

# 3D 引擎开发笔记

Zhang Yf

2021 年 11 月 5 日

# 目录

<b>第一部分 序言</b>	<b>2</b>
<b>第一章 序言</b>	<b>3</b>
1.1 新的开始 . . . . .	3
1.2 初步的写作计划 . . . . .	3
1.2.1 基础系统 . . . . .	4
1.2.2 数学与几何 . . . . .	4
1.2.3 渲染系统 . . . . .	4
1.2.4 物理系统 . . . . .	4
1.2.5 场景管理系统 . . . . .	4
<b>第二部分 基础模块</b>	<b>5</b>
<b>第二章 从零开始</b>	<b>6</b>
2.1 第一个简单项目 . . . . .	6

# 第一部分

## 序言

# 第一章 序言

## 1.1 新的开始

从本科时期开始接触代码后，我便一直希望能够进行自己的项目开发，日常也喜欢写一些个人小程序自娱自乐，因为一直对游戏较为感兴趣，因此初期希望研究并开发类似 Unity 的游戏引擎。困于本科时期知识基础的薄弱，以及相关开发经验的缺乏，四年间都只做过一些较小的个人项目，知识体系尚未形成一个较为系统的架构，没有将个人的思想融入到项目中实现。

研究生阶段，为了能够对读研生涯进行记录，我开始了这个项目的设计。但是由于毕业与工作的压力，之前的文章写了部分就暂时搁浅了，而现在终于有时间来继续推进。我的目标是将此项目作为一个完整的系统，搜集资料后按照个人的想法进行设计和实现，不仅能够锻炼提升个人水平，还能够锻炼阅读与其他能力。同时，这个系列的文章不仅是我个人学习过程的笔记，也将作为后续编写类似程序的相关参考，并在这之中提出一些我个人的想法来抛砖引玉，得到他人的指导或同好的学习与分享。最后希望通过此项目对相关领域做出一些微薄的贡献。

本文使用  $\text{\LaTeX}$  编写。

## 1.2 初步的写作计划

此项目聚焦于 3D 引擎的设计开发，模仿游戏引擎进行模块划分。一个游戏引擎由许多模块组成，每个模块具有相对独立的功能，参考《游戏引擎架构》的内容，本文章将按照以下几个部分设计介绍。

### 1.2.1 基础系统

这个部分用于介绍引擎底层系统的设计，作为学习的必要过程，本项目中将跟随开发进度不断修改引擎底层结构的设计。许多基础的数据结构与算法也会出现在本节。

### 1.2.2 数学与几何

随后的渲染与物理部分，都与数学紧密相关。数学构成了渲染与物理模块的基石。本节主要介绍一些基本的数学内容，作为后续部分的基础。

### 1.2.3 渲染系统

为了能够展示场景的运行效果，需要有直观的方式展现内容，3D 渲染就是最好的方式。这个部分主要介绍渲染的相关内容，包括一个软渲染器以及对应的 GPU 版本渲染器。

### 1.2.4 物理系统

本节展示一个简单的物理仿真模块，用于刚体碰撞与刚体动力学仿真。

### 1.2.5 场景管理系统

这个部分用于介绍较为大型的场景优化所需的场景管理模块的相关内容。

## 第二部分

## 基础模块

## 第二章 从零开始

### 2.1 第一个简单项目

作为笔记的第一部分，本节主要介绍从零开始搭建一个小型项目的过程。大部分的 C++ 项目都需要选择（虽然其他项目也是，但是没有 C++ 这么多的选项）一个基本的开发环境，而 CMake 是其中一个流行的选择。从 github 看去，很多国内外大型的开源项目都选择了 CMake 进行项目的配置和管理。前几年我也是 CMake 的推荐者，但学习过 CMake 的我仍然觉得 CMake 过于死板，需要学习额外的 DSL (Domain Specific Language)，无法轻易更改编译过程。经过查找，我发现了一个国产的开源项目 xmake<sup>[1]</sup>，可以用 Lua 编写项目配置文件，符合 C++ 编译过程的直觉，且可以通过 xmake 避免复杂繁琐的外部库安装配置过程，让人感觉很舒服。最终本项目决定使用 xmake 作为项目配置软件。

当然，本项目从熟悉的 Hello World 开始，一步步搭建起项目框架。首先是函数主体：

```
#include <iostream>

int main() {
    std::cout << "Hello World" << std::endl;
    return 0;
}
```

Engine/main.cpp

这个就是简单的标准的 Hello World，通过 xmake 可以简单的配置项目：

```
target("hello_world")  
    set_kind("binary")  
    add_files("Engine/*.cpp")
```

xmake.lua

在命令行输入