

# Qt 大作业 41 组作业报告

组名: Hello\_world      组员: 王骏达 孟凡荣

## 1 程序功能简介

我们开发的游戏《Oliver》是一个第三人称剧情向解谜游戏，玩家在各个地点探索，与不同的人物对话，阅读作者留下的线索，理解故事、收集宝物、获得能力，最终迎战 boss，走向结局。

地图上有若里村，放逐之地，火焰谷，极寒之地四个可到达地点。每个地点中有一些小地点，道具以及 NPC。若里村是初始地点，放逐之地是最终战斗地点。劳斯镇，谷比恩和灵魂圣地是提示性地点以及展示设定。

首先玩家可以选择新游戏和继续游戏。如果选择继续游戏则会读档。玩家点击新游戏后会得到故事背景的提示。在游戏中，玩家可以随时跳转到不同地图或在同一地图内去不同地点探索，但是访问不同地点有一定的能力要求，部分地区只有当玩家的状态符合要求才能进入。点击玩家头像可以查看当前能力、物品等状态，点击上方选项可以存档并退出或者查看设置，点击帮助可以查看提示以及游戏设定。左下角展示玩家所在地点。玩家需要强化自身，获取装备以进入不同地区，通过提示和对话中的线索进行解谜，最后当玩家前往最终战斗地点时，主角的不同状态会导向不同的结局（3 种）。

## 2 项目模块与类设计细节

模块包括开始界面模块，背景音乐模块，文件加载模块，对话展示模块，人物状态模块，背景与动效模块，场景和场景切换模块，游戏设置模块。

### （一）开始界面模块

开始界面模块包括 Mainwindow 类，实现了界面初始化，资源加

载等功能。其中 `init_window` 函数实现窗口初始化（包括获取时间，初始化窗口及背景，初始化按钮位置，显示游戏名称）。之后依次加载背景及音乐资源，创建若里村子窗口，设置音乐播放器。另有 2 个子函数分别实现更换开始界面背景以及检测开始界面的鼠标事件并更换背景图片。`Mainwindow` 类有 4 个槽函数，分别对应开始游戏，继续游戏，设置，退出 4 个按钮。点击开始游戏即创建新游戏，点击继续游戏可读取存档，点击游戏设置可以调节音量，点击退出游戏即退出。

## （二）背景音乐模块

音乐模块包括 `effectaudio` 类。该类通过 `QThread`, `QSoundeffect` 等 QT 自带类实现了音效播放的功能。该类包括 4 个子函数，分别实现音效播放（传入不同 `voice_type` 信号播放不同音效），次数设置，停止以及设置音量。

在该模块中，我们采取了多线程的方法避免了音效播放卡顿的问题，通过将 `QSound` 播放移到新线程中，有效解决了音画不同步等问题。

同时我们通过使用 `QMediaPlayer` 实现了背景音乐播放，其中正常界面、海边、战斗环节和感谢阶段分别有不同的音乐。在此我们创建了一个静态的 `player` 以便于我们访问它，方便实现音量调节和歌曲更换。

## （三）文件加载模块

文件加载模块包括 `file` 类。`Setfile` 函数实现存档（二进制文件），`init_by_file` 函数实现读取（二进制文件）。其中读取以及存档的内容是人物对所有有关地点的访问状态以及获取能力，武器，魔法的状态。

## （四）对话展示模块

对话展示模块包括 `dialog`, `dialogue` 类。`Dialog` 类实现了对话

框，包括两个槽函数对应继续与退出两个按钮。初始化包括窗口设置（有背景），位置设置以及计时器设置。Set\_text 函数实现类似输入流功能，show\_act 函数实现将输入流中的内容以逐行浮现形式展现，通过计时器计时并以一定速度输出，通过 pos, roll\_sum 等参数控制每次输出量，保证输出不超过指定范围，并且一页仅展示一部分内容，点击继续展示下一部分内容。Dialogue 类包括游戏所有文字内容（约 5k 字）。

### （五）人物状态模块

人物状态展示模块包括 Oliver 类以及 status\_show 类。Oliver 类是本项目的主角类，负责在新场景中创造主角图像以及存储主角状态参数。通过 Oliver 类构造函数实现了设置人物以及头像按钮，status\_show 类则会在点击头像后的对话框中通过存储的状态展示对应的人物状态。

### （六）背景与动效模块

背景与动效模块由 set\_context 文件中的函数实现。Set\_context 函数通过打包 Palette 实现了通用的窗口背景设置功能，grad\_animation 函数则通过 QAbstractAnimation 实现了透明度改变的动效（其中通过传入 judge 参数判断结束过程后是否要 delete 该窗口），set\_button\_context 和 set\_status 设置按钮背景和位置信息展示。

其中，我们运用该模块函数和 QTimer+多帧动画的方式实现了很多特效。如 rolig\_village 中的喷泉通过连续绘制多张喷泉图像实现了流动效果，rolig\_sea 中的船只动态也用了此方法，邪教徒的移动特效通过平移实现。而开始界面建立新游戏后的黑屏淡出效果则采用了将一块黑色 widget 透明度逐渐调整的方式。

### （七）场景和场景切换模块

本项目中的场景模块包括 banished, rolig\_village, iceplace,

fire, end, shop, getmagic, getsword, iceiron, information, ironman house, magichouse, outsea, pass, rolig\_sea 等类。

其中，End 类展示结局之后的作者的话，有背景音乐。Banished, fire, iceplace, rolig\_village 分别对应放逐之地，火焰谷，极寒之地，若里村的背景场景实现，小地点展示以及状态更新。

Getmagic 类，getsword 类对应获得魔法，获得武器（展示对话框，更新状态，检查是否有更高等级状态）。Iceiron 对应极寒之地中的宝箱（展示对话框，检查宝物是否被取走）。Information 是游戏开始时展示的对话框（背景图片与文字），magichouse 是若里村中魔法屋。Outsea, rolig\_sea 类包括若里村遇到渔夫后行动内容。Pass 实现火焰谷中解谜（密码）场景。

地图间跳转模块由 load 类与 map 实现。load 类会展示一个含有进度条和文字的对话框，可视化跳转过程。进度条与实际加载进度无关，由 QTimer 计时控制进度。Map 类则实现地图上某区域被点击后更新位置（pos），更新背景音乐，展示进度条，对特殊的窗口执行额外操作，上个窗口关闭，下个窗口打开等功能。

## （八）游戏设置模块

游戏设置模块主要包含 settings 类。我们可以在开始界面和若里村访问游戏设置，并对游戏设置进行调节。settings 设置包含音量调节功能（可以用来调节游戏中 BGM 的声音大小），一个虚设的游戏难度调节功能和两段作者的话。

## （九）彩蛋设计

在游戏中，有一个一直进不去的灵魂圣殿，直到达成结局一拿到戒指，并且进行二周目继续游戏，才能进入灵魂圣殿，看到最终反派的回忆录。

# 3 分工

王骏达负责了图片及音乐资源，音效动效背景的实现以及所有模块代码的实现。

孟凡荣负责了文字内容部分。

架构设计，作业报告及游戏流程设计为两人共同完成。

## 4 项目总结与反思

在功能层面，本组比较好的实现了大量场景之间的切换、音乐和音效的添加、动画特效的展现等功能，但比较遗憾的是没有来得及实现键盘操纵移动战斗的功能，未来可能会补上。

在代码层面，最开始工作时，由于对 QT 的不熟悉，许多代码抱着试一试的想法，命名和注释较为随意，代码可读性较差。在后续也进行了许多修改和调整，极大的提高了代码的清晰程度。在未来的工程中也需要注意这一点。同时，由于在 QT 中会大量使用指针，最开始我们对于指针的释放不太在意，导致了許多程序崩溃的 bug，这也让我们在后续的开发中着重注意了指针问题，修复了崩溃 bug。

在合作层面，最开始两个人并未使用 github 进行合作，导致双方的修改较难合并，这也降低了我们的效率。并且初步计划时比较仓促，分工不太明确，合作效率也较低。但在路演后，我们进行了调整，明确的责任分工和完成期限，大大提高了我们的工作效率。我们也逐渐熟练运用了 github 和 git 进行版本管理和存储，使得后续程序开发较为方便快捷。

总的来说，这是一次独立开发游戏的难得体验，和同学的合作也十分愉快，在编程中磨练了我们面向对象编程的能力，也掌握了许多合作方法。我们未来还应该继续提高对细节的把控，更加熟练的运用合作方法。

《Oliver》还很简陋，和我们想象中还有不少差距，未来我们也会继续完善这个游戏。最后，感谢所有玩《Oliver》的人，感谢你们玩我们的游戏！