

# 天津大学



## 《 计 算 机 图 形 学 》

学 院： 智能与计算学部

年 级： 2018 级

班 级： 计科 5 班

学 号： 3018216233

学生姓名： 刘煜堃

指导教师： 徐庆

2020 年 11 月 14 日

# 目 录

一 实验目的 .....	1
二 概要设计 .....	1
三 详细设计 .....	1
四 系统实现 .....	5
五 软件测试 .....	6

## 一、实验目的

使用 Unity 软件平台实现城市场景内车辆的控制。

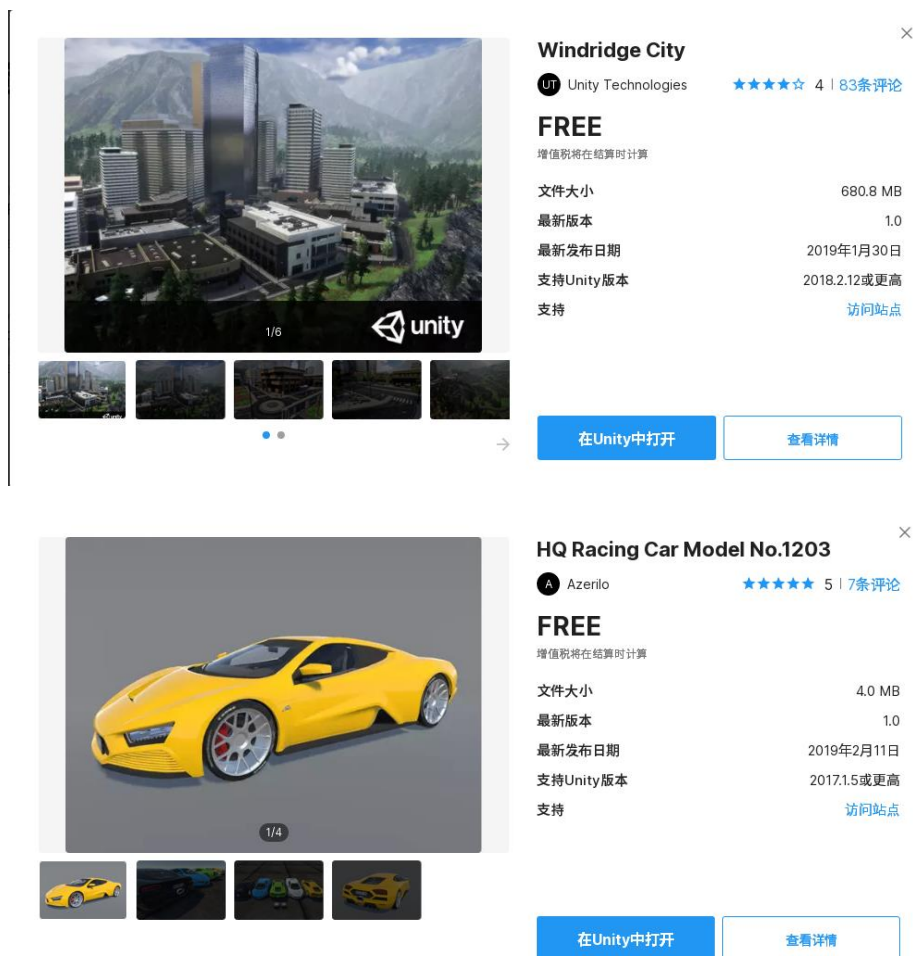
## 二、实验内容

- (1) 选用第三方资源列表，例如美术素材，源代码，API 等；
- (2) 城市场景的搭建，例如灯光的摆放那个，各种场景元素的布置；
- (3) 车辆运动控制的脚本实现，例如车辆的前进，后退，转向等；
- (4) 适当添加其他的额外效果与功能。

## 三、实验步骤

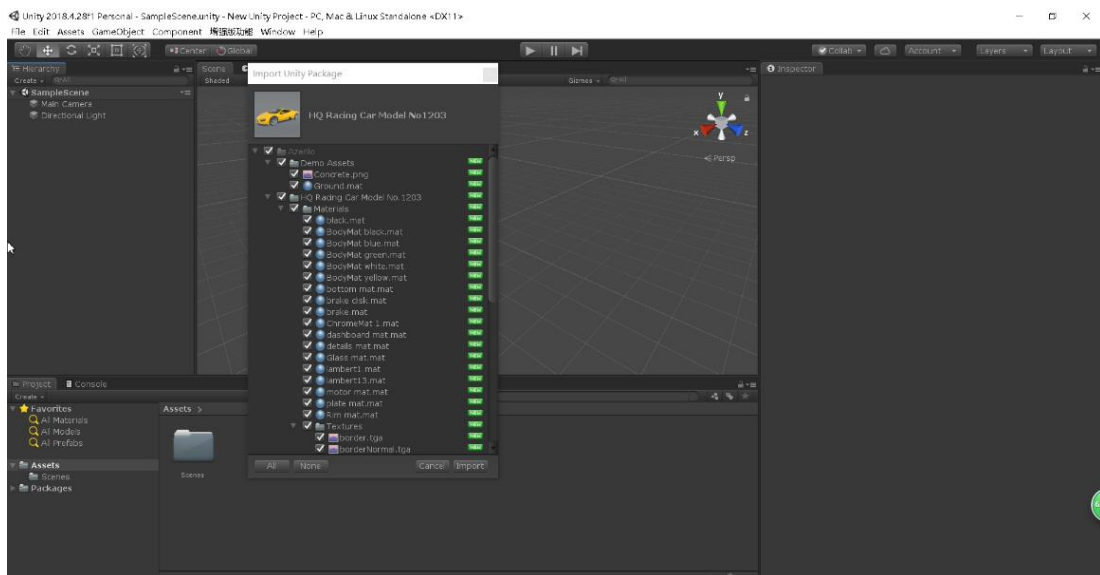
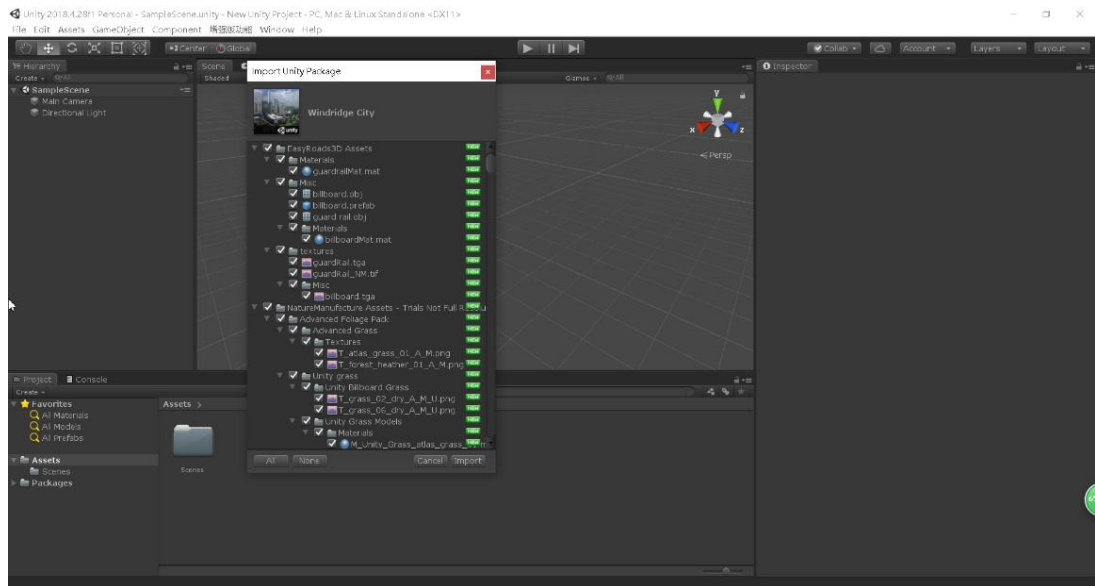
### (1) 第三方资源列表的选用

选用的城市与车辆资源如下：

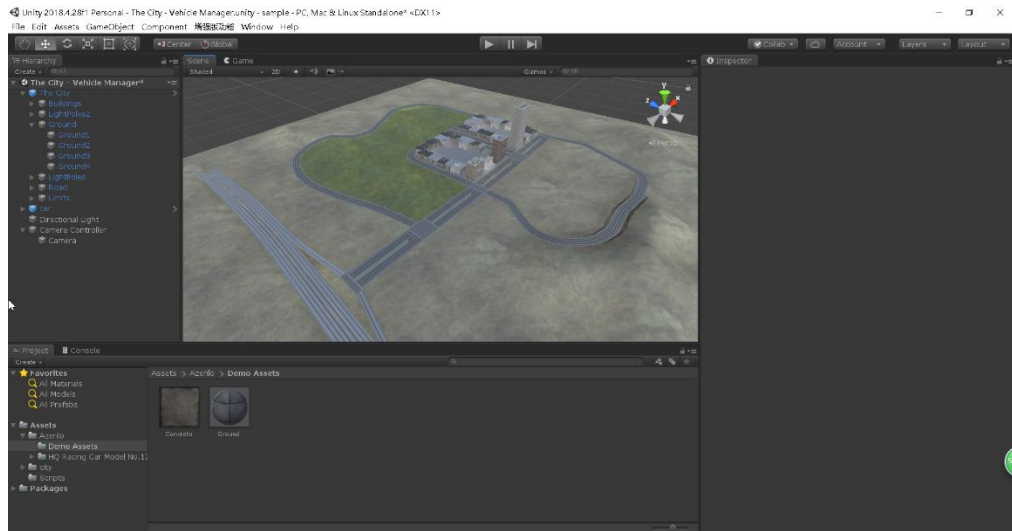


## (2) 场景的搭建

选择完资源后，在 Unity 中进行资源导入：



导入完资源后，从资源列表中选择模块拖入工作区即可：



### (3) 车辆控制脚本的实现：

使用脚本，我们需要实现通过 wasd 键的汽车前进，后退以及左右转弯。通过查阅相关资料，写出的车辆控制脚本如下：

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

public class CarCtrl : MonoBehaviour
{
    float MoveSpeed = 5; //前进速度
    float RotateSpeed = 45; //转向速度

    void Update()
    {
        if(this.transform.up.y > 0 && this.transform.up.y <= 10)
        {
            if(Input.GetKey(KeyCode.W)) //当按下w键时
            {
                if(MoveSpeed <= 80) //当速度不大于200时，按下w键可以使车辆提速
                {
                    MoveSpeed = MoveSpeed + 10 * Time.deltaTime;
                }

                this.transform.Translate(Vector3.forward * Time.deltaTime * MoveSpeed);
            }
        }
    }
}
```

```

        if(Input.GetKey(KeyCode.A))//在按下w键的同时按下a键，车辆向左前方前进
        {
            this.transform.Rotate(Vector3.up * Time.deltaTime * -RotateSpeed);
        }

        else if(Input.GetKey(KeyCode.D))//在按下w键的同时按下d键，车辆向右前方前进
        {
            this.transform.Rotate(Vector3.up * Time.deltaTime * RotateSpeed);
        }
    }

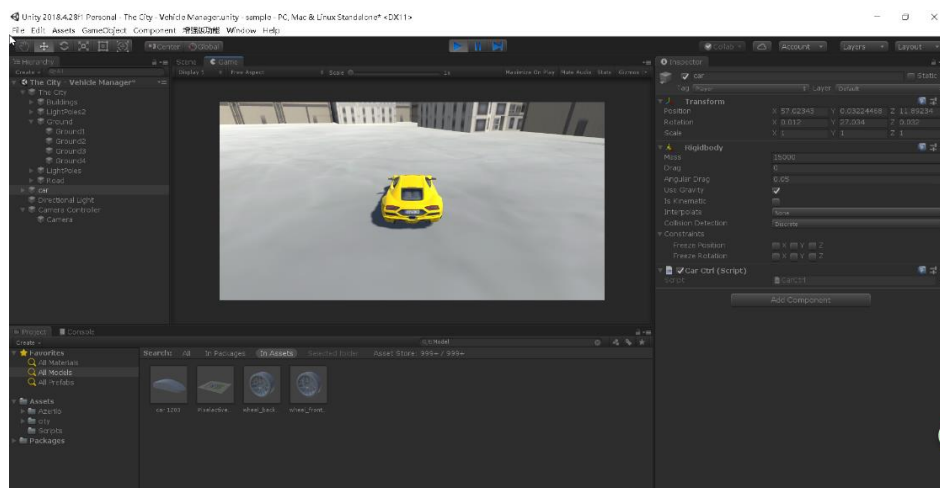
    else if(Input.GetKey(KeyCode.S))//当按下s键时
    {
        MoveSpeed = 15;

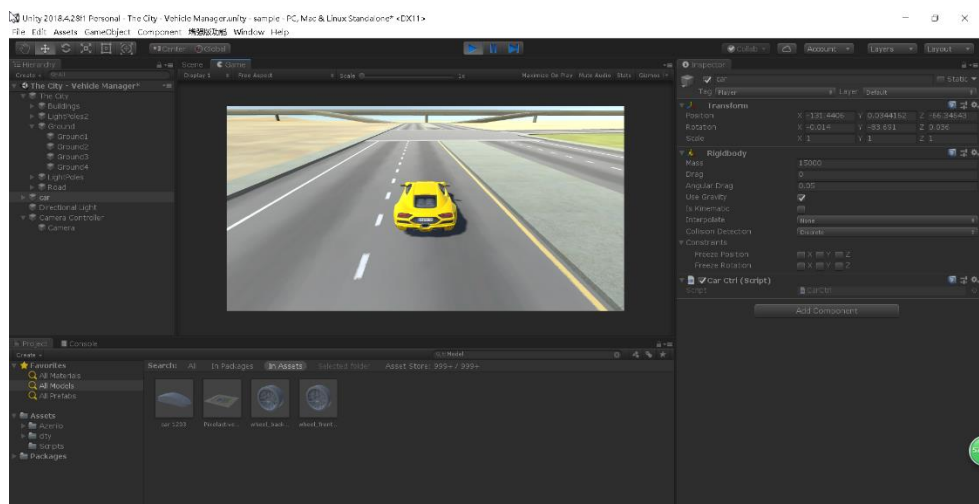
        this.transform.Translate(Vector3.forward * Time.deltaTime * -MoveSpeed);
        if(Input.GetKey(KeyCode.A))//在按下s键的同时按下a键，车辆向左后方倒车
        {
            this.transform.Rotate(Vector3.up * Time.deltaTime * -RotateSpeed);
        }

        else if(Input.GetKey(KeyCode.D))//在按下s键的同时按下d键，车辆向右后方倒车
        {
            this.transform.Rotate(Vector3.up * Time.deltaTime * RotateSpeed);
        }
    }
    //如果车辆并非在行驶过程中，那么只按下a键或d键不会有任何效果
}
}

```

#### (4) 最终实现效果





## 四、实验过程中的问题与解决

(1) 城市场景布置完成后，点击 play 键时，提示 The type or namespace name 'PostProcessing' does not exist in the namespace 'UnityEngine.Rendering'

**解决方法：**

经过查询资料得知出现上述问题的原因为 Unity 平台中没有安装 PostProcessing 组件。相应的解决方法为在 Unity 中点击 window->package manager->找到 Post Processing->install，之后即可正常进行展示效果。

(2) 添加完汽车物体后，为使主摄像机跟随运动中的汽车，需要添加镜头跟随脚本时，但在脚本添加完毕后点击 play，出现目标丢失的情况。

**解决方法：**

经过查阅相关资料以及检查代码内部逻辑后发现出现上述的问题的原因为摄像机跟随目标的脚本中，没有设置相机的初始变量，同时设定的目标跟随距离不合适。在脚本中增设初始变量并改变跟随距离，进行多次调试后，确定了合适的跟踪距离。最终该问题得到了解决。

(3) 在城市场景中添加完车辆物体，并附加车辆控制脚本后，在实际操作中发现：在没有按 w 键或 s 键时，仅按 a 键或 d 键，车辆会原地转向。这一问题并不符合现实生活中的汽车转向操作。

**解决方法：**

通过查阅脚本中的相关内容发现，出现上述问题的原因为脚本逻辑中，没有加入对

应的限定条件。原有的代码只是单独考虑了 `wasd` 四个键会产生的效果，而没有考虑不同键位组合使用的效果。简言之，即相关判断逻辑缺失。在重新安排逻辑结构并更改代码加入限定条件后，转向操作只能在按下 `w` 键或 `s` 键的行驶过程中才可以发生，问题得到解决。

（4）在添加完成所有脚本后，对车辆在各种运动情况下的状态进行测试，发现当车辆以较高的速度跨越障碍物时，车辆会有轻微的失重的现象。

#### **解决方法：**

经过查阅相关资料，重新检查代码内在判断逻辑以及分析为车辆配置的组件后发现，出现这一问题的原因是为车辆添加刚体组件后，为车辆添加的重量不够，因而导致车辆在速度过快时将会发生轻微的失重现象。在确定了问题原因后，经过多次测试，将车辆重量适当提高后，这一问题得到了解决。

## **五、实验总结**

本次实验是我第一次使用 Unity 环境来进行编程实现工作，在实验过程中遇到了很多问题，通过查找相关资料，向老师及同学寻求帮助后，这些问题基本到了解决。

（1）在实验的开始阶段，由于没有进行整体的规划，使得实验进展一度比较混乱，在之后的实验过程中我将注意实验初始阶段的规划环节。

（2）在具体实现的过程中，我也遇到了不少问题，例如演示操作的不顺利，各种效果与预期效果存在较大的偏差。通过不断地查找资料，与同学进行交流询问，这些问题基本得到了解决。同时我对于 Unity 也有了更进一步的了解。

（3）在实现各个控制脚本的时候，由于是第一次使用 C# 语言来实现需求，导致了该阶段初期在查找资料上耗费了一定的时间，同时由于对于 Unity 的不熟悉，这一过程进展不是很顺利。例如在车辆行驶过程中，虽然通过脚本实现了摄像机全程跟随车辆的效果，但是在车辆倒车时，摄像机的视角转换仍然有一些僵硬。另一个问题是，在摄像机跟随车辆的时候，视野会有小幅度的抖动，这一问题经过查找资料以及与同学交流后，虽然有所好转，但是没有得到彻底的解决。

最后，本次 Unity 实验让我对计算机图形学有了更深层次的认识，同时也让我对 Unity 有了一定的了解。在之后的学习生活中，我将充分利用所学到的知识，将其应用到日常的学习生活之中。