



*Zhejiang University*

*College of Computer Science and Technology*

---

## 面向对象程序设计 大程报告

---

*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 6 月 11 日

## 1 大程简介

在课程大程中，我们实现了一个完整的 MUD 游戏。这一游戏的主题灵感来源于 FiraxisGames 的著名游戏《文明》系列，希望实现的是游戏主人公从山洞中走出直到统治文明的过程。虽然本项目的初衷是完成本学期 oop 的课程大程，但是我们的目标不止于此。我们的设计具有强大的可拓展性，实际上，用户只需要修改数据文件即可实现游戏的再生产。本项目开源地址为 [https://github.com/FrightenedFoxCN/OOP\\_capolization](https://github.com/FrightenedFoxCN/OOP_capolization)，对于本项目实现的基本框架感兴趣可以使用，完善，或者添加自己希望实现的功能。

## 2 使用的技术

### 1. 基于 C++ 的面向对象编程技术

在这一大程中，我们基于 C++ 开发，通过几个主要类的设计进行组织，使用了命名空间、虚函数、异常处理、流等 C++ 编程技术。一方面将课程所学知识应用于我们的设计，另一方面也通过网络等途径在编程过程中学习新的特性。

### 2. 基于 Git 的版本管理

我们的项目目前已在 GitHub 上开源，开发过程中也一直使用 git 的方式进行版本管理。

### 3. 基于 JSON 的数据管理

对于这一点，我们需要说明的是，在我们代码的 lib 文件夹中，引入了一个 JSON parser 库便于处理我们的 JSON 数据。我们 JSON 文件的处理方式在一定程度上参考了人生重开模拟器的组织方式。这一设计相对于使用其他文件组织方式更为简便、清晰。

## 3 项目特色

### 1. 可扩展性

这一点已在第一部分大程简介有所描述，此处不再赘述。

### 2. 实现了游戏存档功能

这一实现基于我们的使用 JSON 的数据处理方式，这使得我们能够以很便捷易懂的方式完成游戏存档与重新读入数据的操作。

### 3. 完整的测试代码

我们不仅实现了最后的 release 版本，在开发过程中，我们对于每一个模块，例如 character 类、dialog 类以及我们的 JSON 数据处理，都做了基本的测试。详细的测试代码在 testbench 文件夹中。

### 4. 清晰的开发流程，文件框架，以及代码规范。这一部分在 doc 文档的 general.md 中有所说明。

### 5. 故事设计埋藏深意（或许需要进行很多次游戏遍历大量分支才能有所体会）。

## 4 代码基本框架

```

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      |—— dialog.h
9  |—— lib(外部引用库, 如JSON parser及图形界面等所需外部库)
10     |—— nlohmann
11         |—— ...
12 |—— src(大程源代码)
13     |—— core
14         |—— dialog.cpp
15         |—— character.cpp
16     |—— utils
17     |—— main.cpp(初始执行文件)
18     |—— Makefile
19 |—— testbench(测试文件)
20     |—— xcharacter.cpp
21     |—— xjson.cpp
22     |—— xdialog.cpp
23     |—— xjsonfile.cpp
24 |—— Makefile

```

## 5 基本设计

我们的程序入口为 `main` 函数, 在进行了一系列初始化, 创建新文件或者从文件中读入文件。接下来我们开始游戏。实际上 `main` 中的整体流程是并不复杂的, 实际上就是通过调用 `Dialog` 中的函数得到下一次的分支, 然后通过解析出 `JSON` 中的内容调用构造函数构造出这个 `dialog` 对象, 如此往复, 直到结局。

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, 系统自动根据概率计算创造随机分支
3. `condBranch`: 条件分支, 根据所给条件决定分支 (实际上与角色的属性有关)
4. `nextDialog`: 指向下一 `dialog` 的信息

在角色 `Character` 类的设计中，我们主要实现的是几个属性，以及如何将这些属性保存进 JSON 文件/从 JSON 中读入。以及如何将我们在进入一定分支后，对角色自身属性产生一定的的影响。

在整体设计中，我们设计了四个死亡结局，剩余一个可以苟活的结局。对话作者在其中设计了洛夫克拉夫特式的结局，蕴含深意，与我们选择的主题契合。

## 6 运行与使用方式

运行在 windows 环境下需要 `mingw32-make release` 即可，Linux 环境使用 `make` 命令进行 `make release` 也可以。接下来 `./test` 即可开始游戏。

进入游戏后，首先输入用户名，新用户新建用户名即可。接下来开始游戏，当需要输入时，根据显示输入即可，如果输入 `q` 则会退出并存档。注意，由于游戏中有很多的随机 `branch`，所以运气不好可能会陷入循环，或者很快就会落入死亡结局。

如果遇到编码问题，请将终端中的字符设置为 UTF-8 格式。

## 7 运行结果展示

在以下几项中选择：

- a. 接着在洞穴里转转吧。
- b. 再留在这个山洞里该发霉了。

你选择：a

\* 接着在洞穴里转转吧。

在以下几项中选择：

- a. 先看看旁边带刻痕的石板吧。
- b. 在山洞的角落里，躺着一堆木板，似乎还有燃烧过的痕迹，在那边或许可以找到什么东西。
- c. 石壁上似乎有画着什么东西。

你选择：a

\* 先看看旁边带刻痕的石板吧。

- 石板看起来有点年头了，青苔附着在上面，在黑暗中让你更加难以看清上面的字迹。

在以下几项中选择：

- a. 你离开了石板。
- b. 你用手摩挲着石板上的刻痕，试图识读石板上的痕迹。

你选择：b

- 你用手摩挲着石板上的刻痕，试图识读石板上的痕迹。

- 你无视了诡异的气氛，凑近石板，仔细研读其上的字迹。

- 你发现石板上的字迹似乎曾相识，经过一些尝试，你读出了上面的文字：“梦境.....爱.....手工艺品.....”

\* 接下来要干什么呢？

在以下几项中选择：

- a. 接着在洞穴里转转吧。
- b. 再留在这个山洞里该发霉了。

你选择：a

\* 接着在洞穴里转转吧。

在以下几项中选择：

- a. 先看看旁边带刻痕的石板吧。
- b. 在山洞的角落里，躺着一堆木板，似乎还有燃烧过的痕迹，在那边或许可以找到什么东西。
- c. 石壁上似乎有画着什么东西。

你选择：b

- 在山洞的角落里，躺着一堆木板，似乎还有燃烧过的痕迹，在那边或许可以找到什么东西。

- 你用手轻轻掀开已经被青苔覆盖的木板，在余烬中尝试寻找一些有用的东西。

- 你听见了一些奇怪的响动，似乎由洞内传来.....

## 8 一些展望

虽然本次大程到目前告一段落，但是我们对于这一游戏的设计还没有结束。我们一方面希望能给这个游戏一个更加圆满的设计，当然限于时间和灵感，我们目前的设计还有可以扩展的空间。另一方面，我们希望将这一框架实现为一个能够便捷开发所有具有类似设计思路的 MUD 游戏的基本框架，我们还可以实现将一定格式的流程图转直接转化为 JSON 文件，使得开发更为便捷。除此之外，我们还可以对存档文件做加密，这一点更加接近真实情况，也防止后门。