

hw7_AdPhysics

肖蔚尔 520030910314

1. 头发模拟

。 框架说明

- 算法部分在 HairSimulate.compute 文件中实现，具体要求见注释。
- 涉及 UI 交互部分在 HairController.cs 文件中实现，具体要求见注释。
- Head 运动部分在 HeadMove.cs 文件中实现（可自行进行扩充，如改变视角等），具体按键及对应功能如下：

↑前移 ↓后移 ←左移 →右移 J 上移 K 下移

- 部分参数定义：

```
//上一帧的顶点位置
RWStructuredBuffer<float3> prePositions;
//当前帧的顶点位置
RWStructuredBuffer<float3> positions;
//使用松弛法时迭代的次数
int iterations;
//阻尼系数
float damping;
//重量
float gravity;
//发丝数量
uint hairCount;
//每根头发节点数
uint nodeCount;
//每根头发节点的间距
float nodeDistance;
```

。 实现过程

所谓弹簧质点系统，其实就是仿真一些有质量的粒子（质点），再在粒子之间加入一些无质量的虚拟弹簧。例如要模拟一条绳子，最简单的方法是建立n个粒子，再在每两个连续的粒子之间加入弹簧，即有n-1个弹簧，如图2。



图2：用5个粒子和4个弹簧模拟的绳子

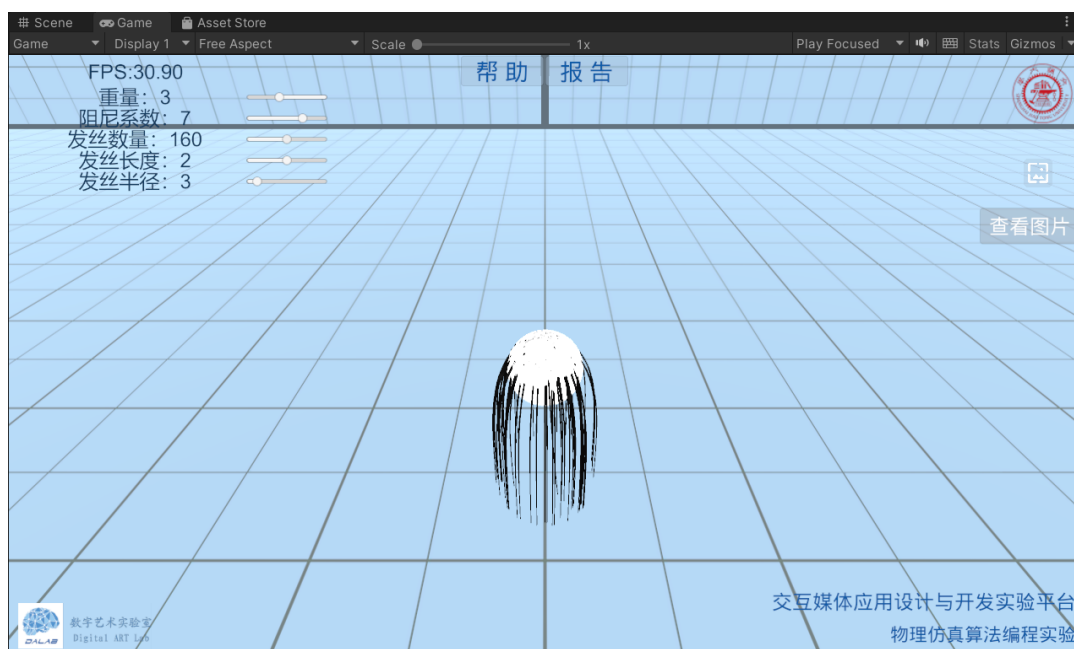
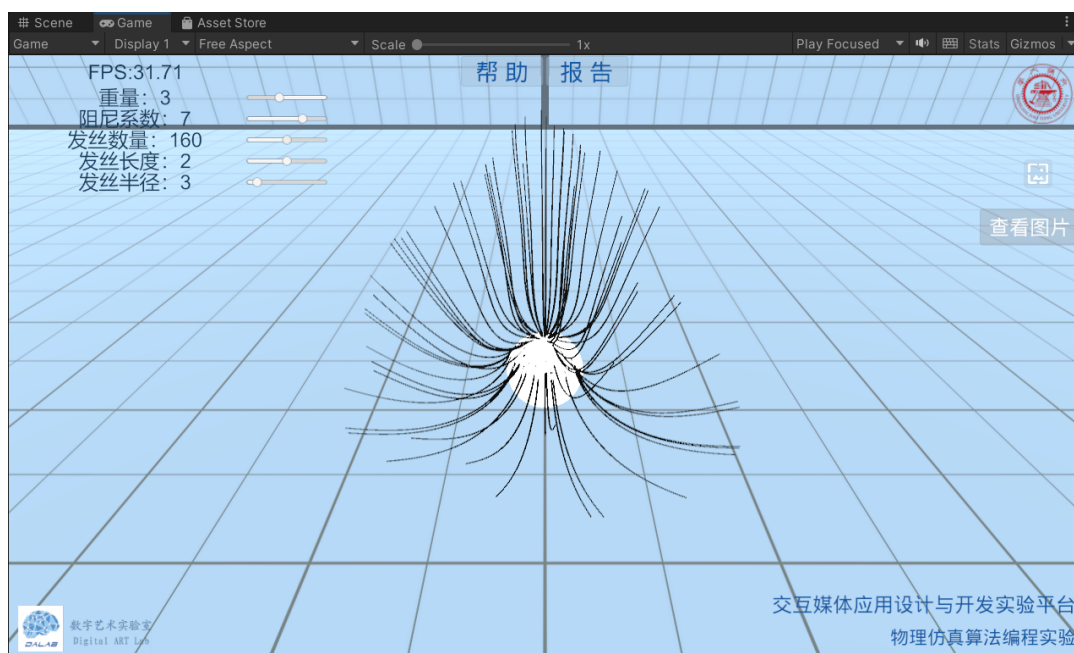
要模拟粒子运动，可使用《[用JavaScript玩转游戏物理（一）运动学模拟与粒子系统](#)》一文中谈及的欧拉方法（Euler method），但[1]里介绍的Verlet数值积分在很多情况下是更好的选择。我使用了含简单阻尼效果的Verlet数值积分方程：

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + d \cdot (\mathbf{x}(t) - \mathbf{x}(t - \Delta t)) + \mathbf{a}(t)\Delta t^2$$

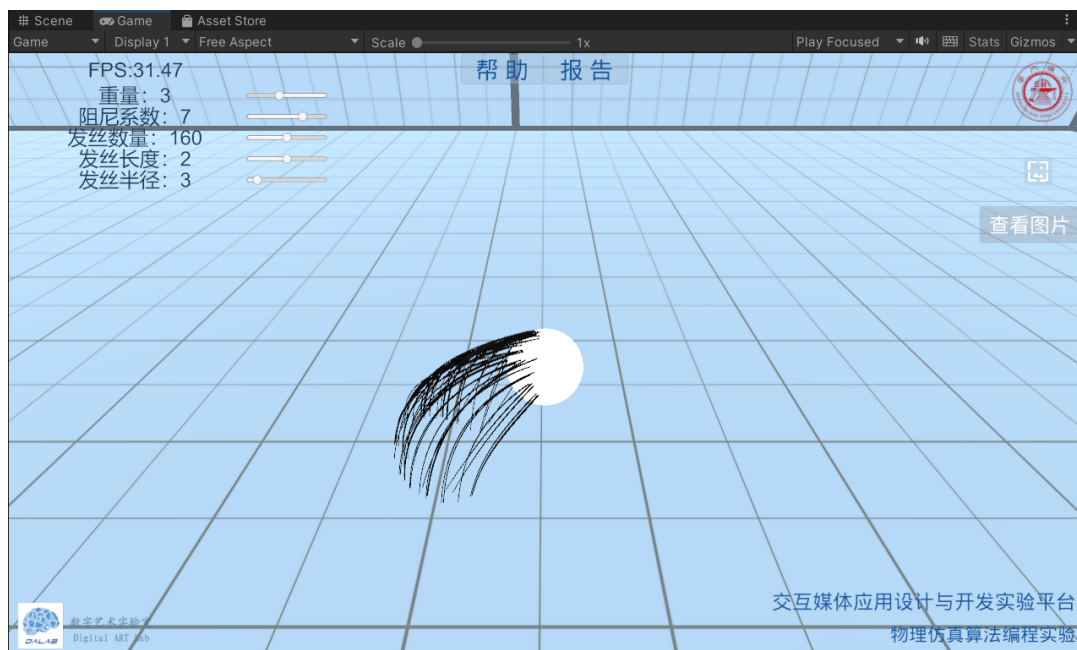
Verlet积分的另一特点，是可以简单地加入各种约束（constraint），例如某粒子在仿真之后，其位置位于地面以下，只需把粒子移至最近地面的点。对于绳子，另一约束就是相邻粒子的距离，在Verlet积分下，此距离约束可以模拟弹簧。假设两个相邻粒子的位置为 x_1 、 x_2 ，两者间的止动长度（rest length）为 l_r ，则可以这样调节两粒子的位置：

$$\mathbf{x}'_1 = \mathbf{x}_1 + (\mathbf{x}_2 - \mathbf{x}_1) \cdot \frac{\|\mathbf{x}_2 - \mathbf{x}_1\| - l_r}{2 \|\mathbf{x}_2 - \mathbf{x}_1\|}$$
$$\mathbf{x}'_2 = \mathbf{x}_2 - (\mathbf{x}_2 - \mathbf{x}_1) \cdot \frac{\|\mathbf{x}_2 - \mathbf{x}_1\| - l_r}{2 \|\mathbf{x}_2 - \mathbf{x}_1\|}$$

。效果展示



正在右移：



2. 布料模拟

。 框架说明

- 算法部分在 ClothSimulate.compute 文件中实现，具体要求见注释。
- 部分参数定义：

```
//顶点位置
RWStructuredBuffer<float3> *positions*;
//顶点速度
RWStructuredBuffer<float3> *velocities*;
//x表示横向顶点数量,y表示纵向顶点数量,z = x * y
uint3 *size*;
//弹性系数.xyz分别对应结构弹簧、剪力弹簧、弯曲弹簧
float3 springs;
//弹簧原长.xyz分别对应结构弹簧、剪力弹簧、弯曲弹簧
uniform float3 restLengths;
```

- 获取顶点速度和位置的函数接口

```
static float3 getVelocity(uint index);
static float3 getPosition(uint index);
```

- 更新顶点速度和位置的函数接口

```
static void setVelocity(uint index, float3 vel);
static void setPosition(uint index, float3 pos);
```

。 实现过程

网格弹簧质点模型

目前使用较多的布料建模方法，是将布料视作一个由多个质点构成的网格，质点之间由弹簧相连。在每个时间步中，需要计算每个质点所受的合力，进而更新速度和位置。本次作业中需要考虑的力包括重力、弹性力和阻尼力。

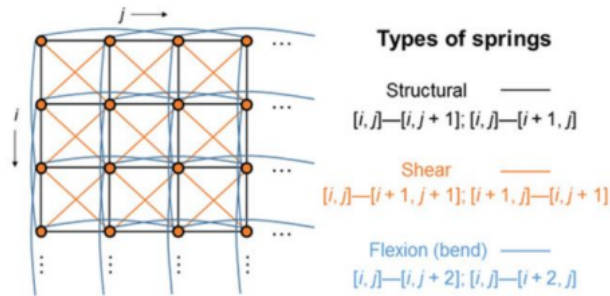
受力计算

1. 重力

每个质点都会受到重力影响，计算公式为： $F_g = mg$

2. 弹性力

质点之间的弹簧连接可以分为三类，具体分布如下图所示：



以上三种连接分别用以模拟材料中的三种力：

- 上下左右相邻质点连接，模拟结构力(Structural)，如拉伸和收缩。
- 对角线连接，模拟剪力(Shear)。
- 上下左右跨一个质点连接，用以模拟材料弯曲的力(Flexion)。

弹簧的力学模型满足胡克定律，即： $F_s = -K_s \Delta x$

其中 Δx 为弹簧的伸缩量， K_s 为弹性系数，对一根弹簧来说，一般为常数。

3. 阻尼力

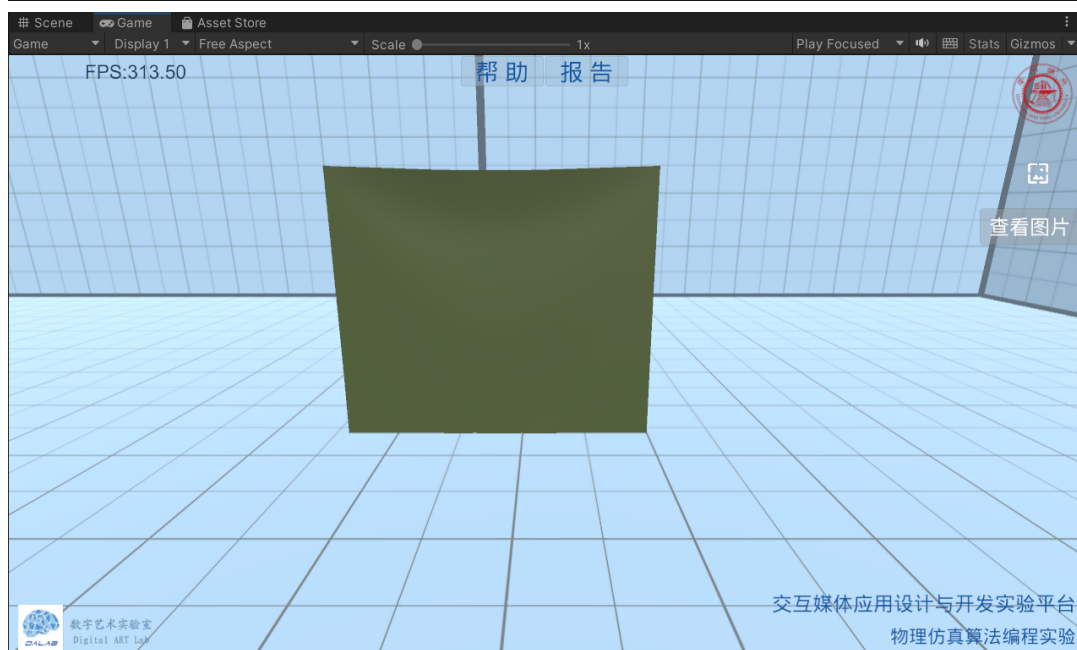
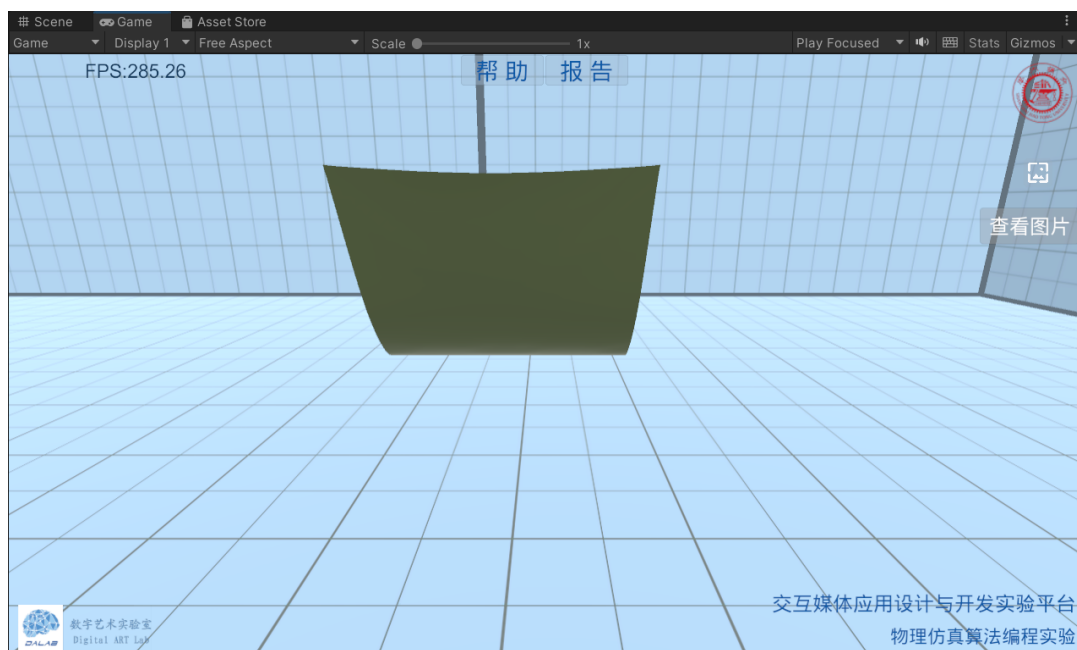
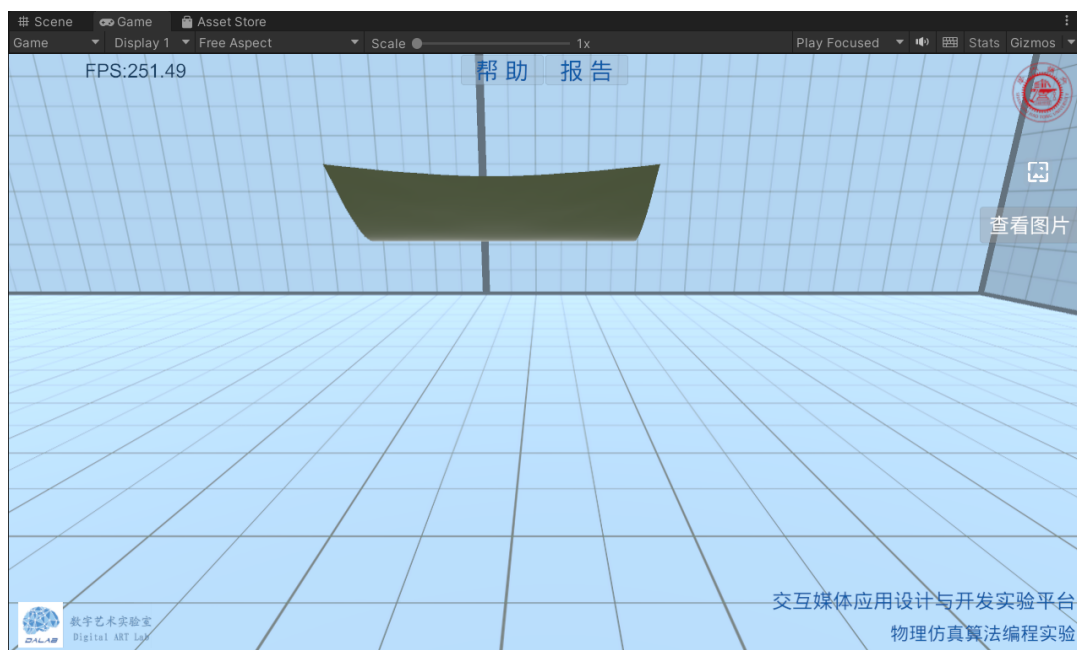
质点在运动的时候，通常是有能量耗散的。如果没有耗散，一个弹簧将会永远震动下去，与现实情况不符。因此需要加入阻尼力来模拟这种耗散，公式为： $F_d = -C_d v$

其中 C_d 为阻尼系数， v 为运动速度。

Semi-implicit Euler(又名 Symplectic Euler)积分

1. 将上述几种力加和可以得到每个质点所受的合力 F
2. 根据牛顿第二定律得到质点的加速度 $a = F/m$
3. 更新质点的速度 $v_{t+1} = v_t + a_t \Delta t$
4. 更新质点的位置 $x_{t+1} = x_t + v_{t+1} \Delta t$

。效果展示



3. 参考文档

- <http://miloyip.com/2011/alice-hair/>