



Tank-Strike 项目开发文档

1. 项目简介

Tank-Strike 是一款基于 Java 的 2D 坦克对战游戏。支持 和 。核心亮点在于：随机生成迷宫算法、智能化人机、道具系统。

- **模式选择：**支持**双人对战 (PVP)**，或**人机对战 (PVE)**。
- **AI 逻辑：**采用状态机建模（躲子弹、防卡死、找道具），利用 **A*** 算法寻路。
- **随机地图：**随机穿凿。
- **物理引擎：**子弹撞墙会反弹。
- **道具：**生命药水、速度靴。
- **记录：**游戏结束后会保存最短通关时间（人机模式）。

2. 项目结构

代码主要分成了三块：界面显示、游戏逻辑控制、实体对象。

```
src/
└── Main.java          # 程序入口
└── com/
    ├── Screen.java      # 窗口容器，负责切换界面（菜单/游戏/结算）
    ├── GamePanel.java    # 游戏核心循环（Loop），负责画图和更新数据
    ├── HomePanel.java    # 开始菜单
    ├── EndPanel.java     # 结束画面
    ├── Tank.java          # 坦克类（坐标、血量、开火等）
    ├── TileManager.java   # 画地图的，把数组画成墙和地
    ├── ShowHeart.java    # 显示血量的UI
    ├── controller/
        ├── TankPanel.java # 接口
        ├── User1.java      # 玩家1 键盘监听
        ├── User2.java      # 玩家2 键盘监听
        └── UserAI.java     # AI 的逻辑代码（重点）
    ├── item/
        ├── Bullet.java     # 子弹
        ├── Item.java        # 道具
        ├── BulletManager.java # 管理所有子弹的list
        └── ItemManager.java # 管理道具生成
    └── util/
        ├── ImageLoader.java # 读图片的
        ├── MazeGenerator.java # 生成随机迷宫算法
        ├── Logger.java       # 读写排行榜文件
        ├── images/
        └── record.txt        # 存记录的文件
```

3. 核心技术实现

3.1 游戏引擎

在 `GamePanel` 里写了一个 `while` 循环来实现游戏主线程，帧率控制在 60 FPS 左右。

- **Update**: 算坐标、算碰撞。
- **Repaint**: 重新画图，用了双缓冲解决闪烁问题。

3.2 智能 AI (UserAI.java)

AI 是写得最复杂的部分。主要思路是给 AI 设定几个行为优先级，让它看起来比较聪明，不仅仅是寻路。

3.2.1 行为逻辑

每一帧 AI 都会判断当前局势，按下面的顺序决定做什么：

1. 保命（优先级最高）：

- **躲子弹**：AI 会遍历场上的子弹，如果发现有子弹离得很近且正冲着自己飞过来，就会往垂直方向移动一下来避开。
- **防卡死**：记录上一帧的坐标，如果发现长时间坐标没变（撞墙了）但还在试图移动，就会随机换个方向走几步，防止卡在一个角落动不了。

2. 决策：

- **残血时**：血量低于 30% 时，不再进攻，而是去地图里找回血道具；如果没道具，就往离玩家最远的角落跑。
- **丢失视野时**：如果中间隔着墙看不见玩家，就优先去找加速道具吃。
- **正常战斗**：保持和玩家一定的距离（风筝战术），不贴脸硬刚。

3.2.2 寻路算法 (A*)

为了让 AI 能绕过墙壁找到玩家，使用 **A* 算法**。

- **优化**：因为 A* 比较耗性能，设置为 500ms 的延迟，不用每一帧都重新算路径。
- **危险判定**：在计算路径代价时，除了距离，还需计算“危险值”。如果某个格子在玩家的直射范围内，那个格子的代价就会变大，使得 AI 就会倾向于走有墙壁掩护的路线。

3.2.3 攻击与移动

- **视线检测**：通过 `hasLos` 方法，判断 AI 和目标之间连线上有没有墙。如果有墙挡着就不开枪，防止需要开枪时处于冷却状态。
- **瞄准**：只有当 AI 和玩家的 X 轴或 Y 轴坐标差不多对齐的时候，才会开枪。
- **走位**：随机走位，让玩家不易命中。

3.3 碰撞检测

- **物体碰撞**: 把坦克、子弹都看作矩形，判断有没有重叠。
- **撞墙检测**: 根据坐标（为增加流畅性，以delta尺度缩小碰撞箱）算出当前在地图数组的哪个格子里，如果那个格子是 1 (墙)，就不能走。
- **反弹**: 子弹碰到墙的时候，把速度方向取反 (`vx = -vx, vy = -vy`)。

3.4 地图生成

- 用了**递归分割**的思路。
- 先生成全是墙的地图，然后随机挖一些房间，再打通路径，最后把死胡同去掉一些，保证地图是通的。

3.5 交互实现

通过 Java 的 `KeyListener` 监听键盘事件，`ActionListener` 处理按钮点击。

4. 游戏操作

按键说明

玩家	移动	开火
Player1	WASD	空格 (Space)
Player2	方向键	回车 (Enter)

人机模式 (PVE) 下，默认玩家为 **Player1**

规则

1. 把对方血量打空就赢了。
 2. 子弹碰到墙会反弹，小心别打到自己。
 3. 看到地上的道具可以去吃，会有 Buff。
-

5. 运行说明

环境

- JDK 8 以上
- 用 Eclipse 或 IDEA 打开项目即可

素材

图片: `src/com/util/images/`。

运行

运行 `Main.java` 启动游戏。

6. 代码统计

语言	文件数	总行数
Java	19	~2,500 行
