基于 QT 的游戏修改器的设计与实现

喻 锐 白俊鸽
成都锦城学院计算机与软件学院 四川 成都 611731

【摘要】随着当今科技的发展，游戏已经成为了大多数人忙碌一天后选择放松的方式，但某些游戏需要达到一定分数才能解锁所有关卡，本文设计并实现的这款游戏修改器旨在让使用者可以直接达到解锁关卡的条件。

【关键词】游戏修改器；QT；多线程

1 引言

在互联网飞速发展的今天，市面上出现了各种各样的休闲游戏，但很多游戏存在需要达到一定的分数目标才能解锁下一关卡的设定，对使用者造成了一定的困扰。本文设计的这款游戏修改器，可以给用户一个轻松，愉快的游戏体验。

2 开发环境介绍

QT 作为一种基于 C++ 的跨平台 GUI 系统，能够给用户提供构造图形界面的强大功能。为满足用户构造复杂的图形界面系统的需求，提高图形用户界面的响应速度，QT 提供丰富的多线程编程支持。[1]

3 主要功能描述

本游戏修改器的功能主要分为两大模块，一为修改游戏比分功能，二为显示历史修改记录功能。在打开本游戏修改器的可执行文件后，用户可以通过在初始的菜单界面中选择进入游戏修改后，根据刷新出的进程列表选择当前正在进行的游戏进程名字，选择打开当前用户正在进行的游戏所对应的系统进程，之后根据当前的游戏得分选择进行第一次查找，这时会显示第一次查找到的地址列表，此时需用户进行第二次查找，将查找到的唯一地址选中后，可以根据当前游戏的设置达到解锁剩余关卡的规则来输入想要得到的分数，点击修改，之后，如果弹出一个窗口显示修改成功，那么所对应的游戏分数则修改成功。此外，可以根据初始的菜单界面中选择点击查看记录，可查询并显示出之前所有的修改记录。

4 界面设计

应用界面分为三块，一为供用户选择功能的菜单界面，用户在此可选择进入游戏修改界面或者显示历史修改记录界面；一为显示历史修改记录界面，整体为一张显示历史修改记录的表，下方为一个返回的按钮；最核心的游戏修改界面整体左边为当前系统的进程列表，左下方为选择刷新或者打开进程，右上角依次为目标进程地址，第一次和第二次查找按钮，第一次查找进度条，查找后的地址列表，目标地址，选中目标地址按钮，修改分数按钮。游戏修改界面如下图（图 1）所示。

图 1 游戏修改界面

5 功能实现过程

5.1 界面控件

在对于核心功能游戏得分修改的界面设计为，地址列表，进程列表等需要将信息显示给用户的使用 QT 提供给用户的表格控件；刷新按钮，打开按钮，第一次查找按钮，第二次查找按钮，选中按钮，修改按钮等需要用户点击的使用 QT 提供给用户的按钮控件；目标进程，查找分数，目标地址，修改为等显示给用户一些信息的使用 QT 提供给用户的文本显示控件。

5.2 游戏分数修改功能

作为一个游戏修改器，最基本的功能实现就是通过打开的游戏进程名和根据当前游戏得分查找到的对应内存地址，来修改游戏比分的功能。具体实现步骤为：首先，通过操作系统提供的对当前进程操作的接口函数来创建一个当前系统进程快照的结构体，并取出此结构体中的内容即当前系统运行的进程唯一标识符和对应的名字，通过设置项函数显示在由 QT 提供的表格控件上，使用 QT 提供的选中表格某一项的接口函数选中用户想要进行修改的游戏进程，由操作系统提供的打开进程的接口函数打开想要进行修改的游戏进程。下一步应该思考的是如何在打开的进程中找到游戏得分的内存地址，并根据修改此内存地址空间里面的数据达到修改游戏分数的功能。鉴于通过读操作系统进程函数读取当前打开进程内存后，第一次搜索到与当前游戏分数相等的有多个地址内存来看，应让游戏继续进行，让分数增加一次，然

后将当前增加后的游戏得分与第一次搜索到的地址列表中的地址空间里面的数值逐一进行对比，将此次对比后相等的地址空间进行标记，并确定为最终目标地址。最后，将此地址选中，通过操作系统提供的写内存地址接口函数来进行对内存空间存储数据的修改，达到最终修改游戏得分的功能。值得一提的是，由于此过程中用到的读写操作系统进程内存函数可能会由于访问或者修改到操作系统内核的一些基本指令或者功能而发生一些意料之外的错误进而导致一定的危险性，因此需要执行由操作系统提供给用户的保护权限函数来提供一定的访问使用权限，防止操作系统判断危险而终止读写内存。

由于在主窗口中只有一个主线程在进行工作，而第一次查找与当前游戏得分相等的地址空间时需要占用操作系统内核的大量时间进行运算，因此此界面的主线程被查找计算功能所独占，此时用户无法再对其他窗口进行操作，必须等待查找结束后才可以进行其他工作，大大影响了用户的使用体验以及降低了操作系统内核的性能和效率。因此可以使用多线程技术，在项目中增加一个子线程，通过对线程类中的运行函数进行重写，专门用于第一次查找的比较内存计算，而将最终计算出的结果通过 QT 独特的信号槽机制先进行子线程中的信号与主线程中槽函数的连接，在子线程计算完成后通过带参数的信号将结果发送给主线程，再由主线程接收后显示在用户界面中。最终效果展示界面如下图（图 2）所示。

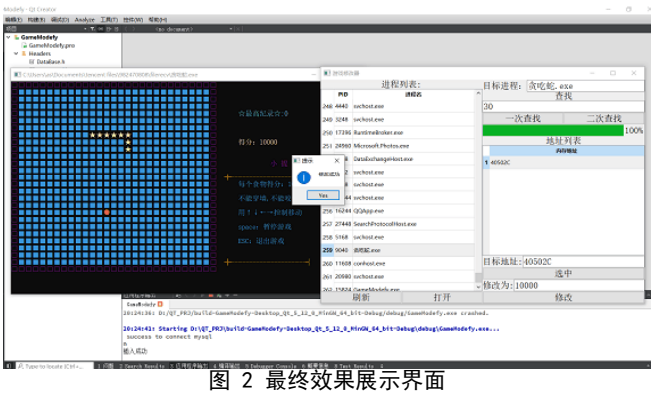

图 2 最终效果展示界面

5.3 显示历史修改记录功能

对于用户想要查看以往修改记录的功能实现。首先选择 Mysql 数据库作为数据的永久存储对象。要访问数据库，需要首先建立到数据库的连接。QT 提供了一个类：QSqlDatabase 来处理数据库的连接。[2] 在修改游戏分数界面成功修改后，通过 QT 提供的对当前系统时间操作函数获取到当前日期，通过获取控件内容的接口函数将进行修改的游戏名，修改前的分数，修改后的分数利用 QT 提供的插入对应数据库及表的接口函数实现保存在数据库中。当用户选择进行查看历史修改记录时，再次使用该接口函数查询数据库中对应表的所有记录后用设置项到表格的函数显示到表格上。

在 QtSql 模块中提供了很多封装好的类，其中最重要的几个类为：QSqlDatabase，QSqlQuery，QSqlQueryModel，QSqlTabelMode,QSqlRelationalTableModel，只要掌握了这几个类的使用方法，那么无论是本地数据库的开发还是网络数据库的开发都变得轻而易举。[3]

6 测试及结论

项目完成之后，对本游戏修改器进行了如下功能测试以及技术总结。

主要测试了本游戏修改器能否正确修改游戏得分的功能，在游戏修改中第一次查找时能否进行当前窗口的拖动，隐藏以便于使用者在等待查询结果的过程中能否进行其他工作的功能，能否保存当前修改记录并且正确显示出历史修改记录的功能。

在此测试过程中，发现如果按照开发者的提示来逐一进行操作，上述功能都能够准确实现，达到预期效果。但如果没有经过开发者的提示，用户在不知道流程的情况下，还未打开目标进程就进行查找功能或者修改得分的功能，那么会造成不但不能实现修改得分的目的，反而会让游戏修改器程序崩溃，用户也并未知道具体发生了什么情况。这对于用户的使用体验来说非常不友好。

在不断进行测试后，此问题的最终解决方案为两条：（1）在开始界面中增加一个操作流程的提示选项，用户点击后可以进入由开发者给出的提示界面，此提示界面中会给出详细，正确的操作使用流程，便于使用者后续的正常使用，在了解了使用流程后可以点击返回按钮退出到选择界面。（2）将一些按钮初始化为不可点击状态进行保护，相当于给一些可能会引发危险的功能上了锁，只有当触发一定条件后此保护状态被解除，状态也随之修改为可点击状态。例如，只有在用户成功打开某进程后，具备了查找进程内存空间的所有条件后，才将第一次查找的按钮设置为可点击使用状态。

由于本游戏修改器的开发成本较低，实现简单易懂，因此还存在一些可以优化的地方。对于本项目的具体优化方向为只通过一次根据当前游戏得分查找就能准确地找到对应内存地址，从而进行修改；以及对于一些有冷却时间的游戏，增加一个可以修改游戏冷却时间的功能。

7 结束语

本文介绍了如何使用 QT 提供的一些基础控件和封装好的接口来实现一款本地游戏修改器，该游戏修改器具有可并发操作，高效，所占内存小，简单易实现的特点，但还存在未能囊括市场上所有游戏的问题。适合于 QT 的初学者进行练手的学习以及部分希望通过修改游戏得分以解锁其他关卡的游戏玩家。

【参考文献】

[1] 黄宇东，胡跃明，陈安．基于 Qt 的多线程技术应用与研究 [J]．教育技术导刊 (10 期):40-42.

[2] 孔翔鸣．Qt 本地数据库开发 [J]．电脑知识与技术，2017，13(010):4-5.

[3] 张治国，董西广．基于 QT 平台的数据库编程 [J]．福建电脑，2011，27(003):165-167.