



*Zhejiang University*

*College of Computer Science and Technology*

---

## 面向对象程序设计 大程 M3 报告

---

*Author:*

刘泓健 吴一航

*Supervisor:*

翁恺

An Assignment submitted for the ZJU:

211C0010 面向对象程序设计

2022 年 5 月 21 日

## 1 概述

在这两周的大程设计中，我们在此前完成的基本类设计基础上，完成了 `dialog` 类以及函数的设计以及相关的测试程序，我们的 MUD 游戏初始场景已经可以成功运行。

## 2 代码基本框架

本月的大程设计实现的代码基本框架如下：

```
1 .
2 |——assets(游戏所需图片、文案等资源)
3     |——dialog.json
4 |——doc(大程文档，如规范性文档以及注释文档等)
5     |——general.md(规范文档)
6 |——include(自主设计的头文件)
7     |——character.h
8     |——skill.h
9     |——community.h
10    |——infrastructure.h
11    |——policy.h
12    |——technology.h
13 |——lib(外部引用库，如JSON parser及图形界面等所需外部库)
14     |——nlohmann
15     |——...
16 |——src(大程源代码)
17     |——core
18     |——dialog.cpp
19     |——utils
20     |——main.cpp(初始执行文件)
21     |——Makefile
22 |——testbench(测试文件)
23     |——xcharacter.cpp
24     |——xjson.cpp
25     |——xdialog.cpp
26 |——Makefile
```

## 3 dialog 设计

在这一部分中，我们完成了开场白的设计。当程序运行时（注意：程序的运行方式在根目录 README 中），会进入游戏初始化界面，然后开始我们的初始对话。主函数的循环中，我们直到没有下一条对话时结束循环，结束游戏。

在对话类的设计中，我们关注以下几个重要的属性：

1. `branch`：分支，根据玩家所作出的选择进入某一分支
2. `randBranch`：随机分支，玩家无需做出选择，系统自动创造随机分支
3. `nextDialog`：指向下一 `dialog` 的信息

当然后续还会有条件分支的设计，这些将会是下一阶段完善剧情时会逐步实现的内容。

我们不难发现，在运行游戏时，主体结构是首先运行当前分支的内容，然后根据玩家选择或者系统设定获取下一分支，接下来构造下一分支继续执行。特别注意的是，我们是通过 `json` 来实现数据的存储的，我们利用上一次报告中提及的 `json parser`，取出我们 `dialog.json` 中的数据然后进行类的初始化等。并且这一 `parser` 还有自带的异常处理，当用户给出错误的输入时会自动抛出异常。当然，这一异常处理我们会在下一次整体框架完善时做进一步的改进。

## 4 下一阶段展望

在下一阶段的设计中，我们将会完成整个游戏的设计。这一方面是会将我们准备好的后续文案放置在 `asset` 中，另一方面是完善 `community`、`infrastructure` 等类的设计，构建起完整的文明游戏框架。除此之外，异常处理部分我们也会对原先的库中的处理做一定的优化，最后会根据时间决定是否使用图形库设计。