

Psu! --osu-mania like music game

北京大学2023春程序设计实习QT大作业

编译环境

Qt 6.5.1 with multimedia, qmake

****System Requirement: **** Windows 10, 11; MacOS (greater than 11.0)

程序功能介绍

支持多曲目的四键音游，现有9首曲目，可以通过修改data文件自行添加曲目。

项目模块和设计细节

1. 主界面

主界面继承自QMainWindow，代码实现见MyMainWindow类，使用QStackedLayout处理不同界面的切换
界面切换的具体实现方式（这样就不用每次新开一个window）

```
QWidget *w = layout->currentWidget(); // layout是QStackedLayout*对象，为
MyMainWindow类的成员变量
layout->removeWidget(w); // 删除当前的界面(QGraphicsView*)
w->deleteLater();
QGraphicsView* game_view = initGame(); // initGame()返回游戏主界面的QGraphicsView*
指针
layout->addWidget(game_view);
layout->currentWidget()->setFocus(); // 强焦点于(Strong Focus on)当前界面，以保证程
序能响应键盘输入
```

主界面和其他界面的相互跳转

开始游戏：跳转至选择界面 设置：跳转至键位设置界面 退出：退出游戏

2. 选择界面

选择界面读入内部的tree.txt，该文本档包含所有曲目信息，也可以通过修改tree.txt增添曲目。选择界面的代码实现位于selection.cpp中。

2.1 选择栏

选择栏包含多个按钮，每个按钮对应一首曲目，通过WS上下滚动，通过Space选择。同时在滑动时增加了动画。

2.2 信息栏

信息栏包含曲目信息，包括曲目长度、谱师、难度等级；同时会读取应用文件前两层的/data文件中的历史记录，记录当前最好分数。

3. 游戏界面

游戏界面继承自QGraphicsScene，代码实现见GameScene类；启动游戏界面后，程序运行的主要步骤如下 (逻辑主要体现在GameScene::startGame函数中)：

1. 从txt文件中读取谱面数据到GameScene的成员变量QVector<QPair<int, int> > tm[11], tm[i]是第i个音轨对应的下落键的QVector数组；tm[i][t].second == -1代表短键，否则为长键
2. 初始化游戏界面的背景 (包括计分栏、背景图、下落轨道)；
3. 初始化下落键
4. 计时器开始运行
5. 播放音乐

3.1 下落键的实现

FallingKey类为下落键的代码实现，继承自QGraphicsPixmapItem, 同时兼具长键和短键的实现，预留了自定义外观的接口 (成员变量stylePath).

FallingKey同时支持长键和短键，通过变量longKey辨别长短键

GameScene中使用keyFallingTimer每隔INTERVAL毫秒判断是否有键下落，使用e_timer记录当前时刻；如果暂停，暂停时间会累加到pauseTime里面 (pauseClock记录上一个开始暂停的时刻)；目前时间方面e_timer和clock()两种方式并存，建议之后统一到e_timer，因为clock记录的是cpu时间，跨平台可能有比较多的问题

与GameScene类的交互如下

1. bool isFalling变量用于判断键是否在下落，方便GameScene的暂停；
2. GameScene类使用QTimer *keyFallingTimer判断当前是否有键下落，QTimer::timeout连接槽函数GameScene::timerFallingKey
3. 键在下落到底部的时候会emit signal: FallingKey::endOfFalling (signal) -> GameScene::handleEndOfFalling (slot)

3.2 暂停功能的实现

使用一个队列 (事实上是QMultimap实现的优先队列) 保存所有在下落的键，当按下暂停时，在当前界面显示暂停小组件并且停止所有动画。

3.3 判定功能的实现

程序读取音乐文件，音乐文件包含BGM，历史记录，内部记录了每个键应该落到底部的时间，即按照时间序列进行判定即可。同时按照与标准时间的时间差制定了Perfect, Good, Normal和Miss四级标准。

3.4 分数栏的实现

分数栏位于右上角，包括当前分数，当前Combo和当前时间 (单位：ms)

4. 结算界面

结算界面会显示通过游戏阶段记录的准确率、最大Combo以及得分，按照准确率判断等级并显示为S, A, B, C, D，同时会记录当前最好分数。（由于QT本身的限制，并不能将该文件内嵌至可执行文件中），详见 `ending.cpp`。从游戏界面过渡到选择界面有一个动画。

5. 设置界面

设置界面可以设置游戏音量，同时具有四组按钮可以设置游戏键位。详见 `settings.cpp`

小组成员分工

龙天龔：判定内核、动画、文件读写；以及设置界面，选择界面的实现，同时负责Debug和调试。冯睿容：UI实现，主界面UI、主游戏界面UI，下落键的实现。楼翰涛：初始框架搭建、同时负责Debug和调试，并且负责音乐文件制作和导入。

项目总结与反思

我们在制作时，一开始遇到的困难包括需要新建窗口，动画效果不尽人意，以及下落判定问题。我们通过查找资料，不断调试最终解决了这些问题。然而现在仍有QT本身的问题无法解决，包括：文件路径问题、下落Interval问题（如果将下落Interval调成小于15，则QT由于在15ms的间隔中需要处理的事件过多，就会产生各种bug），这些问题的解决还需要继续探索。

通过此次QT作业，本组同学对C++多文件编写代码，C++面向对象语言设计有更深入的理解。同时，我们注意到经常会在.h文件中写代码，而且也经常出现野指针的问题。通过此次作业，我们也逐渐改善了代码习惯。