

# 比赛流程

参赛人员以小于或等于三人为一队参加比赛，经工作人员审核确认其参赛资格后，随机分组（分组过程公开透明）进行比赛。

比赛分为三部分：初赛，复活赛和决赛。

## 初赛和复活赛阶段

**初赛：**预计将在十二月中旬在线上进行，赛前 2 天收集参赛代码。初赛采用单淘汰赛制，直到决出前 4 进入决赛。初赛结束后将公布进入复活赛的队伍名单。

**复活赛：**预计将在十二月下旬在线上进行，赛前 2 天收集参赛代码。初赛中被淘汰的选手将进入复活赛，复活赛同样采用单淘汰赛制，决出前 4 进入决赛。复活赛结束后将公布进入决赛的队伍名单。

初赛和复活赛进程中，为确保公平，公示分组后将分时段进行全程直播。直播指派工作人员线上值班，随时解答选手的疑问。直播结束后将通过 QQ 群和微信公众号等渠道公布比赛结果，并上传直播视频以供检查。

针对比赛中可能出现的异常情况，将设置应急预案，准备备用平台以供直播，同时做好代码复审和平台异常情况筛查工作。

## 决赛阶段

**决赛：**预计将在十二月底举办，选手需在限定时间内完成决赛的代码编写。

比赛将以决赛晚会形式邀请诸位嘉宾、参赛选手及同学到场观看。决赛晚会围绕比赛设置决赛正赛、观众互动、致辞颁奖三个环节。此外，决赛将进行全程录制，晚会结束后上传视频以备检查。比赛结束后当场公布比赛结果并举行颁奖仪式。

# 代码提交

## 代码提交

**初赛 12 月 15 日 8:00 截止**

选手将编写好的程序代码(playerA.cpp 和 playerB.cpp)打包成 zip 或 7z 文件

大小小于 50MB

以附件的形式发送到

**TuringCup2024@163.com**

邮件主题为**队伍编号-队伍名** 例子:

200-图灵杯技术部

# 选手程序编译

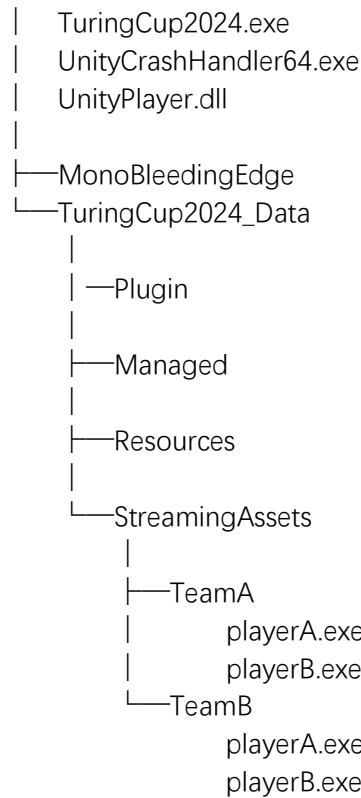
编译方法：

- 1.选手可使用命令行指令 `g++ playerA.cpp -o playerA.exe` 和 `g++ playerB.cpp -o playerB.exe` 完成编译
- 2.选手可以使用 Dev-C++等软件运行编译

编译完成后的可执行文件替换游戏平台中的选手文件  
如果需要复杂的编译过程或者加载 AI 模型请提前和工作人员讲，并且在选手文件中的 Compile.md 文件中 写好步骤

## 程序运行

目录结构：



选手写好代码编译后生成的两个可执行文件复制到  
TuringCup2024/TuringCup2024\_Data/StreamingAssets 中的 TeamA或 TeamB下面

然后运行 TuringCup2024/TuringCup2024.exe后， Unity会自动打开 TeamA/playerA.exe  
和 TeamB/playerB.exe并且开始比赛



# 游戏规则

## 积分与胜利条件

游戏总时长为 150 秒，游戏中积分会根据己方存活角色每人每秒增加 5 分，也可以通过击杀敌人(50)或获取金币(100)得到对应的分数，时间结束后，积分多的一方获胜，一样多则为平局。

## 视野

Unity 端给双方选手的数据都是经过视野处理的，PlayerA 能得到的数据是视野内敌方士兵数据和双方的工事数据

一个士兵的视野半径是 10 单位，同时会被墙阻挡

PlayerA 获得的数据是 TeamA 三个士兵的视野的集合

## 死亡

士兵死亡后 15 秒后会在两个复活点随机复活(阵营 A 为地图左下方和右下方，阵营 B 为地图左上方和右上方)

角色在死亡或不可见时，角色的位置血量等信息不会更新

## 药水

药水在被拾取后 15s 后刷新，药剂效果为随机抽取，效果为如下所示：

- 1.恢复当前角色 15 滴血
- 2.提高角色攻击力 2 点【子弹伤害】，持续时间 5s
- 3.提高角色射速为原来的 1.5 倍，持续时间 5s
- 4.提高角色防御力 2 点，持续时间 5s
- 5.提高角色移速 20%，持续时间 5s

## 金币

拾取金币可以为阵营获得 100 分，金币在被拾取后的 15 秒后在刷新区域内随机刷新

# 子弹伤害

Gunner:5

Hurler:8

Medic:5

## 战术小队介绍

### Gunner

移动速度 5      血量 50  
防御力 4          射程 20

技能 1:

射手提高移速 30%

持续时间为 5s , CD 20s 使用例子:

skill Gunner 1

技能 1:

提高攻击速率为自身原本射速的 200%

持续时间为 5s , CD 30s 使用例子:

skill Gunner 2

### Hurler

移动速度 5      血量 50  
防御力 6          射程 20

技能 1

向目标区域投掷一个范围 3 格的燃烧弹，每 0.5 秒造成 4 点燃烧伤害。(!!! 注意，该技能有友伤，请尽量不要误伤队友。)

持续时间为 3s , CD 25s

例子

skill Hurler 1 3.5 3.5

技能 2

在原地布置一颗隐形地雷，触发时在范围内造成 0-20 伤害不等，根据距离造成伤害，隔墙无法造成伤害，不对友军造成伤害。

例子

skill Hurler 2

# Medic

移动速度 5      血量 50  
防御力 6      射程 20

技能 1  
向附近投射一个治疗光圈，为光圈内的友军恢复 10 点生命值，并在接下来的 3 秒内提供 2 点防御力加成

CD 25s  
skill Medic 1 10 10

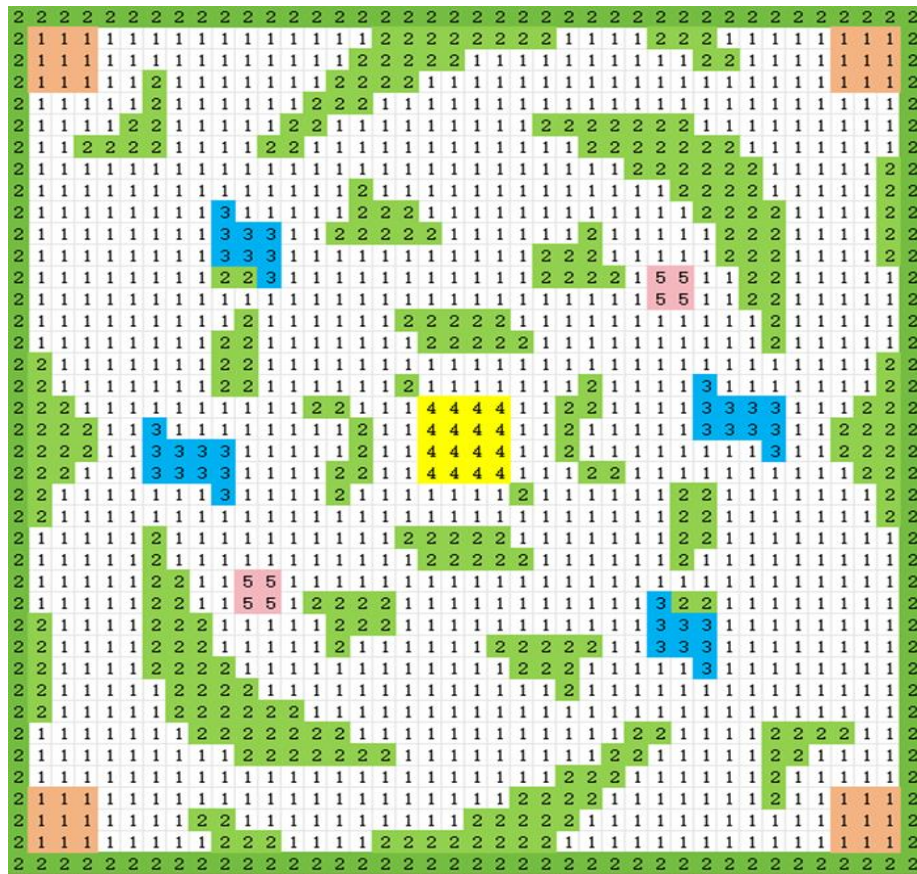
技能 2  
为目标友军或者自身提供一个可抵挡 8 点伤害的能量盾，能量盾不可叠加。

CD 60s  
skill Medic 2 Gunner

# 编码样例

## 地图说明

- 1 是空地
- 2 是墙(会阻挡一切)
- 3 是水(子弹等物体可以穿过，角色不能穿过)
- 4 是金币刷新地
- 5 是药水刷新地



## 游戏数据说明

1. vector<vector<int>> game\_map: 游戏地图, (0,0)为地图左下角
2. pair<float,float> GunnerA\_Pos: GunnerA 位置
3. pair<float,float> HurlerA\_Pos: HurlerA 位置
4. pair<float,float> MedicA\_Pos: MedicA 位置
5. pair<float,float> GunnerB\_Pos: GunnerB 位置
6. pair<float,float> HurlerB\_Pos: HurlerB 位置
7. pair<float,float> MedicB\_Pos: MedicB 位置
8. float GunnerA\_HP: GunnerA 血量
9. float HurlerA\_HP: HurlerA 血量
10. float MedicA\_HP: MedicA 血量
11. float GunnerB\_HP: GunnerB 血量
12. float HurlerB\_HP: HurlerB 血量
13. float MedicB\_HP: MedicB 血量
14. bool has\_coin: 金币存在信息(为 true 则金币存在)
15. pair<float,float> coin\_Pos: 金币位置信息
16. int scoreA: A 阵营得分
17. int scoreB: B 阵营得分
18. vector<Fire> fire\_pos: 场上火焰的位置
19. vector<Potion> potion\_pos: 场上药水的位置
20. int frame\_count: 帧数信息

## 选手程序输出

```
1 skill Gunner 1
2 move Gunner 4 30
3 move Hurler 15 15
4 skill Gunner 2
5 shoot Gunner 1 30
6 End
```

选手可以输出的命令类型一共3种

1. 移动(move)
2. 射击(shoot)
3. 技能(skill)

移动:

例子:

```
move Gunner 15.5 24
```

己方阵营的Gunner向 (15.5, 24)移动 寻路需要由选手完成

射击:

例子:

```
shoot Gunner 15.5 24
```

己方阵营的Gunner向(15.5, 24)射击

技能:

例子:

```
skill Medic 1 3.5 3.5
```

己方医疗兵开启技能1(治疗) 坐标(3.5,3.5)

选手程序每一帧输出命令以 End 结尾,

若程序崩溃, 请减少一帧中输出的命令!