

Qt 程序设计作业报告 组号: 16

Chroma

一、程序简介

这是一个关于“颜色”的故事。独自一人拥有五种色彩的英雄踏上黑白世界的旅途，在路上，ta 与被困在战争中的角色们相遇，帮助他们找回原有的色彩，同时逐步揭开诅咒这个世界的巫师的真相。

二、程序类型：剧情向角色扮演类游戏

三、程序功能介绍

1. 游戏部分

玩家可通过箭头键操纵英雄在地图上左右移动，通过空格键与地图上的人物、点位进行交互。交互以对话框的形式进行，文字滚动出现，通过选择相应选项，英雄可以推进剧情发展、获得不同道具。地图上设有传送点，英雄接触传送点可进入下一个地图或下一个场景。

道具存储在背包中。玩家可打开背包界面查看道具信息及使用道具。每一件道具对英雄有不同的提升效果。最终与巫师的战斗采用回合制，玩家需采取一定的道具使用策略方能获胜。根据战斗的不同结果，剧情会走向不同的结局。

2. 游戏支持及 UI

玩家中途可暂停游戏，返回开始界面后可选择开始新游戏或继续游玩。

在暂停界面及主菜单中，均可打开设置界面，调节游戏 bgm、按键及剧情音量。

所有按钮为自定义按钮，鼠标悬浮与点击有相应音效。窗口实现 BGM 循环播放，不同场景有不同 BGM。

四、窗口设计

由于游戏剧情具有不同场景、每个场景有不同地图，所以采用双核进行管理。游戏中的 QGraphicsView 有两个，一个是 mainwindow 中的 view，另一个是 scene。

游戏主要窗口为 mainwindow，管理两个 QGraphicsView 与所有场景。负责开始菜单与设置界面的呈现、场景的切换、最终 boss 战的启动，及背景音乐的播放。

scene 部分用于显示游戏地图 map 及背包界面、暂停界面。在不同编号 scene 的构造函数中，直接将所有地图及地图中的角色加载完毕。场景编号存储在全局变量中，便于玩家存档与读档。场景结束时，向 mainwindow 发送信号，切换至下一个场景。

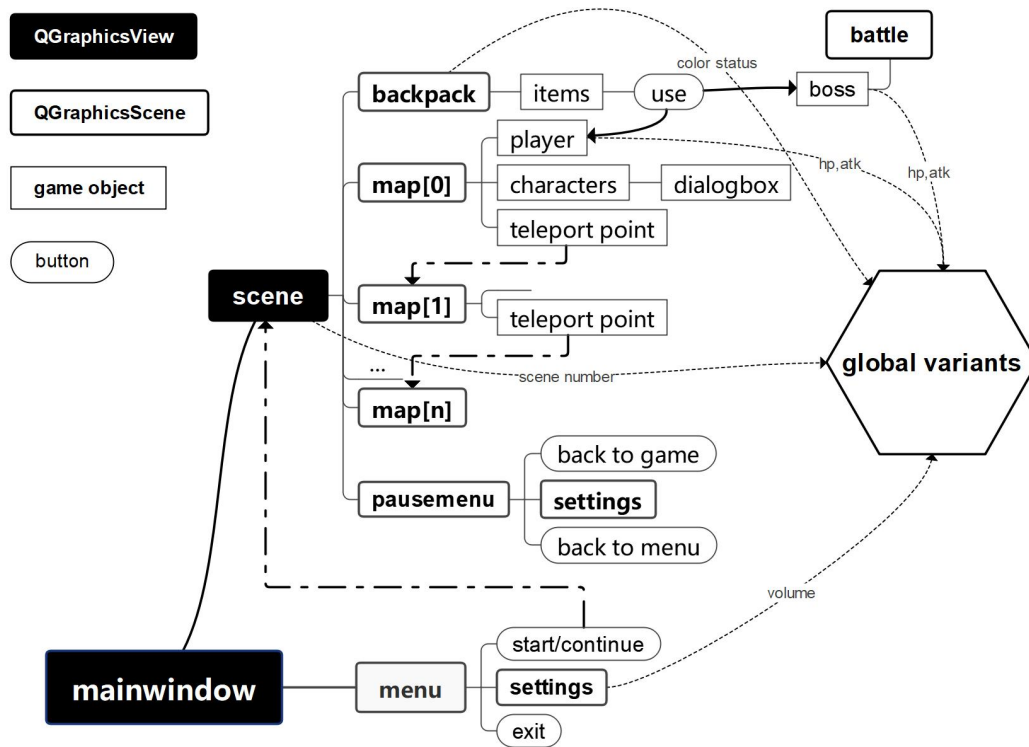
地图：地图上可放置角色、交互点、传送点。玩家与这些点交互时，产生对话窗口。

背包：背包界面显示当前拥有的道具，提供使用选项以修改角色属性。关闭背包后返回当前地图。

暂停：场景切换到暂停界面。选择返回游戏时，向 scene 发送信号，返回当前地图；选择返回主菜单，则向 mainwindow 发送信号，将 view 转换为开始界面。

设置：在主菜单与暂停界面中都可进入设置界面，但这两个界面在不同的 view 中，是不同的 QGraphicsScene，事实上由读取全局变量生成。

Boss 战为单独的 battle 类，具有背包场景与地图场景的双重功能。接收到最后一个 scene 的结束信号后，mainwindow 构造并显示 battle 场景。



五、类设计

Mainwindow

程序的主窗口，继承自 `QWidget`。可接收来自 `scene`、`battle`、`menu`、`settings` 的信号，在成员 `QGraphicsView` 中实现菜单展示、设置界面展示。单击开始游戏或继续游戏，将当前 `view` 隐藏，将游戏场景显示。收到场景结束信号时，构造下一个场景并显示，如当前场景为最后一个场景，则构造战斗场景并显示。包含 `QMediaPlayer`，可循环播放 BGM。

Scene

游戏窗口，继承自 `QGraphicsView`。在构造函数中，根据当前场景编号生成若干张地图，并在地图中添加角色、传送点等。接收来自 `backpack` 和 `pauseMenu` 的信号，实现在 `map`、`backpack`、`pauseMenu` 三个场景之间切换。场景结束时，向 `mainwindow` 发出信号，生成下一个场景。

Startmenu

菜单界面，继承自 `QGraphicsScene`。包含 `new game`、`settings`、`exit` 三个按钮，如当前有正在进行的场景，则添加 `continue` 按钮。`Scene` 内的 `pauseMenu` 也是这一类的实例。不同按钮按下的信号连接到不同的窗口或场景，其中主菜单的 `exit` 按钮用于程序退出。

Setting

设置界面，继承自 `QGraphicsScene`。包含 `BGM`、`buttons`、`story` 三个 `Qslider`，分别用于调节全局变量中的背景音乐音量、UI 音效音量、剧情音量。

Globalvar

全局变量。包含 `colorStatus` 数组，记录颜色的使用情况；`bgmSound`、`toolSound`、`storySound` 记录音量大小设置；`curScene` 与 `prevScene` 记录当前场景与上一次游戏的场景编号；`btnSize`、`boxSize`、`picSize` 用于统一按钮的大小；`heroHP`、`wizHP`、`last_act` 用于记录战斗情况。拥有三个改变音量值的成员函数。

Mybtn

自定义按钮类，继承自 `QPushButton` 和 `QGraphicsPixmapItem`。具有三个构造函数：复制自 `QPushButton`，以字母串初始化，或以图片初始化。鼠标悬浮及按下时，改变背景颜色，并播放悬浮与按下音效。如果一个 `mybtn` 是某个类的成员，该按钮按下的信号由该类的槽接收，再在该槽中发出该类的信号。

Character

人物，继承自 `QWidget` 和 `QGraphicsPixmapItem`。构造时，设置其移动速度与图像。具有 `interaction` 属性，当按下空格键时，设置属性为真。属性为真时，可开启交互。通过箭头键或屏幕上的左右按键进行左右移动，移动时根据移动方向展现不同的图片，并播放行走音效。

Map

一维地图，继承自 `QGraphicsScene`。可通过 `setCharacter` 指定角色初始位置，`addInteractionPoint` 添加交互点并指定对话内容，`addTeleportPoint` 添加传送点并指定目标地图。包含一个定时器，每 100 毫秒检查一次交互或传送。如果有交互点与角色重合，并且角色按下了空格键，则弹出对话框显示对话内容，并停止角色的移动和交互。如果有传送点与角色重合，则将角色从当前地图移除，添加到目标地图中，并停止当前地图的定时器。

Dialoguebox

对话框。包含一个 `QMessageBox` 用于显示对话内容，两个 `QLabel` 用于显示角色立绘。通过读取 JSON 文件，获取说话者及内容，并为每个选项创建一个按钮。从选项对象中获取背景图像文件名用于更新背景图像、`backpack` 指令用于更新背包，或使用选项作为文件名加载和解析一个新的 JSON 文件。通过一个 `textBuffer`，将字符逐个取出，追加到消息框的文本中，从而实现文字逐个显现的特效。

Backpack

一维存储空间，继承自 `QGraphicsScene`，以 `QList` 实现道具的存储与消耗。根据背包内容，在方格中展示道具，利用 `QMap` 将格子位置映射到道具编号。单击道具图片，右方的标签显示道具名称与道具描述。单击下方“使用”按钮，消耗当前选中道具并重新绘制背包界面。

Item

道具，消耗品，使用后修改角色的属性或修改战斗逻辑。内部存储道具编号对应的名称、描述、图标。

Battle

战斗部分总类，管理战斗逻辑与界面，无实际函数，负责发送战斗开始信号。

Battle_panel

实现战斗过程中下方面板。`initialize` 函数初始化面板，通过 `QGridLayout` 实现在 `QMainWindow` 面板上的道具 `button`、战斗 `button`、道具 `label` 与 `hero` 血条的布局；`battle_action` 函数实现了战斗的逻辑，根据 `initialize` 函数中确立的按钮的信号机制以及 `globalvar` 中 `colorStatus` 不断更新面板中按钮状态，并且调用 `update_hp` 函数实现 `hero` 与 `wizard` 血条的更新。

Battle_scene

实现战斗的场景。`showscene` 函数中实现背景渲染，在场景上方绘制 `wizard` 的血条，通过在其中建立 `battle_panel` 对象并利用 `addWidget` 函数将其 `hero` 面板加入场景，同时将 `wizard` 的血条作为参数传入以实现 `battle_panel` 中对于 `wizard` 血条的更新。

六、小组分工

平行：

负责程序整体设计、功能设计、界面设计与类的架构。

负责窗口之间的连接及 UI 特效、游戏音效。

主要编写 `mainwindow`、`scene`、`mybtn`、`globalvar`、`startmenu`、`backpack` 界面部分。

负责程序中所有图片的绘制及音效素材的收集。编写剧情及文案。

负责小组分工与沟通，GitHub 版本管理，制作项目报告及成果视频。

徐佳怡：

负责游戏内物体及相应特效。

主要编写 `map`、`character`、`dialogbox`、`settings`、`backpack` 逻辑部分。

范佳玥：

负责游戏战斗部分制作、呈现及数值策划。

主要编写 `battle`、`battle_scene`、`battle_panel`。

七、项目总结与反思

平行：

本次大作业是我第一次制作一个较大规模的程序，也是第一次与他人合作写程序。由于一直以来有一个做游戏的心愿，所以主动当了队长，负责了程序整体的设计和所有人工作的整合，同时自告奋勇当了美工和文案。程序架构的搭建以及更具体的窗口处理确实很好地锻炼了我的程序设计能力和编程能力；Qt 作为一门我之前不熟悉的语言，锻炼了我的学习能力；而安排进度、与组员沟通，也让我收获了与他人协作的技巧。当然，由于能力与精力有限，最终的呈现效果只实现了一开始的设想的一小部分，也令我体会到游戏制作的不易。本次小组合作确实出现过不少问题，比如组员不怎么会用 GitHub，导致实际上是组长一个人把 GitHub 当云盘，实际上组员之间的版本进度是不统一的，最终还是落为微信发代码；还有前端后端划分不明确，一开始的分工也不是很清晰，导致制作的效率降低——这点我作为程序的设计者和小组的分工者是负有责任的。当然，虽然遇到了种种困难，最终大家也都互相帮助着克服过来了，真的很感谢两位伙伴。最后想说，这个程序最令我骄傲的一点是所有的图片和文案都是原创的，（不存在的）画技似乎也提升了！

徐佳怡：

我们这次用 qt 制作的名为 Chroma 的游戏是一个结合了情节推进和对抗小游戏的有趣的项目。我在这个项目中负责的是，根据平行同学，即游戏的总设计师，对游戏的设计及需要，写出每个 class 的代码框架。在写代码框架的过程中，我遇到了一些困难和挑战，比如如何设计合适的接口，如何保证各个 class 的独立性，以及如何与队友沟通和协调。这是我第一次与他人合作写一个项目，也体会到与他人合作和自己独立完成项目的不同。通过这个项目，我学到了很多关于 qt 编程和团队合作的知识和技能，我的编程水平和沟通能力都得到了提升。总的来讲，我认为大家这次的表现都很棒，但是也有一些可以改进的地方，比如在一开始，由于对 qt 的不熟悉以及此前缺少合作完成项目的经验，我们一直难以做出明确的分工，也导致大家都有些无从下手。

范佳玥：

QT 大作业让我了解到团队合作实现项目的流程，从最初的分工到各自实现再到最后的整合，中间的不断修整与 debug 让我了解到了团队项目中可能存在的问题以及协调的不易。同时，从零开始学习 QT 让我体验了如何从头开始自学一个新软件，学会了通过 documentation 与网上教程迅速上手。