# 高级语言程序设计大作业实验报告

2113839 林帆

## 一. 作业题目

《丁老头脱发记》

## 二. 开发软件

**Qt 5.14.2**

## 三. 课题要求

1) 学习 QT 基本功能

2) 学会窗口创建等游戏展示界面

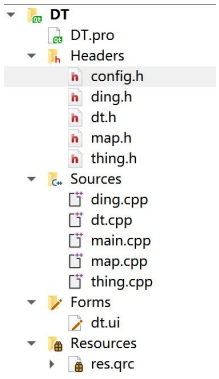
3) 将所学 C++知识与之联系

4) 理解 QT 创作的简单游戏文件、代码

5) 自己利用 QT 编写游戏

## 四. 主要流程

### 1．大体框架/主函数实现



## 为游戏中元素创建多个.h 头文件，对应多个.cpp 源文件实现函数编写，Resources 文件中存储图片资源

## Main 函数为主函数

```
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
     DT w;

    QResource::registerResource(":/im");

    w.show();
    return a.exec();
}
```

其实有效调用的函数是 DT w 和 w.show()
其中 DT 类由<dt.h>头文件创造，dt.cpp 为主窗体类
构造函数 DT：：DT（）中含有多个、多类变量，运行了多类函数、多个函数
其中 initscene();

```
    showbutton();
    playGame();
```

三个函数是实现游戏界面不断更新、有序更新的关键

## 2．头文件构造类的展示

<config.h>资源配置

```
#ifndef CONFIG_H
#define CONFIG_H

#define GAME_WIDTH  700
#define GAME_HEIGHT  1250
#define GAME_TITLE  "丁老头脱发1.0"
#define GAME_RATE 10 //定时器单位毫秒

#define MAP_SPEED 2 //地图滚动速度

#define THING_SPEED 10000;//东西移动速度
#define THING_NUM 20;//总数量
#define THING_INTERVAL 50;//出场时间间隔

#define IFSTART 0;
#include<iostream>

#endif // CONFIG_H
```

<ding.h>老头类

```
class DING
{
public:
    DING();

    //设置老头位置
    void setPosition(int x,int y);

    //老头资源对象
    QPixmap d;

    //老头坐标
    int m_x;
    int m_y;

    //老头矩形边框
    QRect m_Rect;

    //老头头发数量
    float num;

    //头发数量变化
    void ding_add();
    void ding_minus();
    //头发增/减
    int state;//(0,1,2)对应（不变，增，减）

};
```

## <map.h>地图类

```
class map
{
public:
    map();

    //地图滚动坐标计算
    void mapPosition();

    //地图图片对象
    QPixmap map1;
    QPixmap map2;

    //地图Y轴坐标
    int y1;
    int y2;

    //地图滚动幅度
    int map_speed;

};
```

## <thing.h>东西类

```
class thing
{
public:
    thing();

    //类型
    int type;
    //类型设定
    void settype(int n);
    //更新坐标
    void updatePosition();

    //东西资源对象
    QPixmap m_thing;

    //位置
    int m_x;
    int m_y;

    //东西的矩形边框（碰撞检测）
    QRect m_Rect;

    //状态
    bool m_free;

    //速度
    int m_speed;

    //更新时能否改变
    bool m_change;

};
```

## 五.　游戏界面展示


# 六.　收获

## 1. 信号与槽的应用

（1）定时器事件

```
//启动定时器
    m_Timer.start();
    //槽监视信号
    connect(&m_Timer,&QTimer::timeout,[=](){
        //东西出场
        thingtoscene();
        //更新游戏中元素坐标
        updatePosition();
        //绘制
         m_ding.num-=0.01;
        update();
        //碰撞检测
        detect();
       });
```

（2）按键事件实现

```
void DT::showbutton()
{
   //创建按钮
        QPushButton *but=new QPushButton("start",this);

    //调整位置
    but->move(250,700);

    //按钮上文字大小
    but->setStyleSheet("QPushButton{font:20px;}");

    //按钮尺寸
    but->resize(200,200);

    //建立信息和槽 ， 点击释放按钮之后，将该按钮隐藏
    QObject::connect(but,&QPushButton::clicked,[=](){
        but->hide();
        playGame();
    });

    }
```

## 2. 画图工具的应用

在<QPainter>头文件下编写 PaintEvent()函数

```
void DT::paintEvent(QPaintEvent *)
{

    QPainter painter(this);
    //绘制地图
    painter.drawPixmap(0,m_map.y1,m_map.map1);
    painter.drawPixmap(0,m_map.y2,m_map.map2);

   //绘制丁老头
    painter.drawPixmap(m_ding.m_x,m_ding.m_y,m_ding.d);

    //绘制 thing
    for(int i=0;i<20;i++)
    {
        if(ts[i].m_free==false)
        {
           painter.drawPixmap(ts[i].m_x,ts[i].m_y,ts[i].m_thing);
        }
    }

    //显示 num
    painter.drawRect(20,20,200,50);//画矩形框框
    QFont font("Courier",30);
        painter.setFont(font);
        painter.setPen(Qt::black);
        painter.setBrush(Qt::red);
        painter.drawText(20,50,"hair num:");
        painter.drawText(165,50,QString::number((int)m_ding.num));

    //给老头画头发
    if(m_ding.state==1)
    {
         painter.setPen(Qt::black);//设置画笔颜色
         int i=rand()%10;
         int j=rand()%10;

painter.drawLine(m_ding.m_x-i,m_ding.m_y-j,m_ding.m_x,m_ding.m_y);

    }
    else if(m_ding.state==2)
    {
         painter.setPen(Qt::black);//设置画笔颜色
    }
```

```
}
```

### 3. 鼠标拖动物体移动

```
void DT::mouseMoveEvent(QMouseEvent *event)
{
    int x= event->x()-m_ding.m_Rect.width()*0.5;
    int y= event->y()-m_ding.m_Rect.height()*0.5;

    //边界检测
    if(x<=0)x=0;
    if(x>=GAME_WIDTH-m_ding.m_Rect.width())
        x=GAME_WIDTH-m_ding.m_Rect.width();
    if(y<=0)y=0;
    if(y>=GAME_HEIGHT-m_ding.m_Rect.height())
    y=GAME_HEIGHT-m_ding.m_Rect.height();


     m_ding.setPosition(x,y);
    }
```