Astar2010 百度之星程序设计大赛

乐CODE&乐CODE

坦克大战（TankCraft）AI编写快速入门

游戏描述

一个回合制战略游戏——“坦克大战”(TankCraft)。游戏规则参见“Astar2010 坦克大战详细规则.pdf”。

游戏流程

每回合选手能够控制的每一辆坦克，选手程序都会收到平台发送的一个结构体，其中包含了当前赛场上的全部信息。选手的程序根据这些信息进行运算（每次计算限时50 ms，最终比赛时以服务器计时为准），返回一个行动的指令。待平台收集到所有坦克的行动指令后，会更新所有赛场元素，进入下一回合。如此往复，直至游戏结束。下图为某地图的2D显示效果：

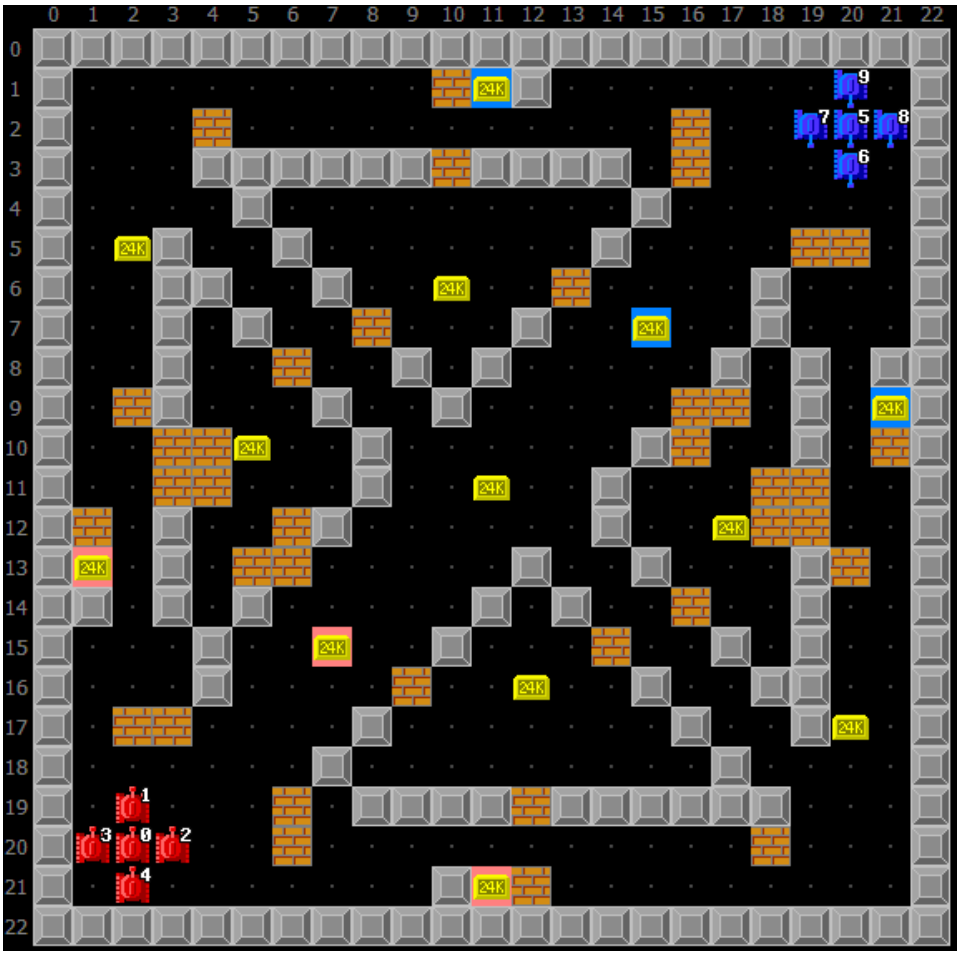

图 1

选手AI工程的结构

主要包括三个文件：Tank.h 和 myAI.cpp

1. **Tank.h**: 由平台组提供给选手的 AI 头文件，主要包括以下内容：

① 常量宏定义，例如：

```
#define MAP_WIDTH          21   //赛场的宽度
#define  MAP_HEIGHT        21   //赛场的高度
//……
```

② 自定义枚举类型，例如：

```
typedef enum {PERVIOUS,    BRICK, BREAKBRICK, STONE} CellType; //地面的类型
typedef enum {STOP,GOUP,GODOWN,GOLEFT,GORIGHT,FIRE} OrderType;//指令的类型
typedef enum {Striker, Pioneer, Sniper} TankType;//坦克的类型
//……
```

③ 定义结构体，存放游戏信息，例如：

```
struct DataForAI                                    //装载传递给坦克 AI 的所有信息
{
      MapCell map[MAP_HEIGHT+2][MAP_WIDTH +2];
      //当前的赛场地图，从上到下，从左到右，行优先
      //比赛场有效区域头尾各多一行，左右各多一列，便于编写算法
      TankData tank[MAP_TANK_NUM];              //所有参赛坦克的信息
      Point source[MAP_SOURCE_NUM];                  //矿点位置信息
      FlagType myFlag;                          //自己属于哪一方
      short myID;                               //自己的 ID 是多少
      short round;                              //当前回合数
      short redGoldNum,blueGoldNum;                 //当前双方的金子数
      short redScore,blueScore;                     //当前双方比分
      short redSource,blueSource,totalSource;       //当前双方占有矿点数 当前总矿点数
      int redTime,blueTime;                         //当前双方 AI 总用时数
};
//……
```

本游戏定义的各种类型、结构体的具体含义在 Tank.h 中都有详尽的注释。

2. **myAI.cpp**: 包括两个函数，chooseType() 和 makeOrder(DataForAI data)。

平台 0 回合时调用 chooseType 函数获取 AI 名称及坦克类型信息，0 回合之后该函数不再被调用；

从第一回合开始，调用 makeOrder 函数，获得选手返回的指令。makeOrder 函数的形参 data 是由平台提供给选手的当前回合游戏信息。选手的任务是：

①.改写 chooseType 函数，在第一回合选择每辆坦克的类型(Striker/Pioneer/Sniper)，为自己的 AI 命名。

②编写 makeOrder 函数，由当前回合游戏信息得出下一步指令（左走、开火等等），

返回至平台。

下面是平台组为选手提供的 myAI.cpp 文件：

```
#include "Tank.h"      //引用游戏头文件 Tank.h
#include <string.h>

Extern “C”
InitiateInfo chooseType()
//平台回合时调用此函数获取 AI 名称及坦克类型信息
//0 回合之后该函数不再被调用
{
    InitiateInfo Info;        //定义一个 InitiateInfo 类型变量 Info，存放函数返回的信息
    Info.tank[0]=Striker; //0 号坦克选择 Striker
    Info.tank[1]=Striker; //1 号坦克选择 Striker
    Info.tank[2]=Striker;        //2 号坦克选择 Striker
    Info.tank[3]=Striker;        //3 号坦克选择 Striker
    Info.tank[4]=Striker;        //4 号坦克选择 Striker
    strcpy(Info.aiName,"myAI"); //设置 AI 名为“myAI”
    return Info;
}

Extern “C”
Order makeOrder(DataForAI data)
//平台从第 1 回合开始调用此函数获得每回合指令
{
    Order order;        //定义一个 Order 类型的变量 order
    order.type=STOP;//对 order 的成员 type 赋值（参考头文件中 struct Order 和 OrderType 的定义
    return order;        //返回 order
}
```

3. **main.cpp**: 由平台组提供的选手 AI 本地编译所需的主函数文件。一般情况下选手无需知晓其代码涵义；**切勿修改**，否则后果自负。

如何让平台调用我的程序

1. 生成 AI 程序文件

在 myAI.cpp 中写好程序之后（注意：请勿自行往工程中添加自己的其他代码文件；请勿修改 Tank.h 与 main.cpp，否则造成的后果自负），编译生成即可。本次大赛最终评测使用的编译器为 GCC。其余编译器也可生成本地 AI 程序文件且可本地调用，但可能与大赛评测使用系统不兼容，届时后果自负。推荐在压缩包中自带的样例工程基础上修改（使用 DevCPP 为 IDE）。

2. 调用 AI

① 打开游戏平台（图 3）。

图 3

② 点击“单机游戏”（图 4）。

图 4

③ 在对话框中选择红蓝两方 AI 路径并点击启用（应能弹出供 AI 输出调试信息的命令行窗口，如图 5）；选择合适的地图，然后选择游戏模式即可开始。“比赛模式”为带有 3D 显示的模式，供选手慢慢欣赏；“调试模式”则不调用 3D 模块，以最快速度完成比赛。选手可通过观看录像回放得知比赛的详细情况。

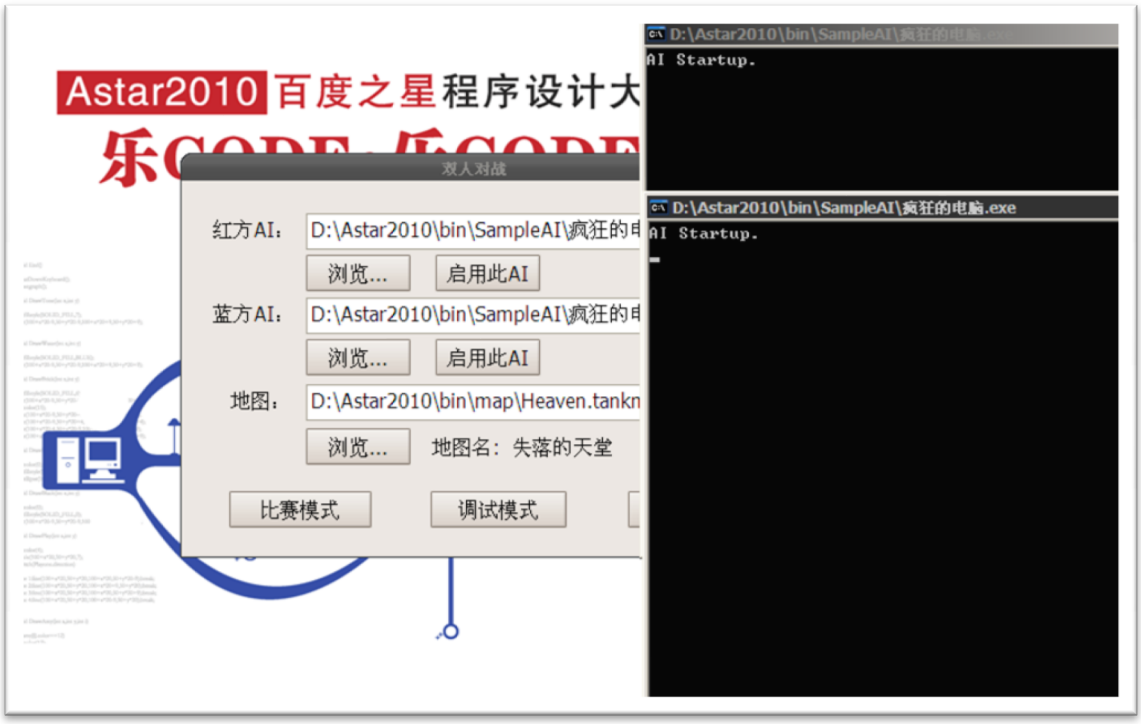

图 5

如何利用Tank.h获得信息

希望能够通过以下几个实例，让选手熟悉如何使用 Tank.h 头文件。

实例 1 判断整形变量 x 是否超过地图宽度

```
if ( x > MAP_WIDTH)
{
   //……
}
```

分析:

头文件中有这么一个宏定义：

```
#define MAP_WIDTH          21
```

实例 2 判断当前回合地图坐标(5, 8)的点的地图属性是否为砖墙

```
if (data.map[5][8].type == BRICK)
{
      //……
}
```

分析:

在 myAI.cpp 中，makeOrder 函数的参数中定义了一个结构体 DataForAI 类型的变量 data

头文件中已定义 DataForAI 结构体如下。

```
struct DataForAI
```

```
{
    MapCell map[MAP_HEIGHT+2][MAP_WIDTH +2];
    //……
}
```

其中嵌套了另一个结构体 MapCell，其定义如下：

```
struct MapCell
{
    CellType  type;                //当前格的地面属性
    //……
};
```

其中的 CellType 是自定义的枚举类型，其定义如下：

```
typedef enum {PERVIOUS,   BRICK, BREAKBRICK, STONE} CellType;
```

实例 3 判断当前回合地图坐标(8, 4)的点上是否有坦克；若有，判断是否是敌方坦克

```
    if (data.map[8][4].whoIsHere != -1)
    {
        int n = data.map[8][4].whoIsHere;
        if (data.tank[n].flag != data.myFlag)
        {     //……
        }
    }
```

请选手查阅 Tank.h，自行分析这段代码。

通过以上三个实例，相信你已经对编写 AI 的形式有了一定的了解，现在就动手编写自己的 AI 吧！