

先输入红方



双方起点：

红：第10列第5行（10，5），方向朝左(rotation = 3)

蓝：第1列第5行（1，5），方向朝右(rotation = 1)

红方先走

坦克的方向用以下数字表示

0: Up

1: Right

2: Down

3: Left

可以使用的方法：

calculateCoordinates(distance=1,direction=None,position=None)

#得到从位置position开始，沿direction方向，距离distance的新位置

#如果省略position,则默认为坦克当前位置

#如果省略direction，则默认为坦克当前方向

#如果省略distance，则默认距离为1

lookInFront():

#返回一个string，代表前面位置的属性，返回值可以是

#me:自己的位置

#bot:对手机器人的位置

#wall:四周的墙

#clear:空的

#HP:加血

#ATK:加攻击力

#SPD:加速器

#ex: obj = self.robot.lookInFront()

lookAtSpace(space):

#返回一个string，查看某一位置的属性，参数space是1个代表位置的长度为2的list

#返回值可以是

#me:自己的位置

#bot:对手机器人的位置

#wall:四周的墙

#clear:空的

#HP:加血

#ATK:加攻击力

#SPD:加速器

#可以通过2层for循环获取整个战场信息，但是请你注意，使用这个方法不能获取对方的血量/攻击力/加速器属性信息；能够得知战场上的物品种类和位置，但不能获取物品的具体数据（例如能知道[x,y]位置上有血包，但不知道+多少血）

#ex: obj = self.robot.lookAtSpace((1,2))

attack():

#攻击，若攻击成功，对手减血量 = 你当前攻击力

#ex: self.robot.attack()

goBack()

#后退一格，方向不变

#ex: self.robot.goBack()

goForth()

#前进一格，方向不变

#ex: self.robot.goForth()

goBack2()

#拥有加速器时可以使用，后退2格（中间不能有障碍），方向不变

goForth2()

#拥有加速器时可以使用，前进2格（中间不能有障碍），方向不变

turnLeft():

#左转弯

#ex: self.robot.turnLeft()

doNothing()

#停一回合，不动也不攻击

#ex: self.robot.doNothing()

turnRight():

#右转弯

locateEnemy():

#得到敌人的位置和方向

#返回一个元组（位置，方向）

#ex: (p, r) = self.robot.locateEnemy()

detectEnemy():

#探测身边的4个格子，若有敌人，得到敌人的位置和方向

#返回一个元组（位置，方向，血量，攻击力，是否有加速器（0/1））

#ex: (p, r, HP, ATK, SPD) = self.robot.detectEnemy()

每个回合，战场上以2%的概率随机出现以下物品之一（位置随机）：

1. 加血包，加血量为randint(0,50)

2. 加攻击力，攻击力提升为randint(0,5)

3. 加速器，得到加速器之后获得goForth2()/goBack2()方法，可以每回合跳2格

\*上述能力加成都是永久性的，并且不设血量上限

游戏开始时，会生成随机地图，地图中随机放置障碍/血包/攻击力/加速4种物品

其他注意事项：

1. 在Run setting中选择“Execute in an external System terminal”

2. 对战时请尽量使用windows电脑，开始对战后不要切换窗口

3. 若500回合之后双方都存活，则首先比较血量（血量多的一方直接获胜），若血量一样再数格子数量判定输赢，此时1个金色格子 = 15个普通格子，金色格子在固定位置=（5，5）

4. 在最后一回合中，如果先手方把后手方打死，后手方还有1次还手的机会，即游戏以“红蓝”2次行动为判定胜负的一个时间单位。

5. 每回合中，以下改变战场状态（位置/方向/攻击）的方法会造成回合转换，也就是每回合中只能执行其中一个方法（如果在一回合中某方使用了1个以上方法，则直接判负）

doNothing()

goForth()

goBack()

goForth2()

goBack2()

attack()

turnRight()

turnRight()

而对于calculateCoordinates()/lookInFront()/lookAtSpace()/locateEnemy()/detectEnemy()这几个方法，在回合中你可以不限次数地任意使用。

6. 如果在一回合中某方程序死循环，则直接判负

7. 请注意你的攻击范围只有正前方一格

8. 如果没有加速器时使用goForth2()/goBack2()，或者面前没有敌人时使用attack()，或者走出边界/走到障碍上，则浪费一个回合，相当于doNothing()