|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 操作 | 命令类型  commandType | 参数列表  parameters | | | | | |
| 参数0 | 参数1 | 参数2 | 参数3 | 参数4 | 参数5 |
| 塔的生产任务 | towerCommand  =1 | 塔操作类型  TProduct = 0 | 塔的序号  <int>  与Info读到的一致 | 塔生产任务类型  <enum productType>  = 0-5 | 空 | 空 | 空 |
| 塔的攻击任务 | 塔操作类型  TAttackCorps = 1 | 塔攻击对象序号  <int>  与Info读到的一致 | 空 | 空 | 空 |
| 兵团移动 | corpsCommand  =0 | 兵团操作类型  CMove = 0 | 兵团的序号  <int>  与Info读到的一致 | 移动方向  <enum corpsMoveDir>  = 0-3 | 空 | 空 | 空 |
| 兵团驻扎（非塔）  这一指令去掉了 | 空 | 空 | 空 | 空 | 空 |
| 兵团驻扎（在塔）  这一指令去掉了 | 兵团操作类型  CStationTower = 2 | 原地命令  去掉塔的序号 | 空 | 空 | 空 |
| 兵团攻击 | 兵团操作类型  CAttackCorps = 3/  CAttackTower = 4 | 兵团攻击对象序号  <int>  与Info读到的一致 | 空 | 空 | 空 |
| 兵团整编  这一指令去掉了 | 空 | 空 | 空 | 空 | 空 |
| 修建新塔 | 兵团操作类型  CBuild = 7 | 空 | 空 | 空 | 空 |
| 修理原来的塔 | 兵团操作类型  CRepair = 8 | 原地命令  去掉塔的序号 | 空 | 空 | 空 |
| 改变方格地形 | 兵团操作类型  CChangeTerrain = 9 | 目标地形类型  <enum terrainType>  = 1-5  （只有平原——森林的互相改变是有效的） | 空 | 空 | 空 |
|  |  |  |  |  |  |  |  |

备注：上表中为参数分配了编号，如兵团攻击兵团操作值为3，而你在编写ai时，可以使用相应的枚举值来代替具体的数字，你可以在definition.h中找到这些枚举值，以及它们具体的含义。

玩家需要进行的操作

Info是一个结构体，给玩家player\_ai的函数传入的参数为info，它是当前游戏Info的引用。这个info包含了玩家可以访问的场上地图信息、防御塔信息、兵团信息和其他玩家信息。此外，Info中还带有一个命令表CommandList，玩家在自己的ai代码中通过

info.myCommandList.addCommand(<命令类型：commandType>, {<参数列表：parameters>});

这个函数向CommandList添加命令，命令参数解释如下表所示。

注意：游戏运行时，只会执行玩家有效的命令，无效的命令将会被丢弃。Info只是游戏中信息的副本，通过修改Info无权修改游戏实际运行的信息。