

大作业报告

我选择的主题是 D. Simulation 中的 1. Position Based Dynamics，根据网上的资料实现了PBD算法。

不同于复杂的隐式欧拉积分，PBD方法更加简单，也不需要很多的物理的力的模拟，只用通过一些约束就可以实现整个仿真的物理模型。

算法流程主要是通过弹簧长度不变这个约束，先通过显式欧拉积分，算出一小段时间内物体的位移，根据约束，调整各个质点的位置，具体来说对于连接 i, j 两个点的弹簧，

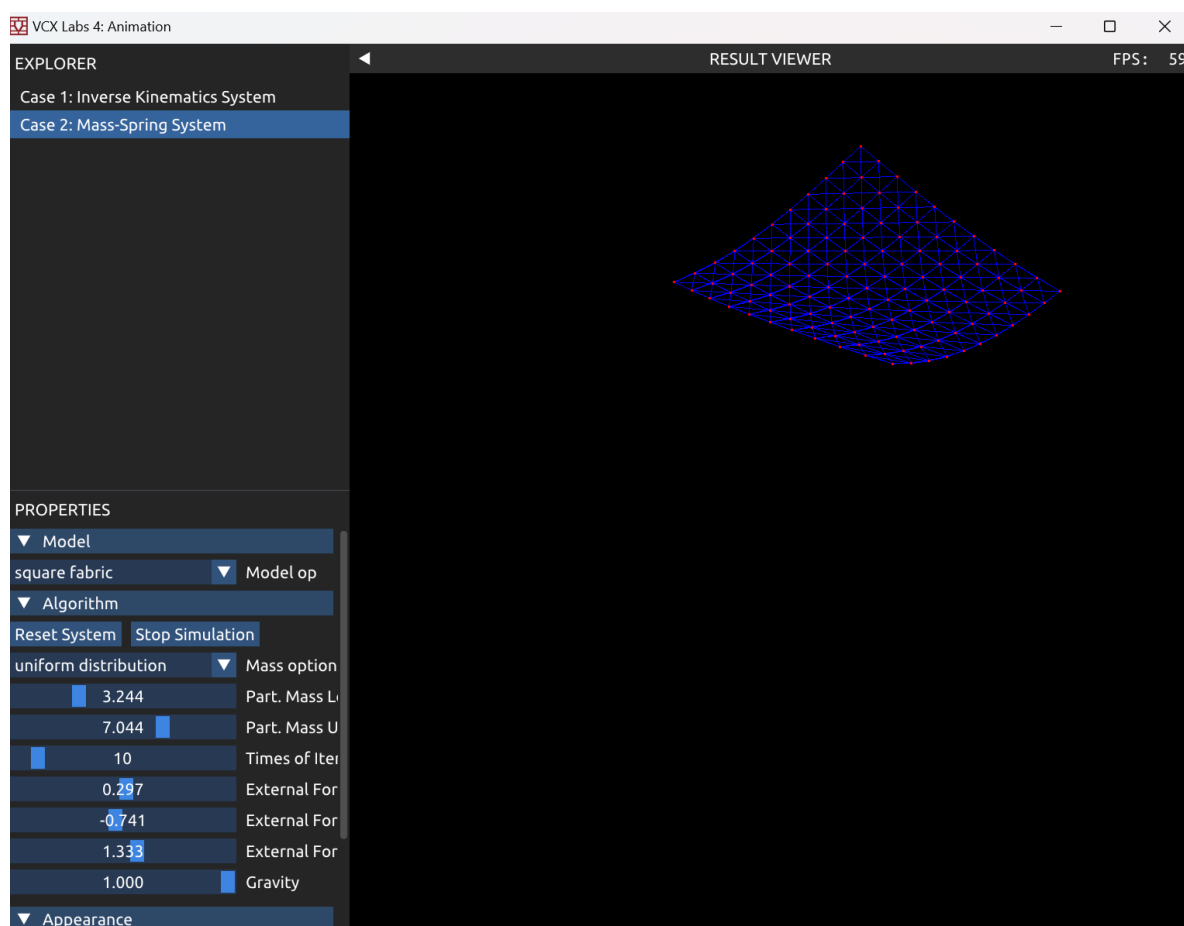
$$x'_i \leftarrow x_i - \frac{m_j}{m_i + m_j} (|x_i - x_j| - L) \frac{x_i - x_j}{|x_i - x_j|}$$

$$x'_j \leftarrow x_j + \frac{m_i}{m_i + m_j} (|x_i - x_j| - L) \frac{x_i - x_j}{|x_i - x_j|}$$

这样通过若干次迭代，每次迭代依次调整每个弹簧所对应质点的位置，最后区域一个比较稳定的状态。

其中迭代次数越少，显示出来的弹性越强；迭代次数越多，显示出来的弹性越弱。

基于lab4的代码的实现，并更改了部分交互的button，以下为实现结果。



修改部分如下：

1. 增加了一个模型，可以在Model op中选择布料模型或者类似于一个蹦床的模型。
2. 可以选择每个点的质量是相同的，或者是一个范围内的均匀分布。
3. 可以选择迭代的次数，来实现不同的弹性功能。
4. 可以选择外力（风力）的大小，来更好的观察模拟的效果。

VCX Labs 4: Animation

EXPLORER

Case 1: Inverse Kinematics System

Case 2: Mass-Spring System

RESULT VIEWER

FPS: 57

Model

trampoline

Algorithm

Reset System

Stop Simulation

same mass

1.000

10

2.370

0.519

1.259

1.000

Appearance

3.444

Part. Mass

Times of Iter

External For

External For

External For

Gravity

Part. Size

