C语言/笔记/项目/

```
/*
========ww==================================ww==============
●本程序由奥利奥制作
    QQ——2783608988
●版本：
井字棋 （Tic Tac Toe）【终极版】
●使用说明：
    【人机对战】
[W][S][A][D]控制玩家方向
[回车键]确定 [Esc]返回
    【双人对战】
玩家1:[W][S][A][D]
[空格]确定
玩家2:[↑][↓][←][→]
[回车键]确定

[Esc]返回
                        时间：2016.10.22
------------------------------------------------------------
漏洞

    1.玩家先手时 电脑的最后一个棋子不会出现a
    2.返回主菜单时 重复选项                          【已修复】
    3.提示语（白子，黑子）难理解                    【已修复】
    4.弹出两个重复的信息框

新增：
    1.方向键也可以控制主菜单                        【已添加】
    2.双人对战 【确定键】增加空格                    【已添加】
======ww==================================ww==============
*/

/*        头文件        */
# include <time.h>
# include <stdio.h>
# include <stdlib.h>
# include <windows.h>

/*        全局变量    */
int chess[3][3]={{0,0,0},{0,0,0},{0,0,0}},step = 0,frist_play,number_time = 1;    //井字棋地图,用户步数,谁先手,draw_map函数运行次数

/*        声明函数    */
```

```
void main_mean(int show);                                   //显示界面
void gotoxy(int x,int y);                                   //光标到指定位置
void a_word_printf(char *a, int time);                       //渐显文字
void computer_play(void);                                   //人机对战
void people_play(void);                                      //双人对战
int check_result(void);                                      //【人机对战，双人对战】检测结果
void draw_map(int model);                                   //【人机对战，双人对战】画井字棋
void computer_play_chess(void);                              //【人机对战】电脑下棋
void check_people_play(int *will_x,int *will_y);           //【人机对战】检测玩家棋子，从而进行判断
void check_computer_play(int *will_x,int *will_y);          //【人机对战】检测电脑棋子，从而进行判断

int main(void)
{
    system("title 井字棋【终极版】");                  //改变窗口标题
    system("mode con: cols=38 lines=24");                  //设置窗口大小

    /*                  隐藏光标              */
    HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);     //获得标准输出设备句柄
    CONSOLE_CURSOR_INFO cci;                                //定义光标信息结构体
    GetConsoleCursorInfo(hConsole, &cci);                  //获得当前光标信息
    cci.bVisible = 0;                                        //为0时光标不可见
    SetConsoleCursorInfo(hConsole, &cci);


    int location = 1;         //选项

    main_mean(1);     //显示主菜单

     while(1)     //死循环
    {
        switch(getch())
        {
            case 'W':case 'w':
            W:                  //GOTO标签
                location--;
                if( location == 0 )
                {
                    location++;
                }
                else if( location == 1 )
                {
                    /* 画出本次选择痕迹 */
                    gotoxy(8,4);
                    printf("●");
                    /* 清除上次选择痕迹 */
                    gotoxy(8,7);
```

```
                printf("  ");
            }
            else if( location == 2 )
            {
                /* 画出本次选择痕迹 */
                gotoxy(8,7);
                printf("●");
                /* 清除上次选择痕迹 */
                gotoxy(8,10);
                printf("  ");
            }
            else if( location == 3 )
            {
                /* 画出本次选择痕迹 */
                gotoxy(8,10);
                printf("●");
                /* 清除上次选择痕迹 */
                gotoxy(8,13);
                printf("  ");
            }
            break;

        case 'S':case 's':
        S:        //GOTO标签
            location++;
            if( location == 2 )
            {
                /* 画出本次选择痕迹 */
                gotoxy(8,7);
                printf("●");
                /* 清除上次选择痕迹 */
                gotoxy(8,4);
                printf("  ");
            }
            else if( location == 3 )
            {
                /* 画出本次选择痕迹 */
                gotoxy(8,10);
                printf("●");
                /* 清除上次选择痕迹 */
                gotoxy(8,7);
                printf("  ");
            }
            else if( location == 4 )
            {
                /* 画出本次选择痕迹 */
```

```
                    gotoxy(8,13);
                    printf("●");
                    /* 清除上次选择痕迹 */
                    gotoxy(8,10);
                    printf("  ");
                }
                else if( location == 5 )
                {
                    location--;
                }
                break;
            case 224:            //方向键上和下
            {
                int d_key = getch();         //记录方向键
                if( 72 == d_key )             //72--方向键上
                {
                    goto W;
                }
                else if( 80 == d_key )          //80--方向键下
                {
                    goto S;
                }

            }break;
            case 13:
            {
                main_mean(0);    //显示标题

                if( location == 1 )             /*    选项—【人机对战】    */
                {
                    computer_play();//人机对战
                }
                else if( location == 2 )    /*    选项—【双人对战】    */
                {
                     people_play();//双人对战
                }
                else if( location == 3 )    /*    选项—【使用说明】    */
                {
                    a_word_printf("●本程序由奥利奥C语言制作\n作者QQ2783608988\n"
                    "【人机对战】\n"
                    "[W][S][A][D]控制玩家方向\n[回车键]确定 [Esc]返回\n\n"
                    "【双人对战】\n"
                    "玩家1:[W][S][A][D]\n[空格]确定\n玩家2:[↑][↓][←][→]\n[回车键]确定 [Esc]返
回\n"
                    ,50);
```

```
                    printf("====================================\n\n          按【ESC】返回主界
面");
                    if( 27 == getch() )     //27是【ESC】
                    {
                        main_mean(1);     //显示主菜单
                        location = 1;
                    }
                }
                else if( location == 4 )     /*     选项——【获取源码】     */
                {
                    system("start http://user.qzone.qq.com/2783608988/infocenter?
ptsig=6WaMCZpAXYGnFFTEpeBbVlVER2TwK6fP45CaFcPZY7Q_");     //例子——打开作者QQ空间

                    printf("====================================\n\n          按【ESC】返回主界
面");
                    if( 27 == getch() )     //27是【ESC】
                    {
                        main_mean(1);     //显示主菜单
                        location = 1;
                    }
                }

                location = 1;      //恢复
            }break;


        } //switch

    }//while

    return 0;
}

void computer_play(void)                //人机对战主函数
{
    int chess_x = 0 ,chess_y = 0,result,choose;//井字棋横纵坐标，游戏结果，选择，棋子种类

    draw_map(1);             //画井字棋
    getch();

    frist_play = MessageBox(NULL,TEXT("是否电脑先手"),TEXT("选择"),MB_YESNO);//弹出信息框
    if( IDYES == frist_play ) //信息框的【是】按钮被选择
    {
        frist_play = 3;             //2表示用户，3表示电脑
        computer_play_chess();
    }
```

```
    else     //IDNO == choose //信息框的【否】按钮被选择
    {
        frist_play = 2;         //2表示用户，3表示电脑
    }

    getch();
    draw_map(1);             //画井字棋

    if( 3 != chess[chess_y][chess_x] )
    {
         chess[chess_y][chess_x] = 1; //画井字棋初始位置
    }
    else
    {
        chess_x = chess_y = 1;
        chess[chess_y][chess_x] = 1; //画井字棋初始位置
    }

    while(1)
    {
        draw_map(1);             //画井字棋

        switch(getch())
        {

            case 'w':case 'W':
            {
                if( 0 == chess_y ) //防止越界
                {
                    chess_y = 1;
                }
                if( 2 != chess[--chess_y][chess_x] && 3 != chess[chess_y][chess_x] )
                {
                    chess[chess_y][chess_x] = 1;
                }
                if( 1 == chess[chess_y+1][chess_x] )
                {
                    chess[chess_y+1][chess_x] = 0;
                }
            }break;
            case 's':case 'S':
            {
                if( 2 == chess_y ) //防止越界
                {
                    chess_y = 1;
                }
```

```
            if( 2 != chess[++chess_y][chess_x] && 3 != chess[chess_y][chess_x] )
            {
                chess[chess_y][chess_x] = 1;
            }
            if( 1 == chess[chess_y-1][chess_x] )
            {
                chess[chess_y-1][chess_x] = 0;
            }
        }break;
        case 'a':case 'A':
        {
            if( 0 == chess_x ) //防止越界
            {
                chess_x = 1;
            }
            if( 2 != chess[chess_y][--chess_x] && 3 != chess[chess_y][chess_x] )
            {
                chess[chess_y][chess_x] = 1;
            }
            if( 1 == chess[chess_y][chess_x+1] )
            {
                chess[chess_y][chess_x+1] = 0;
            }
        }break;
        case 'd':case 'D':
        {
            if( 2 == chess_x ) //防止越界
            {
                chess_x = 1;
            }
            if( 2 != chess[chess_y][++chess_x] && 3 != chess[chess_y][chess_x] )
            {
                chess[chess_y][chess_x] = 1;
            }
            if( 1 == chess[chess_y][chess_x-1] )
            {
                chess[chess_y][chess_x-1] = 0;
            }
        }break;
        case 13:    //回车键
        {
            if(    1 == chess[chess_y][chess_x] )    //判断当前位置是否为1，那么
            {
                chess[chess_y][chess_x] = 2;//确定当前位置为2
                step++;                          //步数+1(棋盘中棋子的数量+1)
                if( step < 9 )
```

```
                    {
                        computer_play_chess();            //用户确定之后，电脑下棋
                    }
                }

            }break;
            case 27:    //Esc键
            {
                choose = MessageBox(NULL,TEXT("游戏还没有结束，是否返回?"),TEXT("提
示"),MB_YESNO);
                if( IDYES == choose ) //信息框的【否】按钮被选择
                {
                        /*         全部初始化         */
                    for( chess_y = 0 ; chess_y < 3 ; chess_y++ ) //清空井字棋的棋子信息
                    {
                        for( chess_x = 0 ; chess_x < 3 ; chess_x++ )
                        {
                            chess[chess_y][chess_x] = 0 ;
                        }
                    }
                    step = 0,number_time = 1;

                    main_mean(1);
                    return;
                }

            }break;

        }

        result = check_result();                //检测结果
        draw_map(1);                            //画井字棋

        if( 2 == result || 1 == result || 9 == step )
        {

            if( 2 == result )    //当电脑胜利
            {
                choose = MessageBox(NULL,TEXT("太遗憾了，你输了！是否再来一局?"),TEXT("提
示"),MB_YESNO);
                if( IDNO == choose ) //信息框的【否】按钮被选择
                {
                    /*         全部初始化         */
                    for( chess_y = 0 ; chess_y < 3 ; chess_y++ ) //清空井字棋的棋子信息
                    {
                        for( chess_x = 0 ; chess_x < 3 ; chess_x++ )
```

```
                    {
                        chess[chess_y][chess_x] = 0 ;
                    }
                }
                step = 0,number_time = 1;

                main_mean(1);
                return;
            }
        }
        else if( 1 == result )     //当玩家胜利
        {
            choose = MessageBox(NULL,TEXT("你胜利了！是否再来一局?"),TEXT("提示"),MB_YESNO);
            if( IDNO == choose ) //信息框的【否】按钮被选择
            {
                /*        全部初始化        */
                for( chess_y = 0 ; chess_y < 3 ; chess_y++ ) //清空井字棋的棋子信息
                {
                    for( chess_x = 0 ; chess_x < 3 ; chess_x++ )
                    {
                        chess[chess_y][chess_x] = 0 ;
                    }
                }
                step = 0,number_time = 1;

                main_mean(1);
                return;
            }
    }
        else if( 9 == step  )
        {
            choose = MessageBox(NULL,TEXT("平局！是否再来一局?"),TEXT("提示"),MB_YESNO);
            if( IDNO == choose ) //信息框的【否】按钮被选择
            {
                /*        全部初始化        */
                for( chess_y = 0 ; chess_y < 3 ; chess_y++ ) //清空井字棋的棋子信息
                {
                    for( chess_x = 0 ; chess_x < 3 ; chess_x++ )
                    {
                        chess[chess_y][chess_x] = 0 ;
                    }
                }
                step = 0,number_time = 1;

                main_mean(1);
                return;
```

```
                }
            }

            /*         全部初始化         */
            for( chess_y = 0 ; chess_y < 3 ; chess_y++ ) //清空井字棋的棋子信息
            {
                for( chess_x = 0 ; chess_x < 3 ; chess_x++ )
                {
                    chess[chess_y][chess_x] = 0 ;
                }
            }
            step = 0,result = 0,chess_x = chess_y = 0;
            draw_map(1);             //画井字棋

            frist_play = MessageBox(NULL,TEXT("是否电脑先手"),TEXT("选择"),MB_YESNO);
            if( IDYES == frist_play ) //信息框的【是】按钮被选择
            {
                frist_play = 3;         //3表示电脑
                computer_play();
            }
            else                        //信息框的【否】按钮被选择
            {
                frist_play = 2;         //2表示用户
            }
            getch();
            draw_map(1);             //画井字棋

            /*         画井字棋初始位置     */
            if( 3 != chess[chess_y][chess_x] )
            {
                 chess[chess_y][chess_x] = 1;
            }
            else
            {
                chess_x = chess_y = 1;
                chess[chess_y][chess_x] = 1;
            }
        }
    }
    return;
}

void people_play()                          //双人对战主函数
{
    int chess_x = 0 ,chess_y = 0,result,choose,chess_kind;//井字棋横纵坐标，游戏结果，选择，棋子种
类
```

```
    draw_map(2);            //画井字棋
    getch();
    draw_map(2);            //画井字棋

    chess[chess_y][chess_x] = 1; //画井字棋初始位置

    while(1)
    {
        //step表示当前下的棋子的数量，以此判断下一步下的棋子的种类 （棋子数量最多9个）
        if( 1 == step || 3 == step || 5 == step || 7 == step || 9 == step  )
        {
            chess_kind = 2 ;//2表示绿棋 (玩家1 右)
        }
        else
        {
            chess_kind = 3 ;//2表示蓝棋 (玩家2 左)
        }

        draw_map(2);            //画井字棋

        if( 2 == chess_kind ) //2表示蓝棋
        {
            switch(getch())
            {
                case 'w':case 'W':
                {
                    if( 0 == chess_y ) //防止越界
                    {
                        chess_y = 1;
                    }
                    if( 2 != chess[--chess_y][chess_x] && 3 != chess[chess_y][chess_x] )
                    {
                        chess[chess_y][chess_x] = 1;
                    }
                    if( 1 == chess[chess_y+1][chess_x] )
                    {
                        chess[chess_y+1][chess_x] = 0;
                    }
                }break;
                case 's':case 'S':
                {
                    if( 2 == chess_y ) //防止越界
                    {
                        chess_y = 1;
                    }
```

```
                    if( 2 != chess[++chess_y][chess_x] && 3 != chess[chess_y][chess_x] )
                    {
                        chess[chess_y][chess_x] = 1;
                    }
                    if( 1 == chess[chess_y-1][chess_x] )
                    {
                        chess[chess_y-1][chess_x] = 0;
                    }
                }break;
                case 'a':case 'A':
                {
                    if( 0 == chess_x ) //防止越界
                    {
                        chess_x = 1;
                    }
                    if( 2 != chess[chess_y][--chess_x] && 3 != chess[chess_y][chess_x] )
                    {
                        chess[chess_y][chess_x] = 1;
                    }
                    if( 1 == chess[chess_y][chess_x+1] )
                    {
                        chess[chess_y][chess_x+1] = 0;
                    }
                }break;
                case 'd':case 'D':
                {
                    if( 2 == chess_x ) //防止越界
                    {
                        chess_x = 1;
                    }
                    if( 2 != chess[chess_y][++chess_x] && 3 != chess[chess_y][chess_x] )
                    {
                        chess[chess_y][chess_x] = 1;
                    }
                    if( 1 == chess[chess_y][chess_x-1] )
                    {
                        chess[chess_y][chess_x-1] = 0;
                    }
                }break;
                case 13:    //回车键
                {

                    if( 2 != chess[chess_y][chess_x] && 3 != chess[chess_y][chess_x] )
                    {
                        chess[chess_y][chess_x] = chess_kind;//确定当前位置
                        step++;    //步数+1(棋盘中棋子的数量+1)
```

```
                }
            }break;
        }
    }
    else if( 3 == chess_kind )
    {
        switch(getch())
        {
            case 224:    //方向键
            {
                switch(getch())
                {
                case 72:     //方向键-上
                {
                    if( 0 == chess_y ) //防止越界
                    {
                        chess_y = 1;
                    }
                    if( 2 != chess[--chess_y][chess_x] && 3 != chess[chess_y][chess_x] )
                    {
                        chess[chess_y][chess_x] = 1;
                    }
                    if( 1 == chess[chess_y+1][chess_x] )
                    {
                        chess[chess_y+1][chess_x] = 0;
                    }
                }break;

                case 80:     //方向键-下
                {
                    if( 2 == chess_y ) //防止越界
                    {
                        chess_y = 1;
                    }
                    if( 2 != chess[++chess_y][chess_x] && 3 != chess[chess_y][chess_x] )
                    {
                        chess[chess_y][chess_x] = 1;
                    }
                    if( 1 == chess[chess_y-1][chess_x] )
                    {
                        chess[chess_y-1][chess_x] = 0;
                    }
                }break;

                case 75:     //方向键-左
                {
```

```
                    if( 0 == chess_x ) //防止越界
                    {
                        chess_x = 1;
                    }
                    if( 2 != chess[chess_y][--chess_x] && 3 != chess[chess_y][chess_x] )
                    {
                        chess[chess_y][chess_x] = 1;
                    }
                    if( 1 == chess[chess_y][chess_x+1] )
                    {
                        chess[chess_y][chess_x+1] = 0;
                    }
                }break;

                case 77:     //方向键-右
                {
                    if( 2 == chess_x ) //防止越界
                    {
                        chess_x = 1;
                    }
                    if( 2 != chess[chess_y][++chess_x] && 3 != chess[chess_y][chess_x] )
                    {
                        chess[chess_y][chess_x] = 1;
                    }
                    if( 1 == chess[chess_y][chess_x-1] )
                    {
                        chess[chess_y][chess_x-1] = 0;
                    }
                }break;
            }
        }break;
        case 32:    ///空格键
        {
            if( 2 != chess[chess_y][chess_x] && 3 != chess[chess_y][chess_x] )
            {
                chess[chess_y][chess_x] = chess_kind;//确定当前位置
                step++;    //步数+1(棋盘中棋子的数量+1)
            }
        }break;
        case 27:    //Esc键
        {
            choose = MessageBox(NULL,TEXT("游戏还没有结束，是否返回?"),TEXT("提
示"),MB_YESNO);
            if( IDYES == choose ) //信息框的【否】按钮被选择
            {
                    /*         全部初始化         */
```

```
                for( chess_y = 0 ; chess_y < 3 ; chess_y++ ) //清空井字棋的棋子信息
                {
                    for( chess_x = 0 ; chess_x < 3 ; chess_x++ )
                    {
                        chess[chess_y][chess_x] = 0 ;
                    }
                }
                step = 0,number_time = 1;

                main_mean(1);
                return;
            }

        }break;
    }
    }

    result = check_result();            //检测结果
    draw_map(2);            //画井字棋

    if( 2 == result || 1 == result || 9 == step )
    {
        if( 2 == result )    //当绿子胜利
        {
            choose = MessageBox(NULL,TEXT("绿子胜利! 是否再来一局?"),TEXT("提示"),MB_YESNO);
            if( IDNO == choose ) //信息框的【否】按钮被选择
            {
                /*         全部初始化         */
                for( chess_y = 0 ; chess_y < 3 ; chess_y++ ) //清空井字棋的棋子信息
                {
                    for( chess_x = 0 ; chess_x < 3 ; chess_x++ )
                    {
                        chess[chess_y][chess_x] = 0 ;
                    }
                }
                step = 0,number_time = 1;

                main_mean(1);
                return;
            }
        }
        else if( 1 == result )     //当蓝子胜利
        {
            choose = MessageBox(NULL,TEXT("蓝子胜利! 是否再来一局?"),TEXT("提示"),MB_YESNO);
            if( IDNO == choose ) //信息框的【否】按钮被选择
            {
```

```
                /*         全部初始化         */
                for( chess_y = 0 ; chess_y < 3 ; chess_y++ ) //清空井字棋的棋子信息
                {
                    for( chess_x = 0 ; chess_x < 3 ; chess_x++ )
                    {
                        chess[chess_y][chess_x] = 0 ;
                    }
                }
                step = 0,number_time = 1;

                main_mean(1);
                return;
            }
        }
        else if( 9 == step )
        {
            choose = MessageBox(NULL,TEXT("平局! 是否再来一局?"),TEXT("提示"),MB_YESNO);
            if( IDNO == choose ) //信息框的【否】按钮被选择
            {
                /*         全部初始化         */
                for( chess_y = 0 ; chess_y < 3 ; chess_y++ ) //清空井字棋的棋子信息
                {
                    for( chess_x = 0 ; chess_x < 3 ; chess_x++ )
                    {
                        chess[chess_y][chess_x] = 0 ;
                    }
                }
                step = 0,number_time = 1;

                main_mean(1);
                return;
            }
        }

    /*         全部初始化         */
    for( chess_y = 0 ; chess_y < 3 ; chess_y++ ) //清空井字棋的棋子信息
    {
        for( chess_x = 0 ; chess_x < 3 ; chess_x++ )
        {
            chess[chess_y][chess_x] = 0 ;
        }
    }
    step = 1,result = 0,chess_x = chess_y = 0;
    draw_map(2);             //画井字棋

    getch();
```

```
        chess[chess_y][chess_x] = 1; //画井字棋初始位置

        }
    }
    return ;
}

void main_mean(int show)                //主菜单
{
     system("cls");    //清屏
    gotoxy(0,0);
     HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(hConsole,FOREGROUND_GREEN|FOREGROUND_RED|FOREGROUND_INTENSITY);
//设置控制台字体颜色
    printf( "                                   \n"
            "         奥利奥井字棋【终极版】     \n"
            "                                   \n");
    int line;
    for( line = 0 ; line < 5 ; line++ )
    {
        if( show == 1 )
        {

SetConsoleTextAttribute(hConsole,FOREGROUND_BLUE|FOREGROUND_GREEN|FOREGROUND_INTENSITY|COMMO
N_LVB_GRID_HORIZONTAL); //设置控制台字体颜色
        printf( "                                       \n" );

SetConsoleTextAttribute(hConsole,FOREGROUND_BLUE|FOREGROUND_GREEN|FOREGROUND_INTENSITY); //
设置控制台字体颜色



        if( line == 0 )
        {
            printf("                人机对战              \n");
        }
        else if( line == 1 )
        {
            printf("                双人对战              \n");
        }
        else if( line == 2 )
        {
            printf("                使用说明              \n");
        }
        else if( line == 3 )
        {
            printf("                获取源码              \n");
```

```
        }

        }
    }

    if( show == 1 )
    {

    /*         随机显示提示     */
    SetConsoleTextAttribute(hConsole,FOREGROUND_GREEN|FOREGROUND_INTENSITY); //设置控制台字体颜
色
    srand((unsigned)time(NULL));                              //随机数种子
    gotoxy(0,16);
    int tips = ( rand() % 3 ) + 1;
    if( tips == 1 )
    {
        printf("=====================================\n                  小提示\n\n
电脑可是很聪明的\n   不信你就去【人机对战】挑战电脑吧");
    }
    else if( tips == 2 )
    {
        printf("=====================================\n                  小提示\n\n
作者QQ 2783608988\n         有建议的小伙伴可以加Q");
    }
    else if( tips == 3 )
    {
        printf("=====================================\n                  小提示\n\n
如果你不知怎么玩\n     玩游戏还是先看一下【使用说明】");
    }
    else if( tips == 4 )
    {
        printf("=====================================\n                  小提示\n\n
本程序由c语言制作\n       想获取源码可点击【获取源码】");
    }


    SetConsoleTextAttribute(hConsole,FOREGROUND_BLUE|FOREGROUND_GREEN|FOREGROUND_INTENSITY);
//设置控制台字体颜色
     gotoxy(8,4);
    printf("●");
    }
    SetConsoleTextAttribute(hConsole,FOREGROUND_BLUE|FOREGROUND_GREEN|FOREGROUND_INTENSITY);
//设置控制台字体颜色

     return;
 }
```

```
void check_people_play(int *will_x,int *will_y)            //【人机对战】检测玩家棋子，从而进行判断 （参数1：电脑将要下棋的位置x ，参数2：电脑将要下棋的位置y ）
{
    int c_x,c_y;

    /*         检测横列     */
    for( c_y = 0 ; c_y < 3 ; c_y++ )
    {
        if( 2 == chess[c_y][0] && 2 == chess[c_y][1] )
        {
            if( 3 != chess[c_y][2] )
            {
                *will_x = 2, *will_y = c_y;
                return;
            }
        }
        if( 2 == chess[c_y][0] && 2 == chess[c_y][2] )
        {
            if( 3 != chess[c_y][1] )
            {
                *will_x = 1, *will_y = c_y;
                return;
            }
        }
        if( 2 == chess[c_y][1] && 2 == chess[c_y][2] )
        {
            if( 3 != chess[c_y][1] )
            {
                *will_x = 0, *will_y = c_y;
                return;
            }
        }

    }

    /*         检测竖列     */
    for( c_x = 0 ; c_x < 3 ; c_x++ )
    {
        if( 2 == chess[0][c_x] && 2 ==  chess[1][c_x] )
        {
            if( 3 != chess[2][c_x] )
            {
                *will_x = c_x, *will_y = 2;
                return;
            }
```

```
        }
        if( 2 == chess[0][c_x] && 2 == chess[2][c_x] )
        {
            if( 3 != chess[1][c_x] )
            {
                *will_x = c_x, *will_y = 1;
                return;
            }
        }
        if( 2 == chess[1][c_x] && 2 == chess[2][c_x] )
        {
            if( 3 != chess[0][c_x] )
            {
                *will_x = c_x, *will_y = 0;
                return;
            }
        }
    }

    /*         检测对角线          */
    if( 2 == chess[0][0] && 2 ==  chess[1][1] )
    {
        if( 3 != chess[2][2] )
        {
            *will_x = 2, *will_y = 2;
            return;
        }
    }
    if( 2 == chess[0][0] && 2 == chess[2][2] )
    {
        if( 3 != chess[1][1] )
        {
            *will_x = 1, *will_y = 1;
            return;
        }
    }
    if( 2 == chess[1][1] && 2 == chess[2][2] )
    {
        if( 3 != chess[0][0] )
        {
            *will_x = 0, *will_y = 0;
            return;
        }
    }
    if( 2 == chess[0][2] && 2 ==  chess[1][1] )
    {
```

```
        if( 3 != chess[2][0] )
        {
            *will_x = 0, *will_y = 2;
            return;
        }
    }
    if( 2 == chess[0][2] && 2 == chess[2][0] )
    {
        if( 3 != chess[1][1] )
        {
            *will_x = 1, *will_y = 1;
            return;
        }
    }
    if( 2 == chess[1][1] && 2 == chess[2][0] )
    {
        if( 3 != chess[0][2] )
        {
            *will_x = 2, *will_y = 0;
            return;
        }
    }


    *will_x = 4;    //检测不到，传递错误信息
    return;
}

void check_computer_play(int *will_x,int *will_y)        //【人机对战】检测电脑棋子，从而进行判断（参数1：电脑将要下棋的位置x ，参数2：电脑将要下棋的位置y ）
{
    int c_x,c_y;

    /*        检测横列    */
    for( c_y = 0 ; c_y < 3 ; c_y++ )
    {
        if( 3 == chess[c_y][0] && 3 == chess[c_y][1] )
        {
            if( 2 != chess[c_y][2] )
            {
                *will_x = 2, *will_y = c_y;
                return;
            }
        }
        if( 3 == chess[c_y][0] && 3 == chess[c_y][2] )
        {
```

```
            if( 2 != chess[c_y][1] )
            {
                *will_x = 1, *will_y = c_y;
                return;
            }
        }
        if( 3 == chess[c_y][1] && 3 == chess[c_y][2] )
        {
            if( 2 != chess[c_y][0] )
            {
                *will_x = 0, *will_y = c_y;
                return;
            }
        }

    }

    /*        检测竖列    */
    for( c_x = 0 ; c_x < 3 ; c_x++ )
    {
        if( 3 == chess[0][c_x] && 3 ==  chess[1][c_x] )
        {
            if( 2 != chess[2][c_x] )
            {
                *will_x = c_x, *will_y = 2;
                return;
            }
        }
        if( 3 == chess[0][c_x] && 3 == chess[2][c_x] )
        {
            if( 2 != chess[1][c_x] )
            {
                *will_x = c_x, *will_y = 1;
                return;
            }
        }
        if( 3 == chess[1][c_x] && 3 == chess[2][c_x] )
        {
            if( 2 != chess[0][c_x] )
            {
                *will_x = c_x, *will_y = 0;
                return;
            }
        }
    }
```

```
    /*         检测对角线          */
    if( 3 == chess[0][0] && 3 ==  chess[1][1] )
    {
        if( 2 != chess[2][2] )
        {
            *will_x = 2, *will_y = 2;
            return;
        }
    }
    if( 3 == chess[0][0] && 3 == chess[2][2] )
    {
        if( 2 != chess[1][1] )
        {
            *will_x = 1, *will_y = 1;
            return;
        }
    }
    if( 3 == chess[1][1] && 3 == chess[2][2] )
    {
        if( 2 != chess[0][0] )
        {
            *will_x = 0, *will_y = 0;
            return;
        }
    }
    if( 3 == chess[0][2] && 3 ==  chess[1][1] )
    {
        if( 2 != chess[2][0] )
        {
            *will_x = 0, *will_y = 2;
            return;
        }
    }
    if( 3 == chess[0][2] && 3 == chess[2][0] )
    {
        if( 2 != chess[1][1] )
        {
            *will_x = 1, *will_y = 1;
            return;
        }
    }
    if( 3 == chess[1][1] && 3 == chess[2][0] )
    {
        if( 2 != chess[0][2] )
        {
            *will_x = 2, *will_y = 0;
```

```
            return;
        }
    }

    *will_x = 4;    //检测不到，传递错误信息
    return;
}

void draw_map(int model)                //【人机对战，双人对战】画井字棋
{
    HANDLE hOut = GetStdHandle(STD_OUTPUT_HANDLE);    //获取输出流的句柄

    int n_x = 13, n_y = 6,map1,map2;

    if( 1 == model )//模式1_人机对战
    {
        SetConsoleTextAttribute(hOut,FOREGROUND_GREEN|FOREGROUND_BLUE|FOREGROUND_INTENSITY);
// 设置颜色
        gotoxy(12,5);
        printf("现在由你出棋");
    }
    else if( 2 == model )//模式2_双人对战
    {
        if( 1 == step || 3 == step || 5 == step || 7 == step || 9 == step  )
        {

SetConsoleTextAttribute(hOut,FOREGROUND_GREEN|FOREGROUND_BLUE|FOREGROUND_INTENSITY); // 设置
颜色
            gotoxy(10,5);
            printf("现在由玩家2出棋");
        }
        else
        {

SetConsoleTextAttribute(hOut,FOREGROUND_GREEN|FOREGROUND_BLUE|FOREGROUND_INTENSITY); // 设置
颜色
            gotoxy(10,5);
            printf("现在由玩家1出棋");
        }
    }

    if( 1 == number_time )        //如果本函数是第一次运行
    {
        SetConsoleTextAttribute(hOut,FOREGROUND_RED|FOREGROUND_GREEN|FOREGROUND_INTENSITY);
        gotoxy(n_x,n_y);
        printf("□□□□□\n");
```

```
        /*         画出井字棋     */

        for( map1 = 0 ; map1 < 3 ; map1++ )
        {
            gotoxy(n_x,++n_y);
            printf("□      ");


SetConsoleTextAttribute(hOut,FOREGROUND_RED|FOREGROUND_GREEN|FOREGROUND_INTENSITY);
            printf("□");
        }
        SetConsoleTextAttribute(hOut,FOREGROUND_RED|FOREGROUND_GREEN|FOREGROUND_INTENSITY);
        gotoxy(n_x,++n_y);
        printf("□□□□□\n");

        if( 1 == model )//模式1_人机对战
        {

            SetConsoleTextAttribute(hOut,FOREGROUND_GREEN|FOREGROUND_INTENSITY); //设置控制台
字体颜色
            gotoxy(0,13);
            printf("=====================================\n               使用说明\n\n"
            "           ↓【人机对战】↓\n"
            "        [W][S][A][D]控制玩家方向\n        [回车键]确定   [Esc]返回\n\n"
            "        作者奥利奥QQ2783608988\n");


        }
        else if( 2 == model )//模式2_双人对战
        {
            SetConsoleTextAttribute(hOut,FOREGROUND_GREEN|FOREGROUND_INTENSITY); //设置控制台
字体颜色
            gotoxy(0,13);
            printf("=====================================\n               使用说明\n\n"
            "           ↓【双人对战】↓\n"
            "        玩家1:[↑][↓][←][→]\n        [空格键]确定          \n玩家2:[W][S][A][D]\n
[回车键]确定   [Esc]返回\n"
            "        作者奥利奥QQ2783608988\n");

        }
        SetConsoleTextAttribute(hOut,FOREGROUND_GREEN|FOREGROUND_BLUE|FOREGROUND_INTENSITY);
// 设置颜色
        gotoxy(8,5);
        printf("     按任何键开始     ");
    }
    else    //不是第一次运行本函数
```

```
    {
        n_x = 15 ,n_y = 7 ;

        /*         画出井字棋     */
        for( map1 = 0 ; map1 < 3 ; map1++ )
        {
            gotoxy(n_x,n_y++);
            for( map2 = 0 ; map2 < 3 ; map2++ )
            {
                if( 0 == chess[map1][map2] )
                {
                    printf("  ");
                }
                else if( 1 == chess[map1][map2] )
                {

SetConsoleTextAttribute(hOut,FOREGROUND_GREEN|FOREGROUND_BLUE|FOREGROUND_INTENSITY); // 设置
颜色
                    printf("·");
                }
                else if( 2 == chess[map1][map2] )
                {

SetConsoleTextAttribute(hOut,FOREGROUND_GREEN|FOREGROUND_BLUE|FOREGROUND_INTENSITY); // 设置
颜色
                    printf("×");
                }
                else if( 3 == chess[map1][map2] )
                {
                    SetConsoleTextAttribute(hOut,FOREGROUND_GREEN|FOREGROUND_INTENSITY); //
设置颜色
                    printf("●");
                }
            }

        }
    }



        number_time++;//运行次数+1
}

void computer_play_chess(void)                    //【人机对战】 电脑下棋
{
     /*
    【人工智能，电脑下棋】
```

```
相关文章：
详解Tic-Tac-Toe人工智能实现_精品文库_IThao123 - IT行业第一站
http://www.ithao123.cn/content-3384427.html
 */
 srand((unsigned)time(NULL));            //随机数种子
 int way = (int)(rand()%3);                  //取随机数（0~2）【随机数让电脑下棋方式不单一】

 if( 0 == step )     //电脑先手
 {
     if( 0 == way )    //占中间
     {
         chess[1][1]=3;
    }
     else if( 1 == way )    //占棱位
     {
         way = (int)(rand()%4);
         if( 0 == way )
         {
             chess[0][1] = 3;
        }
         else if( 1 == way )
         {
             chess[2][1] = 3;
        }
        else if( 2 == way )
         {
             chess[1][0] = 3;
        }
         else if( 3 == way )
         {
             chess[1][2] = 3;
        }
    }
    else if( 2 == way )    //占四角
    {
        way = (int)(rand()%4);
         if( 0 == way )
         {
             chess[0][0] = 3;
        }
         else if( 1 == way )
         {
             chess[0][2] = 3;
        }
        else if( 2 == way )
         {
```

```
                chess[2][0] = 3;
            }
             else if( 3 == way )
             {
                chess[2][2] = 3;
            }
        }
    }

    else if( 1 == step ) //电脑后手
    {
        if( 2 == chess[1][1] )       //玩家落子在中心位置
        {
            /*
            使用随机数，让电脑下棋位置不单一
            下面都是让落子都在角位 （先角原则）
            */
            way = (int)(rand()%4);
            if( 0 == way )
            {
                chess[0][0] = 3;
            }
            else if( 1 == way )
            {
                chess[0][2] = 3;
            }
            else if( 2 == way )
            {
                chess[2][0] = 3;
            }
            else if( 3 == way )
            {
                chess[2][2] = 3;
            }
        }
        else                        //玩家落子在棱位，角位
        {
            chess[1][1] = 3;    //电脑落子在中间
        }
    }
    else
    {
        int c_x,c_y;
        check_computer_play(&c_x,&c_y);//如果电脑可以赢，就攻（任何己方两枚棋子连接在一起，且连线有空位时，落子在空位。）
         if( 4 != c_x ) //如果可以攻
```

```
         {
             chess[c_y][c_x] = 3;
        }
        else
        {
            check_people_play(&c_x,&c_y);//否则如果玩家快要赢，就守(任何对方两枚棋子连接在一起，且连线有空位时，落子在空位。)
            if( 4 != c_x )      //如果可以守
             {
                 chess[c_y][c_x] = 3;
            }
            else
            {
                for( c_y = 0 ; c_y  < 3 ; c_y++ )    //再否则，特殊情况
                {    for( c_x = 0 ; c_x < 3 ; c_x++ )
                    {
                        if( 0 == chess[c_y][c_x])
                        {
                            chess[c_y][c_x] = 3;//从上到下，扫描整个棋盘，然后寻找空位，随便找到一处空白的地方下棋子
                            step++;
                            return;
                        }
                    }
                }
            }
        }
    }
    step++;//步数+1
     return;
 }

int check_result(void)                          //【人机对战，双人对战】检测结果（[3]胜利返回1，[2]胜利返回2，都没胜利返回0。黑子或电脑--[3]，白子或用户--[2])
{
    int c_x,c_y;

    /*         检测横列     */
    for( c_y = 0 ; c_y < 3 ; c_y++ )
    {
        if( 2 == chess[c_y][0] && 2 == chess[c_y][1] && 2 == chess[c_y][2])    //黑子胜利
        {
            return 1;
        }
        else if( 3 == chess[c_y][0] && 3 == chess[c_y][1] && 3 == chess[c_y][2])//白子胜利
        {
```

```
            return 2;
        }
    }

    /*        检测竖列    */
    for( c_x = 0 ; c_x < 3 ; c_x++ )
    {
        if( 2 == chess[0][c_x] && 2 == chess[1][c_x] && 2 == chess[2][c_x] )//黑子胜利
        {
            return 1;
        }
        else if( 3 == chess[0][c_x] && 3 == chess[1][c_x] && 3 == chess[2][c_x] )//白子胜利
        {
            return 2;
        }
    }

    /*        检测对角线          */
    if( ( 2 == chess[0][0] && 2 == chess[1][1] && 2 == chess[2][2] ) ) //黑子胜利
    {
        return 1;
    }
    else if( ( 3 == chess[0][0] && 3 == chess[1][1] && 3 == chess[2][2] ) ) //白子胜利
    {
        return 2;
    }
    if( ( 2 == chess[0][2] && 2 == chess[1][1] && 2 == chess[2][0] ) )//黑子胜利
    {
        return 1;
    }
    else if( ( 3 == chess[0][2] && 3 == chess[1][1] && 3 == chess[2][0] ) )//白子胜利
    {
        return 2;
    }

    return 0;//都没胜利
}

void a_word_printf(char * a, int time)        //渐显文字
{
    int i ;
    for (i = 0; a[i] != '\0'; i++)
    {
        printf("%c", a[i]);
        Sleep(time);
    }
```

```
    return;
}

 void gotoxy(int x,int y)                    //光标到指定位置
{
    HANDLE hOut;
    hOut = GetStdHandle(STD_OUTPUT_HANDLE);         //获得标准输入输出的句柄
    COORD pos = {x,y};                              //表示一个字符在控制台屏幕上的坐标(ASCLL码宽度为1
非ASCLL码宽度为2)
    SetConsoleCursorPosition(hOut ,pos);         //光标定位在对应的位置
    return;
}
```