

高级程序设计课程项目

——塔防游戏实验报告

曹雨森*

南京大学计算机科学与技术系

目录

1	项目构架	1
2	游戏特色	2
2.1	地图	3
2.2	防御塔	3
2.3	敌方单位	3
2.4	UI	3
3	实现特色	4
3.1	GameValue	4
3.2	GameItem	5
4	致谢	7

1 项目构架

(更新自 Project2 的实验报告)

(更新自 Project1 的实验报告,优化了继承结构并添加了许多新的内容)

采用了 Qt 的 Graphics View 框架

关于每个类的功能描述和实现方式我们放在下一个部分进行介绍,本部分仅描述项目的组织体系结构

*yusenCao@outlook.com

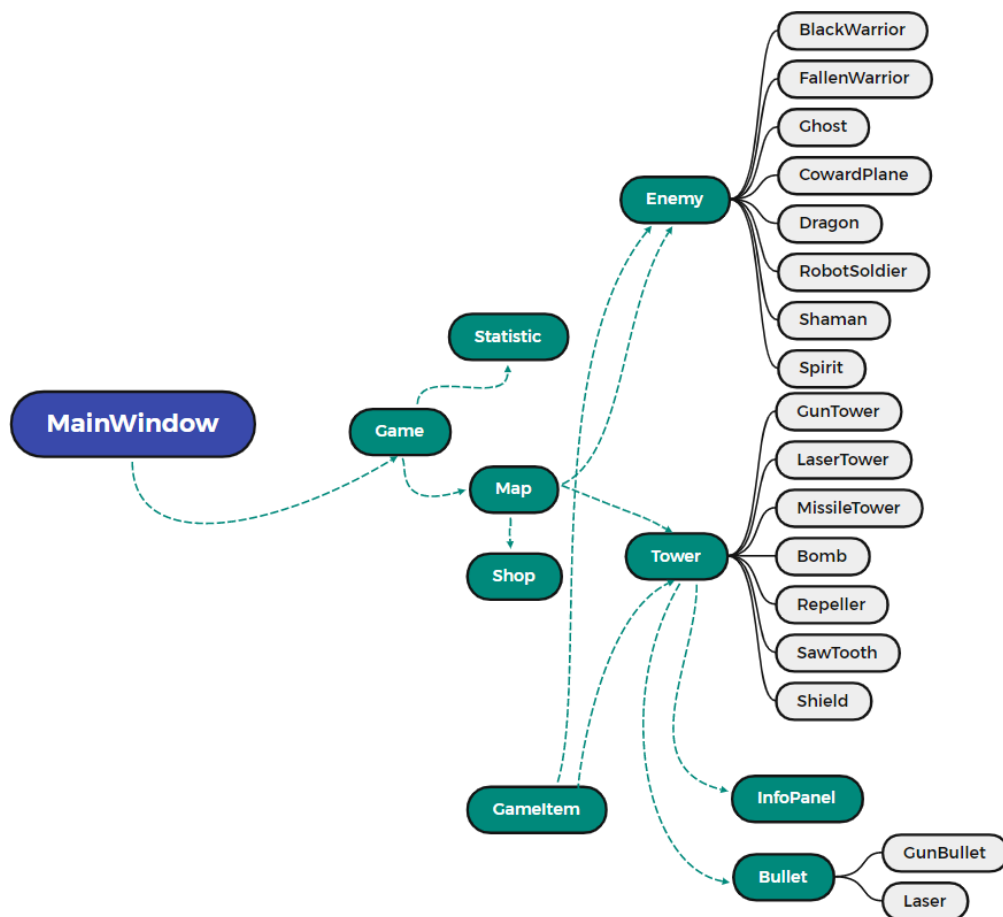


图 1.1: 项目组织关系图

- 蓝色框, 即 `MainWindow`, 表示的是应用的主窗口界面
- 绿色框表示当前类是继承于Qt 的自定义基类, 黑色框为其对应的子类。
而其中, `Tower` 和 `Enemy` 是继承于 `GameItem` (将防御塔和敌人的共同特征做了较大程度的抽象) 的派生类
- 绿色箭头表示被指向的一端是另一端的成员, 是绘图层面或者是对象组织层面的父子关系

2 游戏特色

(更新自 Project2 的实验报告)

(更新自 Project1 的实验报告)

游戏特色: 火力崇拜

个人想把这个游戏做的有趣一点, 自己夜里调 bug 的时候玩的乐, 同时也希望这个小项目更加丰富充实一些, 而不是成为我诸多短命的 repo 之一。

2.1 地图

- 添加地图和导出地图功能 (地图为符合特定语法规则的文本文件)
- 内置了四张地图: 冰原, 丛林, 沼泽和沙漠

2.2 防御塔

防御塔包含三种远程单位和四种近战单位

- 炸弹 (Bomb) 爆炸时击杀范围内的所有地面敌人
- 盾牌 (Repeller) 阻挡地面敌人的移动
- 锯齿 (SawTooth) 对地面敌人造成持续伤害
- 护盾发生器 (Shield) 保护范围内的所有远程炮塔, 承受其所受伤害
- 机枪塔 (GunTower) 发射子弹进行单体攻击 (子弹不可穿透, 考虑弹道命中)
- 激光塔 (LaserTower) 范围攻击地面敌人
- 导弹塔 (MissileTower) 发射跟踪导弹, 爆炸时产生范围伤害,

2.3 敌方单位

丰富的敌方单位

- 鬼魂 (Ghost) 没有特殊能力, 最基本的地面单位
- 幽灵 (Spirit) 具有攻击能力的普通地面单位
- 黑骑士 (BlackWarrior) 具有较快移动速度和较高血量, 地面单位
- 堕落骑士 (FallenWarrior) 具有较快移动速度, 高攻低血的地面单位
- 小飞机 (CowardPlane) 无攻击能力, 随机路径 (干扰我方防御塔), 遇到远程塔和子弹会加速的飞行单位
- 飞龙 (Dragon) 高攻击高防御, 移速慢的飞行单位
- 机械战士 (RobotSoldier) 小 boss 级别的地面单位, 血量极厚, 攻击范围远, 攻击时会逐渐提升攻速 (同时也会加伤)
- 萨满 (Shaman) 为己方队友加血, 不能被近战单位阻挡, 加血时会获得目标单位的移动速度 (比较想做出” 附身” 的感觉, 可惜没有美术素材)

2.4 UI

UI 上进行了更接近于现实游戏的优化

- 添加了好看的启动菜单 (好看的背景和标题 (×), 悬浮高亮的按钮)
- 添加了防御塔和敌人的血条绘制
- 单位受伤时会有闪烁特效
- 添加了敌方单位死亡的金币动画
- 添加了金币和生命值的动态显示
- 为防御塔添加了双击属性面板 (删除塔的功能被移动到属性面板中)
- 为敌人和防御塔添加了攻击范围绘制 (继承于 GameItem)
- 为防御塔增加了放置动画

- 添加了游戏结束的背景和标题
- 添加了暂停画面图标

3 实现特色

主要是谈谈我个人认为项目中的一些设计特色, 以及一些细节的实现。

3.1 GameValue

因为游戏中的很多值都具有一个特点: 拥有一个初始值 (在大多数情况下也即是最大值), 和一个当前状态值, 也可能有一个最小值

- 生命值有最大生命值和当前生命值, 同时也有最小值 0
- 移动速度有初始移动速度和当前移动速度 (受到敌方塔影响减速, 得到己方 buff 加速)
- 攻击范围可能随着等级提升而提升
- 游戏生命条数应该是一个正数, 游戏金币在很多游戏的设计中往往是有上限的
- ...

同时, 不同的值也有可能有不同的数据类型

- 生命值和攻击力往往是相同的数据类型, 为了方便持续伤害型敌人的设计 (攻击频率是由 advance 决定的, 与画面更新率有关, 攻击力仅仅为 1 也会因为持续伤害而产生巨大的威力), 所以为了不让生命值设计成许多大整数, 往往使用浮点数
- 游戏生命条数和游戏金币, 建筑物的造价, 赏金等, 很直观的就能看成应该是整型
- 攻击范围, 攻速这些根据具体实现而定

而为了保护好类的封装性, 我们更倾向于将这些数据设置为 private/protected, 使用

```
double HP;  
double getHP();  
void setHP(double HP);
```

这样的 getter 和 setter 来操纵他们 (事实上, 在 Project1 中我确实是这么做的) 而这样就使得:

一方面, 随着各种数据的引入, 需要增加越来越多的声明, 代码变得臃肿而难以维护

另一方面, 有的数据可能是只需要读的, 只要 getter 不要 setter, 有些数据有最大值 (HP | maxHP), 而有些可能暂时还没有 (在 Project1 里 speed 是初始化时就确定好的), 这使得这些明明具有共性的数据形式上很不统一, 零零散散的

所以我定义了一个模板类 GameValue 来使得游戏中的变量可以统一化

它有两个 private 成员, 一个表示当前值, 一个表示最大值 (我默认它的最小值是该数据类型的默认值), 这样就可以将 getter 和 setter 封装在这个类中了

这个实现的另外一个好处是, setter 将 curValue 约束在了 (Val(),maxValue) 这个区间内, 所以不需要考虑出现负数或者溢出的情况

比如生命值和游戏金币这样的数据, 肯定应该是恒正的, 并且应该有一个上限

```
template<typename Val>
class GameValue
{
private:
    Val curValue;
    Val maxValue;

public:
    GameValue() {
        curValue = Val();
        maxValue = Val();
    };
    GameValue(Val cur, Val max) {
        curValue = cur;
        maxValue = max;
    }
    ~GameValue() {}
    inline const Val &getCurValue() const {
        return curValue;
    };
    inline void setCurValue(const Val &newCurValue)
    {
        curValue = newCurValue;
        if (curValue < Val())
            curValue = Val();
        if (curValue > maxValue)
            curValue = maxValue;
    };
    inline void changeCurValue(Val delta) {
        setCurValue(curValue + delta);
    };
    inline const Val &getMaxValue() const {
        return maxValue;
    };
    inline void setMaxValue(const Val &newMaxValue) {
        maxValue = newMaxValue;
    };
};
```

3.2 Gameltem

仔细观察可以发现, Tower 和 Enemy 具有很多相同的特点

- 他们都具有生命值, 攻击力, 攻速, 攻击范围等
- 他们都会具有各种状态 (受到其他单位影响, 死亡等)

- 他们都会攻击, 都会获取其他目标
- 他们都需要接受鼠标事件 (点击显示攻击范围)
- 他们在 UI 绘制上有很多共同点 (都是一个方格大小的 boundingRect, 都有血量条)

因而将这些诸多要点抽出来, 就可以得到一个两者共同的父类——抽象类 GameItem (这里利用了 GameValue 模板类, 使得为 GameItem 添加数据成员变得相当容易)

```
#include <common.h>

class GameItem : public QGraphicsObject
{
public:
    GameItem();
    virtual ~GameItem();

public:
    QRectF boundingRect() const override;
    void paint(QPainter *painter, const QStyleOptionGraphicsItem
        ↪ *option, QWidget *widget) override;
    virtual void aquireTarget() = 0;
    virtual void attack() = 0;

    virtual bool getIsDead() const;
    virtual void setIsDead(bool newIsDead);

    QPointer<GameItem> getAtkTarget() const;
    void setAtkTarget(QPointer<GameItem> newAtkTarget);

    const QString &getName() const;
    void setName(const QString &newName);

    void mousePressEvent(QGraphicsSceneMouseEvent *event) override;
    void mouseReleaseEvent(QGraphicsSceneMouseEvent *event)
        ↪ override;

protected:
    QString name;

    int money;
    qreal atkRadius;
    bool isDead;

    QPointer<GameItem> atkTarget;

public:
    GameValue<qreal> HP;
    GameValue<qreal> atk;
    GameValue<qreal> speed;
    GameValue<int> blockNumber;
    GameValue<int> aquireCounter;
    QGraphicsEllipseItem *atkArea;
```

```
    QGraphicsRectItem *HPMeter;  
};
```

4 致谢

在此列出我参考过, 也仔细学习过的项目/视频/教程
(博客这种较为琐碎的资料我就不在此单独列出)

1. [GeeekExplorer-PVZ](#) 一份广为人知的 PVZ 的 Qt 项目, 高程入门新手教程
2. [Starting a Tower Defense Game](#) 远程类防御塔和子弹的实现
3. [Qt 学习之路 2](#) 一份较为详尽的 Qt 新手教程

高程 Project 撒花!