五子棋报告

一、 实验要求：

完成一个为 15*15 的五子棋棋盘的五子棋 AI，可以使电脑自由与玩家进行攻防战。

二、 实验想法：

1、我们可以将棋盘看作是一个二维数组，数组大小为 15*15.在棋盘的数组上，用数值的大小来表示棋盘的危急程度。那么相对一边有利对另一边就越危险。

2、整个棋盘上，还用到一个叫做“获胜组合”的结构，即每并排的五个子组成一个可能获胜组合。

3、将所有的下棋角色封装为一个类，方便在 AI 类中使用。

三、 实验步骤：

1、一个位置用一个结构来表示：

```
struct Seat
{
    int x;//x 坐标
    int y;//y 坐标
    int Urgent;
};
```

这个结构包含棋盘上的坐标位置和这个位置的危急程度。

2、

```
struct WinUnion
{
    WinUnion(){}
    int  Type;//      标识组合方向--纵向：1；横向：2；反正斜：3；正斜：4；
    bool HasDanger;//该组合数是否有效
    Seat seat[5];//五个组成一组
};//所有获胜组合数
```

在这个结构中 Type 表示组合的方向，由于有四个可能方向它们是：横向；纵向；左上到右下；左下到右上；分别用 1、2、3、4 来表示。这样只要下了子的位置，这个位置的危急程度就为 0，因为没有人可以通过往这个地方下子可以取胜。

3、封装为一个类：

```
class PlayerStr
{
public:
    PlayerStr(){}

public:
    //标识是己方还是对方－1：对方；1：己方
    int flag;
    //标识是否手禁手约束
```

```
        bool Inhibiting;
        //桌面布子情况：－1：对方；1：己方
        int Table[25][25];
        //棋盘危急程度
        int Urgent[25][25];
        //每个位置指向所有相关联的可能获胜组合,动态创建可能获胜组合，并使相关联五个 AllUnion 指向它
        UnionNode   AllUnion[25][25];

};
```

其中 inhibiting 表示是否禁手约束，先下子者为禁手，如果 inhibiting 为 true 则使用禁手规则，否则不使用。Table 数组表示桌面部子的情况，0 为没有子的情况，-1 为对方的棋子，1 为己方的棋子。Urgent 数组为整个桌面的危急程度，这个危急程度跟 Seat 坐标的危急程度有分别，Seat 坐标结构的危急程度指的是与这个坐标所组成的所有可能获胜组合对这个位置造成的危急程度，与这个位置组成的可能获胜组合中，越多则对方的子在同一条直线上危急程度就越大。而 Urgent 数组则表示这个位置可能获胜组合越多就越危急。即 Urgent 数组是某个位置上所有可能获胜组合的可能获胜组合造成的危急总和。

//AllUnion 存储所有位置上的所有可能获胜组合链表组，从一个坐标查找链表可以找到所有与这个坐标组成的可能获胜组合。我们定义链表节点为：

```
struct UnionNode{
        WinUnion *UnionData;//指向一个组合
        UnionNode *Next;//下一个可能获胜组合
};
```

AllUnion 存储了所有链表的链表头。

一个位置上的 AllUnion[j+k][i]包含所有和这个位置相关的可能获胜组合 WinUnion。