

# 高级程序设计

## 实验报告

南开大学 计算机大类  
2311807 问丕丕 人工智能 1 班  
2024 年 5 月 13 日

## 目录

高级语言程序设计大作业实验报告.....	3
一、作业题目.....	3
二、开发软件.....	3
三、课题要求.....	3
四、主要流程.....	3
1、整体流程	
2、随机玩家颜色	
3、绘制棋盘	
4、人机算法	
五、收获.....	7

# 高级语言程序设计大作业实验报告

## 一、 作业题目

五子棋人机对战

## 二、 开发软件

Microsoft Visual Studio 2022

## 三、 课题要求

- (1) 实现棋盘绘制
- (2) 实现电脑下棋算法
- (3) 实现胜负判断

## 四、 主要流程

### 1、 整体流程：

开始游戏 → 随机玩家颜色 → 开始下棋 → 判断输赢 → 继续或者关闭

### 2、 随机玩家颜色：

```
srand(time(0));  
PlayerColor = rand() % 2; // 玩家颜色
```

### 3、 绘制棋盘：

根据位置不同分为 9 种不同格线，再分别绘制棋盘

### 4、 人机算法：

遍历每一个可以下棋的位置，根据附近已有的棋子来计算假设下在此位置所得的最高分，然后进行排序，最后在分数最高且一致的位置随机选取一个位置下棋

```
seat FindBestSeat(int color)  
{  
    int Score[3][5] = {
```

```

        { 0, 20, 360, 5800, 92840 },
        { 0, 0, 20, 360, 92840 },
        { 0, 0, 0, 0, 92840 }
    };
    seat bestseat;
    int Max_num[361] = { 0 };
    int Max_x[361] = { 0 };
    int Max_y[361] = { 0 };
    int num = 0;
    int res;           // 输出的最佳分数位置
    int nowi = 0;
    int nowj = 0;
    int len[4];        // 四个方向的长度
    int enemy[4];      // 四个方向的敌子
    for (int i = 0; i < 19; i++)
    {
        for (int j = 0; j < 19; j++)
        {
            if (box[i][j].value == -1)
            {
                box[i][j].value = color; // 尝试下在这里
                for (int k = 0; k < 4; k++)
                {
                    len[k] = 0;
                    enemy[k] = 0;
                    nowi = i;
                    nowj = j;
                    while (nowi <= 18 && nowj <= 18 && nowi >= 0 &&
nowj >= 0 && box[nowi][nowj].value == color)
                    {
                        len[k]++;
                        nowj += dx[k];
                        nowi += dy[k];
                    }
                    if (box[nowi][nowj].value == 1 - color || nowi < 0 ||
nowj < 0 || nowi > 18 || nowj > 18)
                    {
                        enemy[k]++;
                    }
                    nowi = i;
                    nowj = j;
                }
            }
        }
    }

```

```

        while (nowi <= 18 && nowj <= 18 && nowi >= 0 &&
nowj >= 0 && box[nowi][nowj].value == color)
        {
            len[k]++;
            nowj -= dx[k];
            nowi -= dy[k];
        }
        if (box[nowi][nowj].value == 1 - color || nowi < 0 ||
nowj < 0 || nowi > 18 || nowj > 18)
        {
            enemy[k]++;
        }
        len[k] -= 2;
        if (len[k] > 4)
        {
            len[k] = 4;
        }
        box[i][j].num += Score[enemy[k]][len[k]] * 4
+ (!len[k]) * 20; //加分
        len[k] = 0;
        enemy[k] = 0;
    }
    // 敌人
    box[i][j].value = 1 - color;
    for (int k = 0; k < 4; k++)
    {
        len[k] = 0;
        enemy[k] = 0;
        nowi = i;
        nowj = j;
        while (nowi <= 18 && nowj <= 18 && nowi >= 0 &&
nowj >= 0 && box[nowi][nowj].value == 1 - color)
        {
            len[k]++;
            nowj += dx[k];
            nowi += dy[k];
        }
        if (box[nowi][nowj].value == color || nowi < 0 ||
nowj < 0 || nowi > 18 || nowj > 18)
        {
            enemy[k]++;
        }
    }

```

```

        nowi = i;
        nowj = j;
        while (nowi <= 18 && nowj <= 18 && nowi >= 0 &&
nowj >= 0 && box[nowi][nowj].value == 1 - color)
        {
            len[k]++;
            nowj -= dx[k];
            nowi -= dy[k];
        }
        if (box[nowi][nowj].value == color || nowi < 0 ||
nowj < 0 || nowi > 18 || nowj > 18)
        {
            enemy[k]++;
        }
        len[k] -= 2;
        if (len[k] > 4)
        {
            len[k] = 4;
        }
        box[i][j].num += Score[enemy[k]][len[k]];
        len[k] = 0;
        enemy[k] = 0;
    }
    box[i][j].value = -1;
}
if (box[i][j].num == Max_num[0])
{
    Max_num[num] = box[i][j].num;
    Max_y[num] = i;
    Max_x[num] = j;
    num++;
}
if (box[i][j].num > Max_num[0])
{
    for (int k = 0; k < num; k++)
    {
        Max_num[k] = 0;
        Max_y[k] = 0;
        Max_x[k] = 0;
    }
    num = 0;
    Max_num[num] = box[i][j].num;

```

```

        Max_y[num] = i;
        Max_x[num] = j;
        num++;
    }
}
srand(time(NULL));
res = rand() % num;
bestseat.i = Max_y[res];
bestseat.j = Max_x[res];
bestseat.num = Max_num[res];

return bestseat;
}

```

## 五、收获

学习到了 Easyx 图形库的基本使用方法，和利用贝叶斯定理来实现透明贴图的插入