**北京科技大学实验报告**

学院：计算机与通信工程学院 专业： 计算机类 班级：计2005

姓名：赵方程 学号：42024137 实验日期：2021 年 4 月 22 日

**实验名称：**

**基于easyX的简单飞机大战**

**实验目的：**

**完成简单的游戏,能控制飞机的位置,发射子弹,击毁敌机,收到伤害等功能**

**实验仪器：**

计算机：Zephyrus G14

CPU：2.9GHz AMD Ryzen 7 4800HS

内存：2×8GB 3200MHz DDR4

硬盘：512GB

显卡：Nvidia Geforce RTX2060 with Max-q design

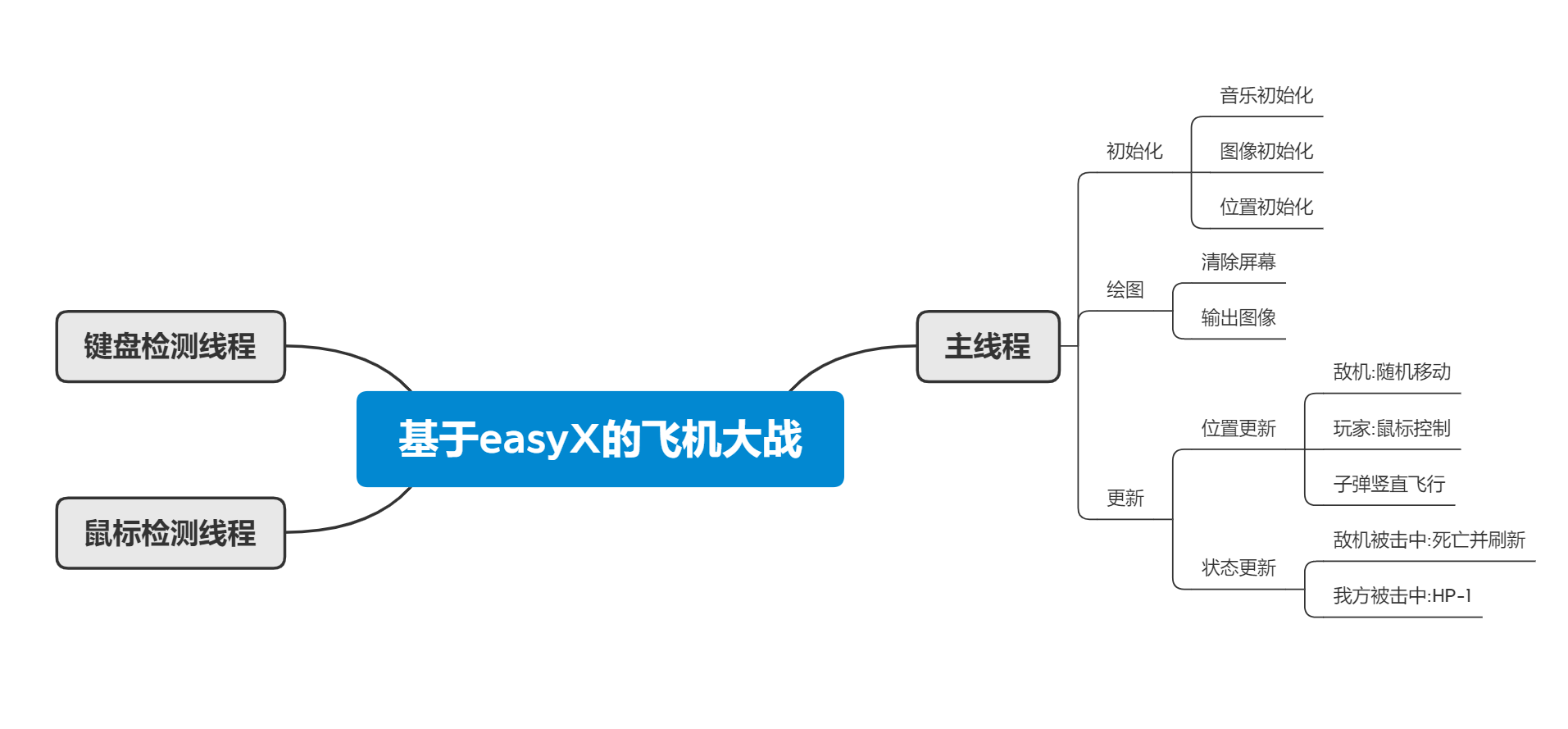
操作系统：Windows 10

编译器：MSVC 16.9.4 X64 Release

集成开发环境：Visual Studio 2019 Version16.9.4

**实验原理：**

主要的流程如下:



**实验内容与步骤：（字体：中文小四，英文Times New Roman小四，均不加粗）**

说明程序实现的步骤

## 相关实现

### 思路

使用easyX提供的API检测鼠标位置来控制飞机位置,监听鼠标点击事件控制子弹发射

玩家飞机、敌方飞机、子弹各为一类, 封装了随机移动、随机位置、随机速度等功能(随机移动效果不佳,未使用)

敌机随机移动,被子弹击中或者超出绘图边界时重置随机位置,获得随机速度

### 主函数

auto main() -> int  
{  
 while (!g\_flag)//g\_flag是一个判断游戏是否进行的bool型全局变量,初始化为false  
 {  
 g\_flag = true;  
 initialization(); // 初始化  
 run();  
 }  
 delete t\_clickMonitor;  
 delete t\_kbMonitor;

delete mouse;  
 return 0;  
}

### 函数initialization中对游戏进行了初始化

auto initialization()  
{   
 srand(static\_cast<unsigned>(time(nullptr)));  
 musicInitial();//音频初始化  
 imageInitial();//图像初始化  
 positionInitial();//位置初始化  
 numInitial(); //敌机数量初始化  
 myPlane.reSetStatus();//重设玩家飞机状态  
 g\_score = 0; //得分初始化  
 mouse = new MOUSEMSG;//实例化mouse  
}

### run函数中进行了游戏主体的运行

auto run()  
{  
 t\_clickMonitor = new std::thread(clickMonitor);  
 t\_kbMonitor = new std::thread(kbMonitor);  
 while (g\_flag)  
 {  
 while (true)  
 {  
 display();//绘制图像  
 if (!g\_flag)  
 {  
 break;  
 }  
 if (g\_pause)//g\_pause是一个判断游戏是否暂停的bool型全局变量,初始化为false  
 {  
 pause();  
 system("pause");  
 g\_pause = false;  
 }  
 refresh(); //刷新  
 }  
 }  
}

#### clickMonitor是检测鼠标点击(滑轮滑动)的函数

t\_clickMonitor是用于检测鼠标点击(滑轮滑动)的线程

[[noreturn]] auto clickMonitor()  
{  
 while (g\_flag)  
 {  
 PeekMouseMsg(mouse);  
 if (mouse->mkLButton || mouse->uMsg == WM\_MOUSEWHEEL)  
 {  
 if (!g\_isFlying)  
 {  
 g\_isFlying = true;  
 }  
 }  
 }  
}

#### kbMonitor是检测键盘事件的函数

t\_kbMonitor是用于检测键盘事件的线程

[[noreturn]] auto kbMonitor()  
{  
 while (true)  
 {  
 while (\_kbhit() && !g\_pause)  
 {  
 g\_pause = true;  
 }  
 }  
}

#### pause是个显示暂停界面的函数

auto pause()  
{  
 BeginBatchDraw();  
 outtextxy(width \* 0.3, height \* 0.5, \_T("Pause, press any key to continue."));  
 FlushBatchDraw();  
}

#### display函数中进行了图像绘制

auto display()  
{  
 cleardevice();  
 BeginBatchDraw();  
 //putimage(0, 0, &g\_backGround);  
 TCHAR scoreDisplay[100] = {'\0'};  
 swprintf\_s(scoreDisplay, \_T("%d"), g\_score);  
 outtextxy(width \* 0.02, height \* 0.02, \_T("Score:")); //显示  
 outtextxy(width \* 0.1, height \* 0.02, scoreDisplay); //分数  
  
 TCHAR hpDisplay[100] = {'\0'};  
 swprintf\_s(hpDisplay, \_T("%d"), myPlane.getHp());  
 outtextxy(width \* 0.02, height \* 0.06, \_T("HP:")); //显示  
 outtextxy(width \* 0.07, height \* 0.06, hpDisplay); //生命  
 if (myPlane.getStatus())  
 {  
 //绘制己方飞机  
 putimage(myPlane.getPositionX(), myPlane.getPositionY(), &g\_myPlane);  
 //绘制敌方飞机  
 for (int i = 0; i < g\_numRandom + g\_numDownOnly; i++)  
 {  
 putimage(enemy[i].getPositionX(), enemy[i].getPositionY(), &g\_enemy);  
 }  
 //绘制子弹  
 if (g\_isFlying)  
 {  
 putimage(g\_bullets.getPositionX(), g\_bullets.getPositionY(), &g\_bullet);  
 }  
 }  
 else  
 {  
 gameOver();  
 }  
 FlushBatchDraw();  
}

#### refresh函数中进行了位置、状态的刷新

auto refresh()  
{  
 refreshPosition();  
 refreshStatus();  
}

##### refreshPosition刷新了位置

auto refreshPosition()  
{  
 PeekMouseMsg(mouse);  
 if (myPlane.getStatus())  
 {  
 myPlane.setPosition(mouse->x, mouse->y);  
 }  
 /\*for (int i = 0; i < g\_numRandom; i++)  
 {  
 enemy[i].randomMove(); //随机运动  
 }\*/

//取消了随机运动

for (int i = 0; i < g\_numRandom + g\_numDownOnly; i++)  
 {  
 enemy[i].moveDownOnly(); //垂直运动  
 }  
   
 if (!g\_isFlying)  
 {  
 g\_bullets.setPosition(mouse->x, mouse->y - 10);  
 }  
 g\_bullets.moveUp();  
}

##### refreshStatus刷新了状态

auto refreshStatus()  
{  
 if (myPlane.getStatus())  
 {  
 g\_numDownOnly = g\_score / 10;  
 int n1;  
 if ((n1 = isHitEnemy()) >= 0) // 击中敌机或与之碰撞  
 {  
 if (isHitBullet())  
 {  
 g\_isFlying = false;  
 }  
 bool collide(false);  
 /\*  
 \* 若敌机碰撞我方,则不加分,敌机重启,我方生命-1  
 \*/  
 if (isHitPlayer() > 0)  
 {  
 myPlane.hpMinus();  
 collide = true;  
 }  
  
 enemy[n1].newPosition(); //重置敌机位置  
  
 /\*  
 \* 子弹击中敌方,加分  
 \*/  
 if (!collide)  
 {  
 g\_score++;  
 putimage(enemy[n1].getPositionX(), enemy[n1].getPositionY(), &g\_explode);  
 }  
 }  
 for (int i = 0; i < g\_numRandom + g\_numDownOnly; i++)  
 {  
 if (enemy[i].getPositionY() >= height)  
 {  
 enemy[i].newPosition();  
 }  
 }  
  
 if (g\_bullets.getPositionY() <= height)  
 {  
 g\_isFlying = false;  
 }  
 }  
}

## 相关函数

###### class bullet

|  |  |  |
| --- | --- | --- |
| Methods | parameter(s) | Description |
| setPositionX | const int &x | set X-position |
| setPositionY | const int &y | set Y-position |
| setPosition | const int &x const int &y | set X&Y Position |
| getPositionX |  | return X-position |
| getPositionY |  | return Y-position |
| moveUp |  | Y-position - 1 |

###### class MyPlane

|  |  |  |
| --- | --- | --- |
| Methods | parameter(s) | Description |
| setPositionX | const int &x | set X-position |
| setPositionY | const int &y | set Y-position |
| setPosition | const int &x const int &y | set X&Y Position |
| setPosition | const MOUSEMSG &m | set X&Y Position as the position of mouse |
| getPositionX |  | return X-position |
| getPositionY |  | return Y-position |
| hpMinus |  | HP\_ -1 and set status of plane |
| getHp |  | return HP\_ |
| getStatus |  | return status (normal or exploded?) |
| resetStaus |  | set HP to 20 and status to alive |
|  |  |  |

###### class enemyPlane

|  |  |  |
| --- | --- | --- |
| Methods | parameter(s) | Description |
| setPositionX | const int &x | set X-position |
| setPositionY | const int &y | set Y-position |
| getPositionX |  | return X-position |
| getPositionY |  | return Y-position |
| moveDown |  | Y-position+1 |
| moveR |  | X-position+1 |
| moveL |  | X-position-1 |
| randomMove |  | move randomly |
| moveDownOnly |  | move down at random speed |
| newPosition |  | set position&speed randomly |

*其他详细代码请参阅*[*main.cpp*](./cpp&header/main.cpp) *&* [*Plane.h*](./cpp&header/Plane.h)

## 实验结果与分析

测试截图:

