

高级语言程序设计

实验报告

南开大学 电子信息与光学工程学院

姓名 付立伟

学号 2212504

班级 电子信息特色班

2023 年 5 月 14 日

高级语言程序设计大作业实验报告

一. 作业题目

FlyingExsusiai——一款基于 QT 的明日方舟二创飞机大战类游戏。

二. 开发软件

Qt Creator 4.11.1(Community)

三. 课题要求

- 1) 使用 C++ 语言进行编程
- 2) 实现图形化界面
- 3) 及时上传源代码至 gitee 或 github 上
- 4) 录制讲解视频

四. 主要流程

1.设计我方主角 Exsusiai

(1) 主要思路：通过设计类 Exsusiai 来实现我方飞机功能的实现，其基本设计思路为：实现飞机模型的载入，设计碰撞箱，实现子弹正常发射，确定飞机位置。主要分为如图所示部分：

```

6 class Exusiai
7 {
8 public:
9     Exusiai();
10
11     //发射子弹
12     void shoot();
13     //设置飞机位置
14     void setPosition(int x,int y);
15     //飞机资源 对象
16     QPixmap m_Plane;
17
18     //飞机坐标
19     int m_X;
20     int m_Y;
21
22     //飞机的矩形边框
23     QRect m_Rect;
24
25     //弹匣
26     Bullet m_bullets[BULLET_NUM];
27
28     //发射间隔记录
29     int m_recorder;
30
31 };
32
33 #endif // EXUSIAI_H
34

```

Exusiai 类

其中涉及到子弹类 Bullet 的设计，其主要设计思路同飞机设计思路，略有所不同的是其具有移动速度以及使用状态。使用状态之用途会在之后的内容进行介绍。

```

class Bullet
{
public:
    Bullet();

    //更新子弹坐标
    void updatePosition();

    //子弹资源对象
    QPixmap m_Bullet;
    //子弹坐标
    int m_X;
    int m_Y;
    //子弹移动速度
    int m_Speed;
    //子弹是否闲置
    bool m_Free;
    //子弹的矩形边框（用于碰撞检测）
    QRect m_Rect;
};

#endif // BULLET_H

```

Bullet 类

两类部分函数具体实现代码如下：

```
void Exusiai::shoot()
{
    //累加时间间隔记录变量
    m_recorder++;

    //如果记录数字未达到发射子弹时间间隔 直接 return 不发射
    if(m_recorder<BULLET_INTERVAL) {
        return;
    }
    //达到发射时间
    m_recorder=0;

    //发射子弹
    for(int i=0;i<BULLET_NUM;i++) {
        //如果是空闲子弹，进行发射
        if(m_bullets[i].m_Free) {
            //将空闲状态改为假
            m_bullets[i].m_Free=false;
            //设置子弹坐标
            m_bullets[i].m_X=m_X+m_Rect.width()*0.25;
            m_bullets[i].m_Y=m_Y-25;
            break;
        }
    }
}

void Bullet::updatePosition()
{
    //空闲状态下的子弹，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);

    //子弹位置 超出屏幕，重新变为空闲状态
    if(m_Y<=-m_Rect.height()) {
        m_Free=true;
    }
}
```

2.设计敌方飞机类

因为敌机与子弹均为自行移动元素，故其设计大框架基本相似。值得注意的是，其与我方飞机一样均需要进行碰撞检测。其基本框架如下图：

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

    //更新坐标
    void updatePosition();

    //敌机资源对象
    QPixmap m_enemy;

    //位置
    int m_X;
    int m_Y;

    //敌机的矩形边框（碰撞检测）
    QRect m_Rect;

    //状态
    bool m_Free;

    //速度
    int m_Speed;
};

class Boss
{
public:
    Boss();

    //更新坐标
    void updatePosition();

    //Boss资源对象
    QPixmap m_boss;

    //位置
    int m_X;
    int m_Y;

    //Boss的矩形边框（碰撞检测）
    QRect m_Rect;

    //状态
    bool m_Free;

    //生命
    int m_life;

    //速度
    int m_Speed;
};
```

本游戏设置了 enemy 与 boss 两种飞机，其设计上唯一不同便在于生命值的不同，故后者增加 m_life 变量。

3.设计主界面以及游戏画面

该部分是本项目中最难的部分。故就几个重点功能进行讲解，详情请看分析视频。

（1）游戏音乐的选择

基本思路：通过 QPushButton 来进行音乐的选择。其流程为

按 btn1 开始游戏→btn1 消失，btn2 与 btn3，btn4 出现→令 btn2 与 btn3 指向不同音乐分支→按 btn2 或 btn3 选择音乐，三个按钮全部消失，游戏开始。实现代码为

```
connect(btn1,&QPushButton::clicked,[=]() {
    //启动游戏
    btn1->hide();

    btn2->show();

    btn3->show();

    btn4->show();

    connect(btn2,&QPushButton::clicked,[=]() {
        btn2->hide();
        btn3->hide();
        btn4->hide();
        SOUND=SOUND_BACKGROUND1;
        playGame();
    });

    connect(btn3,&QPushButton::clicked,[=]() {
        btn2->hide();
        btn3->hide();
        btn4->hide();
        SOUND=SOUND_BACKGROUND2;
        playGame();
    });
});
```

(2) 生命值与积分值的显示

基本思路：对于积分值：令文本直接显示在屏幕上，其字符随击杀飞机数量改变

对于生命值：我相信阿能是不死哒！

实现代码如图：

```
life = new QLabel("生命值：∞，因为阿能是无敌哒!!!",this);
life->setFont(QFont("Algerian",16));
life->move(0,500);
score = new QLabel(QString(SCORE).arg(Score),this);
score->setFont(QFont("Algerian",16));
score->move(0,550);
```

(3) 爆炸效果的实现

一、设计爆炸类

主要思路：确定爆炸位置，播放爆炸动画。其中，利用数组存放爆炸资源。

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

    //更新信息（播放图片下标，播放间隔）
    void updateInfo();

    //放爆炸资源数组
    QVector<QPixmap> m_pixArr;

    //爆炸位置
    int m_X;
    int m_Y;

    //爆炸状态
    bool m_Free;

    //爆炸切图的时间间隔
    int m_Recorder;

    //爆炸时加载的图片下标
    int m_index;
};
```

爆炸类

其中播放函数的定义为：

```
void Bomb::updateInfo()
{
    //空闲状态下的 爆炸效果 直接 return
    if(m_Free) {
        return;
    }

    m_Recorder++;
    //如果记录爆炸的实际未到达爆炸间隔，不需要切图，直接 return
    if(m_Recorder<BOMB_INTERVAL) {
        return;
    }

    //重置记录
    m_Recorder=0;
```

```

//切换爆炸播放的圖片的下標
m_index++;

//數組中下標從 0 開始計算，最後一個圖片的下標為 BOMB_MAX-1
if(m_index>BOMB_MAX-1) {
    m_index=0;
    m_Free=true;
}
}

```

二、碰撞檢測

主要思路：檢測子彈位置與敵機位置。若兩成員碰撞箱有交界，則子彈與敵機全部重置為空閒狀態，爆炸由空閒狀態變為非空閒狀態，播放爆炸。

實現代碼：

```

void MainScene::collisionDetection()
{
    //遍歷所有非空閒敵機
    for(int i=0;i<ENEMY_NUM;i++) {
        //若空閒，執行下一次循環
        if(m_enemys[i].m_Free)
        {
            continue;
        }

        //遍歷所有非空閒子彈
        for(int j=0;j<BULLET_NUM;j++) {
            if(m_exusiai.m_bullets[j].m_Free) {
                continue;
            }

            //如果子彈和飛機相交，則發生碰撞，同時將飛機和子彈空閒狀態設置
            為真

```

```

if(m_enemys[i].m_Rect.intersects(m_exusiai.m_bullets[j].m_Rect)) {
    m_enemys[i].m_Free=true;
    m_exusiai.m_bullets[j].m_Free=true;
    Score+=1;
    //播放爆炸效果
    for(int k=0;k<BOMB_NUM;k++) {
        if(m_bombs[k].m_Free)
        {
            //空閒的爆炸，可以播放了
            m_bombs[k].m_Free=false;
            //更新爆炸坐標

```

注：使用状态之用途。本项目利用数组存储元素，利用使用状态来实现元素的重复利用，即每次更新元素时会重新使用数组中的 Free 状态元素，这样可以有效减小内存的消耗增强了程序的可用性。

本项目较以往所写 C++ 程序不同的，便是利用宏定义来配置全局变量。在本程序中，我特地添加了 `config` 头文件来存储全局变量，并在其中对其进行更新与修改。这样极大提高了代码的更新维护效率，值得在日后的编程中进一步使用。

```
#ifndef CONFIG_H
#define CONFIG_H

/***** 游戏配置数据 *****/
#define GAME_WIDTH 612 //宽度
#define GAME_HEIGHT 835 //高度
#define GAME_TITLE "FlyingExusiai v8.0" //标题
#define GAME_ICON ":/res/p2.jpg"//图标加载路径
#define GAME_RATE 10 //定时器刷新时间间隔 单位ms
#define SCORE "Score: %1 "
#define LIFE "∞, 因为阿能是无敌啦!!!"

/***** 游戏配置数据 *****/
#define MAP_PATH ":/res/background3.jpg"
#define MAP_SCROLL_SPEED 2//地图滚动速度

/***** 飞机配置数据*****/
#define EXUSIAI_PATH ":/res/p3.png"

/***** 子弹配置数据*****/
#define BULLET_PATH ":/res/bullet.png"//子弹图片路径
#define BULLET_SPEED 5

#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/enemy3.png" //敌机资源图片
#define ENEMY_SPEED 3 //敌机移动速度
#define ENEMY_NUM 20 //敌机总数量
#define ENEMY_INTERVAL 80 //敌机出场时间间隔

/*****敌方boss配置数据*****/
#define BOSS_PATH ":/res/boss.F.png"//boss资源图片
#define BOSS_SPEED 4//boss移动速度
#define BOSS_NUM 2 //boss总数量
#define BOSS_INTERVAL 3000 //BOSS出场时间间隔
#define BOSS_LIFE 10 //BOSS生命值

/***** 爆炸资源配置数据*****/
#define BOMB_PATH ":/res/bomb-%1.png" //爆炸资源图片
#define BOMB_NUM 20 //爆炸数量
#define BOMB_MAX 2 //爆炸图片最大索引
#define BOMB_INTERVAL 20 //爆炸切图时间间隔
```


五. 单元测试



简单截取三张运行时图片，具体效果请看讲解视频。

六. 收获

（一）更深刻的理解了类的作用。纸上得来终觉浅。亲身设计项目后才体会到类为大型项目设计所带来的便利。这样的体会是课本上得不到的。

（二）宏定义全局变量。本次多次使用宏定义的全局变量。这是先前编程所没有的。经使用后发现，其维护与使用的确非常之方便，日后会多多使用。

（三）积累了制作游戏，搜索资料的经验。为日后完成更大的项目打下基础。

七. Gitee 链接

<https://gitee.com/nkuflw/nku-c-homework.git>

八. 讲解视频链接