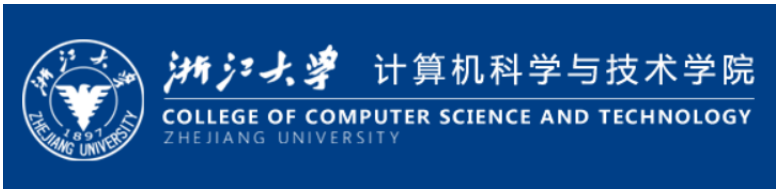

Zhejiang University

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 Firaxis-Games 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，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。

4 代码基本框架

```
.
|——assets(游戏所需图片、文案等资源)
    |——dialog.json
|——doc(大程文档，如规范性文档以及注释文档等)
    |——general.md(规范文档)
|——include(自主设计的头文件)
    |——character.h
    |——dialog.h
|——lib(外部引用库，如JSON parser及图形界面等所需外部库)
    |——nlohmann
        |——...
|——src(大程源代码)
    |——core
        |——dialog.cpp
        |——character.cpp
    |——utils
    |——main.cpp(初始执行文件)
    |——Makefile
|——testbench(测试文件)
    |——xcharacter.cpp
    |——xjson.cpp
    |——xdialog.cpp
    |——xjsonfile.cpp
|——Makefile
```

5 基本设计

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。

6 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。