

CollisionDetection

环境说明

- Windows11
- Visual Studio 2022
- GLU工具库 1.2.2.0 Microsoft Corporation
- CUDA 12.0

VS中运行可能需要管理NuGet程序包，详见[VS2022的openGL环境搭建（完整篇） - 知乎\(zhihu.com\)](#)

程序模块

文件	作用
Point.h	Point类，实际可以作为一个向量使用
Wall.h Wall.cpp	Wall类，用于构造碰撞中的6面墙
Camera.h Camera.cpp	Camera类，实现相机环游
Collision.cu Collision.cuh	gpu上碰撞检测的核心方法实现于此
Ball.h Ball.cpp	Ball类，包含小球的位置、速度、质量、弹性系数等信息
main.cpp	程序入口，并负责图形界面的绘制

运行流程

程序从main函数进入，调用图形界面和相机、墙、小球等对象的初始化方法进行初始化，并绑定glut提供的一些用于交互和更新的回调函数。在glut的循环事件中，每次循环都会调用gpu上的collisionDetection函数，进行碰撞检测并更新碰撞后小球的位置和状态。

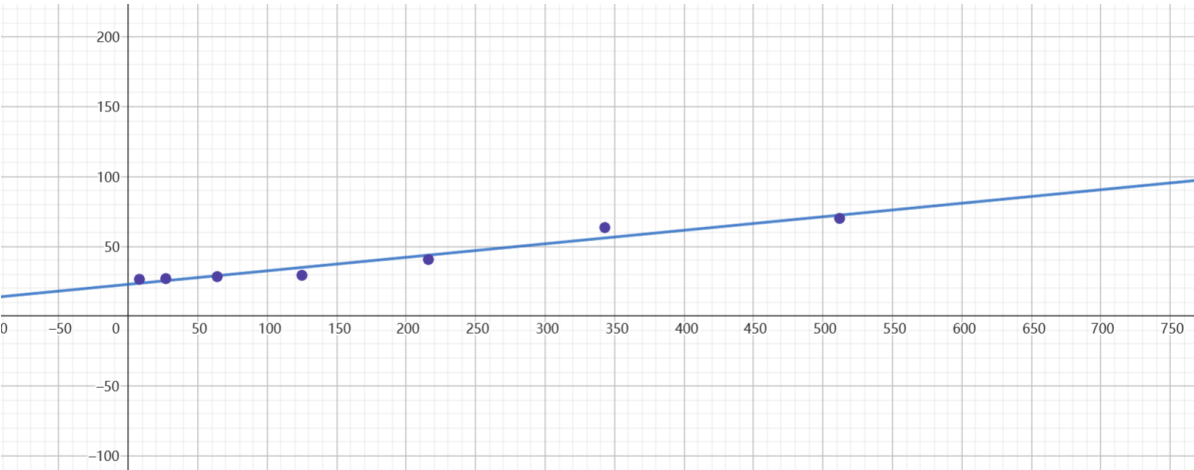
碰撞检测实现方法正如中期报告中叙述，首先构建两个数组存储球体id和cell id。对每个球体进行hashing，得到的hash值存在cell id数组中作为每个物体的第一个cell id，该cell也是物体的H cell，同时更新球体id数组。接下来计算每个球体的P cell并将其id更新到cell id数组中。理论上来说，每个球体最多可以拥有8个cell id，其中一个为H cell，其余的为P cell。然后对数组内容进行排序。由于希望排序后H cell在P cell前面，因此需要采用一种稳定排序，这里选择基数排序。对排序好后的cell id数组进行碰撞检测即可。

算法性能

记录了30*30*30的封闭空间内1000次渲染所耗时间

小球个数(个)	时间(s)
8	26.4242

小球个数(个)	时间(s)
27	26.929
64	28.3534
125	29.3036
216	40.6008
343	63.4627
512	70.045



可以看到，算法的性能大致保持在线性的时间内，这也比较符合中期报告中分析的算法复杂度。

使用方法

成功运行程序后，键盘上WASD前后左右移动相机，按住鼠标左键拖动可以上下左右移动相机。

小球的数量和封闭空间的尺寸通过 `config.txt` 文件读入，文件共4个参数，以空格分隔，依次为封闭盒的长length、宽width、高height及小球列数n。小球的初始化为每个维度上生成n个，因此总数为 n^3 个。**该文件必须与可执行文件置于同目录下。**

GPU上的线程块num_blocks和线程数threads_per_block写死在 `collision.cu` 的 `collisionDetection` 中，

```
unsigned int num_blocks = 128;
unsigned int threads_per_block = 512;
```

因此小球数量不应该设置过大，`config.txt` 中n的值不应该超过8，不应该小于2，且长宽高的数值应至少是n的1.5倍。

初速、质量、弹性系数都随机生成。由于碰撞为非弹性碰撞，因此小球的速度会越来越慢。由于没有考虑重力作用，最终小球的状态表现可能与现实有所差异。

遇到问题

- cuda的核函数无法读取全局变量，必须局部定义传入参数
- 小球越碰越快，最后发现是碰撞公式写错

参考文献

1. cuda安装:

- <https://www.cnblogs.com/arxive/p/11198420.html>
 - 需要先运行 `nvidia-smi` 确认本机支持的cuda版本
 - 已有VS的情况下会自动安装适配的cuda项目模板
- <https://zhuanlan.zhihu.com/p/488518526>

2. 碰撞检测

- <https://developer.nvidia.com/gpugems/gpugems3/part-v-physics-simulation/chapter-32-broad-phase-collision-detection-cuda>

3. C++相关

- <https://www.cnblogs.com/zealousness/p/10324170.html>

4. OpenGL

- 参考本人图形学课程的作业
- glut: [VS2022的OpenGL环境搭建（完整篇） - 知乎 \(zhihu.com\)](#)