**选手接口文档**

1.选手接口

void updateAge() 升级时代

void construct(BuildingType building\_type, Position pos, Position soldier\_pos) 建造命令 参数一次是建造的建筑类型，建造的位置和出兵位置(非造兵建筑空缺这个参数)

void upgrade(int unit\_id) 升级单位命令，参数为目标单位id

void sell(int unit\_id) 出售命令，参数为单位id

void toggleMaintain(int unit\_id) 修理命令，参数为目标单位id

2.游戏静态信息

选手通过bool变量flag获取自己是0/1号玩家

选手通过变量bool\*\* map来获取地图信息，其中map是一个200\*200的bool数组。

3.获取游戏的全局动态信息

struct State

{

int turn;

int winner;

\_resource resource[2];

Age age[2];

vector<Building> building[2];

vector<Solider> soldier[2];

};

游戏信息储存在这样一个结构体中，玩家可以通过指针state获取这些信息。

turn是当前游戏回合数目，winner为当前胜利者，其他为选手的资源和建筑，兵种

其中resource age building soldier这几个数组分别储存了两个选手的信息，其中[0]是player0的信息，[1]是player1的信息

其中building[i] solider[i]储存的是装有结构体Building，Soldier实例的向量（STL的vector）。

4.对应资源信息结构体的实例

struct \_resource

{

int building\_point;

int resource;

};

struct Position {

int x;

int y;

};

struct Soldier {

SoldierName soldier\_name; //soldier的类型

int heal; //soldier的血量

Position pos; //soldier的位置

int flag; //soldier的阵营

int unit\_id; //soldier的ID

};

struct Building {

BuildingType building\_type; //building的类型

int heal; //building的血量

Position pos; //building的位置

int flag; //building的阵营

int unit\_id; //building的id

bool maintain; //building是否被维修

};

注：1.选手是指在player.cpp文件的f\_player()函数中编写自己的代码，请不要在其他文件中写代码。

2.对战方式：

首先打开server.exe程序

运行编译链接好的C++文件生成的.exe文件

运行一次代表连接一个AI，需要且仅需要运行两次。