class Tower : QObject

{

Q\_OBJECT

//Q\_PROPERTY(QPixmap m\_sprite READ currentSprite WRITE setCurrentSprite)

public:

Tower(QPoint pos,MainWindow \*game,const QPixmap &sprite=QPixmap(":/image/chaomengstill.png"));

/\*

\* 为QTimer \*\_fireRateTimer分配内存空间（new QTimer）

\* connect();

\*/

~***Tower***();

/\*

\* delete \_fireRateTimer;

\*/

void **checkEnemyInRange**();//判断攻击范围内是否有敌人

/\*

\* 依据Enemy \*\_chooseEnemy是否为NULL判断攻击范围内是否有敌人

\* 敌人与Tower距离设计函数进行判断两QPoint间距离

\* 1、有敌人：依据敌人位置决定Tower形态

\* 2、无敌人：遍历敌人，看是否有敌人在攻击范围内

\*/

void **draw**(QPainter \*painter) const;//绘制炮塔（攻击范围、Tower图片）

void **attackEnemy**();//攻击敌人，触发QTimer \*\_fireRateTimer

/\*

\* \_fireRateTimer->start(\_fireRate);

\*/

void **chooseEnemyToAttack**(Enemy \*enemy);//选择攻击敌人

/\*

\* 当范围内有多个敌人时，选择第一个进入攻击范围内的敌人

\* \_chooseEnemy=enemy;

\* 与被选中攻击的敌人双向标记（将敌人与塔的联系反馈给MainWindow）

\*/

void **enemyKilled**();//敌人死亡

/\*

\* 1、指针\_chooseEnemy=NULL

\* 2、\_fireRateTimer->stop()停止计时

\* 3、\_sprite恢复原始形态

\*/

void **enemyOutOfRange**();//敌人逃离攻击范围

/\*

\* 1、敌人得到反馈信息

\* 2、指针\_chooseEnemy=NULL

\* 3、\_fireRateTimer->stop()停止计时

\* 4、\_sprite恢复原始形态

\* 注：2、3、4同enemyKilled

\*/

private slots:

void **shootBullet**();//触发子弹运动，QTimer \*\_fireRateTimer槽函数

private:

bool \_attacking;

int \_attackRange;//攻击范围

int \_damage;//伤害

int \_fireRate;//再次攻击敌人的时间间隔

Enemy \*\_chooseEnemy;//被选中的敌人

MainWindow \*\_game;//游戏界面

QTimer \*\_fireRateTimer;//触发子弹Bullet事件

const QPoint \_pos;//tower的位置

QPixmap \_sprite;//tower的图片

static const QSize \_fixedSize;

};