高级语言程序设计

实验报告

南开大学 计算机大类

姓名 李炎龙

学号 2310410

班级 0927

2024年 5月 14 日

目录

# 高级语言程序设计大作业实验报告......................................... 2 一. 作业题目..............................................................................2 二. 开发软件..............................................................................2 三. 课题要求..............................................................................2 四. 主要流程..............................................................................3 1． 整体流程..............................................................................3

# 2． 具体实现............................................................................. 3

1. 主场景的设置.......................................... 3
2. 资源导入...........................................4
3. 地图的设置......................................... 5
4. 创建控制的飞机................................................................. 7
5. 创建敌军飞机......................................................................7
6. 爆炸效果
7. 创建并控制子弹

# 五. 总结......................................................................................9

高级语言程序设计大作业实验报告

一. **作业题目**

经典的游戏：planewar飞机大战游戏的实现

## 开发软件

Qt Creator 6.50

三. **课题要求**

1. 显示出敌我飞机
2. 子弹及击败敌军的效果
3. 音效以及各种界面的美化

## 四. 主要流程

### 1．整体流程

实现思路：mainscene是游戏场景的实现，map是游戏地图，heroplane是友军飞机的设置，enemyplane是敌军飞机的设置，config是游戏中各种资源路径，

bomb是爆炸特效的设置，bullet是子弹的设置，res是导入的资源

**具体实现  
（1）mainscene主场景的实现：**ifndef MAINSCENE\_H

#define MAINSCENE\_H

#include <QWidget>

#include <QTimer>

#include "map.h"

#include "heroplane.h"

#include "enemyplane.h"

#include "bomb.h"

#include <QSound>

class MainScene : public QWidget

{

Q\_OBJECT

public:

MainScene(QWidget\* parent = 0);

~MainScene();

//初始化场景

void initScene();

//启动游戏 用于启动定时器对象

void playGame();

//更新坐标

void updatePosition();

//绘图事件

void paintEvent(QPaintEvent\* event);

//重写鼠标移动事件

void mouseMoveEvent(QMouseEvent\*);

//敌机出场

void enemyToScene();

//碰撞检测

void collisionDetection();

//地图对象

Map m\_map;

//创建飞机对象

HeroPlane m\_hero;

//定时器对象

QTimer m\_Timer;

//敌机数组

EnemyPlane m\_enemys[ENEMY\_NUM];

//敌机出场间隔记录

int m\_recorder;

//爆炸数组

Bomb m\_bombs[BOMB\_NUM];

//测试子弹

//Bullet temp\_Bullet;

};

#endif // MAINSCENE\_H  
**（2）map地图的实现：  
由于游戏框架的设置，地图只需设置相同的两张进行滚动**#include "map.h"

#include "config.h"

Map::Map()

{

//初始化加载地图对象

m\_map1.load(MAP\_PATH);

m\_map2.load(MAP\_PATH);

//设置坐标

m\_map1\_posY = -GAME\_HEIGHT;

m\_map2\_posY = 0;

//设置滚动速度

m\_scroll\_speed = MAP\_SCROLL\_SPEED;

}

void Map::mapPosition()

{

//处理第一张图片滚动

m\_map1\_posY += m\_scroll\_speed;

if (m\_map1\_posY >= 0)

{

m\_map1\_posY = -GAME\_HEIGHT;

}

//处理第二张图片滚动

m\_map2\_posY += m\_scroll\_speed;

if (m\_map2\_posY >= GAME\_HEIGHT)

{

m\_map2\_posY = 0;

}

} **（3）heroplane友军飞机的实现：**

#ifndef HEROPLANE\_H

#define HEROPLANE\_H

#include <QPixmap>

#include "bullet.h"

class HeroPlane

{

public:

HeroPlane();

//发射子弹

void shoot();

//设置飞机位置

void setPosition(int x, int y);

public:

//飞机资源 对象

QPixmap m\_Plane;

//飞机坐标

int m\_X;

int m\_Y;

//飞机的矩形边框

QRect m\_Rect;

//弹匣

Bullet m\_bullets[BULLET\_NUM];

//发射间隔记录

int m\_recorder;

};

#endif // HEROPLANE\_H

**（4）enemyplane敌军飞机的实现：**include "enemyplane.h"

#include "config.h"

EnemyPlane::EnemyPlane()

{

//敌机资源加载

m\_enemy.load(ENEMY\_PATH);

//敌机位置

m\_X = 0;

m\_Y = 0;

//敌机状态

m\_Free = true;

//敌机速度

m\_Speed = ENEMY\_SPEED;

//敌机矩形

m\_Rect.setWidth(m\_enemy.width());

m\_Rect.setHeight(m\_enemy.height());

m\_Rect.moveTo(m\_X, m\_Y);

}

void EnemyPlane::updatePosition()

{

//空闲状态，不计算坐标

if (m\_Free)

{

return;

}

m\_Y += m\_Speed;

m\_Rect.moveTo(m\_X, m\_Y);

if (m\_Y >= GAME\_HEIGHT)

{

m\_Free = true;

}

}

**（5）各种资源路径config的实现：**#ifndef CONFIG\_H

#define CONFIG\_H

/\*\*\*\*\*\*\*\*\*\* 游戏配置数据 \*\*\*\*\*\*\*\*\*\*/

#define GAME\_WIDTH 512 //宽度

#define GAME\_HEIGHT 768 //高度

#define GAME\_TITLE "飞机大战 v1.0" //标题

#define GAME\_RES\_PATH "./plane.rcc" //二进制资源路径

#define GAME\_ICON ":/res/app.ico" //图标路径

#define GAME\_RATE 10 //单位毫秒

/\*\*\*\*\*\*\*\*\*\* 地图配置数据 \*\*\*\*\*\*\*\*\*\*/

#define MAP\_PATH ":/res/img\_bg\_level\_1.jpg" //地图图片路径

#define MAP\_SCROLL\_SPEED 2 //地图滚动速度

/\*\*\*\*\*\*\*\*\*\* 飞机配置数据 \*\*\*\*\*\*\*\*\*\*/

#define HERO\_PATH ":/res/hero2.png"

/\*\*\*\*\*\*\*\*\*\* 子弹配置数据 \*\*\*\*\*\*\*\*\*\*/

#define BULLET\_PATH ":/res/bullet\_11.png" //子弹图片路径

#define BULLET\_SPEED 5 //子弹移动速度

#define BULLET\_NUM 30 //弹匣中子弹总数

#define BULLET\_INTERVAL 20 //发射子弹时间间隔

/\*\*\*\*\*\*\*\*\*\* 敌机配置数据 \*\*\*\*\*\*\*\*\*\*/

#define ENEMY\_PATH ":/res/img-plane\_5.png" //敌机资源图片

#define ENEMY\_SPEED 5 //敌机移动速度

#define ENEMY\_NUM 20 //敌机总数量

#define ENEMY\_INTERVAL 30 //敌机出场时间间隔

/\*\*\*\*\*\*\*\*\*\* 爆炸配置数据 \*\*\*\*\*\*\*\*\*\*/

#define BOMB\_PATH ":/res/bomb-%1.png" //爆炸资源图片

#define BOMB\_NUM 20 //爆炸数量

#define BOMB\_MAX 7 //爆炸图片最大索引

#define BOMB\_INTERVAL 20 //爆炸切图时间间隔

#define SOUND\_BACKGROUND ":/res/bg.wav"

#define SOUND\_BOMB ":/res/bomb.wav"

#endif // CONFIG\_H

**(6)爆炸效果bomb的实现：**#include "bomb.h"

Bomb::Bomb()

{

//初始化爆炸图片数组

for (int i = 1; i <= BOMB\_MAX; i++)

{

//字符串拼接，类似 ":/res/bomb-1.png"

QString str = QString(BOMB\_PATH).arg(i);

m\_pixArr.push\_back(QPixmap(str));

}

//初始化坐标

m\_X = 0;

m\_Y = 0;

//初始化空闲状态

m\_Free = true;

//当前播放图片下标

m\_index = 0;

//爆炸间隔记录

m\_Recoder = 0;

}

void Bomb::updateInfo()

{

//空闲状态

if (m\_Free)

{

return;

}

m\_Recoder++;

if (m\_Recoder < BOMB\_INTERVAL)

{

//记录爆炸间隔未到，直接return，不需要切图

return;

}

//重置记录

m\_Recoder = 0;

//切换爆炸播放图片

m\_index++;

//注：数组中的下标从0开始，最大是6

//如果计算的下标大于6，重置为0

if (m\_index > BOMB\_MAX - 1)

{

m\_index = 0;

m\_Free = true;

}

}

1. **发射子弹bullet的实现：**

#include "bullet.h"

Bullet::Bullet()

{

//加载自动资源

m\_Bullet.load(BULLET\_PATH);

//子弹坐标初始化

m\_X = GAME\_WIDTH \* 0.5 - m\_Bullet.width() \* 0.5;

m\_Y = GAME\_HEIGHT;

//子弹空闲状态

m\_Free = true;

//子弹速度

m\_Speed = BULLET\_SPEED;

//子弹矩形边框 （碰撞检测）

m\_Rect.setWidth(m\_Bullet.width());

m\_Rect.setHeight(m\_Bullet.height());

m\_Rect.moveTo(m\_X, m\_Y);

}

void Bullet::updatePosition()

{

//如果子弹是空闲状态，不需要计算坐标

if (m\_Free)

{

return;

}

//子弹向上移动

m\_Y -= m\_Speed;

m\_Rect.moveTo(m\_X, m\_Y);

if (m\_Y <= -m\_Rect.height())

{

m\_Free = true;

}

## } 五.游戏场景展示 屏幕截图 2024-05-14 234340 六. 总结 在此次大作业完成过程中，自学了一点qt,接着在B站上找一些视频，从零开始制作了一款很简单的游戏，飞机大战，也算是自己第一次肝出一个小游戏，尽管这个游戏很简单，比不上其他同学的精品，我还是收获颇多，锻炼了自己解决编程的能力，希望能在学期学到更多的编程只是，全面完善自己的知识