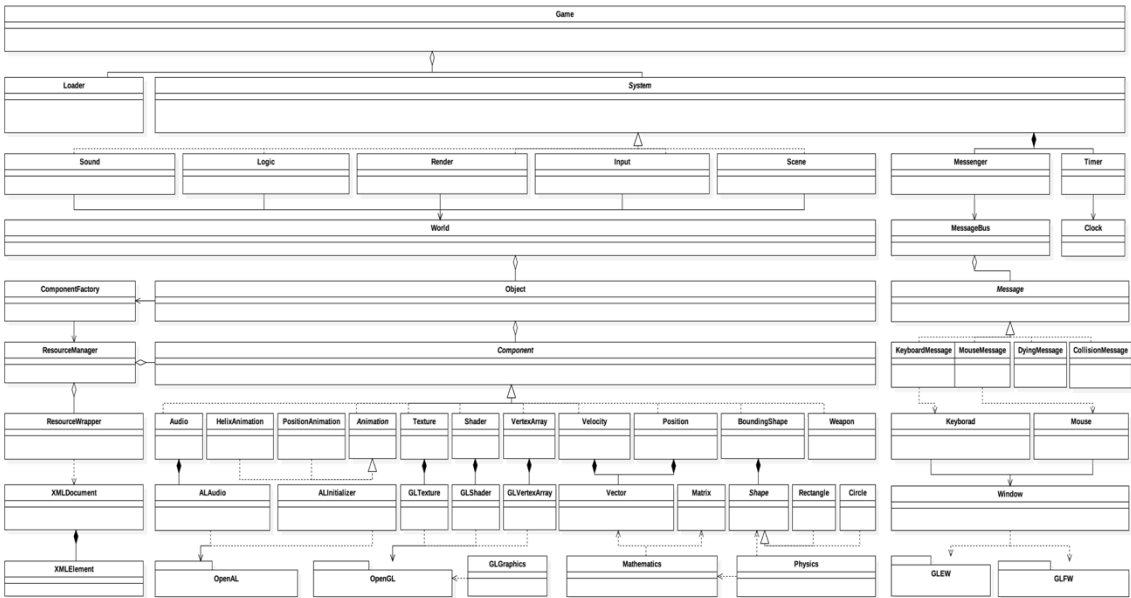


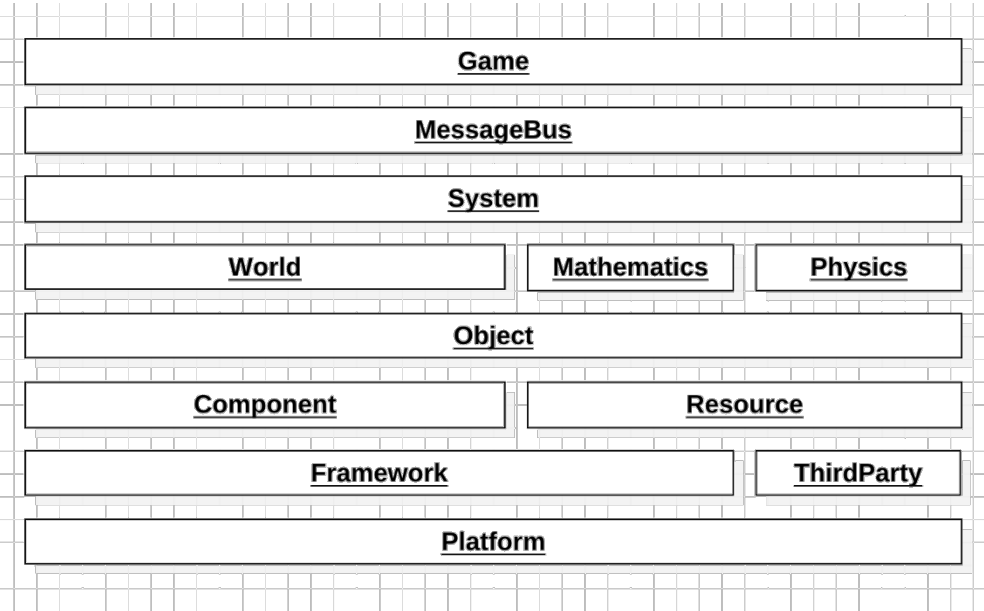
# 游戏引擎制作与成果展示

## 一、 系统架构与运行流程

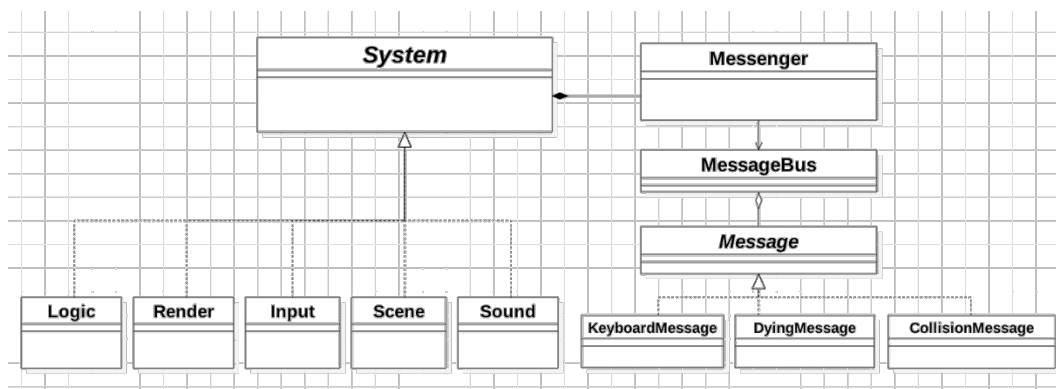
### 1、 整体类图



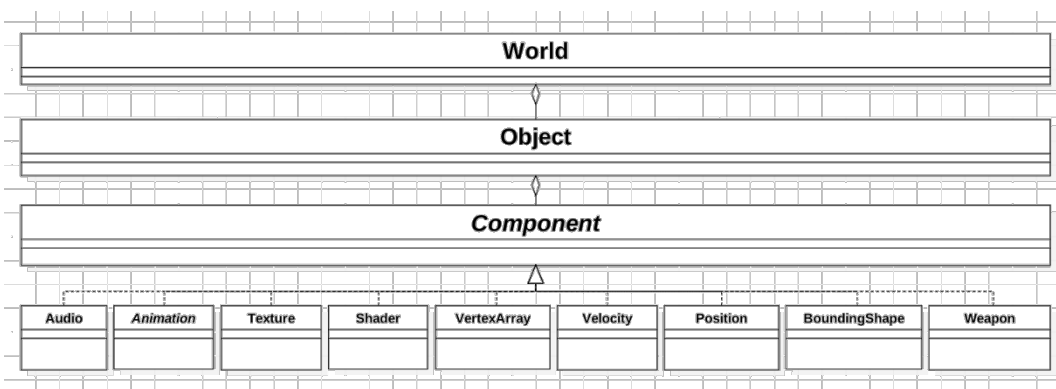
### 2、 分层结构



### 3、 消息驱动



#### 4、 Entity Component System



#### 4、 运行流程

得益于良好的架构设计，我们能以非常优雅的方式创建游戏：

- 获取加载器实例并加载 XML 文件配置游戏；
- 创建游戏对象；
- 创建游戏子系统及游戏逻辑对象；
- 将游戏子系统绑定到游戏对象上；
- 开始游戏。

## 二、 设计模式与游戏逻辑

模式	应用
桥接模式	Object 的 Component 创建接口，Messenger 的收发消息等
工厂模式	ComponentFactory 等
单例模式	Loader, World, Window 等
适配器模式	ResourceWrapper
模版模式	System, Animation, Constraint 等

享元模式	ResourceManager
------	-----------------

### 三、 组员分工

赵鋈峰	游戏逻辑；组件设计等
郑少锐	系统架构；图形引擎（OpenGL）；资源管理等
孙志航	图形引擎（DirectX）；人体工程学设备等
王展鹏	物理引擎；数学引擎等
阎婧雅	游戏设计；音频引擎；网络通信等

### 四、 环境与依赖项

在项目中我们主要是用了以下第三方库：

- [GLFW](#): Open Source, multi-platform library for OpenGL, OpenGL ES and Vulkan development on the desktop
- [GLEW](#): The OpenGL Extension Wrangler Library
- [ASIO](#)（包含在项目文件中）: Cross-platform C++ library for network and low-level I/O programming
- [LodePNG](#)（包含在项目文件中）: PNG image decoder and encoder

项目构建过程中，我们使用了以下 IDE 与工具：

- [CMake](#): Open-source, cross-platform family of tools designed to build, test and package software
- [CLion](#): Cross-platform IDE for C and C++
- [Visual Studio](#)
- [Doxygen](#): De facto standard tool for generating documentation from annotated C++ sources
- [LATEX](#): high-quality typesetting system
- [Git](#)
- [GitHub](#)