南开大学 计算机大类

姓名：周重天

学号：2311082

班级：计算机卓越班

2024年5月12日

高级语言程序设计

实验报告

目录

[高级语言程序设计大作业实验报告 1](#_Toc21286)

[一. 作业题目 1](#_Toc29969)

[二. 开发软件 1](#_Toc18364)

[三. 课题要求 1](#_Toc742)

[四. 主要流程 1](#_Toc25240)

1.整体流程

2.实现思路

[五. 收获 3](#_Toc23116)

高级语言程序设计大作业实验报告

1. **作业题目**

简单飞机大战小游戏

1. **开发软件**

Qt6.7

1. **课题要求**
2. 面向对象。
3. 模型部分
4. **主要流程**
   1. **整体流程**

实现思路：

* + - * 1. 分别设计开始场景、游戏场景、游戏结束场景
        2. 实现敌人的创建、我方飞机、敌方飞机的子弹发射逻辑
        3. 玩家飞机的移动实现
        4. 碰撞逻辑的判断和优化
  1. **算法或公式**
     + - 1. 碰撞之后的操作

void GameControl::Collison()

{

//遍历子弹

for(int i=0;i<mBulletList.size();i++)

{

//遍历敌机

for(int j=0;j<mEnemylist.size();j++)

{

if(mBulletList[i]->collidesWithItem(mEnemylist[j]))//碰撞检测

{

//移除场景回收对象

//mBulletList[i]->GameObjectDelete(&mGameScene);

mEnemylist[j]->GameObjectDelete(&mGameScene);

//移除管理器

//mBulletList.removeOne(mBulletList[i]);

mEnemylist.removeOne(mEnemylist[j]);

}

}

}

for(int i=0;i<mEnemyBulletList.size();i++)

{

if(mEnemyBulletList[i]->collidesWithItem(&mPlane))

{

mEnemyBulletList[i]->GameObjectDelete(&mGameScene);

mEnemyBulletList.removeOne(mEnemyBulletList[i]);

GameOver();

}

}

}

* + - * 1. 游戏进行场景中利用两个相同图片达成背景无限延长的效果

void GameControl::BGMove()

{

mBackGround1.moveBy(0,2);

mBackGround2.moveBy(0,2);

if(mBackGround1.y()>=mBackGround1.pixmap().height())

{

mBackGround1.setY(-mBackGround1.pixmap().height());

}

else if(mBackGround2.y()>=mBackGround2.pixmap().height())

{

mBackGround2.setY(-mBackGround2.pixmap().height());

}

}

1. **收获**

对于一些游戏，可以采用容器的方式来实现对象的生成和移除，例如飞机大战中的敌机和子弹

Enemy\* enemy=(Enemy\*)obj;

//添加到管理器

mEnemylist.append(enemy);

//我的子弹生成函数

void GameControl::PlaneBulletShoot()

{

QPixmap bulletImg(":/img/img/bulletPlane.fw.png");

QPoint pos(mPlane.x()+mPlane.pixmap().width()/2,mPlane.y());

GameObject\* obj=GameObjectPool::Instance()->GetGameObject(GameObject::OT\_BulletPlayer);

PlayerBullet\* bullet=(PlayerBullet\*)obj;

bullet->Init(pos,bulletImg);

//添加到场景

mGameScene.addItem(bullet);

//添加到子弹管理器

mBulletList.append(bullet);

}

void GameControl::EnemyBulletShoot()

{

//pool构建bullet

for(auto & enemy: mEnemylist)

{

QPixmap bulletImg(":/img/img/bulletEnemy1.fw.png");

QPoint pos(enemy->x()+enemy->pixmap().width()/2,enemy->y());

GameObject\* obj=GameObjectPool::Instance()->GetGameObject(GameObject::OT\_EnemyBullet);

EnemyBullet\* bullet=(EnemyBullet\*)obj;

bullet->Init(pos,bulletImg);

//添加到场景

mGameScene.addItem(bullet);

//添加到子弹管理器

mEnemyBulletList.append(bullet);

}

}

void GameControl::Collison()

{

//遍历子弹

for(int i=0;i<mBulletList.size();i++)

{

//遍历敌机

for(int j=0;j<mEnemylist.size();j++)

{

if(mBulletList[i]->collidesWithItem(mEnemylist[j]))//碰撞检测

{

//移除场景回收对象

//mBulletList[i]->GameObjectDelete(&mGameScene);

mEnemylist[j]->GameObjectDelete(&mGameScene);

//移除管理器

//mBulletList.removeOne(mBulletList[i]);

mEnemylist.removeOne(mEnemylist[j]);

}

}

}

for(int i=0;i<mEnemyBulletList.size();i++)

{

if(mEnemyBulletList[i]->collidesWithItem(&mPlane))

{

mEnemyBulletList[i]->GameObjectDelete(&mGameScene);

mEnemyBulletList.removeOne(mEnemyBulletList[i]);

GameOver();

}

}

}

如此即可实现将敌机、双方子弹加入管理容器和从游戏场景中移除（也是从管理容器中移除）的效果