

高级 Physics 原理及框架介绍

头发模拟

弹簧质点模型

参考[爱丽丝的发丝](#)。

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;
```

布料模拟

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

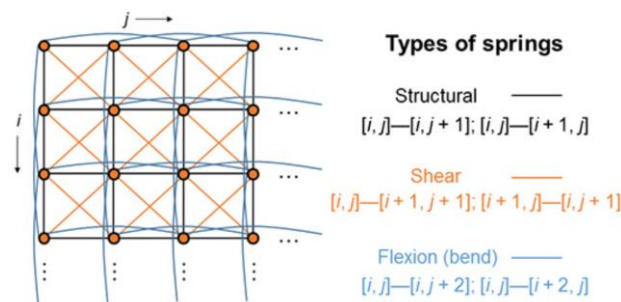
1. 重力

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 $a = \frac{F}{m}$
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 $x_{t+1} = x_t + v_{t+1} \Delta t$

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 springKs;
//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);
```