

QT 大作业：魔塔魔改版

组号：74 组名：人工智能基础

组长：元铭宇(2200013153) 组员：廖丁鸥(2200013147) 梁泽名(2200013195)

一、程序功能介绍

本程序实现了经典游戏魔塔的简易五层版，并在原魔塔游戏中添加了许多不可思议的新功能。在主菜单，玩家可以选择“开始游戏”、“退出游戏”、“背景”三个选项之一。选择“开始游戏”将会开始属于玩家的特别的魔塔之旅，游戏中除了经典的魔塔元素之外还存在着小惊喜，游戏内的怪物数值也都经过了精心设计，希望玩家认真斟酌找出通往最终 boss 的一条正确的路，具体实现将在第二部分阐述；选择“退出游戏”将会退出程序；选择“背景”将会打开新界面来展示原创的背景故事，这是原本魔塔所没有的部分，其实现也将在第二部分完成。**本程序使用 QT6.5.1 对应套件构建。**

二、项目各模块与类设计细节

开始游戏：

我们把模块主要分为了主菜单窗口，游戏界面窗口，NPC 角色交互模块，答题模块

主菜单窗口 中我们设置了三个按钮，第一个按钮直接可以开始游戏，并且无法返回（毫无撤退可言！），第二个是退出游戏，点击后整个程序的生命周期结束。第三个是背景按钮，配备了一段背景音频的播放——dark ♂ 力 ♂ 王以及滚动字幕，还有一个可以往返主菜单的按钮。



游戏界面窗口 我们沿用了网上大多数 qt 开发魔塔的方法，首先自定义了一个 QMainWindow 的子类 main_windows，然后采用一定量的像素来实现地图以及属性界面，我们的窗口是 (14*50) * (20*50) 的大小，采用像素方法我们可以更好的模拟传统魔塔的那种风格，同时也方便后续的动画制作。还有就是由于我们要不断对角色进行移动，所以利用像素的优势可以每次只更新两个像素格，减少了运行的时间空间复杂性。另外我们 main_windows

里面的成员变量非常多，因为我们的代码都是对 main_windows 里的像素来进行操作，所以我们希望在写这个类的成员函数时不会涉及到其他的类，从而避免出现“-1: error: 1 duplicate symbol for architecture arm64”这种天杀的错误。当然这样带来的后果就是代码量很大--我们的 main_windows.cpp 有一千多行代码，这样查看和 debug 不是很方便，但是总体来说还是利大于弊。游戏实现上我们采取自定义 keypress 函数，每次按下按钮之后就进行一次 move 操作（判断是否移动以及进行移动）以及 show_pic () 用于刷新当前游戏界面，还有按下 P 能够弹出怪物的属性面板，再按一次则消失，具体请看 main_windows.cpp。在这个过程中一开始我们发现了一个致命的问题，就是我以为对于一个 QLabel 指针，如果你多次让它=new QLabel 这样操作而不 delete 的话，它是始终占用的都是同一块区域，但是运行游戏之后我发现程序越来越慢，甚至在我按了大概 20 次的移动键后它不仅会移动延迟非常大而且会自动退出。后来我用 mac 的 top 指令查看了内存占用才发现原来是一直不 delete 导致占用内存太大，至此问题才解决。

```
if(start)for(int j=0;j<14;j++)for(int i=0;i<20;i++){delete label[j][i];}
```

NPC 角色交互模块

呢，我们的主角从骑士变成了火柴人，这个是方便后期做动画，而 boss 和怪物我们全部采用传统魔塔的形象（从互联网上截图），然后我们自己设计了一组数值，保证极限状态下的通关。在交互上，第一，我从造梦西游游戏受到启发，设置了角色如果 3s 内不尝试任何移动就会自动进行一段动画，实质上是在合适的位置播放一段合适大小的视频，但是我们的主角的形象可以无缝衔接那段视频所以看上去就像是动画。这是我们的一

个亮点。



←（鸡你太美）

第二，我们还设置了一个火柴人打斗的视频，是一个红色火柴人和主角打斗。这个想法本来是我们想通过做定格动画来实现的，但是后来发现做定格动画不仅要自己画每一帧的图，而且还要实现图片滚动播放，这实质上就和播放视频没有区别嘛，所以我们就决定直接播放视频，但是做了一个优化，具体体现为从主角和另一个火柴人两个像素格的位置逐渐放大两个图片最后变成视频开头的样子，实际上也是想实

现无缝衔接。但是在这其中还有一个插曲就是本来想做一个其他像素格逐渐变透明的动画但是搞了一个下午都没搞出来，后来一查才发现如果 QLabel 以及被添加到父窗口的布局中，那么无法控制其透明度。还有最后与魔王的交流中，我们也设置了“类动画”，本来想通过 QPropertyAnimation 来实现一个个对话框的出现消失，但是不知道为什么实现不了（恼），所以采用了 QTimer 计时的方法，先提前在初始化的时候就布局好图片但是不显示。到了一定时间就让一张图片 hide，一张图片 show，实现动画的效果，

```

if(get_the_book==false){
    qDebug()<<"没有书3";
    QTimer::singleShot(1, [&]() {      Δ Pass a context
        imageLabel[1]->show();
    });
    QTimer::singleShot(2000, [&]() {      Δ Pass a context
        imageLabel[1]->hide();
        imageLabel[2]->show();
    });
    QTimer::singleShot(4000, [&]() {      Δ Pass a context
        imageLabel[2]->hide();
        imageLabel[3]->show();
    });
    QTimer::singleShot(7000, [&]() {      Δ Pass a context
        imageLabel[3]->hide();
        imageLabel[4]->show();
    });
    QTimer::singleShot(9000, [&]() {      Δ Pass a context
        imageLabel[4]->hide();
        imageLabel[5]->show();
    });
    QTimer::singleShot(12000, [&]() {      Δ Pass a context
        imageLabel[5]->hide();
        this->hide();
        for(int ii=0;ii<14;ii++)for(int jjj=0;jjj<20
        main_windows windows2;
    });
}
else{

```

同时在剧情上，我们也设置了两个结局，第一种是你获得了隐藏道具，并且经过计算能够打败打魔王，会出现一种剧情，另一个就是没有打败魔王，或者你的数值可以打败但是没有获得隐藏道具，那你就会重开了。哦，对了，我们打败怪物时会随机触发音效~最后，我们还有一个商人 NPC，你可以选择用金币购买防御力攻击力和生命值。请你务必小心地选择行走路线，一方面有可能拿到能够打败最终 boss 的超级宝典，另一方面钥匙和血量可不是无限的哦~一旦你最后打不过怪物又没有足够的钥匙，说明这个游戏你就失败了哦。我们已经经过多次测试，目前的数据应该是能够在极限状态下走完全过程。

(胜利结算画面)



最后**答题板块**，我们专门写了一个类，通过在初始化的时候传入一个大小为八 QStirng 数组来实现构造题目，都是四个题目和四个选项。有一些是玩笑类的，无论选择哪个选项都是对的，有一些则是从期中考试选择题里面挑了一些作者错的题目考考大家，错了会扣除生命值。当然我们还有一个很**的小游戏（作者自己都觉得傻逼），就是让你在 100 以内随意猜一个数，如果误差小于等于二就算你赢，会获得巨幅加成，否则会扣除少量生命值。

（以下是我们的 main_windows 构造函数）

```

class main_windows : public QMainWindow{
    Q_OBJECT
public:
    explicit main_windows(QMainWindow *parent=nullptr);
    void move(int a,int b);
    int now_text;
    QLabel *textLabel;
    int floorflag=-1;
    QString floorbgmpath[10];
    bool isstart=false;
    ShopDialog *shopper;
    bool hidefirst=true;
    void get_pic(int i,int j);
protected:
    void keyPressEvent(QKeyEvent *event)override;
private:
    QTimer *mo_yu;
    my_label *label[14][20];
    QLabel *label_back;
    main_menu w;
    QWidget *background;
    my_label *imageLabel[10];
    QPushButton *backButton;
    hero *my_hero;
    base_creature *NPC[30];
    my_label *hp_pw_df[20];
    my_label *attribute;
    QMediaPlayer *floorbgm;
    QAudioOutput*floorbgm_out;
    QMediaPlayer *playerv;
    QVideoWidget* videoWidget;

public slots:
    void show_attribute();//属性s
    void show_pic();
    void my_quit();
    void show_background();
    void back_to_menu();
    void add_now_text();
    void handleKeyPress();
    void mo_yu_video();
    void getsignal1();
    void getsignal2();
    void getsignal3();
    void getsignal4();
    void getsignal5();
signals:
    void buttonclicked();

```



```
text[0]="事情要追溯到遥远的1898, <br>"  
      "彬彬来到了北京大学里的塔防游戏";  
text[1]="在这里他遇见了他的小伙伴: <br>"  
      "坤哥, 亡灵凯, 盯针, 大例王.....";  
text[2]="他们会在这里遇到什么样的冒险? <br>"  
      "他们会在这里获得怎样的经历? ";  
text[3]="让我们开始游戏, 进入魔塔! <br>"  
      "一同随它的小伙伴来探险吧! ";  
text[4]="未完待续.....";
```

背景:



三、小组成员分工情况

元铭宇：完成了背景的实现和魔塔主体中大部分类的构建和源文件的编写；编写了主菜单和主界面的框架和结构并完成大部分填充，为魔塔中的众多小惊喜(具体可见第二部分和演示视频)提供想法并进行实现；作为组长和组员研讨大作业内容实现并对组员任务进行分工。

廖丁鸥：对游戏主体中怪物的数值进行了详尽设计并为游戏提供了精美的地图设计和部分怪物资源，设计游戏通关路线并为游戏细节提供切实的建议并进行实现，对程序运行中出现的细节 bug 进行调试并修改。

梁泽名：对游戏音视频进行添加和调试，提供初始战斗代码；为程序内资源文件修改路径和名称使程序可移植地运行；完成游戏中新增怪物和道具的类、金币的编写，对程序中细节处进行修改。

四、项目总结和反思

总结：对于这种多文件的项目需要开始时构建一个良好的框架，这个框架既需要避免 bug 的产生也要易于将其中添加新内容。多动脑筋，在开发的过程中我们其实遇到了很多不太好实现的功能，尤其是涉及动画和视频播放的，但是我们从另外的角度解决了问题。然后其实 PS 的能力还是挺重要的，很多东西都需要用到 PS 技术（比如属性图的制作）

反思：

仍有原本想要实现的联机留言板内容尚未实现；多人参与同一项目程序的编写时需要提前统一命名方式、资源添加方式等。还有就是做动画的时候，很容易陷在一个地方出不来，尤其是当你的动画不能显示时，哪怕用 qDebug 也 de 不出来的时候，善用搜索！有些错误你是真的发现不了的。然后我们其实应该把 main_windows 的代码分散一点，这样查看以及纠错的时候就方便多了，可读性也就增强了。