北京林业大学 2023--2024 学年第 2 学期实验任务书

课程名称：三维动画原理与制作　　开课学院：信息学院

考试班级：数媒 21-1、2　　命题人：杨　猛

实验环境：Unity 3D 2019 或以上　实验学时：4

实验题目(范围)：实验 3　基于物理的动画

请详细说明该设计的方案、内容、要求、进度等

严禁剽窃、抄袭等作弊行为！

实验目的：

1. 理解基于物理的动画的基础思想。
2. 了解物理引擎背后实现的原理。
3. 初步掌握刚体的平移和旋转。

实验内容：

1. Unity 3D 的动画编程方法，主要包括：
   （1） 熟悉 Unity 3D，导入新项目的方法。
   （2） 熟悉 Unity 3D 脚本编辑方法。
   （3） 重点熟练掌握 Unity 3D 的动画编程方法。
2. 依照“**实验方法**”中步骤依次进行实验操作，并给所添加/修改的每一句代码添加注释。

实验环境要求：

1. 实验环境：建议使用 Unity 3D 2019 或更高版本，可在 https://unity.cn/releases 路径下载。

实验方法：（注：以下演示步骤均在 Unity2019.4.17f1c1 环境中完成）

1. 导入 Unity 项目，初始化实验环境

本次实验不再从头开始创建项目，而是导入已有的项目文件进行代码的添加。打开 UnityHub，选择，新建项目，新项目->3D 核心模板->创建项目 （图 1）。

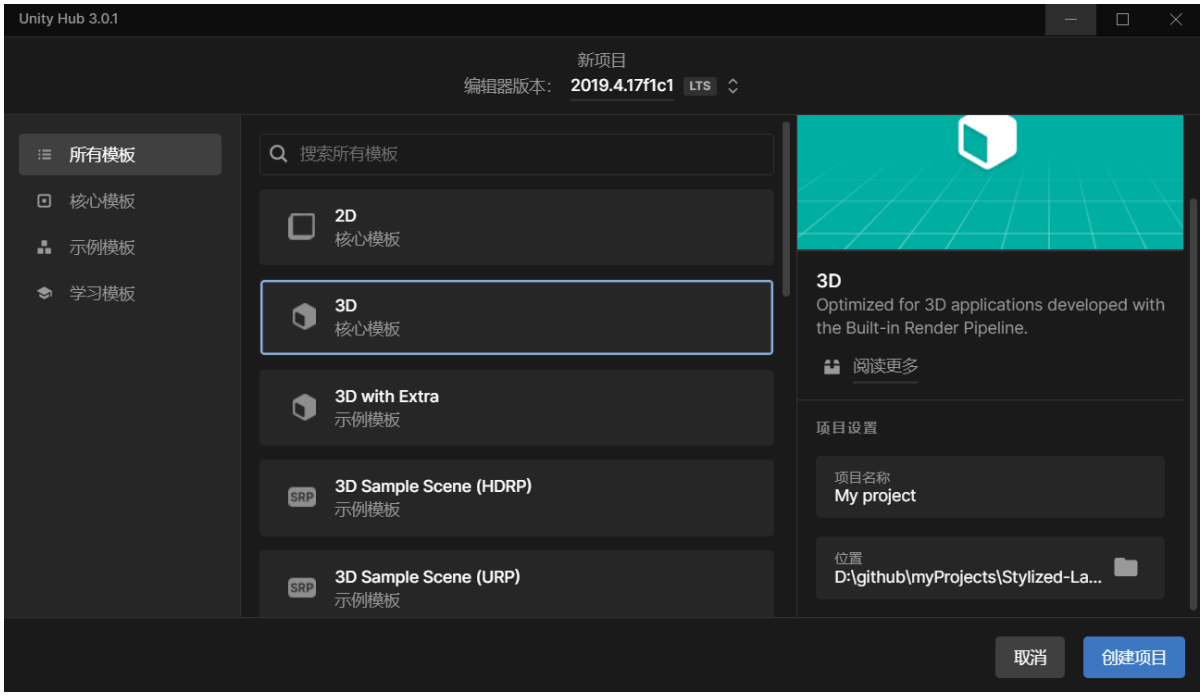

图 1 新建项目菜单

将 Assets 中的 Scenes 文件夹删除（如图 2 所示）。

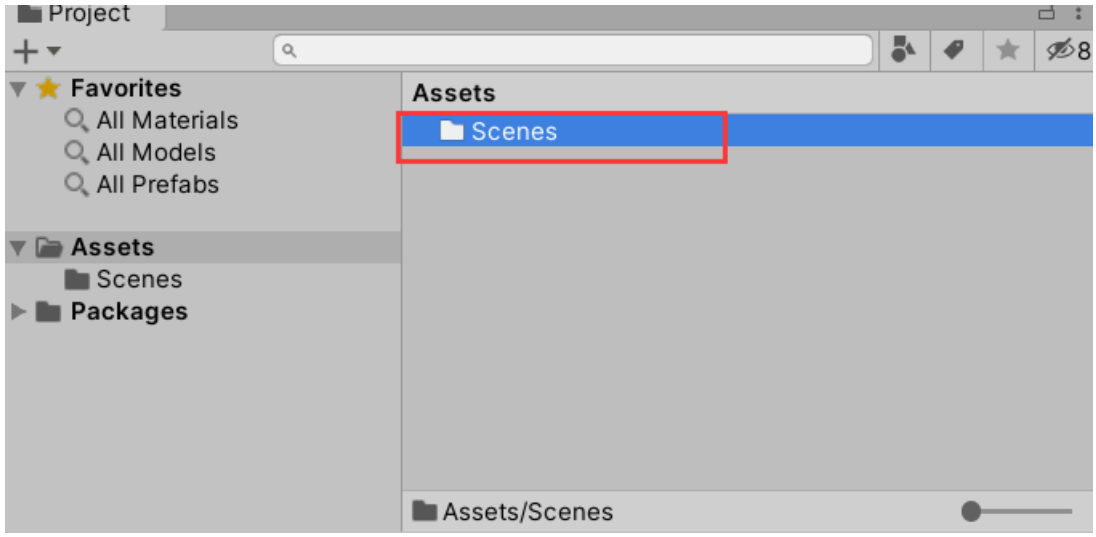

图 2 删除 Scene 文件夹

右键 Import Package->custom package 导入实验提供的项目文件包。

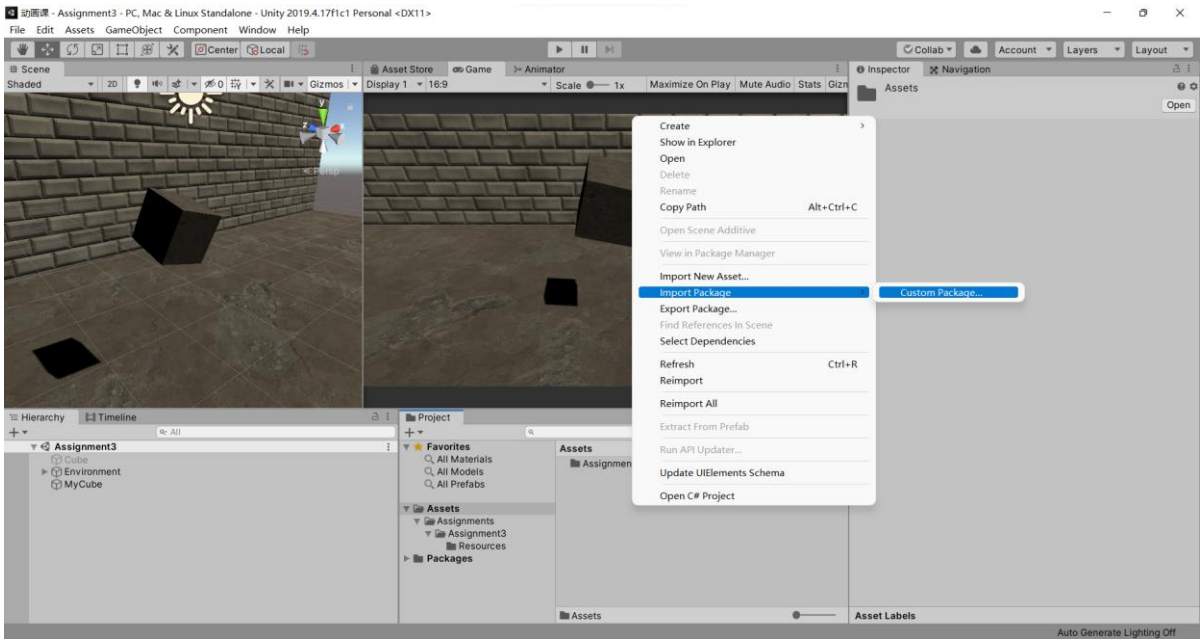

图 3 导入文件包

运行 Unity，使用小键盘的方向键和“r”键控制场景中的立方体移动和旋转。

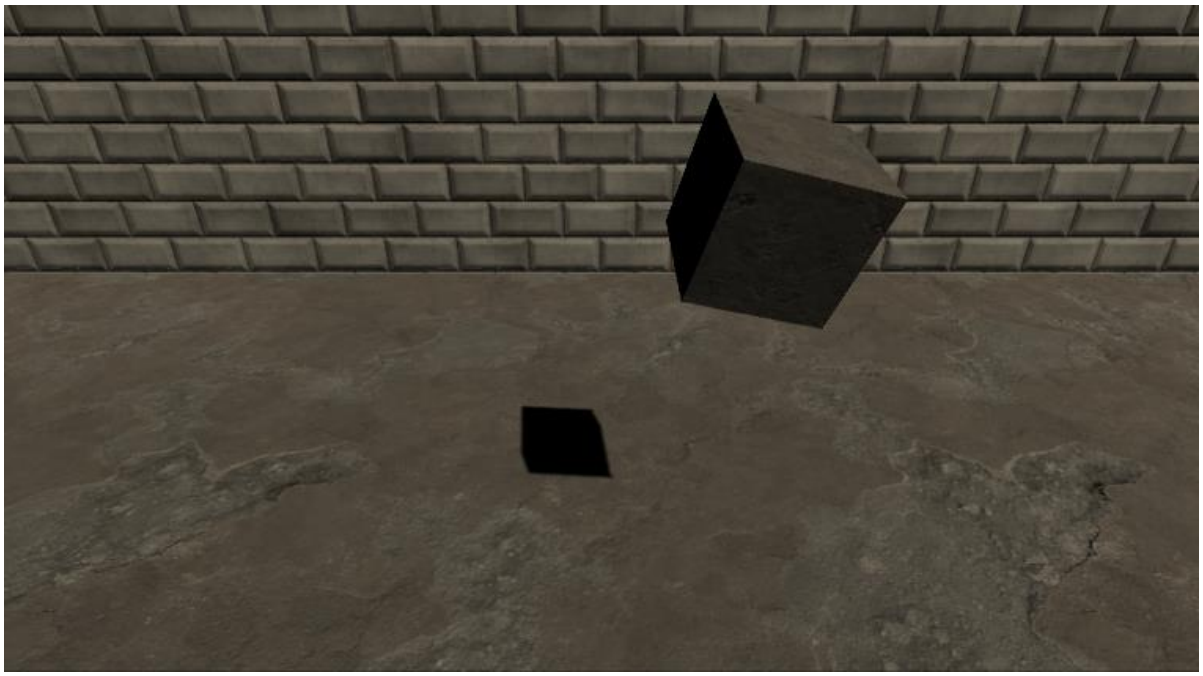

图 4 刚体运动

2. 观察刚体运动原理

打开当前立方体上的脚本 DefaultRigidbody，观察实现的原理。

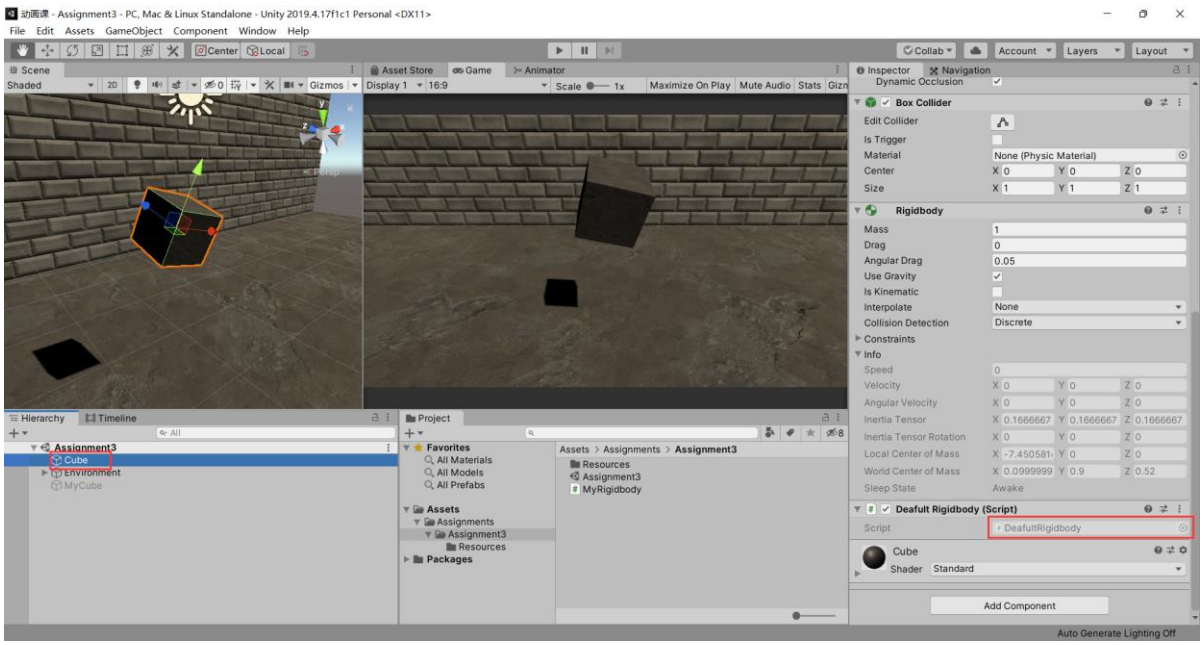

图 5 DefaultRigidbody 脚本

根据下图代码可以看到，当按键盘上的“r”时刚体组件会增加一个扭矩（Torque）扭矩是导致物体旋转的物理量，你可以从力的角度来理解他，如果说力会导致物体平移，那么扭矩就会增加物体的角速度让它旋转。

而按小键盘的方向键时，刚体组件会增加一个力，这就是立方体运动的原因，我们并没直接更改它的位置，而是为它添加力，然后产生加速度，进而改变速度，导致立方体位移。

```
// Update is called once per frame
void Update()
{
    if (Input.GetKey(KeyCode.R))
    {
        torque = new Vector3(Mathf.Sin(Time.time), Mathf.Sin(Time.time), Mathf.Sin(Time.time));
        _rigidbody.AddTorque(torque);
    }

    force= Vector3.zero;
    if (Input.GetKey(KeyCode.LeftArrow))
        force = new Vector3(-5,0, 0);
    if (Input.GetKey(KeyCode.RightArrow))
        force = new Vector3(5, 0, 0);
    if (Input.GetKey(KeyCode.UpArrow))
        force = new Vector3(0, 5, 0);
    if (Input.GetKey(KeyCode.DownArrow))
        force = new Vector3(0, -5, 0);
    _rigidbody.AddForce(force);
}
```

图 6 DefaultRigidbody 脚本控制移动的代码

3. 实现刚体运动脚本

在这部分我们要实现一个自己的刚体脚本

打开 Assets 中的 MyRigidbody 脚本，本次实验需要在脚本中进行代码的填空。

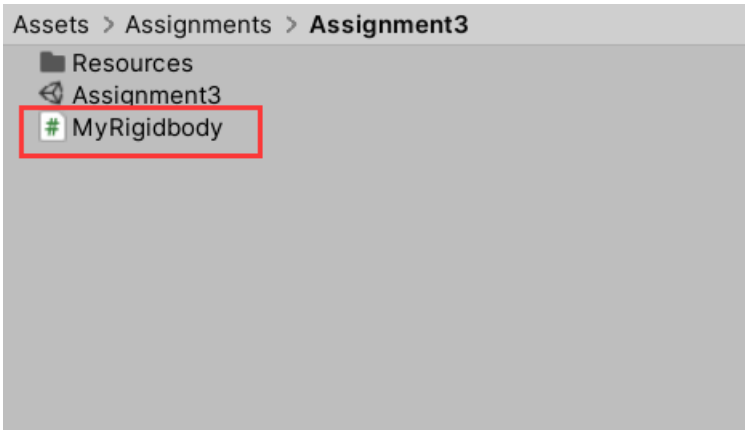

图 7 新建 Graph 对象

首先观察一下脚本中的变量，每一个变量的含义都在注释中有所标注。

```
[SerializeField] private float Mass = 1;                                    // 质量  Unchanged
[SerializeField] private float dt = 0.015f;                                 // 时间步长  Unchanged

private Vector3 torque;
private Vector3 force;

private Vector3 v           = new Vector3(x:0, y:0, z:0);                   // 速度
private Vector3 w           = new Vector3(x:0, y:0, z:0);                   // 角速度

private float mass_inv;                                                     // 质量的倒数
private Matrix4x4 I_ref;                                                    // 转动惯量

float linear_decay  = 0.98f;                                    // 速度衰减
float angular_decay = 0.98f;                                    // 角速度衰减
```

图 8 图表动画

再来看一下 void update（）函数，在这里，共有 5 段代码需要填写

```
void Update()
{
    torque = Vector3.zero;
    if (Input.GetKey(KeyCode.R))
    {
        torque = new Vector3(x:Mathf.Sin(Time.time), y:Mathf.Sin(Time.time), z:Mathf.Sin(Time.time));
    }

    force = Vector3.zero;
    if (Input.GetKey(KeyCode.LeftArrow))
        force += new Vector3(x:-5, y:0, z:0);
    if (Input.GetKey(KeyCode.RightArrow))
        force += new Vector3(x:5, y:0, z:0);
    if (Input.GetKey(KeyCode.UpArrow))
        force += new Vector3(x:0, y:5, z:0);
    if (Input.GetKey(KeyCode.DownArrow))
        force += new Vector3(x:0, y:-5, z:0);

    // TODO 更新速度

    // TODO 更新角速度

    // TODO 更新位置

    // TODO 更新方向

    // TODO 衰减速度及角速度

}
```

图 9 Update 函数

3.1 更新速度

首先更新速度，根据牛顿第二定律：

$$F = ma$$

根据实验代码中提供的 force 遍历，可以的简单的求得加速度 a，并且由此更新加速度

$$v = v_0 + a \cdot \triangle t$$

3.2 更新角速度

然后进行角速度的更新，角速度的更新相较速度来说要显得更加复杂一点，在这里，我们要引入几个新的定义。

τ 扭矩(Torque)：扭矩是导致物体旋转的物理量，你可以从力的角度来理解他，如果说力会导致物体平移，那么扭矩就会增加物体的角速度让它旋转。

I 转动惯量（Inertia）： 转动惯量是一个 3x3 的矩阵，是阻止物体运动状态变化的物理量，如果一个物体处于禁止状态，那么 Inertia 希望这个物体继续处于禁止状态，Inertia 的大小取决于物体的质量。你可以简单的把它理解为旋转的“质量”。

转动惯量的计算公式如下，**默认状态下的转动惯量已经在 void Start（）中进行了计算，当在 update 函数中更新角速度时，需要求出当前物体的旋转矩阵，然后根据**

下述公式求出当前时刻物体的转动惯量：

$$I_{ref} = \sum m_i(r_i^T r_i 1 - r_i r_i^T)$$
$$I = RI_{ref}R^T$$

其中，R 是物体当前的旋转矩阵，上述每一个定义都可以在搜索引擎中找到更加详尽的说明，在这里，我们只需要考虑如何使用他们，更新角速的方法与更新速度的方法相似

$$\omega = \omega_0 + \triangle t(I)^{-1}\tau$$

更新加速度和角速度后，我们进行位置和方向的更新：

3.3 位置更新

先进行相对简单的位置的更新，更新公式如下所示：

$$v^{[n+1]} = v^{[n]} + a^{[n]}\triangle t$$
$$x^{[n+1]} = x^{[n]} + v^{[n]}\triangle t$$

3.4 方向更新

更新方向相对复杂一点，我们用四元数来表示物体的方向。

四元数：四元数最早被提出用来描述 3D 空间中的一个点，并定义它的加减乘除运算，unity 脚本中的 transform.rotation 就是一个四元数

四元数表示为 q =[s, v]，其中，s 是标离，v 是一个三维向量，四元数的加减乘除定义如下：

$$aq = [as \quad av]$$
$$q_1 \pm q_2 = [s_1 \pm s_2 \quad v_1 \pm v_2]$$
$$q_1 \times q_2 = [s_1 s_2 - v_1 \cdot v_2 \quad s_1 v_2 + s_2 v_1 + v_1 \times v_2]$$
$$||q|| = \sqrt{s^2 + v \cdot v}$$

更新方向的公式如下，这里的 X 表示的是四元数的乘法。

$$q^{[n+1]} = q^{[n]} + [0 \quad \frac{\triangle t}{2}\omega^{[1]}] \times q^{[0]}$$

3.5 速度和角速度衰减

最后，进行速度和角速度的衰减，我们使用一种非常简单的方法来实现：

$$v^{n+1} = v^n * 0.98$$
$$\omega^{n+1} = \omega^n * 0.98$$

结论分析：

列举实验中遇到的问题、解决的方法，总结实验的收获和体会以及尚存在的问题。

实验要求：

1. 确认机器已经安装 Unity2019（或更高版本）；参见“**实验环境要求**”。
2. 根据“**实验方法**”中步骤依次实践计算机动画编程方法，并达到**熟练掌握**程度。
3. 给实验中添加/修改的代码逐句添加注释，可参考“参考书目”中教材或网络上资料。
4. 提交材料：本次实验(实验 3)需要填写实验报告并且提交*.Unitypackage文件。实验报告内容包括实验步骤、结果图、注释后的代码以及“**结论分析**”中所述内容等。实验报告必须是在 **Microsoft Office Word** 中进行格式排版后的报告。
5. 命名规则：学号_姓名，注意中间为“下划线”，**未按规则命名者按未交作业处理**。
6. 本课程**所有实验的报告**要求集成在一个文档中，并在课程结束后规定时间内提交即可，即课程结束前不必分开提交“实验 1”、“实验 2”等实验材料。
7. 提交地址：ftp://211.71.149.149/yang_meng/homework/相应目录下。

参考书目：

- 计算机动画算法与编程基础，雍俊海 著，2008.7 ，清华大学出版社。

教研室主任意见：

签字：_____ 年 月 日

学院负责人意见：

签字：_____ 年 月 日

注：此表一式两份，一份于考前交到考试中心，一份随学生课程设计材料上交学院。