

FC18 参赛规则

内测版

开发者 自动化系科协竞赛部
比赛网址 ?
? ?
更新日期 2021 年 2 月 4 日

目录

1	游戏说明	2
2	你需要做什么	3
3	角色说明	3
3.1	塔	3
3.1.1	概述	3
3.1.2	具体说明	4
3.2	作战兵团	6
3.2.1	概述	6
3.2.2	具体说明	7
3.3	工程兵团	7

FC18 为四方势力在地图上进行回合制对战的策略游戏，玩家需要力求攻占其他玩家的塔、占领尽可能大的领地、消灭或俘虏其他玩家的兵团以获得胜利。每个玩家需要编写 AI，根据裁判程序提供的场地信息，决策己方势力在该回合的行动，并返回给裁判程序，以控制己方的行为。

1 游戏说明

作为塔防游戏，每个势力在开始时各在一角拥有一座塔，塔周围的一定区域是自己的领地。玩家需要利用塔的生产力，完成生产兵团或升级塔的任务。生产出的作战兵团可以在场上移动，攻击其他势力的塔或兵团（减少他们的生命值）；生产出的工程兵团则可以完成修改地形、修理塔（恢复塔的生命值）的任务。其中，塔的等级越高，就会拥有更高的生产力、战斗力、生命值上限。塔也可以攻击敌方的兵团，如果塔内有己方兵团驻扎，则塔的战斗会更强。

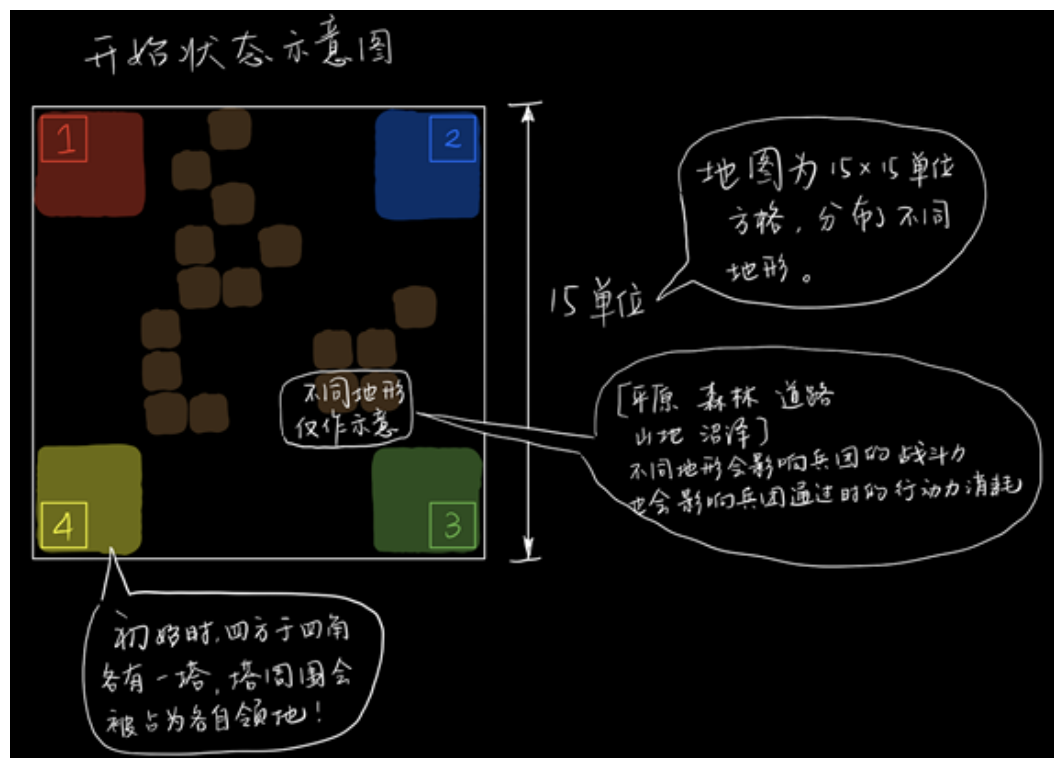


图 1: 游戏场地示意图：开始状态

2 你需要做什么

所有玩家的 AI 都可以从 Info 中读取当前场上各方势力的兵团、塔的信息，并设计算法，并按照统一的接口 CommandList 给裁判程序回传命令，操控己方势力；在游戏中，选手只需要在 ai.cpp 文件中的 void player_ai(Info& info) 函数中填写自己的代码，并最终只需要提交 ai.cpp 文件。

```
1 class CommandList
2 {
3 public:
4     void addCommand(commandType _FC18type, initializer_list<int>
        _FC18parameters);
5     void removeCommand(int n); //【FC18】移除第n条命令
6     vector<Command> getCommand() { return m_commands; } //【FC18】获取所有命令
7 };
```

添加命令有关代码

3 角色说明

游戏中有以下角色：塔 Tower，作战兵团（分为战士 Warrior、弓箭手 Archer、法师），工程兵团（分为建设者 Builder、开拓者 Extender）。

3.1 塔

3.1.1 概述

塔是在地图上固定的，具有对一定范围内（距离为 2 及以上，即 5*5 的方形内）敌对势力塔或兵团自动攻击能力的建筑，每个势力最初时拥有 1 座塔，每个地图方格内最多有 1 座塔，每个势力最多拥有 10 座塔。当某势力没有塔时，该势力将被判为失败。塔有以下属性：

- (1) 等级 N 。
- (2) 生产力 W_N 。
- (3) 等级生命力上限 HP_N ，当前生命力 hp 。
- (4) 等级战斗力 F_N ，实际战斗力 f 。

2	2	2	2	2
2	1	1	1	2
2	1	单位	1	2
2	1	1	1	2
2	2	2	2	2

图 2: 距离计算示意图

```

8
9 struct TowerInfo {
10
11     TTowerID      ID;    //防御塔ID
12     TPlayerID     ownerID; //所属玩家ID
13     TPoint        position; //位置
14     TProductPoint productPoint; //生产力
15     TProductPoint productConsume; //当前生产任务尚需完成的剩余生产力值
16     TBattlePoint  battlePoint; //战斗力
17     THealthPoint  healthPoint; //生命值
18     TLevel        level;    //等级
19     productType   pdtType;  //当前生产任务类型
20 };
21

```

防御塔结构体

3.1.2 具体说明

- 1) 塔的等级 N 为 1-8 的正整数, 从等级 N 升级到 $N+1$ 需要消耗的生产力值为 $40 \cdot N$ 。
- 2) 塔会根据自身的等级状况获得一个生产力数值 W_N , 代表其一个回合能产生的生产力, 具体的数据如表1所示。生产力 W_N 每回合更新, 如果该回合没有使用生产力, 下一回合不会累积。在同一时刻, 塔可以选择一种生产任务 (包括生产战士、生产弓箭手、生产法师、生产建造者、生产开拓者、升级塔自身) 每一种任务需要消耗不同的生产力点数, 具体情况见表2。兵团生产任务完成之后, 塔所在的方格将立即生成一个对应兵团。防御塔所在方格内允许存在多个兵团, 不必遵循同一单元格仅能存在一个兵团的限制。在 `struct TowerInfo` 中, 可以访问到该塔当前未完成的生产任务 `productType pdtType;` 和该任务有待完成的工作余量 `TProductPoint productConsume;`
- 3) 等级生命力上限 HP_N 仅与等级正相关, 具体的数据如表1所示。实际生命力 hp 会

受到进攻而减小（具体结算方式如式2和式5所示）、由于建设者的维护而增加。当实际生命力 hp 被进攻方降低至 0 以下时，对方获得 5 分的击杀分，我方丧失该塔，塔等级下降 4 级；等级下降后，若不足 1 级，则塔消失，该单位恢复原来地形；若塔尚存在，则降级后的塔主权归对方所有（塔被俘虏）。

4) 实际战斗力 f 为战斗中的战斗力。 f 是 F_N 、当前生命值 hp 、等级生命值上限 HP_N 、兵团驻扎情况带来战斗力增益 f_c （具体增益规则如表3所示）的函数。具体计算方法如下：

$$f = F_N \cdot \frac{hp}{HP_N} + \Sigma f_b \quad (1)$$

兵团攻击塔时，生命值的结算方式如下：

$$\begin{aligned} hp_{\text{塔}} - &= 30 \cdot k_c \cdot e^{0.04(f_{\text{兵}} - f_{\text{塔}})}; \\ hp_{\text{兵团}} - &= 28 \cdot e^{0.04(f_{\text{塔}} - f_{\text{兵}})}, \text{当兵团非弓箭手} \\ hp_{\text{兵团}} - &= 0, \text{当兵团为弓箭手} \end{aligned} \quad (2)$$

塔只能攻击距离为 2 及以内的兵团。塔攻击兵团时，生命值的结算方式如下：

$$\begin{aligned} hp_{\text{兵团}} - &= 30 \cdot e^{0.04(f_{\text{塔}} - f_{\text{兵}})}, \text{对于所有兵团种类} \\ hp_{\text{塔}} - &= 0 \end{aligned} \quad (3)$$

你可以为塔添加以下操作：

(1) 生产：在命令中指定命令类型为塔命令，塔操作类型为生产，塔 ID，塔生产任务类型（<enum productType>）。

(2) 攻击：在命令中指定命令类型为塔类型，塔操作类型为攻击，塔 ID，塔的攻击对象序号，即可指定某一做塔攻击某一个对象。

另外，请注意：

(1) 每回合，每座塔最多可以添加一个操作。如果该回合玩家给一个塔添加了多个操作，则第二个及以后的操作会被自动忽略，且占用 50 个操作的余额（相当于填了废志愿），所以请不要添加大于一个操作。

(2) 如果你的操作是无效操作，则也不会起任何作用。无效操作包括试图攻击自己的对象、试图攻击超出攻击范围的对象、不存在对应序号的对象等。

(3) 如果上一回合塔选择了生产，但未完成该生产任务，本回合选择攻击，则上一回合的生产进度会被保留，这样在下一回合假如继续添加生产该任务的命令，会在上一回合基础上继续生产。

(4) 若防御塔的某个生产任务尚未完成，玩家又指定了新任务，则未完成的生产任务的完成度将被缓存起来，然后进行新的任务。之后再选择有一定完成度的任务的时候，只需要完成未完成的部分，即完成剩余的生产力消耗值。即如果上一回合塔选择了生产任务 A，但未完成该生产任务 A，本回合选择生产任务 B，则上一回合的生产进度也会被保留，这样在下一回合假如继续添加生产任务 A 的命令，会在上一回合基础上继续生产。但是，由于接口所限，你只能通过 TowerInfo 访问到最近一次尚未完成的任务类型和余量，所以我们推荐一旦开始一种生产任务，中途不要切换别的生产任务（虽然你仍可以正常地生产它们）。

3.2 作战兵团

在同一个时间，同一个地图方格（防御塔所在方格除外）内的作战兵团、工程兵团数量各自不能超过 1 个。当某个方格内同时存在一个势力的一个作战兵团和工程兵团，则称工程兵团被作战兵团护卫。任何战斗，都优先与作战兵团结算。若在某次战斗中，该护卫队中的作战兵团阵亡。此时进行一次判定，若敌方作战兵团也在作战中阵亡，则护卫队中的工程兵团仍属于原来的玩家；若敌方作战兵团未阵亡，则护卫队中的工程兵团将被敌方俘虏，所属玩家变更为敌方。

3.2.1 概述

兵团有三个种类：战士（近战单位）、弓箭手（远程单位）、法师（高级进攻单位）。其中，弓箭手可以远程轰炸其他单位且自身不受伤害。法师具有较高的移动力。

作战兵团有以下属性：

- (1) 行动力 M_c 。
- (2) 满血战斗力 F_c ，实际战斗力 f 。
- (3) 生命力上限 HP_c ，当前生命力 hp 。
- (4) 攻击距离 d_c 。
- (5) 所属玩家 ID。

```
22 struct CorpsInfo
23 {
24     //不需要，如果不存在就不录入信息了 bool exist;    //是否存在
25     TPoint pos;      //兵团坐标
26     int level;       //兵团等级
27     TCorpsID ID;     //兵团ID
28     THealthPoint HealthPoint; //生命值
29     TBuildPoint BuildPoint;   //劳动力
```

```

30  TPlayerID    owner;          //所属玩家ID
31  corpsType    type;           //兵团种类
32  TMovePoint   movePoint;      //行动力
33  battleCorpsType m_BattleType; //战斗兵用
34  constructCorpsType m_BuildType; //建造兵用
35  };
36

```

防御塔结构体

3.2.2 具体说明

1) 行动力 M_c 表示兵团在某一回合的进行行动的能力，具体见表3。兵团从一个单元格移动到另一个相邻单元格（即上下左右四个方向）将消耗一定行动力。（取决于单元格地形情况，计算方式为：一次移动经过的两个单元格的行动力消耗取平均值，并向上取整，具体见表4）值得注意的是，如果移动后行动力至少还有 1 点则还能发起进攻，而一旦选择进攻将消耗所有行动力。行动力在新回合开始将重置。

2) 生命力上限 HP_c 仅与兵团种类有关，具体见表3。实际生命力 hp 会受到进攻而减小（具体结算方式如式5所示），直至实际生命力 hp 被进攻方降低至 0 以下时，兵团死亡，对方获得 5 分的击杀分。

3) 实际战斗力 f 为战斗中的战斗力。 f 是满血战斗力 F_c 、当前生命值 hp 、生命值上限 HP_c 、兵团所处地形情况带来战斗力增益 f_t （具体增益规则如表4所示）的函数。具体计算方法如下：

$$f = F_c \cdot \frac{hp}{HP_c} + f_t \quad (4)$$

兵团只能攻击在攻击范围内的对象，不同兵团的攻击距离 d_c 如表3所示。兵团 B 受到兵团 A 攻击时，生命值的结算方式如下：

$$\begin{aligned}
 hp_B - &= 30 \cdot e^{0.04(f_A - f_B)}; \\
 hp_A - &= 28 \cdot e^{0.04(f_B - f_A)}, \text{ 当兵团 A 非弓箭手} \\
 hp_A - &= 0, \text{ 当兵团 A 为弓箭手}
 \end{aligned} \quad (5)$$

作战兵团能进行的操作有：在地图上移动、停在某一方格内驻扎、驻扎己方势力的塔、对敌方军团发起进攻、对敌方防御塔发起进攻。去掉兵团整编、兵团原地驻扎的命令（兵团驻扎到塔的指令还保留着）@TODO

3.3 工程兵团

工程兵团分为：建造者、开拓者。建造者用来进行特定的工程建造，而开拓者则用于修建新的防御塔。

工程兵团有以下属性：

- (1) 行动力 M_c 。
- (2) 劳动力 B 。
- (3) 所属玩家 ID。

表 1: 塔等级表

等级 N	生产力 W_N	等级战斗力 F_N	等级生命力上限 HP_N
1	10	25	100
2	15	27	120
3	20	29	140
4	25	32	160
5	30	35	180
6	35	38	200
7	40	41	220
8	45	45	240

表 2: 塔生产任务表

生产任务	所需的生产力值
战士	40
弓箭手	60
法师	100
建造者	40
开拓者	40
升级 (N 升级到 N+1)	$N*40$

表 3: 兵团参数表

兵种 Crops	战士	弓箭手	法师
战斗力增益系数 f_c	2	2	4
攻城系数 k_c	0.4	0.7	0.5
攻击距离 d_c	1	2	1
行动力 M_c	2	2	4
生命力上限 HP_c	60	50	70
满血战斗力 F_c	36	30	44

表 4: 地形参数表

地形	平原	山地	森林	沼泽
地形战斗力增益 f_t	0	5	3	-3
地形行动力消耗 m_t	2	4	3	4