

# 项目申请书

项目名称：为 SimplePhysicsEngine 添加软体仿真

项目主导师：young

申请人：黄长鑫

日期：2024年6月4日

邮箱：[changxinhuang@zju.edu.cn](mailto:changxinhuang@zju.edu.cn)

## 1 项目背景

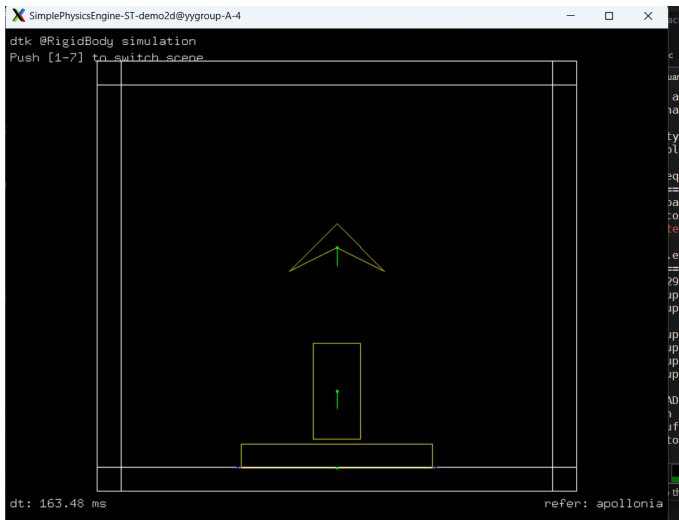
### 1.1 项目基本需求

项目仓库：<https://github.com/Simple-XX/SimplePhysicsEngine>

项目要求：

1. 加载mesh 实现软体仿真
2. 弹簧质点模型
3. 实现基本demo
4. 添加基本测试
5. 尝试使用并行加速

### 1.2 SimplePhysicsEngine编译运行



### 1.3 个人能力相关

1. 图形与几何处理基础：自从我读研快一年以来，所做的多个横向项目就是借助CGAL库对建筑CAD图纸设计几何处理算法，因此我对CGAL库及其文档是非常熟悉的；此外，我还参与了腾讯的四角网格简化课题；而我自身所参与的一个科研课题，也是对于牙齿的CBCT与Mesh模型的配准融合。

虽然我之前没有过物理模拟仿真方面相关的基础，但是这也是我想要参与该项目的动力之一，我希望借助该项目对物理模拟仿真领域探知一二。

## 2 算法选型

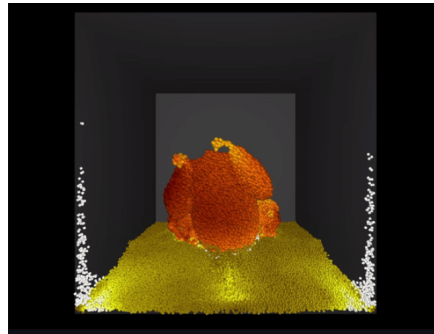
### 2.1 Finite Element Method(FEM)

有限元法将物体分割成有限数量的小单元，通过求解偏微分方程来模拟物体的弹性、塑性等特性。它的优势是高精度，适用于复杂几何形状和大变形，但缺陷是实现复杂，计算量大，且需要对物体进行网格划分，前处理工作量大。

### 2.2 Material Point Method(MPM)

物质点法是一种混合的拉格朗日-欧拉数值方法，适用于模拟大变形和复杂材料行为。MPM算法相较于FEM和FDM，它的核心优势是对大变形或者有复杂拓扑结构变换的物理仿真，且视觉效果十分出色，但实现复杂，计算成本较高。

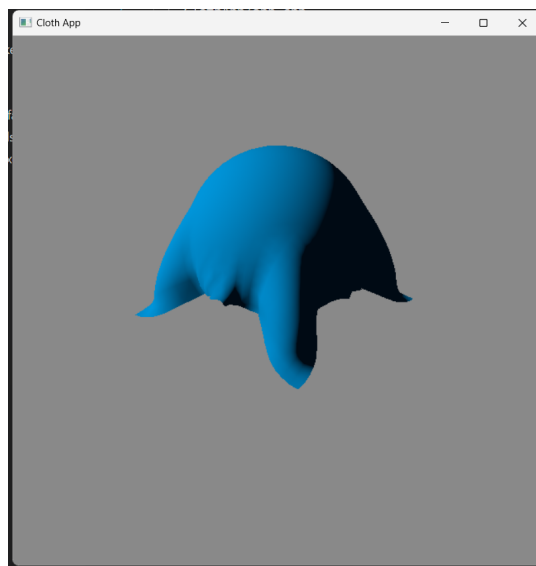
- <https://github.com/chetan-parthiban/WebGPUMPM.git>演示



### 2.3 Mass-Spring Systems(MSM)

相较于前面的几种方法，质点弹簧系统是一种**简单且直观**的方法，将物体建模为由质点（点质量）和弹簧（连接质点的弹性连接）组成的网络。其优势就是实现简单、计算速度快，但是对复杂变形的精度较低以及难以处理大变形和复杂接触

- <https://github.com/sam007961/FastMassSpring.git>编译运行结果

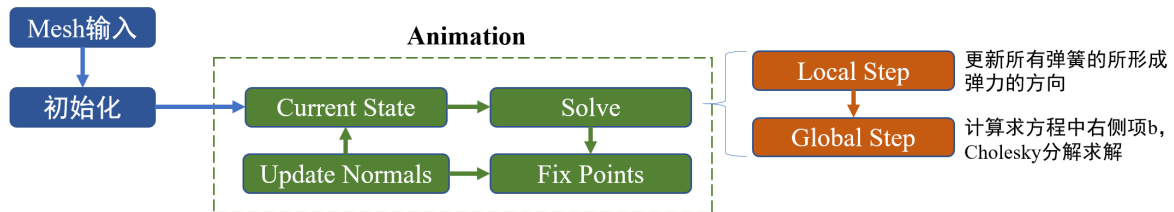


## 3 项目详细方案

### 3.1 MassSpringSolver设计

为了保证项目按时完成，整个MSM求解器的写法还是以**Fast simulation of mass-spring systems**[1]这篇论文中的方法为基础。若能够提前完成项目目标，也可以进一步使用更先进更复杂的写法。

整个项目流程可以简单抽象为 传入mesh->MassSpringSystem将mesh转换为一个弹簧质点模型系统->MassSpringSolver求解整个系统



1. 首先输入mesh，初始化Mass-Spring System（`dtkPhysMassSpring`类）
2. 在每一帧的模拟中，首先获取目前的状态，通过求解Local中所有弹力的方向，再在Global中依据论文中Cholesky分解的方法求解整个系统
3. 在Fix Points中执行满足约束，防止物体穿透
4. 最后更新法向量进入下一个状态求解

### 3.2 并行加速设计

- `dtkPhysMassSpring.h` 中的 `dtkPhysMassSpring` 类提供了使用OpenCL来并行加速弹簧-质点系统的计算，包含了系统的初始化、数据传输、内核执行和结果回传等操作
- `dtkPhysMassSpring` 类中整体设计思路如下，后期我也将测试后调整相应的并行设计思路：
  1. **数据和计算分离**：将质点和弹簧的数据分离，并为它们创建专用的内存缓冲区。
  2. **初始化OpenCL环境**：获取平台和设备信息，创建上下文和命令队列，编译内核程序。
  3. **创建和管理缓冲区**：为质点和弹簧数据创建缓冲区，并在主机和设备之间传输数据。
  4. **设置内核参数和执行内核**：设置内核参数，执行并行计算内核。
  5. **数据传输和结果回传**：在计算前后传输数据，确保主机和设备之间的数据一致性。
  6. **分步骤进行计算**：将计算过程分成多个步骤，逐步进行预处理、内核执行和后处理。
  7. **错误检查和事件管理**：确保每个OpenCL API调用成功完成，并正确管理事件。

### 3.3 SimplePhysicsEngine相关

- `dtkPhysMassPoint.h`：定义了引擎中的质点类
  - `dtkPhysMassPoint` 类中提供了更新位置、速度、加速度，设置和获取质点属性，处理冲量、力和碰撞等功能
- `dtkPhysSpring.h`：定义了引擎中的弹簧类
  - `dtkPhysSpring` 类提供了弹簧的属性（如刚度、阻尼、原始长度等）和方法（如更新弹簧状态、设置和获取属性等）
- `dtkPhysMassSpring.h`：定义了引擎中的质点-弹簧系统类

- `dtkPhysMassSpring` 类提供了设置和获取质点、弹簧及其属性的功能，还更新质点和弹簧的状态，包括位置、速度、加速度等，还提供了用于OpenCL相关的并行计算，加速质点-弹簧系统的计算
- `dtkPhysMassSpringCollisionResponse.h`：定义了处理质点-弹簧系统碰撞响应的类
  - `dtkPhysMassSpringCollisionResponse` 类提供了包括碰撞响应的类型、添加和移除质点-弹簧系统、更新碰撞响应状态以及获取刺穿结果等功能，用于模拟和处理物理系统中质点-弹簧之间的碰撞响应。
- `dtkPhysMassSpringThread.h`：定义了螺纹弹簧类
- `dtkPhysMassSpringThreadCollisionResponse.h`：定义处理螺纹弹簧碰撞响应类

## 4 项目开发时间计划

---

### 4.1 项目研发第一阶段（07 月 01 日 - 08 月 15 日）：

- ☐ 完成弹簧质点模型框架的搭建
- ☐ 加载布料模型
- ☐ 添加基本的测试：施加不同的力，与不同的物体碰撞
- ☐ 借助引擎实现并行加速

### 4.2 项目研发第二阶段（08 月 16 日 - 09 月 30 日）：

- ☐ 加载Deformation Soild模型
- ☐ 将以上的布料模型、Deformation Soild模型、以及相关的并行加速结果都设定成多个scene，整合到gui中通过按键进行switch
- ☐ 对第一阶段的完成内容进行总结，并输出相关文档内容
- ☐ 思考可以改进或者补充的地方

## 参考资料

---

[1] Liu, T., Bargteil, A. W., Obrien, J. F., & Kavan, L. (2013). Fast simulation of mass-spring systems. *ACM Transactions on Graphics*, 32(6), 1-7. doi:10.1145/2508363.2508406

[2] <https://github.com/kuiwuchn/GPUMPM.git>

[3] <https://github.com/chetan-parthiban/WebGPUMPM.git>