

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 4 布料模拟

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. 了解游戏中布料背后实现的原理。
3. 初步掌握 PBD 首先布料的算法实现。

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）。

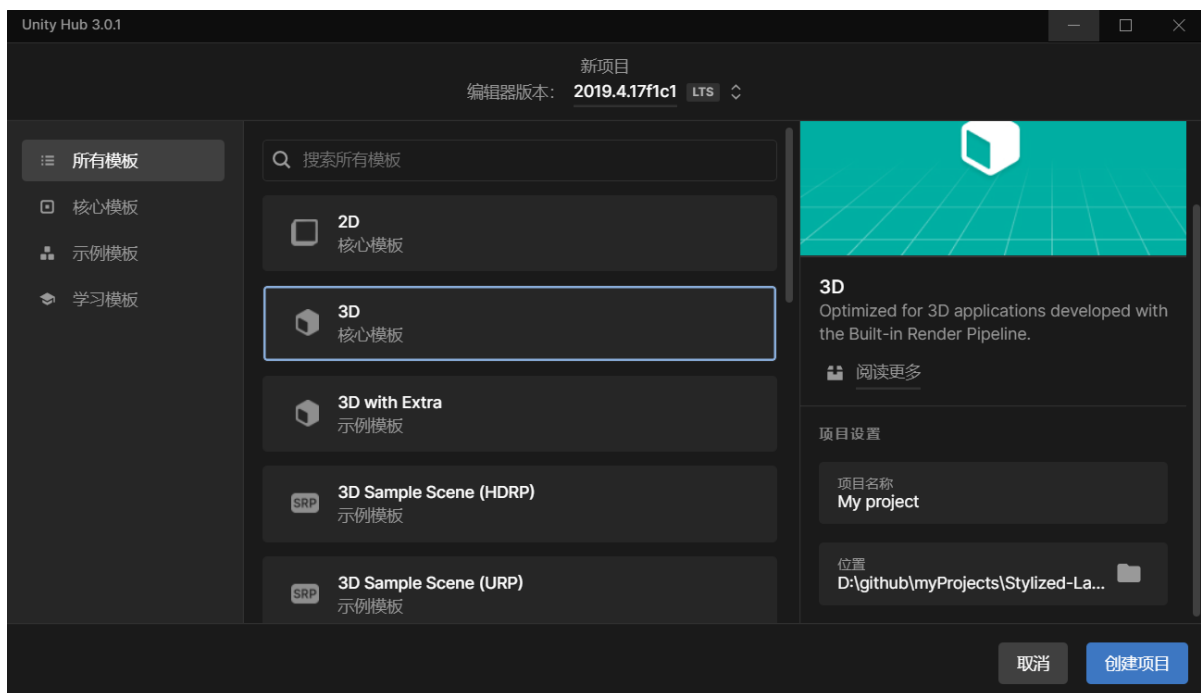


图 1 新建项目菜单

将 Assets 中的 Scenes 文件夹删除（如图 2 所示）。

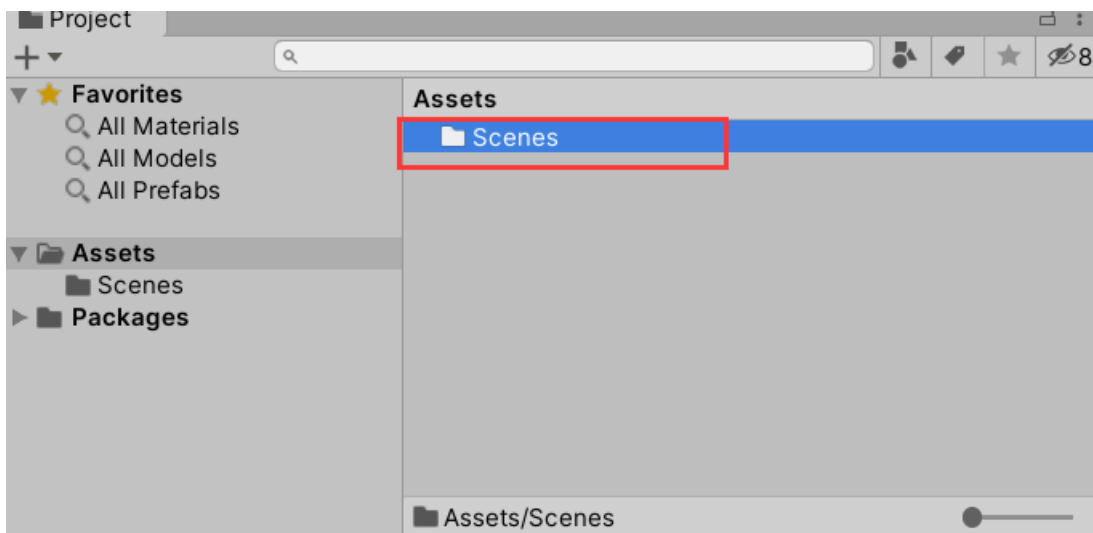


图 2 删除 Scene 文件夹

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

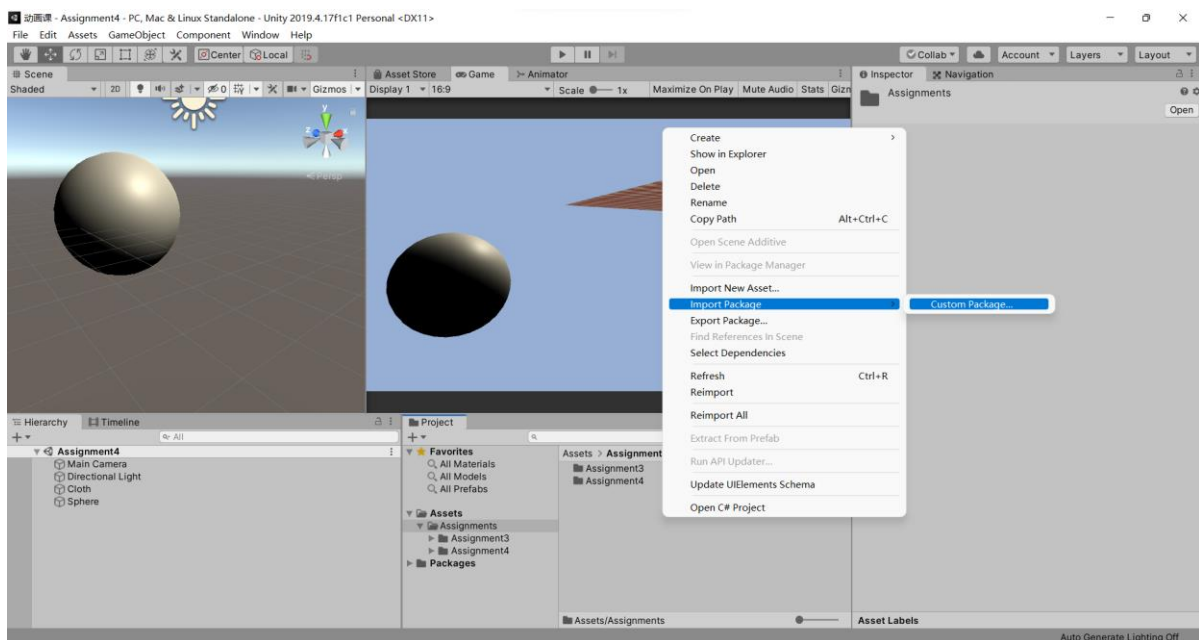


图 3 导入文件包

运行 Unity，你会发现布料掉了下去。

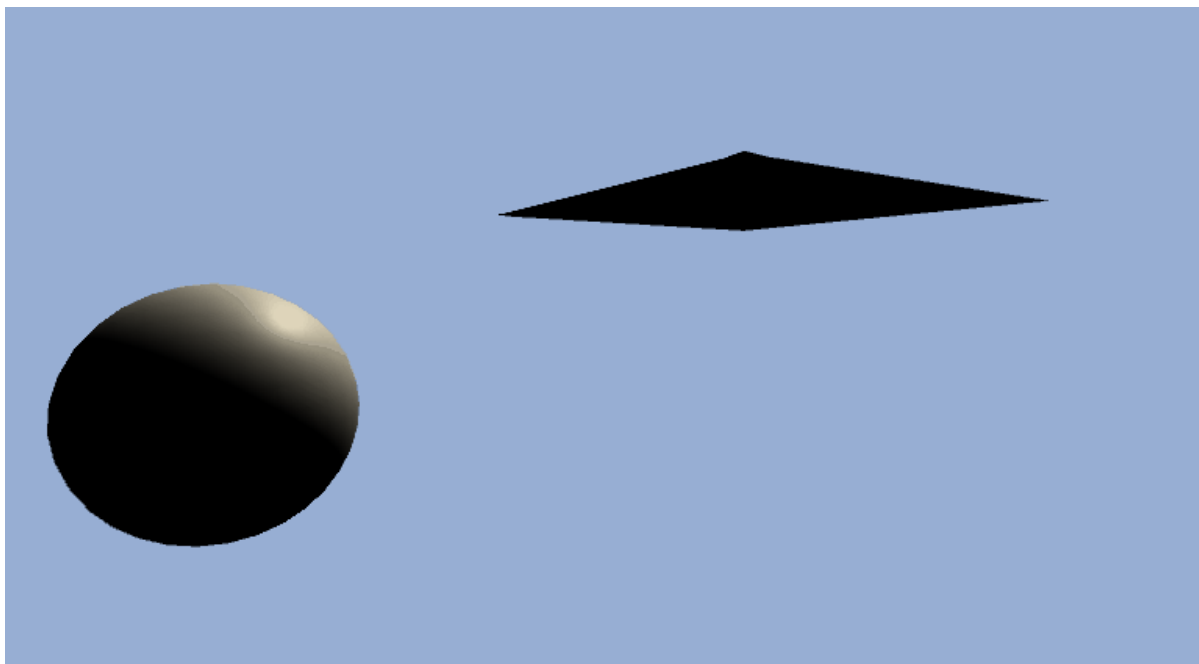


图 4 默认效果

2. 观察脚本属性

打开当前 cloth 上的脚本 PBD_model，观察其内部属性。

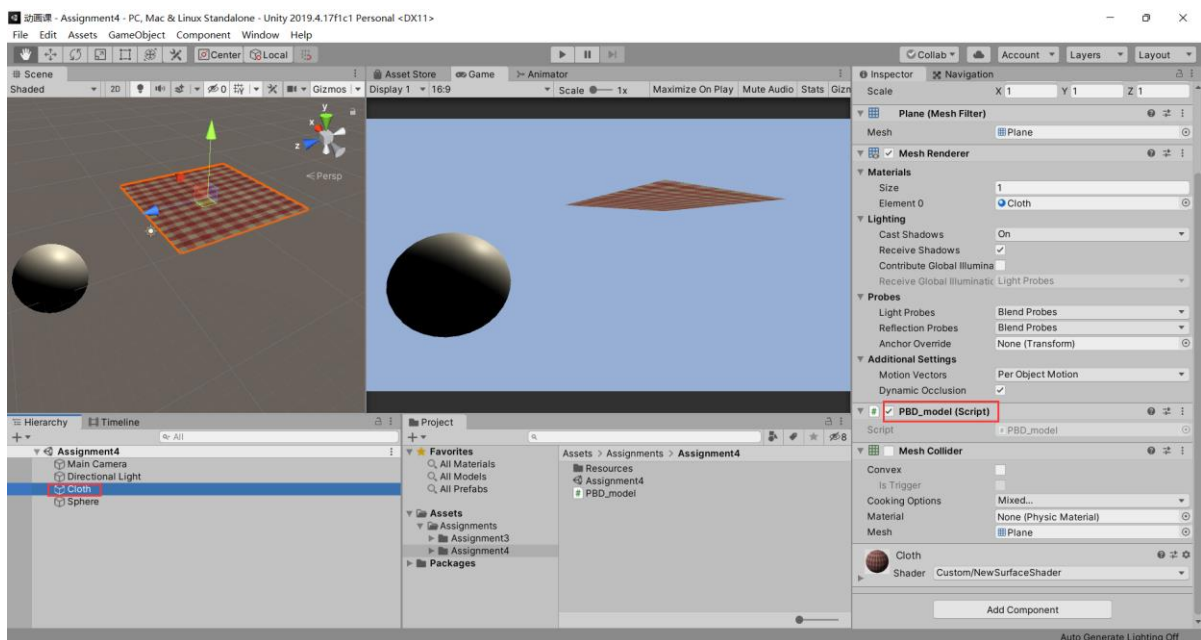


图 5 PBD_model 脚本

观察“PBD_model”脚本中的属性，如果你已经完成了实验 3，那么你应该能够理解这几个参数， t 是时间步长， inv_t 是时间步长的倒数， $damping$ 是速度衰减的参数，剩下三个数组 E 是模型边的关系， L 是每一条边的长度， V 是每一个顶点的位置，下图中红框标记的位置是初始化的函数，本次实验不要求掌握，你也可以查看各种属性初始化的方式，用来深入学习 PBD 布料模拟。

```
public class PBD_model: MonoBehaviour {

    float      t= 0.0333f;
    float      inv_t = 30.03f;
    float      damping= 0.99f;
    int[]      E;
    float[]    L;
    Vector3[]  V;

    MyRegion
```

图 6 PBD_model 中的属性

模型边的关系 E 的调用方式如下图所示， i 和 j 是每个边对应的两个顶点。

```
for (int e=0; e<E.Length/2; e++)
{
    int i = E[e*2+0];
    int j = E[e*2+1];
}
```

图 7 模型边的关系 E 的调用方式

每一条边的长度 L 的调用方式如下图所示：

```
for (int e=0; e<E.Length/2; e++)
{
    float length = L[e];
}
```

图 8 每一条边的长度 L 的调用方式

3. 实现 PBD 布料模拟

在这部分我们要实现一个布料脚本，本次实验只需要在两个部份填写代码即可。

```
// TODO 求出投影总位置
for (int e = 0; e < E.Length/2; e++)
{
}

// TODO 根据总位置求出平均位置，赋值给顶点，并根据位移变化求出速度
for (int i = 0; i < vertices.Length; i++)
{
}
```

图 9 需要填写代码的部份

求出投影总位置的基础思想是新生两个临时数组，分别用来存储顶点更新后的和位置 and 当前顶点连接的其他顶点的数量。

```
Vector3[] sum_x = new Vector3[vertices.Length];
int[] sum_n = new int[vertices.Length];
```

图 10 两个临时数组

在每次更新开始，将这两个数组设为 0，然后利用每一条边 e 连接的顶点 i, j 对这两个数组进行计算，他们的计算公式如下图所示：

$$\begin{aligned} \text{sum_x}_i &\leftarrow \text{sum_x}_i + \frac{1}{2} \left(\mathbf{x}_i + \mathbf{x}_j + L_e \frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|} \right), & \text{sum_n}_i &\leftarrow \text{sum_n}_i + 1, \\ \text{sum_x}_j &\leftarrow \text{sum_x}_j + \frac{1}{2} \left(\mathbf{x}_i + \mathbf{x}_j - L_e \frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|} \right), & \text{sum_n}_j &\leftarrow \text{sum_n}_j + 1. \end{aligned}$$

上述计算的步骤需要在第一个 TODO 的位置进行，你可以从找到到顶点的下标开始，如下图所示。

```
// TODO 求出投影总位置
for (int e = 0; e < E.Length/2; e++)
{
    int i = E[e*2];
    int j = E[e*2 + 1];
}
```

图 11 求得顶点下标

最后，更新每一个顶点的速度和位置，如下公式所示：

$$\mathbf{v}_i \leftarrow \mathbf{v}_i + \frac{1}{\Delta t} \left(\frac{0.2\mathbf{x}_i + \text{sum_}\mathbf{x}_i}{0.2 + \text{sum_}n_i} - \mathbf{x}_i \right), \quad \mathbf{x}_i \leftarrow \frac{0.2\mathbf{x}_i + \text{sum_}\mathbf{x}_i}{0.2 + \text{sum_}n_i}.$$

注意，在这个布料模型上，有两个顶点是固定的，这可以让我们更好的观察现象。

```
// TODO 根据总位置求出平均位置，赋值给顶点，并根据位移变化求出速度
for (int i = 0; i < vertices.Length; i++)
{
    if (i == 0 || i == 20) continue;
}
```

图 12 更新速度和位置

如果你的实现正确，你会看到下图的效果

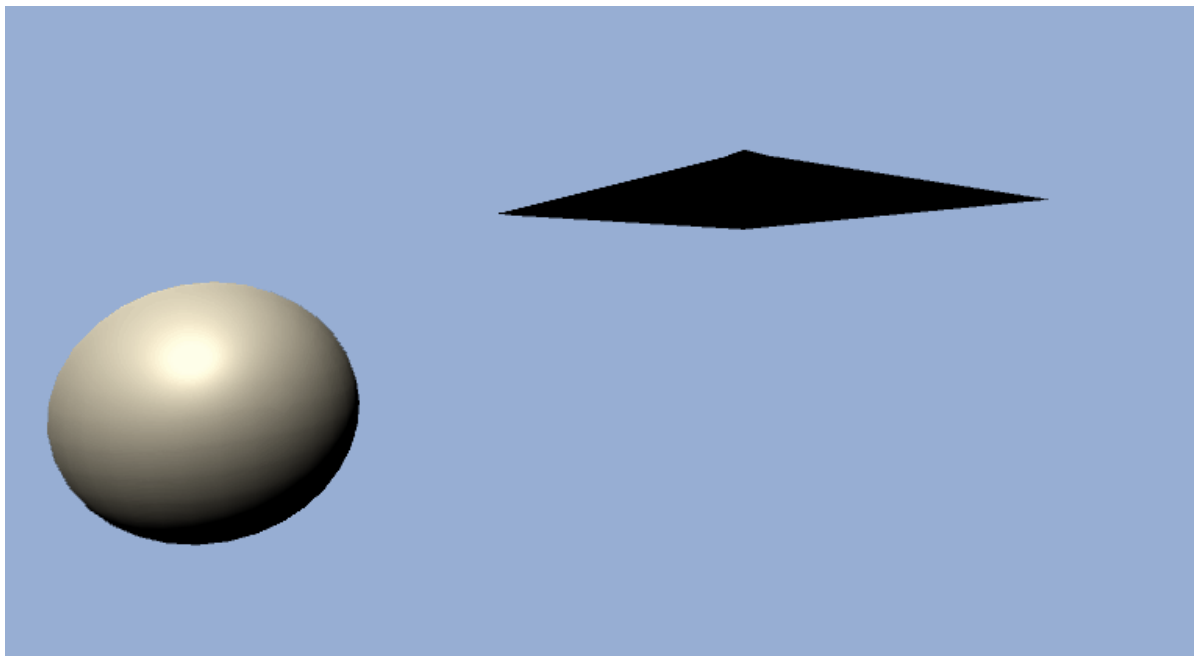


图 13 最终结果

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 4)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。