**大作业报告——TowerDefence**

学号:2019202196

姓名：王浩

1. 游戏概述

名称：Electronic core defense

开发工具：Qt Creator 4.5.0

是否基于他人程序开发：是。

增加了可安放炮塔的位置

增加绘制了塔的攻击范围

优化了炮塔攻击敌人时的旋转角度

增加了消灭敌人可以获得赏金的设定，使建造塔需要花钱

增加了add enemy button，用于提升敌人波数来增加游戏难度

项目Github地址：https://github.com/theemptybox1/TowerDefence3.git

基本功能介绍：

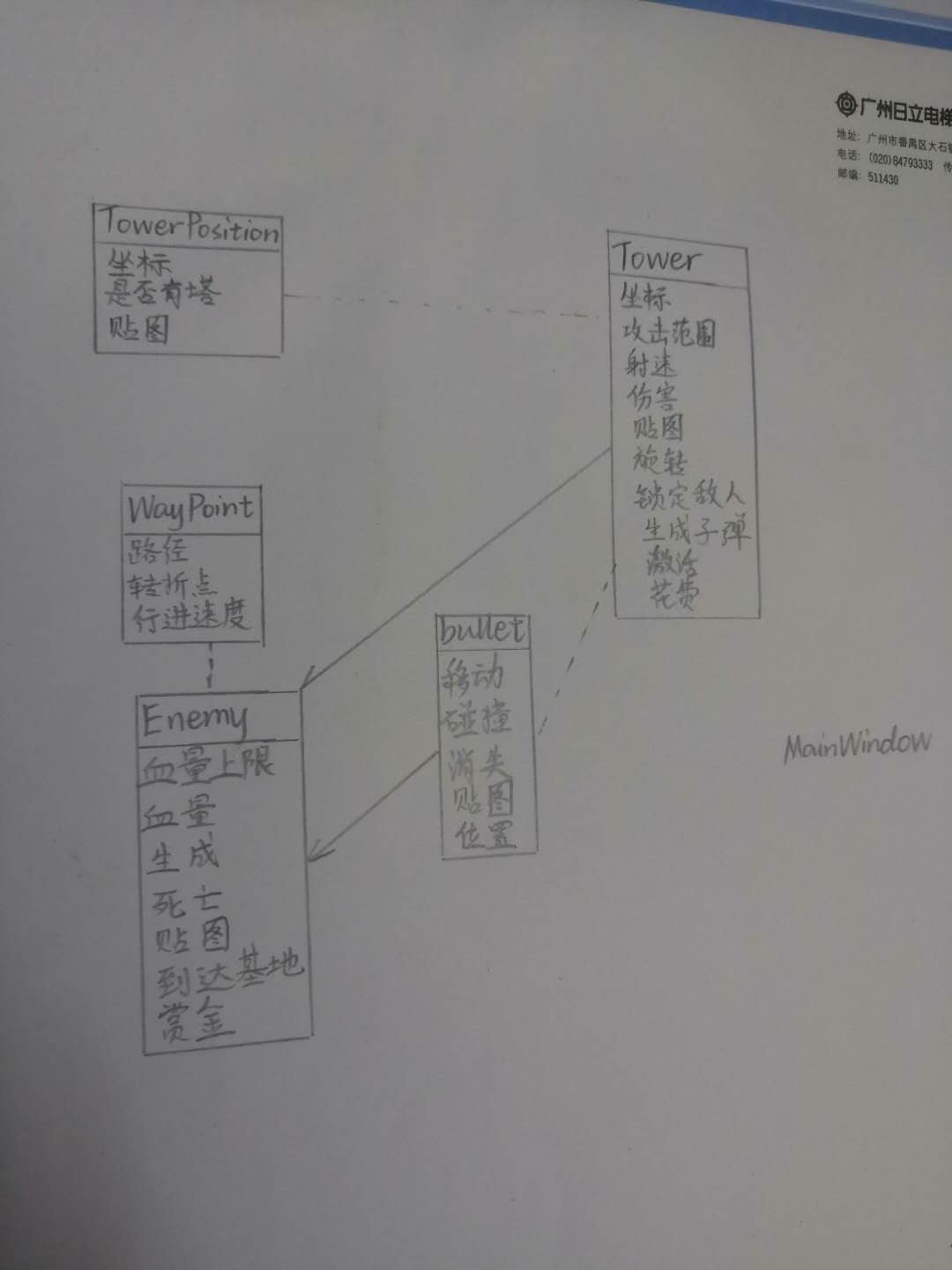
有地图和3\*4个槽点可放置炮塔，有攻击炮塔和子弹以及爬虫敌人，敌人路线和4个转向点，用QTimer来控制攻击间隔和移动速度，还有计算血量的基地及其衍生的胜利或失败界面，有金钱和赏金设置。

1. 游戏视频链接

链接：https://pan.baidu.com/s/17DpTGNWRY0PMz5wkOFQYkw

提取码：ybx9

1. 类图



1. 核心代码说明

1.void MainWindow::drawPlayerGold(QPainter \*painter)

{

painter->setPen(QPen(Qt::red));

painter->drawText(QRect(200,5,200,25), QString("GOLD : %1").arg(m\_playrGold));

} //对玩家进行经济限制，通过重载QPainter实现

2.如何实现购买炮塔时减钱？

if (canBuyTower() && it->containPoint(pressPos) && !it->hasTower())

{

m\_playerGold -= TowerCost;

it->setHasTower();

} //通过mousePressEvent是否实现来判别

3.贴图坐标变换

static const QPoint offsetPoint(-ms\_fixedSize.width() / 2, -ms\_fixedSize.height() / 2);

//获取坐标后将左上角移到中心

4.子弹飞行过慢怎么办？

void Bullet::move()

{

static const int duration = 100;

QPropertyAnimation \*animation = new QPropertyAnimation(this, "m\_currentPos");

animation->setDuration(duration);

animation->setStartValue(m\_startPos);

animation->setEndValue(m\_targetPos);

connect(animation, SIGNAL(finished()), this, SLOT(hitTarget()));

animation->start();

} // 100毫秒内击中敌人

5.炮塔旋转角度优化

if (m\_chooseEnemy)

{

QVector2D normalized(m\_chooseEnemy->pos() - m\_pos);

normalized.normalize();

m\_rotationSprite = qRadiansToDegrees(qAtan2(normalized.y(), normalized.x()))

} //标准化向量

6.如何绘制塔的攻击范围？

painter->save();

painter->setPen(Qt::white);

// 绘制攻击范围

painter->drawEllipse(m\_pos, m\_attackRange, m\_attackRange);

//中心+偏移=左上

// 尺寸大小派上用场了,当然也可以直接获取图片大小,是一样的

static const QPoint offsetPoint(-ms\_fixedSize.width() / 2, -ms\_fixedSize.height() / 2);

// 绘制炮塔并选择炮塔

// 这里将坐标原点移到m\_pos,加上偏移点到左上角

painter->translate(m\_pos);

painter->drawPixmap(offsetPoint, m\_sprite);

painter->restore();

1. 感想

这次大作业对我来说是一次不小的挑战，为了实现一些功能（课上没有涉及的知识点），我先是问老师加问同学，无果后只好向网上求助，日复一日地自学。不得不说，编写这样的程序十分枯燥，但在一点一点完成的过程中也充满了趣味。