

Information

"Construct 3D World" 说明文档

开发环境版本

- 操作系统版本: WSL: Ubuntu-20.04LTS (Windows11 家庭中文版-64bit)
- 编辑器版本: Visual Studio Code 1.92.1
- 编译器版本: gcc 9.4.0

设计思路

整体架构

使用MVC结构，共分为模版、控制器、命令行界面三部分，通过在main.cpp中调用界面类的启动函数开启程序，使用Makefile进行编译

Model

Model模块包含了三维模型处理器的核心类，如Model3D, Face3D, Line3D, Point3D等，为了实现概念的抽象，增强核心类的泛用性，Model模块中还设计了Vector类模板、Group类模板等较为抽象的基类，这些核心类的具体关系为：

- Model3D **HAS-A** Face3D, Line3D
 - Face3D, Line3D **HAS-A** CountFixedSet(of Point3D)
 - CountFixedSet **IS-A** Set
 - Set **IS-A** Group
 - Point3D **IS-A** Vector3D
 - Vector3D **IS-A** Vector
 - Face3D, Line3D **IS-A** Element3D

这些类既服务于大作业，又具有一定的泛用性。Vector、Vector3D是向量类，具有向量的坐标和运算特性；Point3D为Vector3D的派生类，隐藏了基类中关于向量运算的接口。Set类在“组”的基础上实现了集合的特性，而CountFixedSet则限制了元素个数不可改变，使得Face3D和Line3D更加安全。

Controller

控制器模块中只含有Controller一个类，实现了三维模型编辑器的具体功能：

- 储存Model3D实例
- 对文件读入、写入类的调用
- 对模型的更改，包括对线段、平面进行添加、删除、修改
- 对模型参数的显示，包括对线段、平面上的点及序号的显示
- 将运行错误转化为自定义枚举类型，供界面类显示

此外，控制器模块使用了单例模式，防止创建多个控制器对象，造成控制混乱。

Data

Data模块包含了数据文件，导入、导出模型的类。严格来说，导入、导出功能也属于控制器的一部分，但与Controller的关系相对独立，所以与数据文件一起保存到了Data模块中。Data模块中的类关系比较简明：

- OBJImporter **IS-A** Importer
- OBJExporter **IS-A** Exporter

从概念上看，导入、导出类应当适配多种文件格式，因此设计了Importer与OBJImporter的继承关系；Importer包含导入文件的基本操作，OBJImporter则根据.obj文件的具体格式对文件进行逐行读入，导出类的设计思想与此相同。

View

View模块是命令行界面，与控制器结合比较紧密，其中只包含了View类。View类的主要功能包括：

- 形成用户界面，实现用户与软件的交互
- 处理由控制器传出的操作结果信息，并反馈给用户
- 从用户输入中读取指令，调用相关接口并获得程序反馈
- 保持程序健壮性，对非法用户输入进行处理