

坦克大战最终开发说明文档

- 一、软件需求分析
- 二、游戏开发进度与管理说明
- 三、主要技术模块说明
- 四、游戏主要部分白盒测试
- 五、游戏最终呈现形式与总结分析

一、软件需求分析

- 1.用户需求分析概况
- 预期用户调研
- 同类软件分析

需求分析

◆ 手游玩家对网络手游的不满和单机游戏模式的厌倦

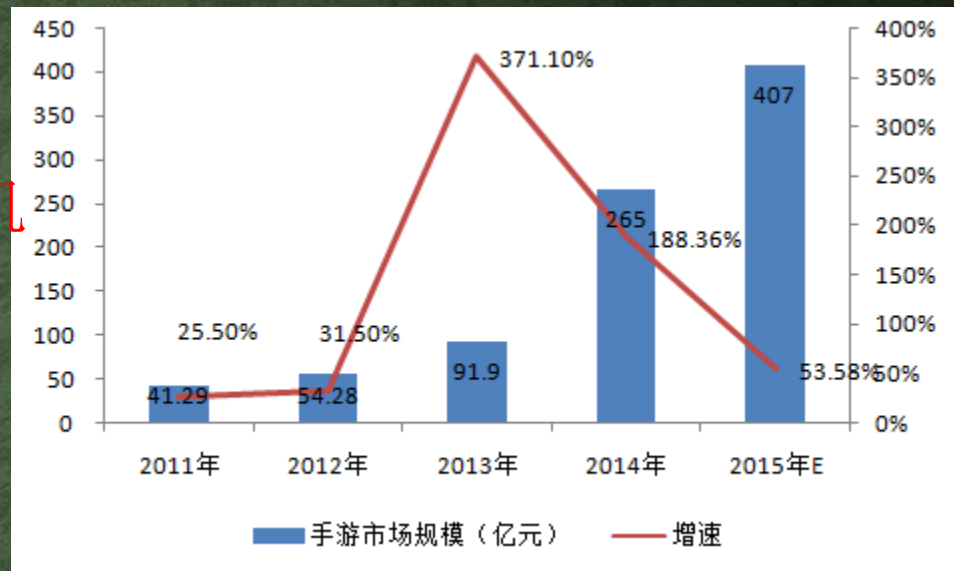
◆ 经典版本的地图太简单

◆ 现今游戏的画风千篇一律

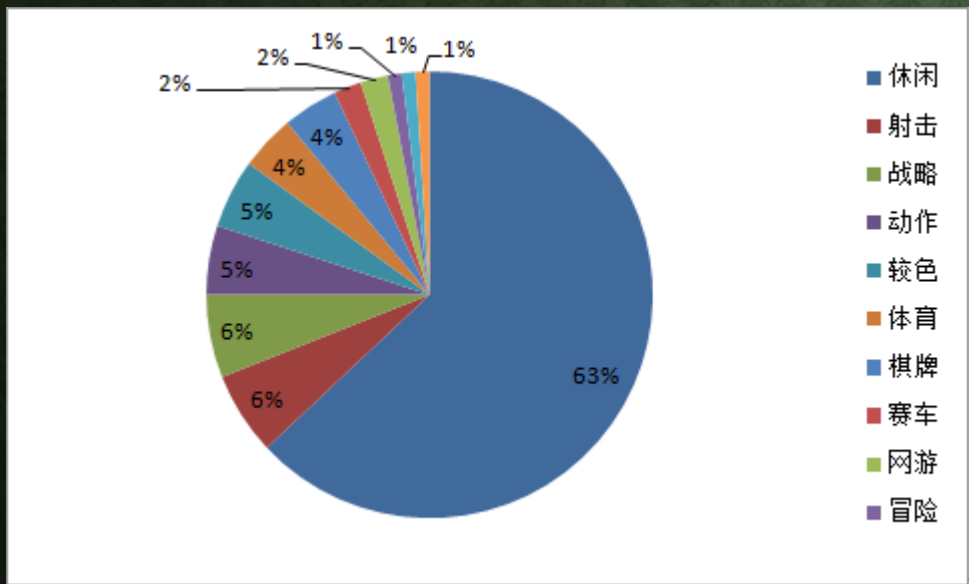


预期用户

随着网络和手机性能的提升，手机成了人们娱乐耗时的主要终端。



2015年中国手游市场不同类型游戏市场份额状况



2011-2015年中国手游市场规模及增长状况

休闲类游戏很受欢迎，我们将把经典带入休闲，为用户提供一款高质量的手游。

同类调查

◆ 系统复杂，收费过多，玩家成本偏大

◆ 缺少AI

◆ 早期怀旧党基数大



二、游戏开发进度计划书

- 进度、分工

- 版本管理

- 风险评估

任务划分

- 1.1 基本功能
 - 1.1.1 坦克、地形、升级道具等元素类的实现
 - 1.1.2 地图的绘制
- 1.2 运动实现
 - 1.2.1 虚拟按键、坦克的移动
 - 1.2.2 开炮、炮弹与其他元素的相互作用
 - 1.2.3 判断胜负条件
- 1.3 丰富游戏功能
 - 1.3.1 敌方坦克AI
 - 1.3.2 调试及添加其他功能
- 1.4 尝试实现对战平台

任务分配

任务 人员	1.1		1.2			1.3		1.4*
	1.1.1	1.1.2	1.2.1	1.2.2	1.2.3	1.3.1	1.3.2	
刘子腾	×		×			×	×	×
张振羽	×		×			×	×	×
张朔文		×		×	×	×	×	×
侯伯男		×		×		×	×	×

任务进展

ID	任务名称	开始时间	完成	持续时间			2016年11月				2016年12月			
					10/23	10/30	11/6	11/13	11/20	11/27	12/4	12/11	12/18	
1	1.1.1 坦克、地形、道具等元素类的实现	2016/10/24	2016/11/6	14天	<div></div>									
2	1.1.2 地图的绘制	2016/10/24	2016/11/4	12天	<div></div>									
3	1.2.1 虚拟按键、坦克的移动	2016/11/7	2016/11/20	14天	<div></div>									
4	1.2.2 开炮、炮弹与其他元素的相互作用	2016/11/4	2016/11/18	15天	<div></div>									
5	1.2.3 判断胜负条件	2016/11/19	2016/11/20	2天	<div></div>									
6	1.3.1 敌方坦克AI	2016/11/21	2016/12/2	12天	<div></div>									
7	1.3.2 调试及添加其他功能	2016/12/3	2016/12/11	9天	<div></div>									
8	1.4 实现对战平台	2016/12/12	2016/12/25	14天	<div></div>									

软件开发管理——配置管理

主要工具：github，
Travis CI。本次以
github为主



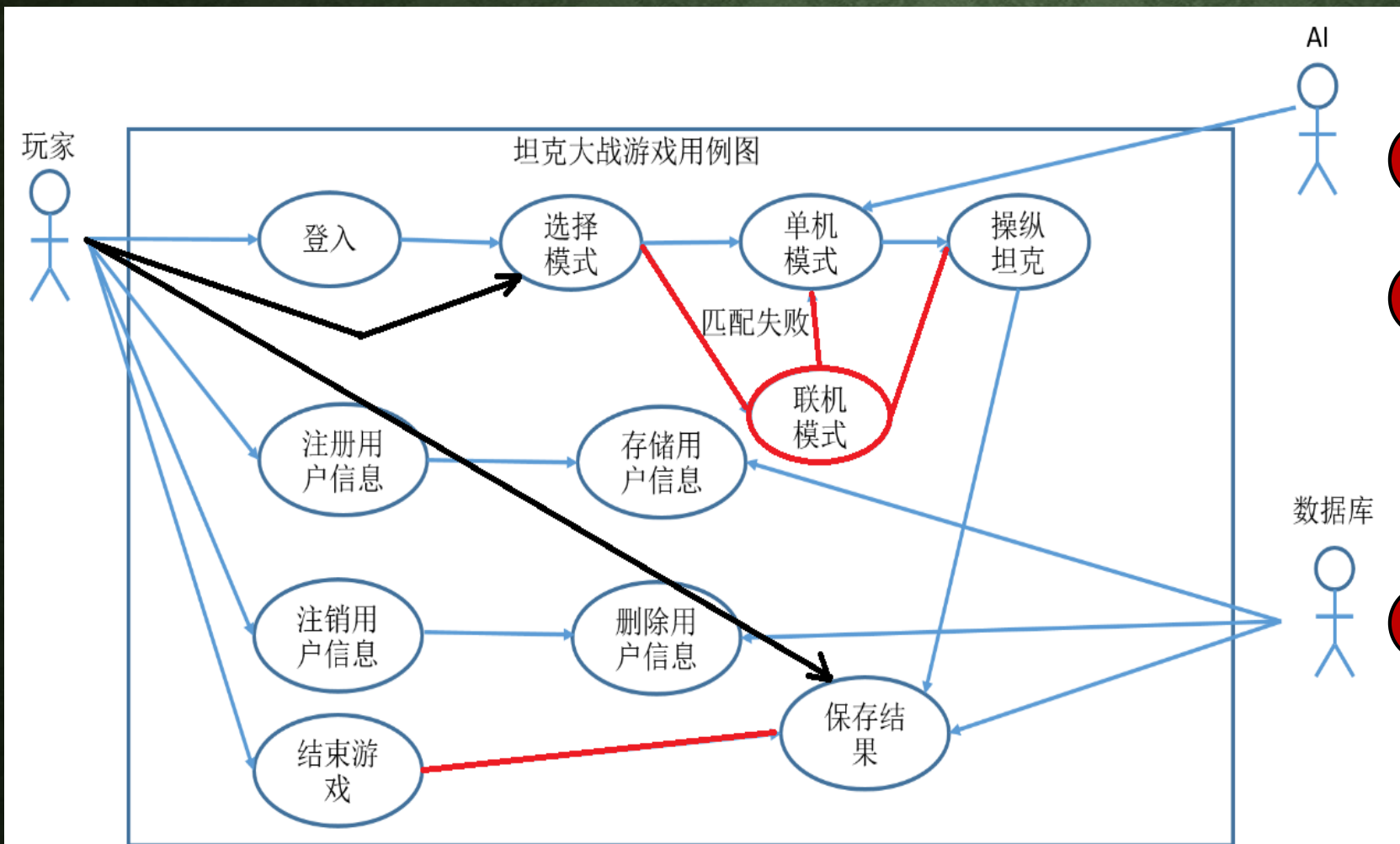
对软件的监督，项目
策划，**bug**调试，版
本控制

1. 一到两周为一个发布周期，对版本进行更新。
2. 需要提交说明文档，对本次更新及相关过程做描述和心得体会。

三、主要技术模块说明

- 1.用例分析
- 2.面向对象的整体规划
- 3.部分实现细节说明

最终用例图



1

网络模块被放弃了

2

用户录入信息后选择游戏基本设置，随即开始游戏；
用户通过登录信息作为凭证在本机上存储相应的游戏状态一共下次开始

3

游戏主要形式为AI与人组队进行人机对战。

用例分析

- 1 简介

一款兼有人机和人人对战的坦克大战**PC**休闲游戏。

- 2 角色

玩家、**AI**、数据库。

- 3 前置条件

联机时要确保两人及其以上在线。

- 4 后置条件

对结果进行保存，匹配不成功处理。

- **5 Basic Flow**

- (1) 登入
- (2) 选择游戏模式
- (3) 多人模式匹配玩家
- (4) 进行游戏
- (5) 判断胜负并结束游戏
- (6) 保存游戏结果

- **6 Sub Flow**

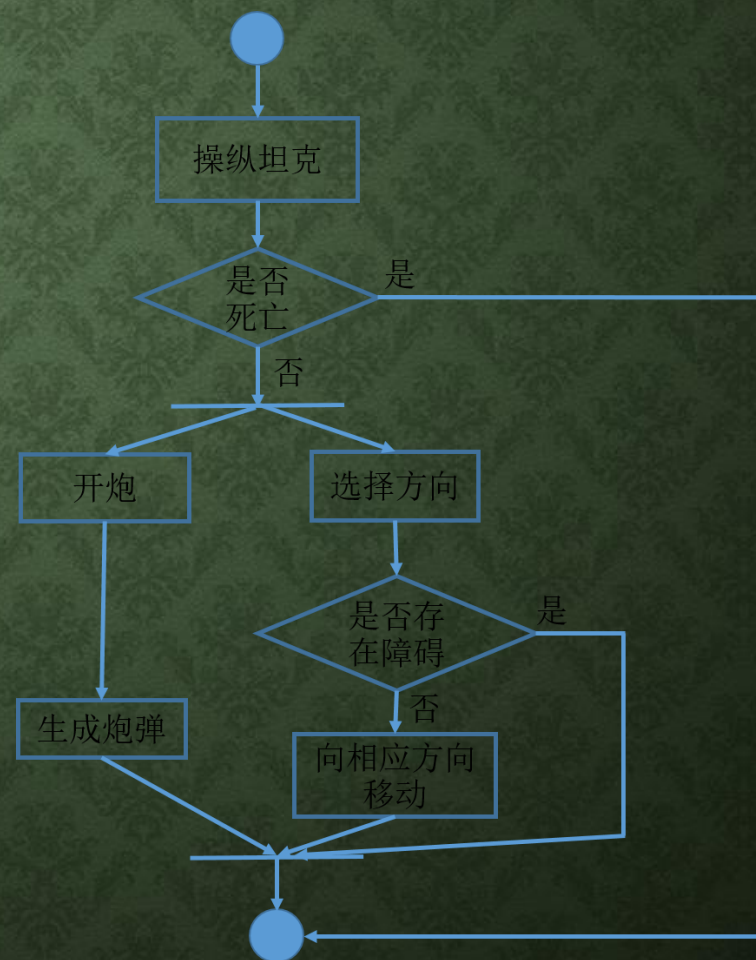
- (2) 1单人模式 (2) 2多人模式

- **7 Exception Flow**

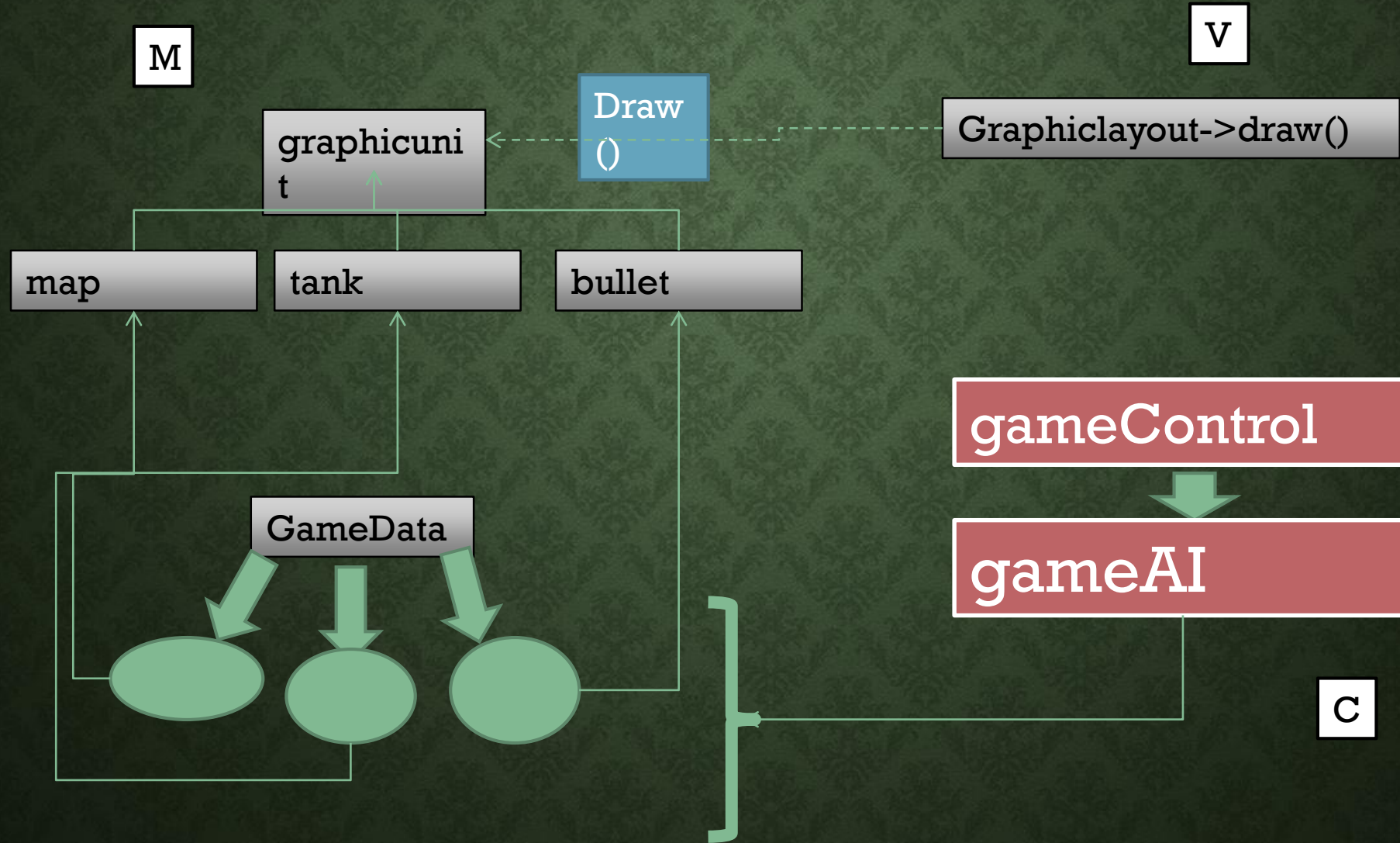
- (1) 登入信息匹配错误则登陆失败
- (2) 匹配玩家失败则推荐进行单人游戏

整体规划

ACTIVITY DIAGRAM——操纵坦克

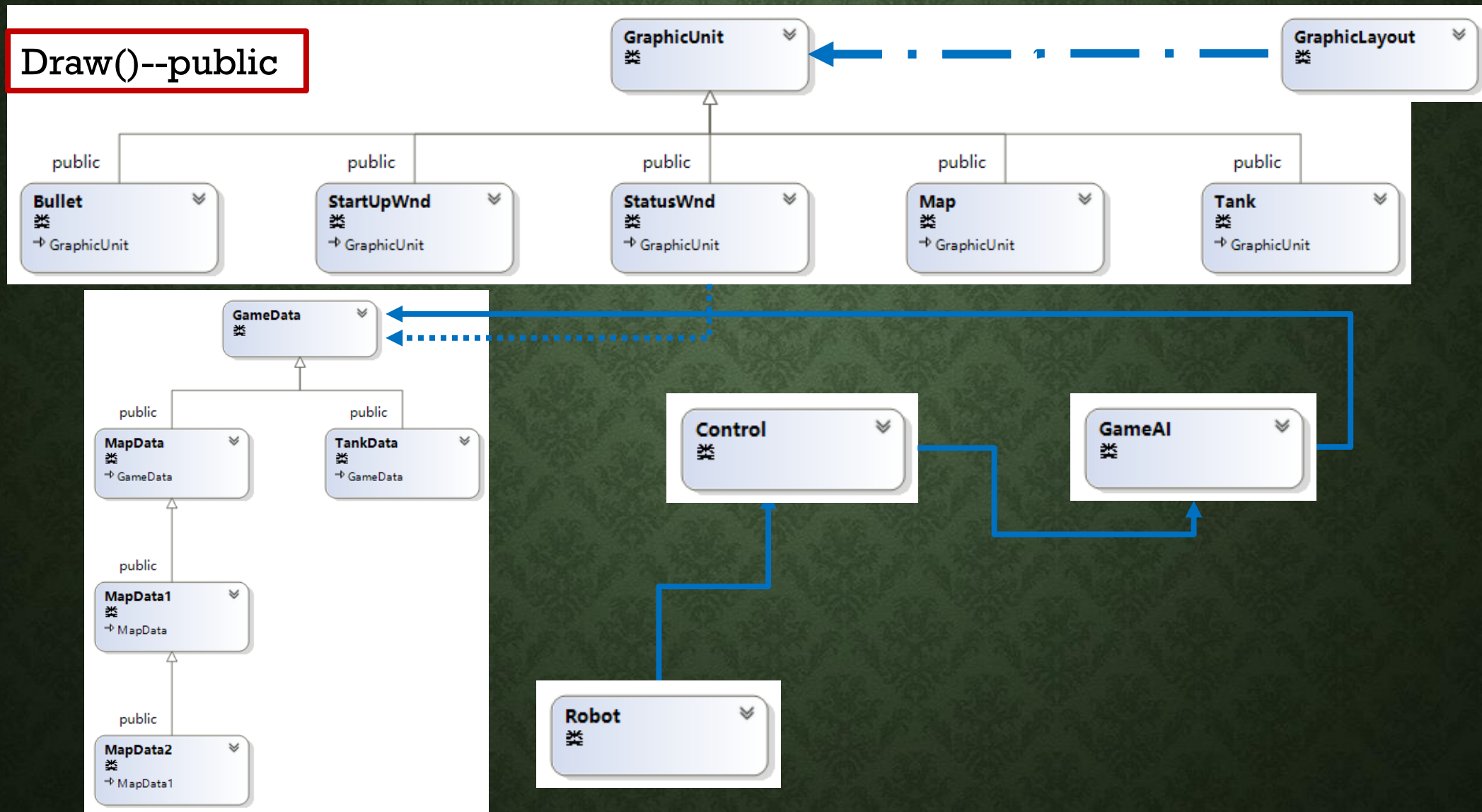


类设计总体简述



软件设计

Draw()--public



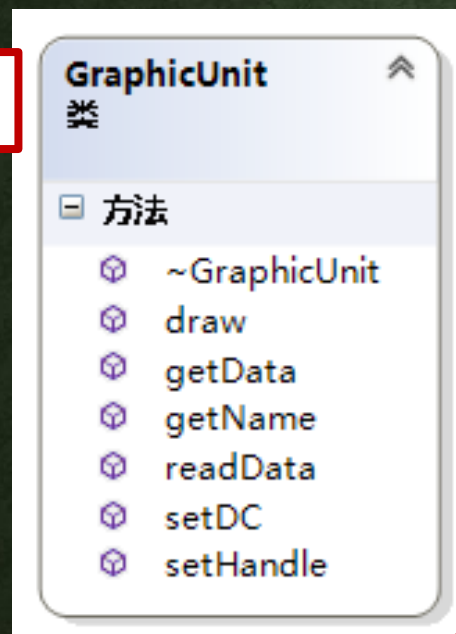
软件设计

关于Input数据到坦克数据之间数据流模型的重构改型

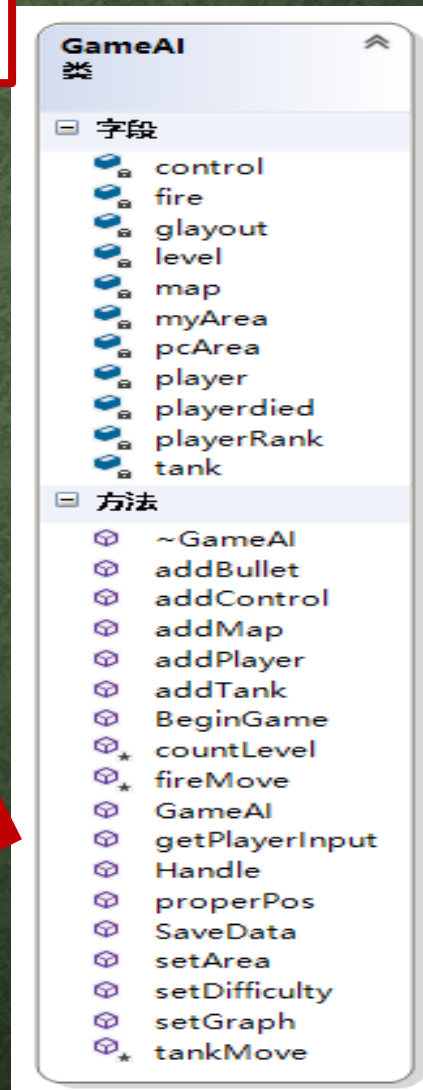


软件设计

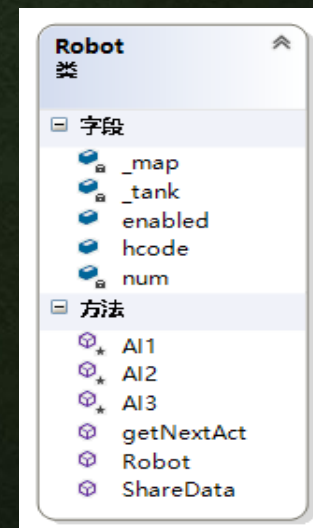
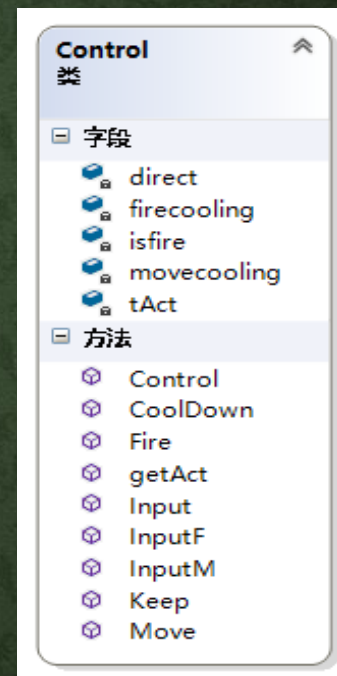
View



Controller



Model



部分具体实现细节说明

地图生成

地图元定义: `struct Brick {size, health, style}`

地图规划: `generateMap`

地图使用: `readData`进行动态绑定

相应: 子弹射击, 移动阻隔等内容

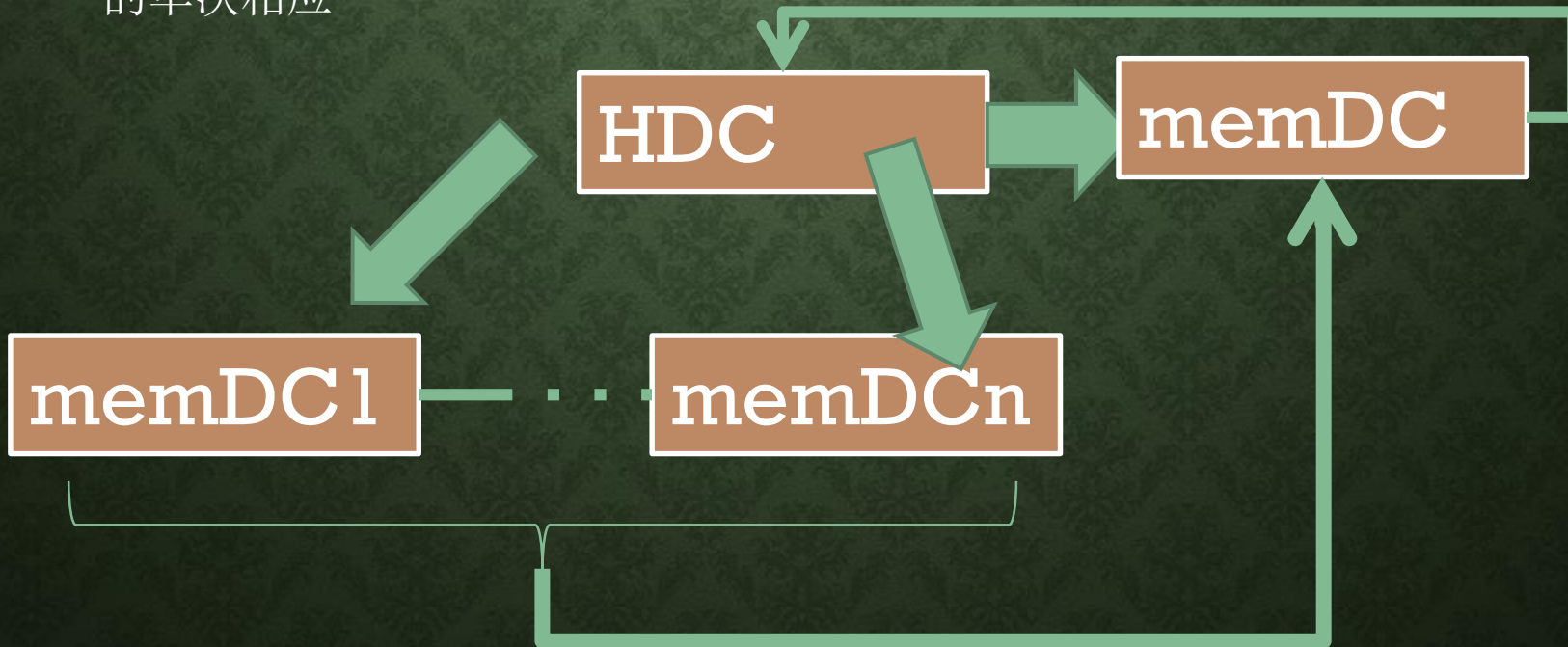


基于提升绘图速度的复制绘图

以多次内存操作换取对屏幕IO的较少次数操作

***data** 存放数据，根据对象特点有相应接口，绘图时遵循已生成数据进行图片加载与布局

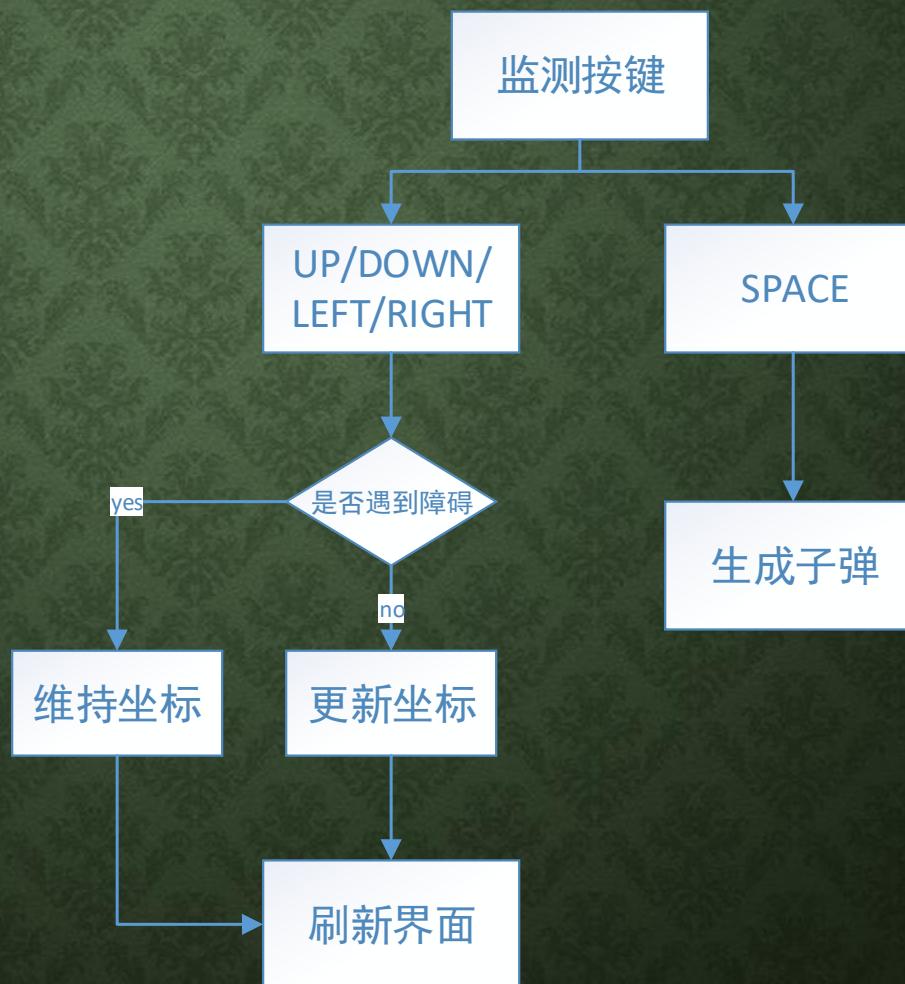
GraphicLayout中的**draw**方法由设备**HDC**获取相应格结构化内存，委派给**GraphicUnit**类进行**draw**，从而实现屏幕对单次刷新的单次相应



坦克的控制

TankData
posx, posy;
direction;

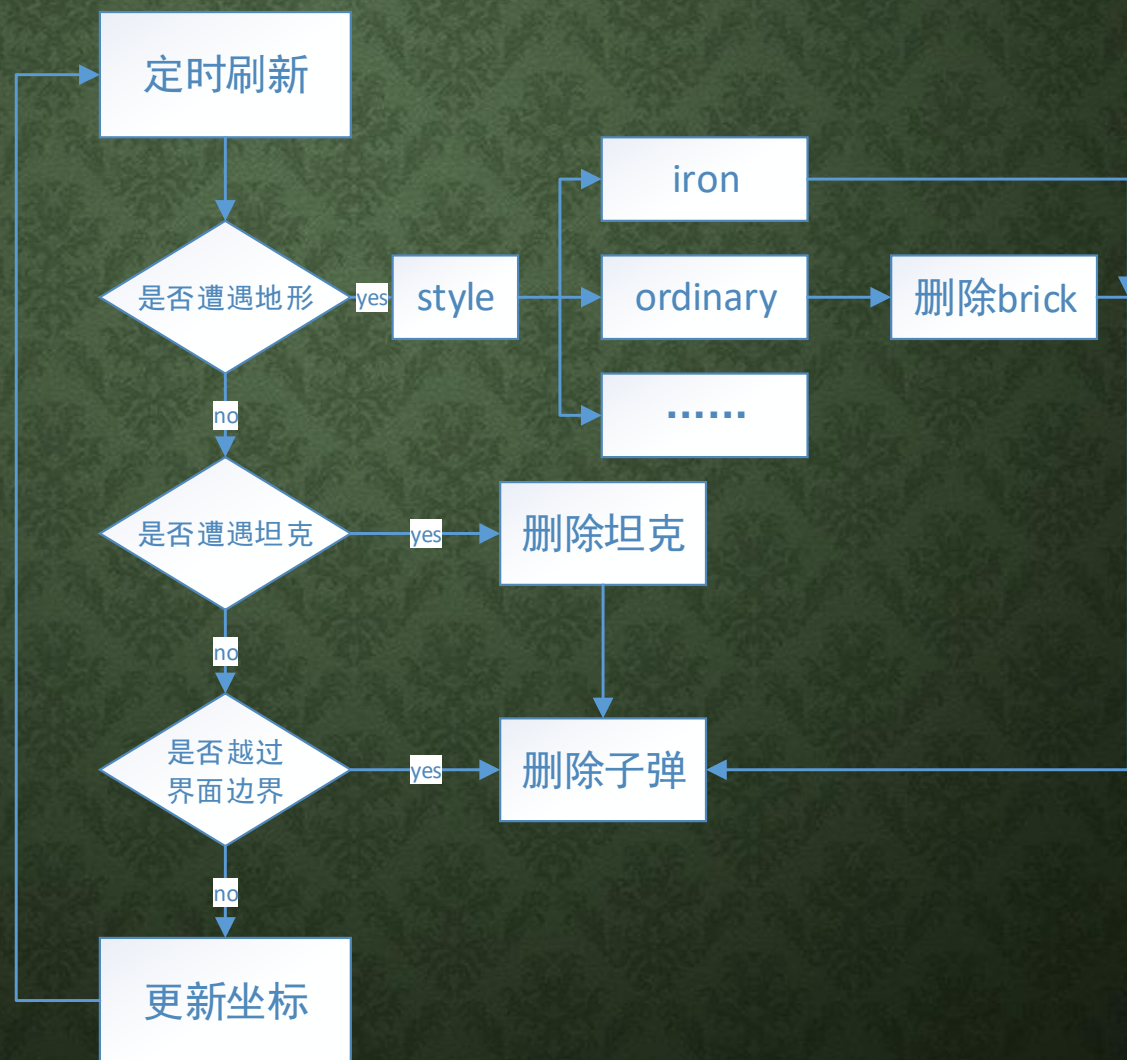
MapData
vector<Brick> ;



子弹的控制

Bullet

posx, posy;
direction;
style;

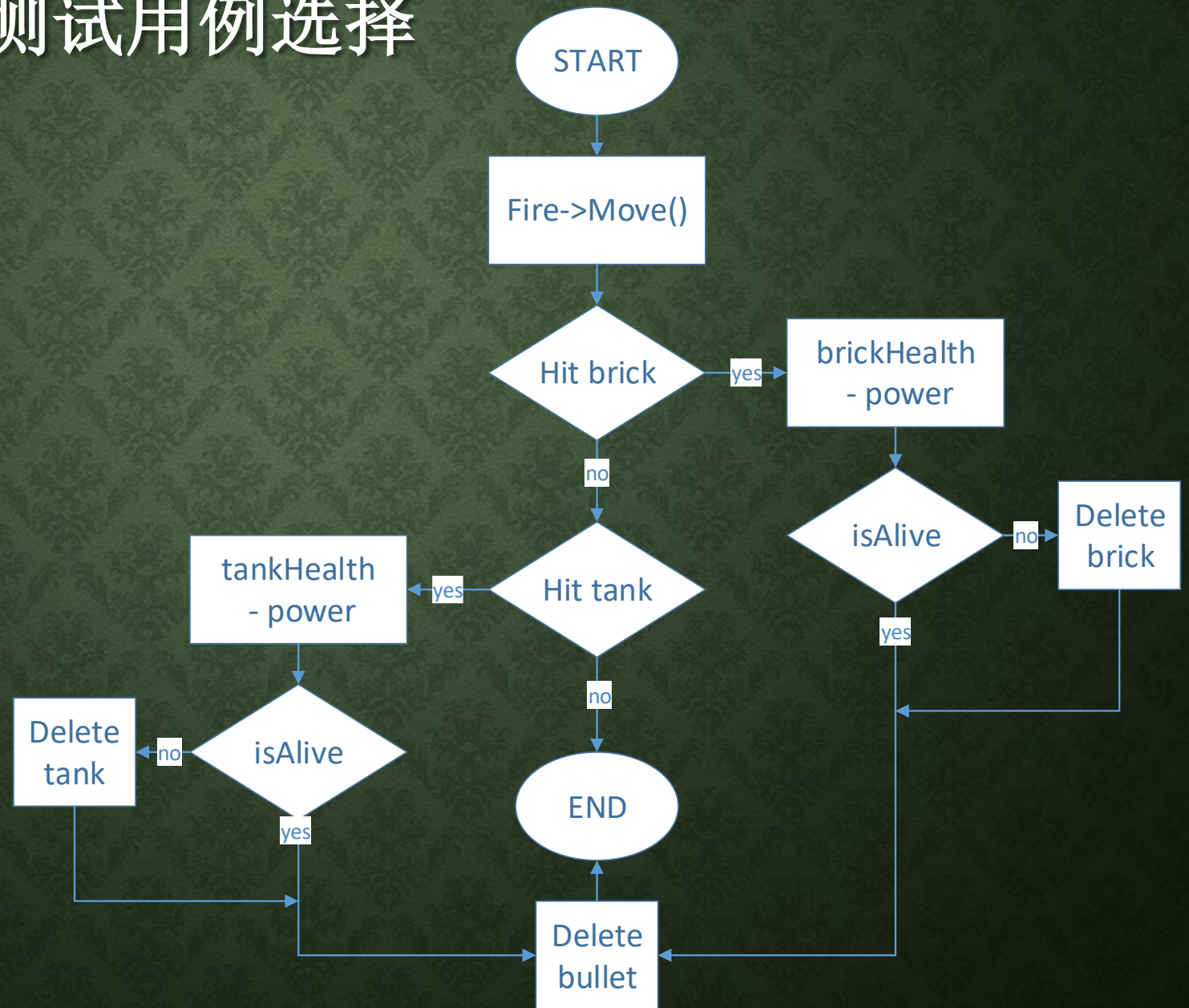


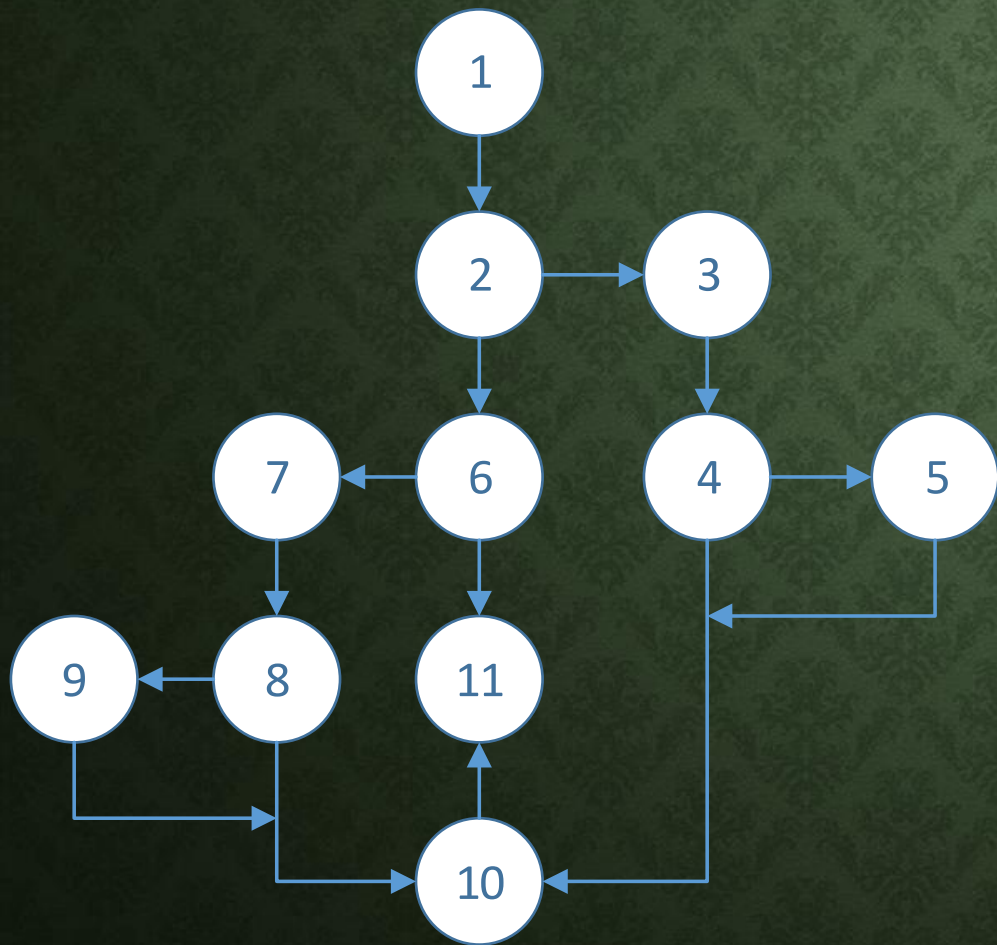
四、游戏白盒测试说明

- 1.测试用例选择分析
- 2.准备测试用例
- 3.测试环境与测试结果详解

测试用例选择

子弹控制部分
流程图





基本路径

- $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 10 \rightarrow 11$
- $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 10 \rightarrow 11$
- $1 \rightarrow 2 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 10 \rightarrow 11$
- $1 \rightarrow 2 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11$
- $1 \rightarrow 2 \rightarrow 6 \rightarrow 11$

测试用例准备

基本路径	元素类型	元素血量	元素正确处理	子弹正确处理
1 → 2 → 3 → 4 → 10 → 11	Brick	>bullet.power()	health – power	deleted
1 → 2 → 3 → 4 → 5 → 10 → 11	Brick	≤bullet.power()	deleted	deleted
1 → 2 → 6 → 7 → 8 → 10 → 11	Tank	>bullet.power()	health – power	deleted
1 → 2 → 6 → 7 → 8 → 9 → 10 → 11	Tank	≤bullet.power()	deleted	deleted
1 → 2 → 6 → 11	\	\	\	\

VS UNIT测试环境测试结果

```
namespace UnitTest1 {  
    TEST_CLASS(UnitTest1) {  
    private:  
    public:  
        TEST_METHOD_INITIALIZE(setUp) {}  
        TEST_METHOD(testmethod1) {}  
        TEST_METHOD(testmethod2) {}  
        TEST_METHOD(testmethod3) {}  
        TEST_METHOD(testmethod4) {}  
        TEST_METHOD(testmethod5) {}  
    }
```


断言ASSERT

```
TEST_METHOD (testmethod1) {  
    ...  
    ...  
    Assert::AreEqual(map->size(), ExpectedNumber, L"test failed");  
    Assert ::IsTrue(AI->isEmptyBullet (), L"test failed");  
}
```

全部运行 | 运行...

已通过测试 (5)

✓ testmethod1	< 1 毫秒
✓ testmethod2	< 1 毫秒
✓ testmethod3	< 1 毫秒
✓ testmethod4	< 1 毫秒
✓ testmethod5	< 1 毫秒

testmethod1

testmethod1

源: unittest1.cpp, 第 34 行

✓ 测试 已通过 - testmethod1

运行时间: < 1 毫秒

→ GameAI

fireMove()

```
RECT rt = {0, 0, 0, 0};
BOOLEAN hitOn = FALSE;
for (int i = 0; i < fire.size(); i++) {
    hitOn=FALSE;
    fire[i]~>Move();
    RECT bulletrect = { fire[i]~>Xpos() - 2, fire[i]~>Ypos() - 2,
        fire[i]~>Xpos() + 2, fire[i]~>Ypos() + 2 };

    //map bricks
    for (int j = 0; j < map1~>size(); j++) {
        Brick brick = map1~>getPos(j);

        if (IntersectRect(&rt, &brick.rect, &bulletrect)) {

            //try to delete the brick
            map1~>deleteBrick(j, fire[i]~>power());
            //delete the bullet
            //glayout~>deleteGraphicUnit(fire[i]);
            it = fire.begin() + i;
            fire.erase(it);
            hitOn=TRUE;
            break;
        }
    }
    if(hitOn)
        continue;
    //hit other tanks
    for(int j=0;j<tank.size();j++){
        if(tank[j]~>isAlive() && IntersectRect(&rt, &tank[j]~>getPos(), &bulletrect)){
            //
            tank[j]~>HitByBullet(fire[i]~>power());

            //
            //glayout~>deleteGraphicUnit(fire[i]);
            it=fire.begin()+i;
            fire.erase(it);
            hitOn=TRUE;
            break;
        }
    }
}
```

56 %

输出

显示输出来源(S): 生成

代码覆盖率结果

zhangsw13_ZHANGSW13-PC 2016-12-01

层次结构	未覆盖(块)	未覆盖(% 块)	已覆盖(块)	已覆盖(% 块)
fireMove	14	18.92%	60	81.08%
isEmptyBullet	1	16.67%	5	83.33%

全部运行 | 运行...

已通过的测试 (5)

✓ testmethod2	< 1 毫秒
✓ testmethod1	< 1 毫秒
✓ testmethod3	< 1 毫秒
✓ testmethod4	< 1 毫秒
✓ testmethod5	< 1 毫秒

testmethod2

testmethod2

源: unittest1.cpp, 第 46 行

✓ 测试 已通过 - testmethod2

运行时间: < 1 毫秒

→ GameAI

fireMove()

```
RECT rt = {0, 0, 0, 0};
BOOLEAN hitOn = FALSE;
for (int i = 0; i < fire.size(); i++) {
    hitOn=FALSE;
    fire[i]->Move();
    RECT bulletrect = { fire[i]->Xpos() - 2, fire[i]->Ypos() - 2,
                       fire[i]->Xpos() + 2, fire[i]->Ypos() + 2 };

    //map bricks
    for (int j = 0; j < map1->size(); j++) {
        Brick brick = map1->getPos(j);

        if (IntersectRect(&rt, &brick.rect, &bulletrect)) {

            //try to delete the brick
            map1->deleteBrick(j, fire[i]->power());
            //delete the bullet
            //glayout->deleteGraphicUnit(fire[i]);
            it = fire.begin() + i;
            fire.erase(it);
            hitOn=TRUE;
            break;
        }
    }
    if(hitOn)
        continue;
    //hit other tanks
    for(int j=0;j<tank.size();j++){
        if(tank[j]->isAlive() && IntersectRect(&rt, &tank[j]->getPos(), &bulletrect)){
            //
            tank[j]->HitByBullet(fire[i]->power());

            //
            //glayout->deleteGraphicUnit(fire[i]);
            it=fire.begin()+i;
            fire.erase(it);
            hitOn=TRUE;
            break;
        }
    }
}
```

56 %

输出

显示输出来源(S): 生成

代码覆盖率结果

zhangsw13_ZHANGSW13-PC 2016-12-01

层次结构	未覆盖(块)	未覆盖(% 块)	已覆盖(块)	已覆盖(% 块)
fireMove	14	18.92%	60	81.08%
isEmptyBullet	1	16.67%	5	83.33%

全部运行 | 运行...

已通过测试 (5)

- ✓ testmethod3 < 1 毫秒
- ✓ testmethod1 < 1 毫秒
- ✓ testmethod2 < 1 毫秒
- ✓ testmethod4 < 1 毫秒
- ✓ testmethod5 < 1 毫秒

testmethod3

testmethod3

源: unittest1.cpp, 第 59 行

✓ 测试 已通过 - testmethod3

运行时间: < 1 毫秒

→ GameAI

fireMove()

```
RECT rt = {0, 0, 0, 0};
BOOLEAN hitOn = FALSE;
for (int i = 0; i < fire.size(); i++) {
    hitOn = FALSE;
    fire[i]~>Move();
    RECT bulletrect = { fire[i]~>Xpos() - 2, fire[i]~>Ypos() - 2,
                        fire[i]~>Xpos() + 2, fire[i]~>Ypos() + 2 };

    //map bricks
    for (int j = 0; j < map1~>size(); j++) {
        Brick brick = map1~>getPos(j);

        if (IntersectRect(&rt, &brick.rect, &bulletrect)) {

            //try to delete the brick
            map1~>deleteBrick(j, fire[i]~>power());
            //delete the bullet
            //glayout~>deleteGraphicUnit(fire[i]);
            it = fire.begin() + i;
            fire.erase(it);
            hitOn = TRUE;
            break;
        }
    }
    if (hitOn)
        continue;
    //hit other tanks
    for (int j = 0; j < tank.size(); j++) {
        if (tank[j]~>isAlive() && IntersectRect(&rt, &tank[j]~>getPos(), &bulletrect)) {
            //
            tank[j]~>HitByBullet(fire[i]~>power());

            //
            //glayout~>deleteGraphicUnit(fire[i]);
            it = fire.begin() + i;
            fire.erase(it);
            hitOn = TRUE;
            break;
        }
    }
}
```

56 %

输出

显示输出来源(S): 生成

代码覆盖率结果

zhangsw13_ZHANGSW13-PC 2016-12-01

层次结构	未覆盖(块)	未覆盖(% 块)	已覆盖(块)	已覆盖(% 块)
addTank	0	0.00%	3	100.00%
fireMove	27	36.49%	47	63.51%

全部运行 | 运行...

已通过测试 (5)

testmethod4	< 1 毫秒
testmethod1	< 1 毫秒
testmethod2	< 1 毫秒
testmethod3	< 1 毫秒
testmethod5	< 1 毫秒

testmethod4

testmethod4

源: unittest1.cpp, 第 70 行

测试 已通过 - testmethod4

运行时间: < 1 毫秒

GameAI

fireMove()

```
RECT rt = {0, 0, 0, 0};
BOOLEAN hitOn = FALSE;
for (int i = 0; i < fire.size(); i++) {
    hitOn = FALSE;
    fire[i] -> Move();
    RECT bulletrect = { fire[i] -> Xpos() - 2, fire[i] -> Ypos() - 2,
                        fire[i] -> Xpos() + 2, fire[i] -> Ypos() + 2 };

    //map bricks
    for (int j = 0; j < map1->size(); j++) {
        Brick brick = map1->getPos(j);

        if (IntersectRect(&rt, &brick.rect, &bulletrect)) {

            //try to delete the brick
            map1->deleteBrick(j, fire[i] -> power());
            //delete the bullet
            //layout->deleteGraphicUnit(fire[i]);
            it = fire.begin() + i;
            fire.erase(it);
            hitOn = TRUE;
            break;
        }
    }
    if (hitOn)
        continue;
    //hit other tanks
    for (int j = 0; j < tank.size(); j++) {
        if (tank[j] -> isAlive() && IntersectRect(&rt, &tank[j] -> getPos(), &bulletrect)) {
            //
            tank[j] -> HitByBullet(fire[i] -> power());

            //
            //layout->deleteGraphicUnit(fire[i]);
            it = fire.begin() + i;
            fire.erase(it);
            hitOn = TRUE;
            break;
        }
    }
}
```

56 %

输出

显示输出来源(S): 生成

代码覆盖率结果

zhangsw13_ZHANGSW13-PC 2016-12-01

层次结构	未覆盖(块)	未覆盖(% 块)	已覆盖(块)	已覆盖(% 块)
fireMove	27	36.49%	47	63.51%
isEmptyBullet	1	16.67%	5	83.33%

全部运行 | 运行...

已通过测试 (5)

✓ testmethod5	< 1 毫秒
✓ testmethod1	< 1 毫秒
✓ testmethod2	< 1 毫秒
✓ testmethod3	< 1 毫秒
✓ testmethod4	< 1 毫秒

testmethod5

testmethod5

源: unittest1.cpp, 第 82 行

✓ 测试 已通过 - testmethod5

运行时间: < 1 毫秒

→ GameAI

fireMove()

```
    BOOLEAN hitOn = FALSE;
    for (int i = 0; i < fire.size(); i++) {
        hitOn=FALSE;
        fire[i]->Move();
        RECT bulletrect = { fire[i]->Xpos() - 2, fire[i]->Ypos() - 2,
                           fire[i]->Xpos() + 2, fire[i]->Ypos() + 2 };

        //map bricks
        for (int j = 0; j < map1->size(); j++) {
            Brick brick = map1->getPos(j);

            if (IntersectRect(&rt, &brick.rect, &bulletrect)) {

                //try to delete the brick
                map1->deleteBrick(j, fire[i]->power());
                //delete the bullet
                //glayout->deleteGraphicUnit(fire[i]);
                it = fire.begin() + i;
                fire.erase(it);
                hitOn=TRUE;
                break;
            }
        }
        if(hitOn)
            continue;
        //hit other tanks
        for(int j=0;j<tank.size();j++){
            if(tank[j]->isAlive() && IntersectRect(&rt, &tank[j]->getPos(), &bulletrect)){
                //
                tank[j]->HitByBullet(fire[i]->power());

                //
                //glayout->deleteGraphicUnit(fire[i]);
                it=fire.begin()+i;
                fire.erase(it);
                hitOn=TRUE;
                break;
            }
        }
    }
}
```

56 %

输出

显示输出来源(S): 生成

代码覆盖率结果

zhangsw13_ZHANGSW13-PC 2016-12-01

层次结构	未覆盖(块)	未覆盖(% 块)	已覆盖(块)	已覆盖(% 块)
addTank	0	0.00%	3	100.00%
fireMove	27	36.49%	47	63.51%

全部运行 | 运行...

已通过的测试 (5)

✓ testmethod1	< 1 毫秒
✓ testmethod2	< 1 毫秒
✓ testmethod3	< 1 毫秒
✓ testmethod4	< 1 毫秒
✓ testmethod5	< 1 毫秒

摘要

上次测试运行 已通过 (总运行时间 0:00:03)

✓ 5 个测试 已通过

→ GameAI

fireMove()

```
RECT rt = {0,0,0,0};
BOOLEAN hitOn = FALSE;
for (int i = 0; i < fire.size(); i++) {
    hitOn=FALSE;
    fire[i]->Move();
    RECT bulletrect = { fire[i]->Xpos() - 2, fire[i]->Ypos() - 2,
                       fire[i]->Xpos() + 2, fire[i]->Ypos() + 2 };

    //map bricks
    for (int j = 0; j < map1->size(); j++) {
        Brick brick = map1->getPos(j);

        if (IntersectRect(&rt, &brick.rect, &bulletrect)) {

            //try to delete the brick
            map1->deleteBrick(j, fire[i]->power());
            //delete the bullet
            //glayout->deleteGraphicUnit(fire[i]);
            it = fire.begin() + i;
            fire.erase(it);
            hitOn=TRUE;
            break;
        }
    }
    if(hitOn)
        continue;
    //hit other tanks
    for(int j=0;j<tank.size();j++){
        if(tank[j]->isAlive() && IntersectRect(&rt, &tank[j]->getPos(), &bulletrect)){
            //
            tank[j]->HitByBullet(fire[i]->power());

            //
            //glayout->deleteGraphicUnit(fire[i]);
            it=fire.begin()+i;
            fire.erase(it);
            hitOn=TRUE;
            break;
        }
    }
}
```

56 %

输出

显示输出来源(S): 生成

代码覆盖率结果

zhangsw13_ZHANGSW13-PC 2016-12-01

层次结构	未覆盖(块)	未覆盖(% 块)	已覆盖(块)	已覆盖(% 块)
fireMove	0	0.00%	74	100.00%
isEmptyBullet	0	0.00%	6	100.00%

五、游戏最终呈现形式



- 基本操作流程介绍
- 游戏规则简介
- 创新与不足分析

游戏基本操作流程

1 注册界面

LOG IN

✕

Username

Password

注册

确定

取消

2 玩家、关卡和难度选择

SELECT GAME MODE

✕

Select Game Parameter

玩家数目

1

add

minus

难度选择

1

add

minus

关卡选择

1

add

minus

确定

重置参数

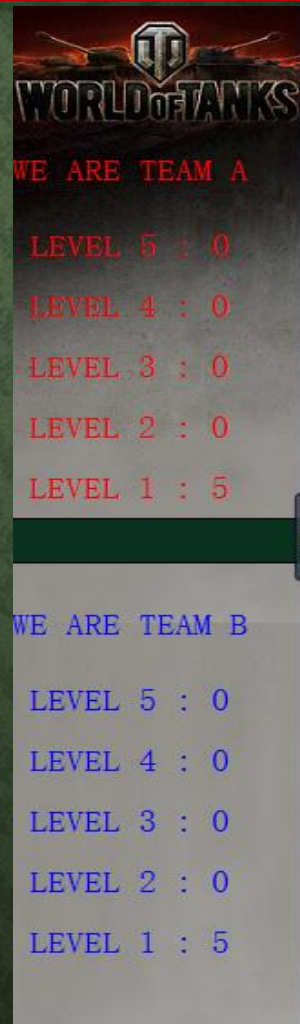
游戏体验—规则简介

1 游戏界面



通过上下左右控制移动，空格控制发炮；
等级1-**heath**=1，**speed**=1；
升级一次生命值加一，升级两次速递加以；
升级后**2**秒内坦克无敌；
有主角光环的是玩家。

2 状态栏



状态栏地区统计了人机双方目前在场的玩家分布，当真实剩余坦克数目超过五后玩家死亡后可从己方阵营再生

创新与不足--GUI设计的思考

1

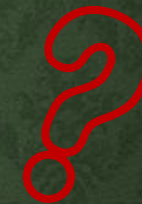
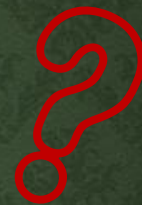
起初进行需求分析时，对于系统需求的分析不足，调研不足，**IDE**选择不佳

2

后期美工设计投入不足，没有合理的安排界面布置

3

对利用**DirectX**进行底层绘图代码迁移部分没有完成



游戏玩法与功能体验思考体验

1

使用**VS**对战模式取代原有经典模式，结合了时下流行的**xV**
x玩法，应用到坦克大战游戏中

2

可以为每位用户保存现有游戏的对战状态，方便下次继续

3

基于网络和数据库的在线匹配对战没有完成，只完成了本地
功能

