

五子棋AI算法——实验报告

第三周

姓名：王广建 学号：09016417 班级：计科四班

1. 对战情况：暂未对战

2. 算法描述：

- (1) 采用网站提供的JSON交互方式
- (2) 此版本只返回下一步棋子位置，不能进行未来情况预测
- (3) 计算落子位子的步骤：

将所有以往下子的位置用Grid数组和pieces数组储存，Grid不分别棋子，pieces的数据类型为ADD（req表示对手，resp表示己方，none表示空），记录黑白子情况

将现有棋子的周围八格中无子的位在Neighbor数组中记录下来，并将所有可以落子位置储存在vector xs和ys中，然后依次对其进行估值，选取最高的位置放置，估值方法如下：

我们关注的是某一个位置的横排，竖排和左斜右斜四种情况，每种情况检查两边各五格，从第二个相邻位置开始，若和上一个位置相同，相应的值加二，

长度加一，直到遇到不同的棋子或已经检查五格，如实空位value不变，长度加一，若遇到的是不同的棋子，则value减一，是空位则不减，若连续的是对手的棋子，则分数减二，然后对比两边，若两边第一位相同，则value相加，不同选取最大值，同时判断长度相加是否大于四或是空，是则value为0.

最后选择是个方向最大值。同时注意如果遇到边界，停止估值。

代码如下（一竖排为例）：

//竖排情况////////////////////////////////////

```
int i = 0;
//竖排上部五格
for (i = 1; i <= 5; i++)
{
    if (y + i > 14)
        break;
    if (i = 1)
    {
        rightValue = 2;
        rightLen = 1;
    }
    else
    {
        if (pieces[x][y + i] == pieces[x][y + i-1])
```

```

        {
            rightValue += 2;
            rightLen += 1;
        }
    else
    {
        if (pieces[x][y + i] != none)
        {
            rightValue -= 1;
        }
        else
        {
            rightLen++;
        }
    }
}

//竖排下部五格
for (i = 1; i <= 5; i++)
{
    if (y - i < 0)
        break;
    if (i = 1)
    {
        leftValue = 2;
        leftLen = 1;
    }
    else
    {
        if (pieces[x][y - i] == pieces[x][y - i + 1])
        {
            leftValue += 2;
            leftLen += 1;
        }
        else
        {
            if (pieces[x][y - i] != none)
            {
                leftValue -= 1;
            }
            else
            {
                leftLen++;
            }
        }
    }
}

```

```

        }
    }
}

//为对方棋子的部分估值减2
//如果是空则估值为零
if (y + 1 < 15)
{
    if (pieces[x][y + 1] == req)
    {
        rightValue -= 2;
    }
    if (pieces[x][y + 1] == none)
    {
        rightValue = 0;
    }
}
else
    rightValue = 0;

if (y - 1 >= 0)
{
    if (pieces[x][y - 1] == req)
    {
        leftValue -= 2;
    }
    if (pieces[x][y - 1] == none)
    {
        leftValue == 0;
    }
}
else
    leftValue = 0;
//如果两边棋子相同
if (y + 1 < 15 && y - 1 >= 0)
{
    if (pieces[x][y + 1] == pieces[x][y - 1])
    {

        if (rightLen + leftLen >= 4)
        {
            revalue = rightValue + leftValue;
        }
        else

```

```

        revalue = 0;
    }
    else//两边不同
    {
        if (leftValue > rightValue&&leftLen>=4)
        {
            revalue = leftValue;
        }
        if (leftValue <= rightValue && rightValue >= 4)
        {
            revalue = rightValue;
        }
    }
}
else if (y + 1 >= 15)
{
    if (leftLen >= 4)
    {
        revalue = leftValue;
    }
}
else
{
    if (rightValue >= 4)
    {
        revalue = rightValue;
    }
}

//value与revalue比较
if (revalue > value)
{
    value = revalue;
}
revalue = 0;

```