

五子棋游戏双人对决的设计与实现

吕橙

(北京建筑大学计算机系,北京 100044)

摘要:

运用 C 语言在 DEV C++ 平台上设计开发一款能够实现双人对决模式五子棋游戏,通过游戏的开发,达到学习 C 语言和熟悉软件开发流程的目的。

关键词:

DEV C++; 五子棋游戏; 双人对战

基金项目:

北京建筑大学校级教研项目(No.Y17-32)

0 引言

五子棋是全国智力运动会竞技项目之一,是一种两人对弈的纯策略型棋类游戏。玩法是游戏双方分别使用黑白两色的棋子,下在棋盘直线与横线的交叉点上,先形成五子连珠者获胜,双人模式通常是黑子先手,白子后手。五子棋容易上手,老少皆宜,而且趣味横生,引人入胜;它不仅能增强思维能力,提高智力,而且富含哲理,有助于修身养性。五子棋已在各个游戏平台有应用。

文献[1]使用 Android Studio 为开发工具,开发了一个基于 Android 的五子棋游戏。文献[2]通过电子技术、LabVIEW 技术和投影技术设计投影按键显示五子棋棋盘,真实环境的棋局对弈。文献[3]用 Eclipse 工具和 Java 语言实现了五子棋游戏单机版的人人对弈的过程,其中对实现双方落子的代码做了详细的描述。文献[4]使用 MYSQL 数据库、V-Play 开发框架对五子棋游戏进行了设计与实现,该游戏有三种游戏模式:双人对战模式、联机对战模式与人机对战模式,玩家可以根据自己的需要选择不同的模式进行游戏,并可以和网络上的其他玩家对弈。文献[5]利用了 HTML5 新增的 Canvas 标签实现了五子棋游戏的开发。

五子棋游戏双人对决模式常用于学生课设、实验

等实践环节,本文利用 Dev C++软件实现了五子棋游戏双人对决模式。

1 五子棋游戏双人对决模式设计分析

1.1 游戏规则

对局双方各执一色棋子,空棋盘开局,黑先、白后,交替下子,每次只能下一子,棋子下在棋盘的空白点上,棋子下定后,不得向其它点移动,不得从棋盘上拿掉或拿起另落别处。黑方的第一枚棋子可下在棋盘任意交叉点上,无禁手。

1.2 功能分析

游戏功能设计为 3 个模块,主要有:

- (1)初始化模块,主要包括设置初始化参数和绘制棋盘等;
 - (2)游戏控制模块,主要包括下子、运行整个对局等;
 - (3)检查模块,主要是输赢判断。
- 五子棋游戏功能如图 1 所示。

2 五子棋游戏双人对决模式功能实现

2.1 初始化模块

游戏可以设置黑棋先手,无禁手,功能设计主要包

括设置初始化参数(如设置棋盘尺寸、棋盘初始数据、先手玩家、回合数、加载背景音乐等)和绘制棋盘。

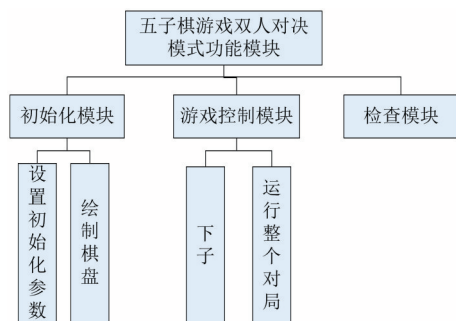


图1 五子棋游戏双人对决模式的功能模块图

2.2 游戏控制模块

游戏执行过程中,通过键盘上(↑)、下(↓)、左(←)、右(→)方向键操控键盘,通过空格键完成落子,当五子连珠时,本局游戏结束,弹出提示对话框。

游戏流程如图2所示。

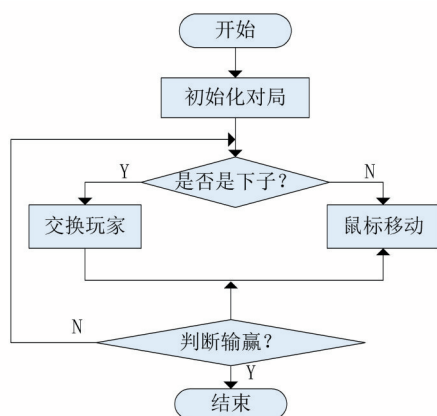


图2 五子棋游戏程序流程图

2.3 检查模块

游戏过程中,当玩家一方落子后,随即进行检查判断是否五子连珠,如果五子连珠则游戏结束,否则继续等待移动鼠标或落子。

3 五子棋游戏双人对决模式代码实现

3.1 初始化模块代码实现

绘制棋盘核心代码如下:

```
// 绘制棋盘
void DrawCheckboard() {
    int i,j;
```

```
system("cls"); // 清屏
for (i = 0; i < CHECKBOARDSIZE; i++)
{
    for (j = 0; j < CHECKBOARDSIZE; j++)
    {
        // 光标
        if (i == g_cursorX && j == g_cursorY)
        {
            printf(" ");
        }
        else if (g_checkboard[i][j] == 1)
        {
            printf("●");
        }
        else if (g_checkboard[i][j] == 2)
        {
            printf("○");
        }
        else if (i == 0 && j == 0)
        {
            printf("┐");
        }
        else if (i == 0 && j == CHECKBOARDSIZE - 1)
        {
            printf("└ ");
        }
        else if (i == CHECKBOARDSIZE - 1 && j == 0)
        {
            printf("┌");
        }
        else if (i == CHECKBOARDSIZE - 1 && j == CHECKBOARDSIZE - 1)
        {
            printf("┘ ");
        }
        else if (i == 0)
        {
            printf("┌");
        }
        else if (i == CHECKBOARDSIZE - 1)
        {
            printf("└");
        }
    }
}
```

```

else if (j == 0)
{
    printf("┌");
}
else if (j == CHECKBOARDSIZE - 1)
{
    printf("┐");
}
else
{
    printf("+");
}
}
printf("\n");
}
}

void DrawCheckboard()
{
    int i, j;
    system("cls"); // 清屏
    for (i = 0; i < CHECKBOARDSIZE; i++)
    {
        for (j = 0; j < CHECKBOARDSIZE; j++)
        {
            // 光标
            if (i == g_cursorX && j == g_cursorY)
            {
                printf(" ");
            }
            else if (g_checkboard[i][j] == 1)
            {
                printf("●");
            }
            else if (g_checkboard[i][j] == 2)
            {
                printf("○");
            }
            else if (i == 0 && j == 0)
            {
                printf("┌");
            }
            else if (i == 0 && j == CHECKBOARD-
SIZE - 1)

```

```

{
    printf("┐");
}
else if (i == CHECKBOARDSIZE - 1 && j ==
= 0)
{
    printf("┌");
}
else if (i == CHECKBOARDSIZE - 1 && j ==
CHECKBOARDSIZE - 1)
{
    printf("┐");
}
else if (i == 0)
{
    printf("┌");
}
else if (i == CHECKBOARDSIZE - 1)
{
    printf("┐");
}
else if (j == 0)
{
    printf("┌");
}
else if (j == CHECKBOARDSIZE - 1)
{
    printf("┐");
}
else
{
    printf("+");
}
}
printf("\n");
}
}

```

3.2 游戏控制模块代码实现

该模块需要进行下子控制,运行整个对局等操作。下子函数:功能是判断该位置是否有下子。代码如下:

```
// 下子
```

```

// int Put() // 成功:1,失败:0(该位置已经有子)
{
    // 判断该位置是否有子
    if (g_checkboard[g_cursorX][g_cursorY] == 0)
    {
        g_checkboard[g_cursorX][g_cursorY] = g_currentGamer; // 黑
        子白子
        return 1;
    }
    else
    {
        return 0;
    }
}
{
    // 判断该位置是否有子
    if (g_checkboard[g_cursorX][g_cursorY] == 0)
    {
        g_checkboard[g_cursorX][g_cursorY] = g_current-
        Gamer; // 黑子白子
        return 1;
    }
    else
    {
        return 0;
    }
}

```

整个对局函数核心代码如下:

```

// 进行整个对局
int RunGame() {
    char input;
    int nWinner = 0;
    Init(); // 调用初始化函数
    while (1)
    {
        DrawCheckboard(); // 调用绘制棋盘
        input = _getch(); // 获取输入
        switch (input)
        {
            case 32: // 空格
                if (1 == Put())
                {
                    nWinner = check(); // 判断输赢
                }
            }
        }
    }
}

```

```

// 交换玩家
if (g_currentGamer == 1)
    g_currentGamer = 2;
else
    g_currentGamer = 1;
if (nWinner == 1)
{
    MessageBox(NULL, "黑子赢了", "五子棋", MB_OK);
}
else if (nWinner == 2)
{
    MessageBox(NULL, "白子赢了", "五子棋", MB_OK);
}
}
break;
case 72: // 上键 ↑
    g_cursorX--; // 行减减
    if (g_cursorX < 0)
        g_cursorX = CHECKBOARDSIZE - 1;
    break;
case 80: // 下键 ↓
    g_cursorX++;
    if (g_cursorX >= CHECKBOARDSIZE)
        g_cursorX = 0;
    break;
case 75: // 左键 ←
    g_cursorY--;
    if (g_cursorY < 0)
        g_cursorY = CHECKBOARDSIZE - 1;
    break;
case 77: // 右键 →
    g_cursorY++;
    if (g_cursorY >= CHECKBOARDSIZE)
        g_cursorY = 0;
    break;
}
}
return 1;
}
int RunGame()
{
    char input;
    int nWinner = 0;
}

```

```

Init(); // 调用初始化函数
while (1)
{
    DrawCheckboard(); // 调用绘制棋盘
    input = _getch(); // 获取输入
    switch (input)
    {
        case 32: // 空格
            if (1 == Put())
            {
                nWinner = check(); // 判断输赢
                // 交换玩家
                if (g_currentGamer == 1)
                    g_currentGamer = 2;
                else
                    g_currentGamer = 1;
                if (nWinner == 1)
                {
                    MessageBox(NULL, "黑子赢了", "五子棋", MB_OK);
                }
                else if (nWinner == 2)
                {
                    MessageBox(NULL, "白子赢了", "五子棋", MB_OK);
                }
            }
            break;
        case 72: // 上键 ↑
            g_cursorX--; // 行减减
            if (g_cursorX < 0)
                g_cursorX = CHECKBOARD_SIZE - 1;
            break;
        case 80: // 下键 ↓
            g_cursorX++;
            if (g_cursorX >= CHECKBOARD_SIZE)
                g_cursorX = 0;
            break;
        case 75: // 左键 ←
            g_cursorY--;
            if (g_cursorY < 0)
                g_cursorY = CHECKBOARD_SIZE -
1;

```

```

        break;
        case 77: // 右键 →
            g_cursorY++;
            if (g_cursorY >= CHECKBOARD_SIZE)
                g_cursorY = 0;
            break;
    }
}
return 1;
}

```

3.3 检查功能代码实现

该模块的代码主要是输赢判断,分别检查水平方向、垂直方向、正斜向、反斜向是否五子连珠,核心代码如下:

```

// 判断输赢
int check() {
    int i, j;
    int nHorizontal = 1; // 水平方向
    int nVertical = 1; // 垂直方向
    int nPositiveDirection = 1; // 正斜向
    int nSkewDirection = 1; // 反斜向

    // 水平 向右检测, g_cursorY 列变化
    for (i = 1; i < 5; i++)
    {
        if (g_cursorY + i < CHECKBOARD_SIZE && g_checkboard[
g_cursorX][g_cursorY + i] == g_currentGamer)
            nHorizontal++;
        else
            break;
    }
    // 水平, 向左检测
    for (i = 1; i < 5; i++)
    {
        if (g_cursorY - i >= 0 && g_checkboard[g_cursorX][g_cursorY - i] == g_currentGamer)
            nHorizontal++;
        else
            break;
    }
    if (nHorizontal >= 5)
        return g_currentGamer;
}

```

```
// 垂直,向下检测,g_cursorX 行变化
for (i = 1; i < 5; i++)
{
    if (g_cursorX - i >= 0 && g_checkboard[g_cursorX - i][g_cursorY] == g_currentGamer)
        nVertical++;
    else
        break;
}
// 垂直,向上检测,g_cursorX 行变化
for (i = 1; i < 5; i++)
{
    if (g_cursorX + i < CHECKBOARDSIZE && g_checkboard[g_cursorX + i][g_cursorY] == g_currentGamer)
        nVertical++;
    else
        break;
}
if (nVertical >= 5)
    return g_currentGamer;
// 正斜向,向斜下检测,g_cursorX 行变化,g_cursorY 列变化
for (i = 1; i < 5; i++)
{
    if (g_cursorX - i >= 0 && g_cursorY - i >= 0 && g_checkboard[g_cursorX - i][g_cursorY - i] == g_currentGamer)
        nPositiveDirection++;
    else
        break;
}
// 正斜向,向斜上检测,g_cursorX 行变化,g_cursorY 列变化
for (i = 1; i < 5; i++)
{
    if (g_cursorX + i < CHECKBOARDSIZE && g_cursorY + i < CHECKBOARDSIZE && g_checkboard[g_cursorX + i][g_cursorY + i] == g_currentGamer)
        nPositiveDirection++;
    else
        break;
}
if (nPositiveDirection >= 5)
    return g_currentGamer;
// 反斜向,向斜上检测,g_cursorX 行变化,g_cursorY 列变化
for (i = 1; i < 5; i++)
```

```
{
    if (g_cursorX + i >= 0 && g_cursorY - i >= 0 && g_checkboard[g_cursorX + i][g_cursorY - i] == g_currentGamer)
        nSkewDirection++;
    else
        break;
}
// 反斜向,向斜下检测,g_cursorX 行变化,g_cursorY 列变化
for (i = 1; i < 5; i++)
{
    if (g_cursorX - i < CHECKBOARDSIZE && g_cursorY + i < CHECKBOARDSIZE && g_checkboard[g_cursorX - i][g_cursorY + i] == g_currentGamer)
        nSkewDirection++;
    else
        break;
}
if (nSkewDirection >= 5)
    return g_currentGamer;
return 0;
}
```

4 五子棋游戏双人对决模式测试与运行

游戏实现了五子棋游戏双人对决模式的基本功能,可以通过操纵“↑”、“↓”、“←”、“→”实现鼠标的上下左右转动,以及“空格”键进行落子,玩家交替落子,运行界面如图 3 所示。

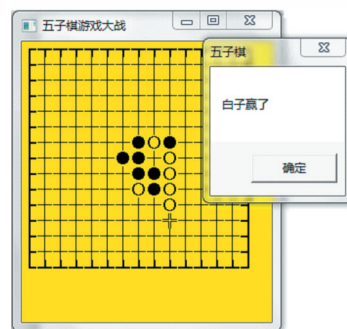


图 3 游戏界面

游戏功能还可以进一步完善。例如:

- (1) 可以设置统计功能,记录玩家胜负盘数。
- (2) 可以修改为通过鼠标进行控制移动光标和落子,而非键盘“↑”、“↓”、“→”、“←”键和空格进行移动

光标和落子。

(3)引入 AI 算法,修改为人机对决模式。

5 结语

本文以五子棋游戏为原型,利用 DEV C++完成了

双人对决游戏模式的基本运行,包括游戏的初始化设置、游戏控制模块和检查模块,游戏操作简单,界面效果良好,基本上能够满足学生课设要求,提高学生的编程兴趣,带给学生具有挑战度的游戏类编程体验。

参考文献:

- [1]黄建,段志举,陈杉杉,等. 基于 Android 的五子棋游戏设计[J]. 数字技术与应用,2020(02):161-162.
- [2]黄旭阳,冯玉松,徐佳璇,等. 一种投影按键显示五子棋棋盘系统的设计[J]. 数码世界,2020(02):88-88.
- [3]刘倚彤. 用 Java 实现五子棋人人对弈[J]. 电脑编程技巧与维护,2019(11):11-12+37.
- [4]万坤,苏雪莲,冉瑞生. 基于 C/S 架构的五子棋游戏软件的设计与实现[J]. 数字技术与应用,2015(04):164-164.
- [5]董春侠,司占军. 基于 HTML5 技术的五子棋游戏的设计与开发[J]. 微型机与应用,2017(11):12-14.

作者简介:

吕橙(1969-),男,辽宁沈阳人,硕士,副教授,研究方向为信息化教育

收稿日期:2020-10-09 修稿日期:2020-10-28

Design and Implementation of Two-Player Mode in Five in a Row Game

LV Cheng

(School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044)

Abstract:

Five in a Row game is designed and developed on the platform of dev C++ with C language. Through the development of the game, we can learn C language and be familiar with the software development process.

Keywords:

DEV C++; Five in a Row Game; Two-Player Mode