南开大学 计算机科学与技术

姓名 朱乐晨

学号 2312194

班级 计卓

2024年5月11日

高级语言程序设计

实验报告

目录

[高级语言程序设计大作业实验报告 1](#_Toc21286)

[一. 作业题目 1](#_Toc29969)

[二. 开发软件 1](#_Toc18364)

[三. 课题要求 1](#_Toc742)

[四. 主要流程 1](#_Toc25240)

[1． 整体流程 2](#_Toc20452)

[2． 算法或公式 2](#_Toc25354)

[（1） 自动探索迷宫出口算法 2](#_Toc22508)

[（2） 自动生成迷宫地图算法 2](#_Toc28665)

[3． 单元测试 3](#_Toc24674)

[五. 单元测试 3](#_Toc28265)

[六. 收获 4](#_Toc23116)

[1．](#_Toc10799) 以QSoundEffect为例的Qt音频类头文件的使用

[2．](#_Toc16275) 基于QTimer的计时器添加和使用

高级语言程序设计大作业实验报告

1. **作业题目**

**《寻找失落的篮球》——基于qt和c++的迷宫探索小游戏。**

1. **开发软件**

**Qt 6.7.0、photoshop2018、visual studio2022等**

1. **课题要求**
2. 使用可视化图形界面和c++语言进行编程。
3. 对程序进行测试和调整。
4. 建立良好的程序架构。
5. 设计可行的实现算法。
6. **主要流程**
   1. **整体流程**

实现思路：

游戏界面主要由4个窗口组成：

1. 主窗口MainWindow:形成游戏界面的框架，负责展示游戏的开始界面
2. 关卡选择窗口ChooseLevelWindow:显示选择关卡时的界面
3. 游戏窗口PlayWindow:显示游戏过程的实时界面
4. 地图编辑窗口MapEditWindow:负责生成和设计游戏地图

游戏组件主要有两类：

1. 按钮组件ImgButton:将基于QPushButton封装的图片按钮与鼠标点击操作进行交互，从而可以控制界面的切换和游戏功能的实现。
2. 地图模块Tile:主要是组成迷宫地图的基本图块，赋予不同类型的图块以不同的性质，实现隔离玩家和判断是否达到游戏完成条件的作用。

其他辅助类文件：

1. 迷宫数据记录工具MazeData:负责记录现已存在的迷宫信息和游戏进度，还包括玩家新创建的地图，可以用于存档和读档。
2. 资源素材res:包括用于游戏的按钮、封面、地图、背景、音效等。
   1. **算法或公式**
3. 自动探索迷宫出口的设计算法实现：

先将地图上的位置都标记为false,并用false代表未被走过的路径。随后，从玩家位置向四个方向进行遍历，当遇到墙时停止此方向的遍历，由每一小格不断向四周扩散，并判定此格是否为终点，如果是终点就停止遍历。当达到除了传递给自己的小格其他三面都是墙时（死角）也会停止遍历。停止遍历时如果此格不是终点，则将起点到该点的所有格标记为true，即已被走过的路径。最终会得到所有的死角和终点，而只有从玩家位置到终点的那条路径还是false的状态，因此将这条路径上的每一个小格标记出来作为自动探索出的出口路径。当不存在这样的出口路径，就会直接触发失败条件作为替代。(详见playwindow.cpp)

1. 自动生成地图的设计算法实现：

最初先将地图的所有格初始化为墙，然后从坐标为（1，1）的格开始变为路并进入栈中，依次为基点判断此点是否与周围连通（周围相邻的格子中至少有一格为路），如果不连通就继续生成新的路。此过程中也会不断判断整个地图是否已被连通，如果已连通，将直接结束生成，如果未被连通，则会设置当前点为最后一个点，寻找该点的“潜在邻居”（上下左右的墙均未被打通）。如果存在“潜在邻居”，则将任意一个“潜在邻居”为“真邻居”并将当前点与“真邻居“之间的墙打通，否则将栈顶元素弹出并再次判断地图是否连通，依次循环直至最终地图完全连通。在结束时还会对地图是否连通再进行一次验证，判断是对地图的奇数行和奇数列均进行遍历，如果其上的小格都是路则地图为连通的，否则还需退回地图并重新生成。(详见mapeditwindow.cpp)

* 1. **单元测试**

程序运行过程：

14:48:59: 为项目MazeMaster执行步骤 ...

14:48:59: 配置没有改变, 跳过 qmake 步骤。

14:48:59: 正在启动 "C:\Qt\Tools\mingw1120\_64\bin\mingw32-make.exe" -j16

C:/Qt/Tools/mingw1120\_64/bin/mingw32-make -f Makefile.Release

mingw32-make[1]: Entering directory 'C:/Users/朱乐晨/Desktop/Qt-MazeMaster-master/build-MazeMaster-Desktop\_Qt\_6\_7\_0\_MinGW\_64\_bit-Release'

g++ -Wl,-s -Wl,-subsystem,windows -mthreads -o release\MazeMaster.exe @release\object\_script.MazeMaster.Release C:\Qt\6.7.0\mingw\_64\lib\libQt6Multimedia.a C:\Qt\6.7.0\mingw\_64\lib\libQt6Widgets.a C:\Qt\6.7.0\mingw\_64\lib\libQt6Gui.a C:\Qt\6.7.0\mingw\_64\lib\libQt6Network.a C:\Qt\6.7.0\mingw\_64\lib\libQt6Core.a -lmingw32 C:\Qt\6.7.0\mingw\_64\lib\libQt6EntryPoint.a -lshell32

mingw32-make[1]: Leaving directory 'C:/Users/朱乐晨/Desktop/Qt-MazeMaster-master/build-MazeMaster-Desktop\_Qt\_6\_7\_0\_MinGW\_64\_bit-Release'

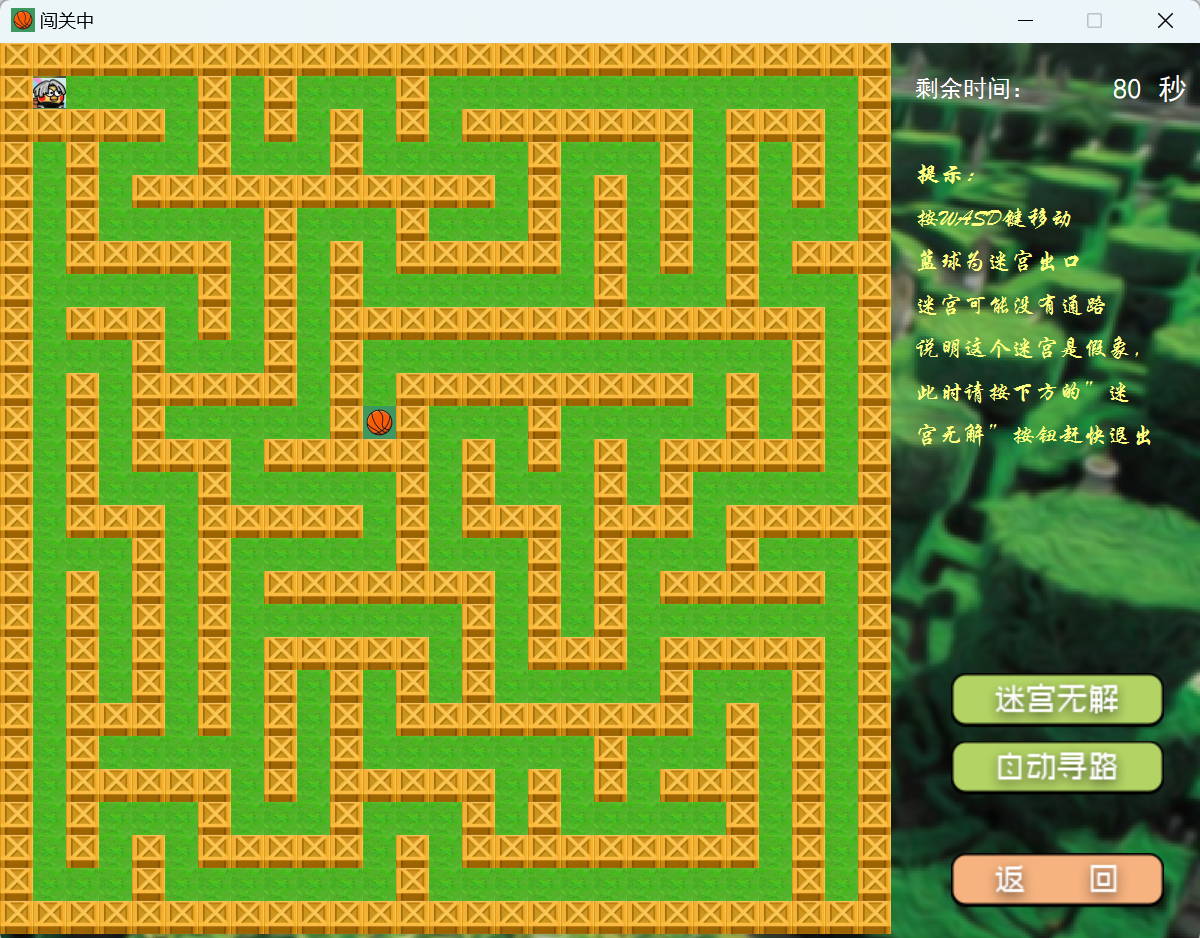
14:49:00: 进程"C:\Qt\Tools\mingw1120\_64\bin\mingw32-make.exe"正常退出。

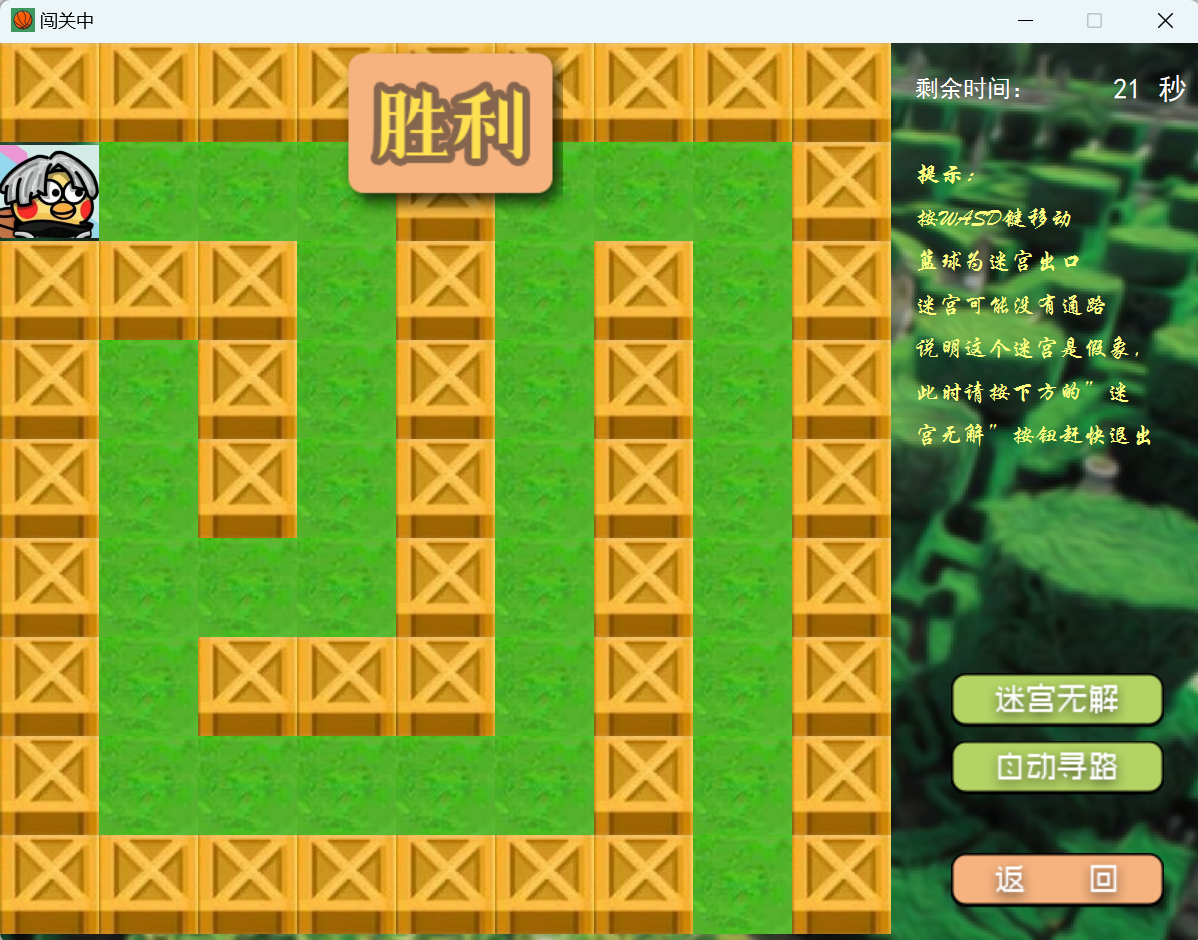
14:49:00: Elapsed time: 00:01.

1. **单元测试**

### 测试结果

程序能够顺利运行，并且具有一定的趣味性和可玩性，可以作为一个比较典型的休闲类迷宫小游戏。

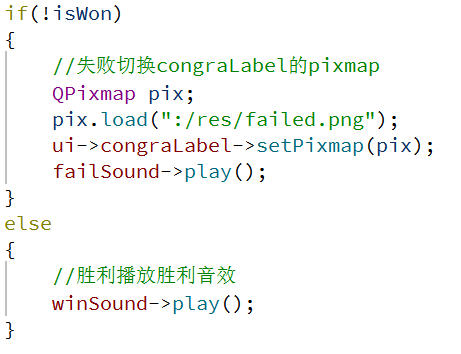
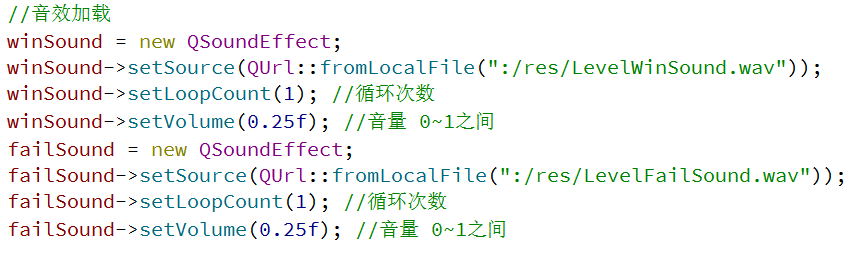
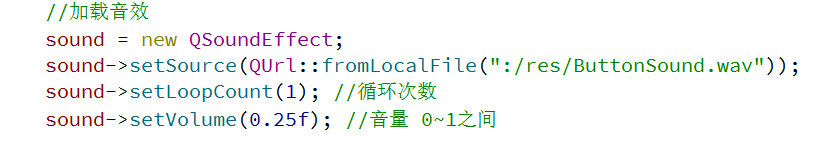




1. **收获**
   1. **QSoundEffect的使用**

在此之前，我一直对如何插入和释放音频感到十分困惑，在完成大作业的过程中，我通过查找学习明白了如何使用QSoundEffect、QSound、QMediaplayer等播放音频文件。在这个小游戏中主要用到的都是较为短促的音乐片段，因此我选择了QSoundEffect作为工具。另外，我也了解到qt只能接受wav类型的音频，因此我舍弃了不少mp3、flac等类型的音效，花了更多时间来寻找适合Qt背景的wav音效。

具体使用有如下几个例子：



包括按钮音效、胜利音效、失败音效三种特定音效，会在条件触发时自动播放。

* 1. **基于QTimer的计时器**

此游戏限定了最长通关时长，而要实现这一功能需要通过计数器来实现。

在此过程中，会先根据地图复杂程度生成一个对应的计数总时长，然后再通过每秒递减的方式进行计数，当计数器的值为0时如果还未通关将判定为通关失败。另外，计时器只有在玩家正在闯关时才会进行计时，退出关卡时会保存现在的数值并静止计数，直到再次进入此关卡才会再次开始计时。在关卡成功时，计数器也会停止计时。当游戏在某一地图已经确定成功或失败时，再次进入该地图会重置玩家的起点和计时，为玩家提供多次闯关的机会。

在程序中的实际使用如下：

