

基于C语言的五子棋游戏的设计与实现

贺 胜 段华琼*

(成都锦城学院 计算机与软件学院, 四川 成都 611371)

摘 要: 五子棋是一款老少皆宜的策略游戏。本文基于C语言并使用EasyX图像库在Visual Studio 2019平台下, 进行五子棋游戏的开发构想与实现。首先, 本文对五子棋游戏进行了介绍; 其次, 论述了游戏的设计方案; 最后, 详细介绍了在实现游戏时的几个重要问题及其解决方法。

关键词: 五子棋游戏; C语言; EasyX

中图分类号: TP317 **文献标识码:** A **文章编号:** 1003-9767(2021)22-108-03

Design and Implementation of Gobang Game Based on C Language

HE Sheng, DUAN Huaqiong*

(School of Computer and Software, Chengdu Jincheng University, Chengdu Sichuan 611371, China)

Abstract: Gobang is a strategy game suitable for all ages. Based on C language and EasyX image library, this paper carries out the development conception and implementation of Gobang game on the platform of visual studio 2019. Firstly, the Gobang game is introduced in this paper; Secondly, it discusses the design scheme of the game; Finally, several important problems and their solutions in the implementation of the game are introduced in detail.

Keywords: gobang game; C language; EasyX

0 引言

五子棋是一款老少皆宜的策略性游戏, 十分受大众喜爱^[1]。五子棋不会受到场合的限制, 只需要一个棋盘加上黑白棋就可以进行对弈。更广泛的五子棋玩法, 甚至连棋盘都可以不需要, 在任何平台都可以进行。五子棋游戏独具特色的玩法, 通俗易懂的规则, 使得该游戏成为了一代又一代人的回忆。

C语言是绝大多数入门计算机编程的人首先学习的一门语言, 它是一门面向过程的计算机语言^[2]。程序员使用C语言特定的语法, 告诉计算机要实现的内容是什么, 如何去实现该内容, 这体现了一个程序开发者的逻辑思维, 并可以细致到实现过程的每一步。下面将对五子棋的开发实现方案进行讲解。

1 游戏设计

1.1 游戏简介

本五子棋游戏基于C语言进行开发, 是一款进行人机对

弈的游戏。下面简单介绍一下五子棋的游戏规则: 玩家为黑色棋子并先落子, 计算机为白色棋子, 双方可在棋盘内的任意位置落子^[3]; 双方以自己的棋子为中心, 从8个方向进行查看, 首先达到5个棋子连成一线的为本轮游戏的胜者。

1.2 游戏设计

本文设计的五子棋游戏包括3个功能界面, 分别是游戏的欢迎界面、下棋互动界面以及最终评分界面^[4]。

1.2.1 游戏开始界面

首先, 绘制一个800×800像素的图形化窗口, 并在后台启用音乐播放函数, 以达到背景音乐功能的实现; 其次, 采用图形库中的loadimage()函数加载背景图片; 最后, 为使开始界面有一段停留时间, 使用System("PAUSE")指令对该程序进行暂停操作, 用户按下回车键时可进入游戏。

1.2.2 游戏操作

进入游戏开始界面后, 敲击回车进入游戏互动界面(下

作者简介: 贺胜(2001—), 男, 四川泸州人, 本科在读。研究方向: 软件工程。

通信作者: 段华琼(1980—), 女, 四川资中县人, 硕士研究生, 副教授。研究方向: 软件开发研究。E-mail: 112111164@qq.com。

棋对弈)。每当一局完成后,自动进入评分界面,之后可通过回车键返回游戏界面。若要退出游戏,可随时点击游戏窗口右上角的关闭按钮。

在游戏开始界面和结束界面下,用户需要用到键盘操作(回车键),其余的游戏操作均是通过鼠标进行的。需要用到 GetMouseMsg() 函数捕获用户对鼠标的操作信息,如果没有捕获到,函数就会一直等待用户输入^[10]。可以使用 graphics.h 文件中的 MOUSEMSG 结构体类型对捕捉到的鼠标信息进行存储。

1.2.3 评分方法

游戏的评分采用积分的方式进行分数计算,包括计算机方的积分规则和用户方的积分规则。用户的初始分数为0分,取得游戏的最终胜利后将进行加分,反之则为扣分^[9]。下面介绍具体的积分规则。

计算机积分规则:计算机每落一个子,进行10分的加分;当计算机落下该子成两点连接时(连2)将有15分的加分;若计算机落下该子成三点连接且连线方向无黑子时(活3)将有55分的加分,反之若周围有黑子在三点连线的一端(死3)则计算机将有30分的加分;若计算机落下该子成4点连线且连线方向均无黑子(活4)将有400分的加分,反之若有黑子阻拦(死4)将有60分的加分;若计算机最终五子连线(连5),则最后将有5000分的加分。

玩家的积分规则:当用户落下该子成两点连接时(连2)将有15分的加分;若用户落下该子成三点连接且连线方向无白子时(活3)将有45分的加分,反之若周围有白子在三点连线的一端(死3)则用户将有35分的加分;若用户落下该子成4点连线且连线方向均无白子(活4)将有300分的加分,反之若有白子阻拦(死4)将有65分的加分;若用户最终五子连线(连5),则最后将有4000分的加分。

2 游戏实现

2.1 程序的结构逻辑

main 函数^[5]中的语句十分的简单,仅仅只有一个顺序逻辑和一个循环逻辑。首先,游戏通过用户自定义 start() 函数进行窗口绘制并展现开始界面;其次,当用户敲击回车后开始执行游戏的初始化,其中起重要作用的 initializeData() 函数针对游戏进行初始化操作,该函数包括加载操作棋盘界面、游戏开始提示音、预加载游戏黑白棋操作棋子图片、初始化棋盘数据以及初始化开始分数,之后便进入 for 循环。

这个 for 循环是实现游戏反复轮换下棋操作的关键,进入循环后首先会设定一个鼠标结构体变量,等待用户点击,用户在进行点击后系统将进行鼠标信息的收集并将这些数据存储到鼠标结构体变量。之后会进行 if 判断,若进行了点击操作并且合法将会激活下一个代码块。

在 if 判断语句中的 manGo() 函数将开始对用户所点击

的位置进行白子图片的渲染并作棋盘更新,之后检查游戏是否结束;若游戏没有结束,将由 computerGo() 函数重复类似 manGo() 函数的操作,不同的是它将开始对整个棋盘数据进行分析,数据处理后开始进行黑子图片渲染并更新棋盘,并再次进行游戏数据判断是否结束游戏;若游戏结束,将转入游戏结算评分界面,并作分数数据的存储和游戏初始化操作;当用户敲击回车后将继续执行本 for 结构,这样就实现了反复轮换下棋的操作。

2.2 游戏运行的实现

具体实现过程中有如下几个方面是难点:准确进行棋子位置的渲染操作、机器下棋算法的设计、界面的构建以及 initializeData() 函数如何进行数据初始化。

2.2.1 棋子渲染操作

为了使棋子在棋盘进行更好的渲染,需要对鼠标信息进行一些处理,通过 chessDown() 函数对用户实际点击的操作进行数据整理,棋盘局部介绍如图1所示。

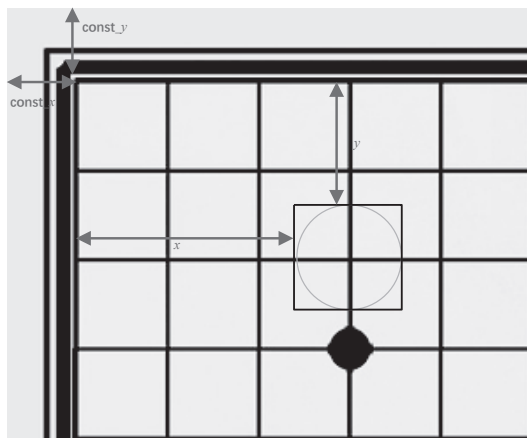


图1 棋盘局部介绍

图1中,为使棋子的图片正确对准每个格子的中心,这里需要首先测量出 const_x 和 const_y 的值。这两个值分别是棋盘框架距离图片的边距,这里可以用诸如 PhotoShop 等图片处理软件进行测量,要求是以像素为单位的值,这里测得 const_x 与 const_y 的值的大小为41个像素距离,此外还需要测量每个小格子的大小。通过测量可知,小格子的大小为51.5×51.5像素。接着只需要使棋子图片的右上角位于格子中心即可,计算方式:棋盘边框距离图片边距的大小(图中为 const_x 与 const_y)加上鼠标获取信息的横坐标数据(图中为 x)或纵坐标数据(图中为 y)并乘上棋盘格子边框的大小(bsize)并减去半个棋盘格子的大小。具体实现代码为:

```
void chessDown(int across, int longit, chess_kind_t kind) {
    mciSendString("play res/down8.WAV", 0, 0, 0);
    int x = const_x + longit * bsize - 0.5 * bsize;
    int y = const_y + across * bsize - 0.5 * bsize;
    if (kind == chess_w) drawPNG(&whitechess, x, y);
}
```

```
else drawPNG(&blackchess, x, y);
```

在进行渲染的时候将鼠标所得到的相对位置 across 和 longit 以及棋子的种类传入该函数即可实现棋子位置的准确渲染。

2.2.2 计算机下棋算法的设计

计算机要对用户下棋的操作进行判断,并作出自己的下棋选择,是游戏的核心算法,实现的基本思路是采用一个二维数组,对计算机和用户所做的操作进行标识。将计算机的棋子在二维数组中存储为 -1 值,用户为 1,未进行操作的区域全部初始化为 0。具体操作就是以用户或者计算机棋子为中心,在二维数组中以标号为中心,向 4 个方向进行权值计算,其中的权值大小即为计算机积分规则,计算机将为分数较高的那个位置进行棋子渲染并更新二维数组中的值。

2.2.3 背景音乐及数据初始化

2.2.3.1 背景音乐

背景音乐在游戏中分为 3 种,一种是游戏进行时一直播放的背景音乐,第二种是游戏点击音效,第三种是游戏评分时的音乐。背景音乐需要一开始就加载,否则无法达到全局背景音乐的效果。点击音效在棋子进行渲染时播放^[6-8]。评分音乐在加载评分界面时播放,实现命令为:

```
if (game.playerFlag == false) {
    mciSendString( "play res/win.
WAV" , 0, 0, 0);
    drawscore();
}
else .....
```

2.2.3.2 数据初始化

对于游戏的初始化,本程序需要进行 3 个初始化操作:图形化窗口的初始化、评分数据的初始化以及记录棋盘数据的二维数组初始化。

在本程序开始正式运行的时候便需要进行图形化窗口的初始化,该初始化的主要目的是为了绘制图像化窗口,并加载游戏运行时所需要的各种素材,为游戏运行调用作准备。

评分数据的初始化需要在两个地方进行运行:首先是程序一开始运行时需要进行评分数据的初始化;其次是每轮评分显示输出后将进行一次评分数据的初始化。主要目的也有两点:一是更好地准备计算机权值计算;二是更新积分数据做出准确的最终评分。

游戏的棋盘数据是用一个二维数组进行记录的,所以一开始需要将 15×15 的二维数组进行初始化,使得其中的每一个值为 0 (即没有任何棋子),这样才能准确记录游戏互动时的棋子数据,相关的实现代码为:

```
void init(){
    loadimage(0, "res/chess_starblackground.jpg" );
    mciSendString( "play res/start.wav" , 0, 0, 0);
    loadimage(&chessBlackImg, "res/blackchess.jpg" ,
```

```
BLOCK_SIZE, BLOCK_SIZE, true);
    loadimage(&chessWhiteImg, "res/whitechess.jpg" ,
BLOCK_SIZE, BLOCK_SIZE, true);
    if (!data) return;
    memset(data->chessMap, 0, sizeof(data->chessMap));
    memset(data->scoreMap, 0, sizeof(data->scoreMap));
    data->playerFlag = true;
    setbkcolor(WHITE);
    settextrcolor(BLUE);
    settextrstyle(50, 0, "微软雅黑" );
    ..... }
```

2.3 评分

本游戏采用积分的形式进行评分,游戏开始时的初始分为 0 分。采用文件形式存储每一次得分,使得每次进行评分结算时,不会因为程序的结束而导致游戏评分数据在最后被清空,类似数据库的方式存储每次游戏的评分数据。

3 结 语

本游戏基于 C 语言开发,并使用了第三方文件,完善了编译器自身不足。开发中调用了文件中的窗口绘制函数,使得本五子棋游戏具有更加美观的操作界面。游戏加入了背景音乐,可以在游戏进行时通过声音向用户进行反馈。游戏引入了机器学习的相关知识,使游戏中的机器方看上去具有人的意识,增加了游戏的趣味性与耐玩性。希望通过本文的讲解给 C 语言学习者提供一些新的游戏开发思路。

参考文献

- [1] 何丽平. 基于 Nios II 软核的 SOPC 五子棋游戏设计 [C]//2008 年“ICT 助力两型社会建设”学术研讨会论文集,2008:4.
- [2] 周厚年. 论计算机技术的重要性 [J]. 电脑知识与技术,2016,12(31):34-35.
- [3] 邓嘉怡. 基于 C++ 的五子棋 AI 算法 [J]. 通讯世界,2019,26(1):281-282.
- [4] 陈慧杰,郭占祥. 基于 C 语言的五子棋游戏程序设计 [J]. 宁波职业技术学院学报,2012,16(2):41-44.
- [5] 谭浩强. C 语言程序设计 [M]. 北京:清华大学出版社,2018:69.
- [6] 王小妮. 嵌入式五子棋游戏设计 [J]. 电脑开发与应用,2014,27(10):64-66.
- [7] 于涵. 网络游戏的互动性因素探究 [D]. 青岛:青岛理工大学,2016:41.
- [8] 奇牛编程. 五子棋人机对战 C 语言 [EB/OL]. (2021-06-01)[2021-11-09]. <https://www.bilibili.com/video/BV1A5411M7iM?from=search&seid=8949369137017454347>, 2021-06-01/2021-08-18