\*实验报告

徐延成

31601010851

\*\*题目

我们的选题是雷霆战机，是一种控制自己的战斗机躲避关卡中的敌人发射的子弹，并击败他们的的2D游戏。

\*\*需求分析

实现这个游戏的需求可以从分析最简化版本开始，找出这个游戏的必要需求，然后考虑可以添加以增强游戏体验的项目作为选做需求。

最基础的要求；

+ 进入游戏的界面

+ 可以操作的己方战斗机

+ 不可操作的boss

+ 直线飞行的子弹

+ 胜利/失败结算画面

\*\*实现方法

我们使用qt的图形功能。整个界面套用其中的qmainwindow类。游戏画面部分使用qgraphicsScene类，子弹、飞机等都编写为flyingObject的子类。而这个类又是qt中的qgraphicsObject的子类，以继承其中的碰撞检测方法。

\*\*分工

+ 界面是由 负责

+ 游戏逻辑部分由徐延成和徐延康负责。这一部分主要在于在己方和敌方飞机类中创建子弹，同时检测自身是否和其他子弹或是飞行物碰撞，并吸收伤害，同时血量减少。

\*\*关键算法

关键在于重新编写显示和碰撞检测的方法。

我们通过qpixmap的类将图片带入创建的类中，通过qpainter中带有的方法将图片显示出来。同时，重写碰撞范围的判定函数，把他们和图片的形状关联起来。

以flyingObject为例：

class FlyingObject : public QGraphicsObject {

Q\_OBJECT

signals:

void notify(int);

signals:

void sig\_fall();

public:

FlyingObject(qreal wvalue, qreal hvalue, qreal speed, const QPixmaps &pic, QGraphicsScene \*scene, QGraphicsItem\* parent);

~FlyingObject();

virtual void hitCtrl() = 0;

virtual void destroy() = 0;

virtual flyingObjectType getType() = 0;

bool checkPos();

bool checkPos(int dx, int dy);

//x, y, z value is included in the qgraphicobject class

qreal width;

qreal height;

qreal maxspeed;

qreal pixpos;

uint maxstep;

QPixmaps pixmaps;

};

在这个虚类中，定义了这个飞行物的大小和贴图，同时拥有判断自身是否在合法的位置的方法，而其子类视其具体类型判断他的类型（通过getType函数），通过实现其中的hitctrl函数等做出反应

其子类以玩家飞机为例：

class PlayerFighter :public FlyingObject

{

protected:

void hitCtrl();

public:

PlayerFighter(qreal wvalue, qreal hvalue, qreal speed, QPixmaps &pic, QGraphicsScene \*scene, QGraphicsItem\* parent, int healthvalue);

~PlayerFighter();

flyingObjectType getType();

//继承自QgraphicsObject的纯虚函数

QRectF boundingRect() const;

QPainterPath shape() const;

void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget);

void Attack(QGraphicsScene \*ptrsence);

void destroy();

//移动函数，请每个时钟周期检查一次键盘，并调用一次此函数

void MoveLeft();

void MoveRight();

void MoveUp();

void MoveDown();

void keyPressEvent(QKeyEvent\* Event);

void keyReleaseEvent(QKeyEvent\* Event);

void advance(int);

friend PlayerFighter\* createPlayerFighter(QGraphicsScene\* scene);

private:

int health;

bool W, A, S, D;

int step;

};

玩家的飞机中advance函数可以通过一个时钟的回掉函数控制它的变化。通过qwidget类定义的按钮暂停和开始可以控制这个timer的启动和暂停

paint定义了这个类如何将自身显示的方法。通过这个方法，就可以让playerFighter显示在scene指针指向的画面。

同时在回掉函数kepressevent和keyreleaseevent中定义了改变它自身位置的方法，通过其基类的位置合法性判定函数判断其位置是否可以前进到相应位置。

在attack函数中定义了飞机创造他相应的子弹的方法，向上直线射击。而在hitCtrl函数中定义了它和其他物体碰撞的反应：如果是子弹的话，那么血量中减少子弹标识的伤害值，同时令子弹消失并释放。

BoudingRect和shape函数则是从qgraphicsObject中继承的虚函数，这个函数定义了这个类的实体的边界，通过这个函数qt引擎可以判断是否碰撞。如果不实现这个函数，这个类是无法创建的

定义的friend函数createPlayerFighter将这个函数封装，可以无视其中的过多参数，通过预设创建飞机。同理也有createBullet和createBoss，这样可以把这些类封装起来，方便简单的创建这个类的实体。

\*\*成果图片



