f,1.5f)), transform.rotation);
        Rigidbody rgd = b.GetComponent<Rigidbody>();
        rgd.velocity = transform.forward * bulletSpeed;
        nextFire = Time.time + fireRate;
    }
}

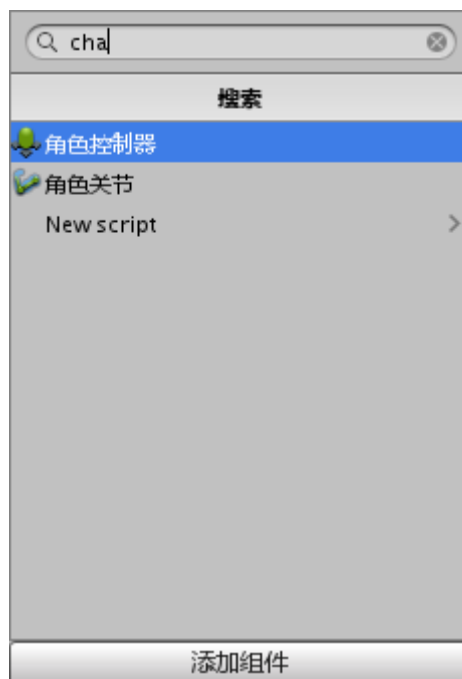
```

在player中添加相应函数，然后在Update()方法里调用就行

在unity界面中需要将创建的bullet预制体拖入进来



## 创建移动动作



在Player中添加一个角色控制器，并编写相应代码（参照unity文档）如下

```

void Move(){
    if (characterController.isGrounded)
    {
        // we are grounded, so recalculate
        // move direction directly from axes
    }
}

```

```

        moveDirection = new Vector3(Input.GetAxis("Horizontal"), 0.0f,
Input.GetAxis("Vertical"));
        moveDirection = transform.TransformDirection(moveDirection);
        moveDirection *= walkSpeed;

        if (Input.GetButton("Jump"))
        {
            moveDirection.y = jumpSpeed;
        }
    }

    // Apply gravity. Gravity is multiplied by deltaTime twice (once here, and
once below
    // when the moveDirection is multiplied by deltaTime). This is because
gravity should be applied
    // as an acceleration (ms^-2)
    moveDirection.y -= gravity * Time.deltaTime;

    // Move the controller
    characterController.Move(moveDirection * Time.deltaTime);
}

```

## 让鼠标可以控制摄像头转动

摄像头添加一个脚本，代码如下

```

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

public class mouseMove : MonoBehaviour
{
    public GameObject player;
    Vector3 rot = new Vector3(0, 0, 0);
    public float speed;
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void LateUpdate()
    {
        float MouseX = Input.GetAxis("Mouse X")*speed;
        float MouseY = Input.GetAxis("Mouse Y")*speed;
        rot.x = rot.x - MouseY;
        rot.y = rot.y + MouseX;
        rot.z = 0;
        //限制上下角度
        if (rot.x<=-70) {
            transform.eulerAngles = new Vector3(-70, rot.y, 0);
        }
        else if (rot.x >= 11) {

```

```

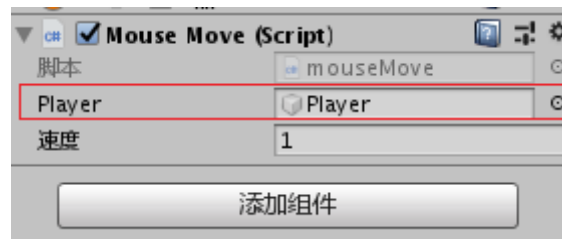
        transform.eulerAngles = new Vector3(11, rot.y, 0);
    }
    else{
        transform.eulerAngles = new Vector3(rot.x, rot.y, 0);
    }
    player.transform.eulerAngles = new Vector3(rot.x, rot.y, 0);
}
}

```

摄像头应该作为player的子对象，如下图



同时需要将player对象拖进来



## 子弹定时消除及破坏效果

给子弹添加一个脚本组件，代码如下

```

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

public class Destroy : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void update()
    {
        Destroy (gameObject,5);
    }

    void OnCollisionEnter(Collision other)
    {
        if (other.gameObject.CompareTag ("goals"))
        {
            other.gameObject.SetActive (false);
            Destroy(gameObject);
        }
    }
}

```

为了使得子弹击中砖块时，砖块会消失，给砖块加上goals的标签，在添加标签处添加goals标签然后选中即可



效果示意图：

