游戏报告

作业题目：直升机大战ikun

开发软件：visual studio 2022

课题要求：1，面向对象

2，单元测试

3，模型部分

4，验证

整体流程：bool PointInRect(int x, int y, RECT& r)

{

return (r.left <= x && x <= r.right && r.top <= y && y <= r.bottom);

}

bool RectDuangRect(RECT& r1, RECT& r2)

{

RECT r;

r.left = r1.left - (r2.right - r2.left);

r.right = r1.right;

r.top = r1.top - (r2.bottom - r2.top);

r.bottom = r1.bottom;

return (r.left < r2.left && r2.left <= r.right && r.top <= r2.top && r2.top <= r.bottom);

}

通过这两个函数，将飞机的产生位置确定下来且不重叠，从而达到有序产生的目的

void Welcome()

{

LPCTSTR title = \_T("飞机大战");

LPCTSTR tplay = \_T("开始游戏");

LPCTSTR texit = \_T("退出游戏");

RECT tplayr, texitr;

BeginBatchDraw();

setbkcolor(WHITE);

cleardevice();

settextstyle(60, 0, \_T("黑体"));

settextcolor(BLACK);

outtextxy(swidth / 2 - textwidth(title) / 2, sheight / 5, title);

settextstyle(40, 0, \_T("黑体"));

tplayr.left = swidth / 2 - textwidth(tplay) / 2;

tplayr.right = tplayr.left + textwidth(tplay);

tplayr.top = sheight /4;

tplayr.bottom = tplayr.top + textheight(tplay);

texitr.left = swidth / 2 - textwidth(texit) / 2;

texitr.right = texitr.left + textwidth(texit);

texitr.top = sheight / 2;

texitr.bottom = texitr.top + textheight(texit);

outtextxy(tplayr.left, tplayr.top, tplay);

outtextxy(texitr.left, texitr.top, texit);

EndBatchDraw();

开始界面，通过设定将界面的开始窗口设置完成，达到目的

void Over(unsigned long long& kill)

{

printf\_s("o");

TCHAR\* str = new TCHAR[128];

\_stprintf\_s(str, 128, \_T("击杀数：%llu"), kill);

settextcolor(RED);

outtextxy(swidth / 2 - textwidth(str) / 2, sheight / 5, str);

// 键盘事件 （按Enter返回）

LPCTSTR info = \_T("按Enter返回");

settextstyle(20, 0, \_T("黑体"));

outtextxy(swidth - textwidth(info), sheight - textheight(info), info);

while (true)

{

ExMessage mess;

getmessage(&mess, EM\_KEY);

if (mess.vkcode == 0x0D)

{

return;

}

}

}

控制键盘颜色和鼠标控制模式

class BK

{

public:

BK(IMAGE& img)

:img(img), y(-sheight)

{

}

void Show()

{

if (y == 0) { y = -sheight; }

y += 4;

putimage(0, y, &img);

}

private:

IMAGE& img;

int y;

};

class Hero

{

public:

Hero(IMAGE& img)

:img(img), HP(SHP)

{

rect.left = swidth / 2 - img.getwidth() / 2;

rect.top = sheight - img.getheight();

rect.right = rect.left + img.getwidth();

rect.bottom = sheight;

}

void Show()

{

setlinecolor(RED);

setlinestyle(PS\_SOLID, 4);

putimage(rect.left, rect.top, &img);

line(rect.left, rect.top - 5, rect.left + (img.getwidth() / SHP \* HP), rect.top - 5);

}

void Control()

{

ExMessage mess;

if (peekmessage(&mess, EM\_MOUSE))

{

rect.left = mess.x - img.getwidth() / 2;

rect.top = mess.y - img.getheight() / 2;

rect.right = rect.right = rect.left + img.getwidth();

rect.bottom = rect.top + img.getheight();

}

}

bool hurt()

{

HP--;

return (HP == 0) ? false : true;

}

RECT& GetRect() { return rect; }

private:

IMAGE& img;

RECT rect;

unsigned int HP;

};

class Enemy

{

public:

Enemy(IMAGE& img, int x, IMAGE\*& boom)

:img(img), isdie(false), boomsum(0)

{

selfboom[0] = boom[0];

selfboom[1] = boom[1];

selfboom[2] = boom[2];

rect.left = x;

rect.right = rect.left + img.getwidth();

rect.top = -img.getheight();

rect.bottom = 0;

}

bool Show()

{

if (isdie)

{

if (boomsum == 3)

{

return false;

}

putimage(rect.left, rect.top, selfboom + boomsum);

boomsum++;

return true;

}

if (rect.top >= sheight)

{

return false;

}

rect.top += 3;

rect.bottom += 3;

putimage(rect.left, rect.top, &img);

return true;

}

void Isdie()

{

isdie = true;

}

RECT& GetRect() { return rect; }

private:

IMAGE& img;

RECT rect;

IMAGE selfboom[3];

bool isdie;

int boomsum;

};

class Bullet

{

public:

Bullet(IMAGE& img, RECT pr)

:img(img)

{

rect.left = pr.left + (pr.right - pr.left) / 2 - img.getwidth() / 2;

rect.right = rect.left + img.getwidth();

rect.top = pr.top - img.getheight();

rect.bottom = rect.top + img.getheight();

}

bool Show()

{

if (rect.bottom <= 0)

{

return false;

}

rect.top -= 3;

rect.bottom -= 3;

putimage(rect.left, rect.top, &img);

return true;

}

RECT& GetRect() { return rect; }

protected:

IMAGE& img;

RECT rect;

};

class EBullet : public Bullet

{

public:

EBullet(IMAGE& img, RECT pr)

:Bullet(img, pr)

{

rect.left = pr.left + (pr.right - pr.left) / 2 - img.getwidth() / 2;

rect.right = rect.left + img.getwidth();

rect.top = pr.bottom;

rect.bottom = rect.top + img.getheight();

}

bool Show()

{

if (rect.top >= sheight)

{

return false;

}

rect.top += 5;

rect.bottom += 5;

putimage(rect.left, rect.top, &img);

return true;

}

};

确定了英雄类，敌机类，子弹类，同时规定了子弹的攻击速度，发射轨迹

bool AddEnemy(vector<Enemy\*>& es, IMAGE& enemyimg, IMAGE\* boom)

{

Enemy\* e = new Enemy(enemyimg, abs(rand()) % (swidth - enemyimg.getwidth()), boom);

for (auto& i : es)

{

if (RectDuangRect(i->GetRect(), e->GetRect()))

{

delete e;

return false;

}

}

es.push\_back(e);

return true;

}

bool Play()

{

setbkcolor(WHITE);

cleardevice();

bool is\_play = true;

IMAGE heroimg, enemyimg, bkimg, bulletimg;

IMAGE eboom[3];

loadimage(&heroimg, \_T("./images/laoda.png"));

loadimage(&enemyimg, \_T("./images/kun.png"));

loadimage(&bkimg, \_T("./images/majiaqi.png"));

loadimage(&bulletimg, \_T("./images/qiu.png"));

loadimage(&eboom[0], \_T("./images/enemy1\_down2.png"));

loadimage(&eboom[1], \_T("./images/enemy1\_down3.png"));

loadimage(&eboom[2], \_T("./images/enemy1\_down4.png"));

BK bk = BK(bkimg);

Hero hp = Hero(heroimg);

vector<Enemy\*> es;

vector<Bullet\*> bs;

vector<EBullet\*> ebs;

int bsing = 0;

clock\_t hurtlast = clock();

unsigned long long kill = 0;

for (int i = 0; i < 5; i++)

{

AddEnemy(es, enemyimg, eboom);

}

while (is\_play)

{

bsing++;

if (bsing % 10 == 0)

{

bs.push\_back(new Bullet(bulletimg, hp.GetRect()));

}

if (bsing == 60)

{

bsing = 0;

for (auto& i : es)

{

ebs.push\_back(new EBullet(bulletimg, i->GetRect()));

}

}

BeginBatchDraw();

bk.Show();

Sleep(2);

flushmessage();

Sleep(2);

hp.Control();

if (\_kbhit())

{

char v = \_getch();

if (v == 0x20)

{

Sleep(500);

while (true)

{

if (\_kbhit())

{

v = \_getch();

if (v == 0x20)

{

break;

}

}

Sleep(16);

}

}

}

hp.Show();

auto bsit = bs.begin();

while (bsit != bs.end())

{

if (!(\*bsit)->Show())

{

bsit = bs.erase(bsit);

}

else

{

bsit++;

}

}

auto ebsit = ebs.begin();

while (ebsit != ebs.end())

{

if (!(\*ebsit)->Show())

{

ebsit = ebs.erase(ebsit);

}

else

{

if (RectDuangRect((\*ebsit)->GetRect(), hp.GetRect()))

{

if (clock() - hurtlast >= hurttime)

{

is\_play = hp.hurt();

hurtlast = clock();

}

}

ebsit++;

}

}

auto it = es.begin();

while (it != es.end())

{

if (RectDuangRect((\*it)->GetRect(), hp.GetRect()))

{

if (clock() - hurtlast >= hurttime)

{

is\_play = hp.hurt();

hurtlast = clock();

}

}

auto bit = bs.begin();

while (bit != bs.end())

{

if (RectDuangRect((\*bit)->GetRect(), (\*it)->GetRect()))

{

(\*it)->Isdie();

delete (\*bit);

bs.erase(bit);

kill++;

break;

}

bit++;

}

if (!(\*it)->Show())

{

delete (\*it);

es.erase(it);

it = es.begin();

}

it++;

}

for (int i = 0; i < 5 - es.size(); i++)

{

AddEnemy(es, enemyimg, eboom);

}

EndBatchDraw();

}

printf\_s("e");

Over(kill);

return true;

}

规定了子弹击中的效果，同时插入图片，显示出攻击的程度

int main()

{

// “重生锚”，切换到程序所在路径

wchar\_t\* wcmdline = GetCommandLineW();

int argc; wchar\_t\*\* wargv;

wargv = CommandLineToArgvW(wcmdline, &argc);

std::wstring self\_path = wargv[0];

PathCchRemoveFileSpec(&self\_path[0], self\_path.size());

self\_path.resize(wcslen(self\_path.data()));

SetCurrentDirectoryW(self\_path.c\_str());

// easyx初始化

initgraph(swidth, sheight, EW\_NOMINIMIZE);

bool is\_live = true;

while (is\_live)

{

Welcome();

// Play

is\_live = Play();

}

return 0;

}

修改重生瞄，去掉运行框