

程设大作业报告

张泽 游皓宇 吴小宇

2025 年 6 月 30 日

1 功能介绍

本小组通过使用 pyqt、api、pymunk、多线程等技术实现了一个希望对标数学几何画板的物理 2D 引擎实验室，同时集成了一个基于 deepseek 的聊天窗口。

相比于视频录制版 demo，我们又进一步实现了一些项目细节。

2 模块 1-pymunk

2.1 物体添加

通过控制中心坐标，重量，转动惯量等物理参数来建立圆，多边形，有质量线段等。同时提供了添加各种约束的接口，弹簧 (成功集成)，杆，齿轮 (未成功集成)。并提供在添加后更改属性的接口。

2.2 物理运动模拟

通过 pymunk.space 中的各用法、属性，通过离散化时间序列 (基础 60fps/s) 的物理计算刷新，不断返回当前物体的运动状态，位置关系，(有摩擦) 碰撞计算。

同时，提供了各种场力计算接口 (重力场等非保守场已实现，而电场等保守场已提供接口)；也提供了控制时间流速的接口。

3 模块 2-物理引擎的渲染和 ai 窗口

3.1 物理引擎界面

在这个模块中我们使用了 pyqt5 本身的众多模块调用，定义了众多的物体类，(如线段、矩形、圆形、弹簧)，同时通过线性衰减的一个平缓化处理实现了运动渲染的丝滑过程。同时我们极大的丰富了界面的趣味性和功能性，我们在 toolbar 一栏添加了众多的工具，比如控制运动的开始与结束按钮、重置、删除选中物体、取消选中、编辑选中物体的属性等功能。同时我们在最上面的 navbar 一栏添加了文件系统，可以实现一个模拟过程的物理数据的文件读取和保存。并且优化了物体添加体验。于此同时，在界面的左侧，我们添加了历史数据栏，使得随时随地可以读取历史数据，探究物体运动的具体过程。

3.2 ai 界面

最新的更新是我们在右侧栏显示添加了一个水平力控制调，更加方便物理过程的控制。同时我们添加了 ai 窗口，通过用户自己上传 deepseek 的 api key，来实现一个 ai 助手窗口，极大的丰富了我们界面的功能。

4 模块 3-模拟数据的处理与保存

这个模块的主要任务是处理物体运动的数据，以便模拟的数据能够重复利用。

4.1 模拟数据储存

datahandler.py 在模拟进行的过程中收集各个时刻每个物体的运动数据，包括物体的位置和速度，并暂时储存。当需要保存数据时，暂存的物体数据就存储到.db 文件中，并在左侧的数据栏中有所显示。

4.2 模拟数据可视化

存储的数据可以通过 matplotlib 的方法进行可视化。文件 drawmap.py 实现了读取数据文件并根据文件中的数据绘制图像的功能。主界面的数据

文件栏提供了图像绘制的接口，在点击文件的时候就会显示由该文件数据绘制出的图像。

5 类设计细节

5.1

在主界面程序 `mianwindow.py` 中，我们使用了类 `ChatAPICaller`，用于实现 ai 聊天窗口，类 `VideoView` 用于开始界面的视频播放，类 `HomePage` 作为程序的开始界面，类 `PhysicsSimulatorWindow` 创建了一个集成各种功能的主界面。这个文件中的类实现了 UI 界面的生成。

5.2

在 `simulator.py` 中创建的类 `PhysicsSimulator`，实现了创建、修改和删除各种物体的接口，场力的创建与修改，并对所有物体的运动状态进行模拟，完成了物理引擎重要的后端。

5.3

在 `datahandler.py` 中创建的类 `DataHandler` 主要用于保存物体的运动状态，实现运动状态的储存和重新调用。主界面中新建、保存等与文件相关按钮的接口也在该类中完成。

6 小组分工情况

张泽：负责的是 `pyqt` 的图形界面的开发，各种物体类的实现和物理过程的渲染，以及早期 ai 窗口的开发。

游皓宇：前期主要负责 `simulator`，`datahandler` 中全部基于 `pymunk` 的底层物理实体添加，物理状态更新计算，并建立好通向主窗口的接口。后期负责开始界面 (`HomePage`) 建立，优化存储逻辑，以及调整 ai 对话窗口 `api`。

吴小宇：负责 `datahandler` 中核心存储功能的完成、图形界面中与文件相关部分的开发和使用 `matplotlib` 的数据可视化功能。

7 总结与反思

下面先阐述我们目前完成的项目文件结构

7.1 Simulator.py

Class PhysicsSimulator 各种添加，物理场景离散时间更新逻辑

7.2 DataHandler.py

Class DataHandler 实现数据收集，数据存储

7.3 MainWindow.py

Class ChatApiCaller 与 deepseek api 的直接接口

Class ViewVideo homepage 的背景视频播放

Class Homepage homepage 生成

Class PhysicsSimulatorWindow 主窗口部分

窗口部件部署

添加接口

运动过程平滑化处理

7.4 DrawMap.py

matplotlib 应用

7.5 我们的项目经历和总结

我们的团队来自树洞组队，且在较早 (4.5) 就进行了第一次项目组会，并确定了我们的项目目标。此处笔者 (游皓宇) 基于作为一个数竞生对“几何画板”的喜爱，提出了做一个类似于几何画板的物理实验室。我们认为这个想法很有创意，且可操作性强

在 5.15 之前的漫长期中季，我们的进度慢慢推进。期中结束后，我们加快项目推进速度，在 5.31, 6.3 这两个关键时间节点基本完成了代码主体。很遗憾由于几位同学都时间上繁忙没能参加大作业路演。

在更加繁忙的期末季结束后，我们进一步处理清楚了项目的细节，基于自己使用体验的不佳之处作了进一步优化（比如 ai 应答窗口回复卡顿易失败等问题的优化）。

但是我们能力有限，还是没能完成一些复杂的工作。比如初期，我们希望能实时记录我们的实验过程并能通过存储实时回溯到某一点并从其开始实验。后来注意到，这其中要实现的回放逻辑过于复杂，内存管理我们都很难控制，并且在其中我们需要记录外界实时拖拽，其中对受力情况，速度情况的瞬时改变很难回溯模拟，引擎本身也可能不支持，超出了我们的能力范围。

但遗憾的是，我们由于时间原因，还没有处理好一些基础逻辑。比如齿轮等约束的添加逻辑。

希望在本次大项目后，我们组内三名成员能从中汲取时间管理，代码建设中类管理，代码中文件沟通，pyqt 窗口建立，运动图形渲染等多方面经验，为今后投身更大项目作好铺垫。