QT 作业报告

(1)程序功能介绍

本程序为二维平面上基于凸多边形刚体的简单动力学约束模拟，可以实现如下几个基本功能：
1.通过设置顶点对导入的图片进行凸多边形的建模
2.通过利用凸多边形体的 AABB 盒完成 BVH 树的构建，从而实现对场景内凸多边形体之间碰撞的粗略检测
3.通过 GJK、EPA 算法实现对凸多边形碰撞的精确检测，并给出穿透向量，生成碰撞点对
4.通过对凸多边形上点对的约束方程求解实现碰撞、轻杆、轻绳、弹簧的模拟，可以基于这几类的自由组合实现对真实物理的模拟效果。

(2)项目各模块与类设计细节

本项目大致可分为凸多边形的构建，凸多边形的碰撞检测，动力学模块。其中凸多边形的构建包含 Point, Polygon 以及演示所用 Circle, Rope, Block, LongBlock 类，实现对凸多边形的建模。碰撞检测包含 BVH, PreciseBounceJudge 类，实现凸多边形的碰撞检测及碰撞点生成。动力学模块包含 Polygon, Constraint 类，实现凸多边形刚体的平动、转动以及各类约束。

类设计细节：
Point 是本项目最为基础的类，它继承于 QT 中 QVector2D 类，可以实现对二维向量的基本运算。其中 SetRXY 函数实现计算此点到给定点的相对位置并存储于 Rx, Ry 中。Move 函数实现给定角速度及转轴的位置变化。Vertical 函数实现将当前向量顺时针转动 90°，cross 函数实现给定两个二维向量的叉乘运算。
Polygon 是构造刚体及实现刚体运动最基本的类，其继承于 QT 中 QGraphicsPixmapItem 类，场景中图形的运动基于此类实现。

成员变量：

- points: 构成凸多边形的顶点
- companyPoints: 伴随刚体运动的点
- forces: 刚体所受冲量的集合
- map, mapPos: 刚体贴图及贴图坐标
- position: 刚体质心坐标
- velocity, angularV: 刚体速度及角速度
- Mass, Inertia,inverseMass,inverseInertia:刚体质量，转动惯量，逆质量，逆转动惯量
- Left, right, up, down, S: AABB 包围盒

成员函数：

- overlaps: 判断包围盒之间是否发生重叠
- caculateIP: 基于凸多边形的顶点，计算刚体的转动惯量及质心位置
- setPoint: 设置多边形的顶点
- addPoint: 添加刚体上的伴随点
- setVelocity, setAngularV: 设置刚体初速度及初角速度

addForce: 添加作用于刚体上给定位置及方向大小的冲量

actForce: 基于刚体动力学实现冲量的作用效果

move: 基于现有角速度及速度实现刚体的运动及贴图运动效果

BVH 是实现快速粗略碰撞检测的重要技术，也成为层次包围体技术。本项目中所用包围体为 AABB 盒，其由结构体 BoundingVolume 实现。层次包围体技术（BVH）指的是将所有包围体分层逐次地再次包围，获得一个更大的包围体，直到包围住所有物体。实际上，它是一个树形结构，因此可以仿照树的结构，将两个或三个小的包围体包围成一个更大的包围体，以此类推。

成员变量：

Children: 指向两个子节点。该指针不为空的 BVH 节点称为分支节点，即包围两个 AABB 的 AABB

Parent: 此节点的父节点

Volume: 此节点的 AABB 盒

Body: 指向包含的刚体。若该指针不为空，则此节点为叶子节点

成员函数：

isLeaf: 判断是否为叶子节点

getPotentialContact(With): 通过遍历子节点，找到所有可能发生碰撞的刚体，并将其存放于 PotentialContact 的数组中

recaculateBoundingVolume: 重新计算当前节点的 AABB 盒

insert: 插入新的叶子节点

~BVH: 删除节点并更新 BVH 树

PreciseBounceJudge 是实现刚体之间精确碰撞的类，基本原理是利用 GJK 算法构造闵可夫斯基差的单纯形，再通过 EPA 算法拓展单纯形得到最大穿透深度及穿透向量。利用穿透向量，可以计算得到刚体之间的碰撞点对，从而为碰撞约束的计算提供基础数据支撑。

成员变量：

S: 单纯形顶点，利用结构体 Minkowski_t 存储得到此结果的两点

Nearest: 存储闵可夫斯基差中离远点最近的一条边

BouncPoint: 存储生成的碰撞点对

U: 存储穿透向量

成员函数：

GJK, support, nearestSimplex: GJK 算法的实现

EPA: EPA 算法的实现

createBouncePoint(Pair): 生成碰撞点对

Constraint 实现各种刚体之间的约束的动力学计算，通过对约束方程的求解，得到各种物理约束效果。

成员变量:

potentialBound, potentialBoundNum: 存储可能发生碰撞的刚体

rigids: 存储有轻绳约束的刚体

elastics: 存储有弹簧约束的刚体

bars：存储有轻杆约束的刚体

成员函数:

setRopeConstraint, actRopeConstraint: 设置轻绳约束，实现轻绳约束

setElasticConstraint, actElasticConstraint: 设置弹簧约束，实现弹簧约束

setBarConstraint, actBarConstraint: 设置轻杆约束，实现轻杆约束

getPotentialBound, actBounce：获取可能发生碰撞的刚体，实现刚体之间的碰撞
Const 头文件设置一些常用常数，便于后续更改

(3)小组成员分工

刘子健：负责实现 BVH 及精确碰撞检测算法
王振霖：负责实现约束方程算法求解
张艺豪：负责实现整体代码的合并及编写演示程序

(4)项目总结与反思

首先需要指出的是本项目只是一个非常粗略的物理模拟，对于静态物体之间的接触未写相关的睡眠函数防止物体的频繁抖动，对于快速运动的物体缺乏连续碰撞检测(CCD)，约束方程求解只争对单点对求解导致对一些复杂场景运动收敛效率较低，致使整个系统稳定性较差。整个项目仍然需要大量的打磨才能实现真正的物理效果模拟。
在整个项目开发中我们也遇到许多问题，接下来将一一说明：

1. 在编写代码之前应当谨慎地考虑方案的可行性，避免走回头路导致项目进展不顺
2. 小组内变量、函数、类的命名应当有统一的命名规则，并且名称最好与对应功能相符，避免出现混淆。
3. 函数的输入与输出最好符合惯例以及经验直觉，对于某一同类事物有统一标准的处理方式，并且最好注释说明函数的用法。
4. 代码应当逐一调试，调试成功后再整体合并，以提升调试的效率。
5. 小组内需要规定统一的精度需求（比如 float 和 double），或是用统一的预处理规范代码书写，便于后续更改
6. 小组成员应当多加交流