

计算机软件开发系列丛书

自己动手设计 Windows 趣味游戏程序



王 敏 编著

学苑出版社

TP312
W/M/1

计算机软件开发系列丛书

自己动手设计 Windows 趣味游戏程序

王 敏 编著

熊可宜 审校

学苑出版社
126410

(京)新登字 151 号

内 容 提 要

本书主要由两大部分组成,一部分是 Windows 下趣味程序,另一部分是 Windows 下游戏程序。

本书中无论是趣味程序,还是游戏程序,每个例子都是经过精心挑选的,具有一定的代表性。

在第一部分里,主要介绍了一些经典的趣味程序,如四色定理的验证,哥德巴赫猜想的验证,汉诺塔等等,同时还介绍了如何利用分形来生成自然奇观,如树、原始森林、岩石、海豹等等。

第二部分,首先介绍了在 Windows 下进行游戏程序设计需注意的几个方面,后面附有 10 个实例。

总之,本书将知识、趣味、游戏三者结合在一起,寓教于乐,从中可以获得许多有益的启示。

本书中的所有程序均采用 Borland C++ 2.0 编写而成。

欲购本书的用户,请直接与北京海淀 8721 信箱书刊部联系,邮政编码:100080,电话:2562329。

JS588/12

计算机软件开发系列丛书

自己动手设计 Windows 趣味游戏程序

编 著:王 敏

审 校:熊可宜

责任编辑:甄国宪

出版发行:学苑出版社 邮政编码:100036

社 址:北京市海淀区万寿路西街 11 号

印 刷:北京太和印刷厂

开 本:787×1092 1/16

印 张:25 字 数:587 千字

印 数:1~5000 册

版 次:1994 年 3 月北京第 1 版第 1 次

ISBN7-5077-0779-2/TP·11

本册定价:42.00 元

学苑版图书印、装错误可随时退换

目 录

第一章 Windows 下的趣味程序	1
1.1 万年历	1
1.2 关键路径	12
1.3 迷宫探道	20
1.4 魔方阵	31
1.5 汉诺塔	39
1.6 自动发牌	49
1.7 四方定理	55
1.8 波瓦松分酒	62
1.9 约瑟夫问题	67
1.10 哥德巴赫猜想	72
1.11 怎样存钱利最大	78
1.12 杨辉三角形	84
1.13 四色定理	89
1.14 骑士周游世界	96
1.15 一片绿叶	106
1.16 一片森林	112
1.17 岩石	117
1.18 饥饿的海豹	122
1.19 一棵大树	127
1.20 完美正方	133
1.21 北京地区公共交通查询	141
第二章 Windows 下游戏程序设计概述	165
2.1 电脑游戏概述	165
2.2 定时器与系统时钟	167
2.3 BITBLT 动画	169
2.4 键盘	194
2.5 鼠标器	198
2.6 如何利用扬声器演奏音乐	199
第三章 游戏程序设计实例	207
3.1 三维井字棋	207
3.2 模拟钢琴	230
3.3 打臭虫	241
3.4 巧搭积木	258

3.5	智击伞兵	285
3.6	法国钉	299
3.7	遨游九宫	319
3.8	八皇后	331
3.9	精灵吃豆	351
3.10	智击敌舰	369

第一章 Windows 下的趣味程序

1.1 万年历

项目文件 CALENDAR.PRJ 包括文件

CALENDAR.DEF CALENDAR.RC CALENDAR.C

经编译运行后,利用对话框输入所要的年代,再输入所要的月份(若为 0 则打印全年)。

该程序原理是利用输入年代,代入公式,求出该年第一天是星期几,并算出该年是否为闰年,然后将月份,日期及星期几放入矩阵中,再将其打印出来。下面附上 1995 年的年历。

1995 JANUARY							FEBRUARY							MARCH						
SU	MO	TU	WE	TH	FR	SA	SU	MO	TU	WE	TH	FR	SA	SU	MO	TU	WE	TH	FR	SA
1	2	3	4	5	6	7				1	2	3	4				1	2	3	4
8	9	10	11	12	13	14	5	6	7	8	9	10	11	5	6	7	8	9	10	11
15	16	17	18	19	20	21	12	13	14	15	16	17	18	12	13	14	15	16	17	18
22	23	24	25	26	27	28	19	20	21	22	23	24	25	19	20	21	22	23	24	25
29	30	31					26	27	28					26	27	28	29	30	31	

1995 APRIL							MAY							JUNE						
SU	MO	TU	WE	TH	FR	SA	SU	MO	TU	WE	TH	FR	SA	SU	MO	TU	WE	TH	FR	SA
					1		1	2	3	4	5	6		1	2	3				
2	3	4	5	6	7	8	7	8	9	10	11	12	13	4	5	6	7	8	9	10
9	10	11	12	13	14	15	14	15	16	17	18	19	20	11	12	13	14	15	16	17
16	17	18	19	20	21	22	21	22	23	24	25	26	27	18	19	20	21	22	23	24
23	24	25	26	27	28	29	28	29	30	31				25	26	27	28	29	30	
30																				

1995 JULY							AUGUST							SEPTEMBER						
SU	MO	TU	WE	TH	FR	SA	SU	MO	TU	WE	TH	FR	SA	SU	MO	TU	WE	TH	FR	SA
					1		1	2	3	4	5			1	2					
2	3	4	5	6	7	8	6	7	8	9	10	11	12	3	4	5	6	7	8	9
9	10	11	12	13	14	15	13	14	15	16	17	18	19	10	11	12	13	14	15	16
16	17	18	19	20	21	22	20	21	22	23	24	25	26	17	18	19	20	21	22	23
23	24	25	26	27	28	29	27	28	29	30	31			24	25	26	27	28	29	30
30	31																			

1995 OCTOBER							NOVEMBER							DECEMBER						
SU	MO	TU	WE	TH	FR	SA	SU	MO	TU	WE	TH	FR	SA	SU	MO	TU	WE	TH	FR	SA
1	2	3	4	5	6	7				1	2	3	4	3	4	5	6	7	8	9
8	9	10	11	12	13	14	5	6	7	8	9	10	11	10	11	12	13	14	15	16
15	16	17	18	19	20	21	12	13	14	15	16	17	18	17	18	19	20	21	22	23
22	23	24	25	26	27	28	19	20	21	22	23	24	25	24	25	26	27	28	29	30
29	30	31					26	27	28	29	30			31						

图 1.1

2 自己动手设计 Windows 趣味游戏程序

有关程序的说明请见程序清单。

下面附上程序清单。

CALENDAR.DEF

```
NAME            CALENDAR

DESCRIPTION      'produce CALENDAR'

EXETYPE         WINDOWS

STUB            'WINSTUB.EXE'

CODE            PRELOAD MOVEABLE
DATA            PRELOAD MOVEABLE MULTIPLE

HEAPSIZE        1024
STACKSIZE       8192

EXPORTS
    MainWndProc    @1
    PARADLG        @2
```

CALENDAR.H

```
// Constants defined for Line Dialog box
```

```
#define IDM_INPUT      110
#define IDM_EXIT       111

#define DI_LLEFT       400
#define DI_LTOP        410
#define DI_LRIGHT     420
#define DI_LBOTTOM    430

#define DI_LOK         440
#define DI_LCANCEL     450
```

CALENDAR.RC

```
#include "calendar.h"
```

```
Calendar MENU
```

```
BEGIN
```

```
    POPUP        "&Calendar"
```

```
        BEGIN
```

```
            MENUITEM    "&Input Parameter"    IDM_INPUT
```

```
            MENUITEM    "&Exit"                IDM_EXIT
```

```
        END
```

```
END
```

```
PARADLG DIALOG DISCARDABLE LOADONCALL PURE MOVEABLE
```

```
    68, 57, 140, 70
```

```
STYLE WS_POPUP | WS_CAPTION | WS_SYSMENU | 0x80L
```

```
CAPTION "Input Parameter"
```

```
BEGIN
```

```
    CONTROL "" DI_LLEFT,    "EDIT", WS_CHILD | WS_VISIBLE |  
                                WS_BORDER | WS_TABSTOP, 47, 10, 31, 12
```

```
    CONTROL "" DI_LRIGHT,  "EDIT", WS_CHILD | WS_VISIBLE |  
                                WS_BORDER | WS_TABSTOP, 47, 28, 31, 12
```

```
    CONTROL "YEAR",        -1, "STATIC", WS_CHILD | WS_VISIBLE,  
                                7, 10, 19, 9
```

```
    CONTROL "MONTH"        -1, "STATIC", WS_CHILD | WS_VISIBLE,  
                                7, 28, 28, 9
```

```
    CONTROL "OK"           DI_LOK,    "BUTTON", WS_CHILD | WS_VISIBLE |  
                                WS_TABSTOP | 0x1L, 24, 49, 32, 12
```

```
    CONTROL "Cancel"       DI_LCANCEL, "BUTTON", WS_CHILD | WS_VISIBLE |  
                                WS_TABSTOP, 74, 49, 40, 12
```

```
END
```

```
CALENDAR.C
```

```
#include <windows.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
#include "calendar.h"
```

```
#define MSG2 "SU MO TU WE TH FR SA"
```

```
static int md[12][6][7];
```

```
static char *mm[12] = {
```

```

    " JANUARY ","FEBRUARY "," MARCH "," APRIL ",
    " MAY "," JUNE "," JULY "," AUGUST ",
    "SEPTEMBER "," OCTOBER ","NOVEMBER","DECEMBER "
    } ;

static int dd[12] = {
    31,28,31,30,31,30,31,31,30,31,30,31 } ;

int t , a , m , i , j , date ;
float w ;
char szBuffer[100] ;

int Year , Month ;
static          xChar , yChar ;
TEXTMETRIC tm ;
static short nVscrollPos = 0 ;
static POINT Client ;

int PASCAL WinMain(HANDLE, HANDLE, LPSTR, int);
long FAR PASCAL MainWndProc(HWND, unsigned, WORD, LONG);
BOOL FAR PASCAL ParaDlgProc(HWND, unsigned, WORD, LONG);

FARPROC lpParaDlgProc;
void DrawGraph(HDC);

/ * * * * *
/ * WinMain() * /
/ * * * * *

int PASCAL WinMain(HANDLE hInstance, HANDLE hPrevInstance,
                  LPSTR lpszCmdLine, int nCmdShow)
{
    WNDCLASS wclass;
    MSG msg;
    HWND hWnd;
    char szName[] = "Calendar";
    short xScreen , yScreen ;

    if (! hPrevInstance)
    {
        wclass.style = CS_HREDRAW | CS_VREDRAW ;
        wclass.lpfnWndProc = MainWndProc;
        wclass.cbClsExtra = 0;
        wclass.cbWndExtra = 0;
    }

```

```

wclass.hInstance      = hInstance;
wclass.hIcon          = LoadIcon(NULL, IDI_APPLICATION);
wclass.hCursor        = LoadCursor(NULL, IDC_ARROW);
wclass.hbrBackground  = GetStockObject(WHITE_BRUSH);
wclass.lpszMenuName    = szName;
wclass.lpszClassName  = szName;
if (! RegisterClass (&wclass))
    return (FALSE);
}

xScreen = GetSystemMetrics(SM_CXSCREEN);
yScreen = GetSystemMetrics(SM_CYSCREEN);

hWnd = CreateWindow(
    szName,
    "Calendar",
    WS_OVERLAPPEDWINDOW | WS_VSCROLL,
    0,
    0,
    xScreen,
    yScreen,
    NULL,
    NULL,
    hInstance,
    NULL);

if (! hWnd)
    return (FALSE);

ShowWindow(hWnd, nCmdShow);
UpdateWindow(hWnd);

while (GetMessage(&msg, NULL, NULL, NULL))
{
    TranslateMessage(&msg);
    DispatchMessage(&msg);
}

return (msg.wParam);
}

/*****
/ *                               * /
/*****

```

```

long FAR PASCAL MainWndProc(HWND hWnd, unsigned message,
                             WORD wParam, LONG lParam)
{
    HDC          hDC;
    HMENU         hMenu;
    PAINTSTRUCT ps;
    static int     GraphID;
    RECT          Rect;
    static HANDLE  hInst;
    TEXTMETRIC    tm;
    switch (message)
    {
        case WM_CREATE;
            Year = 1994;
            Month = 0;
            hDC = GetDC(hWnd);
            GetTextMetrics(hDC,&tm);
            xChar = tm.tmAveCharWidth;
            yChar = tm.tmHeight;
            ReleaseDC(hWnd,hDC);
            SetScrollRange(hWnd,SB_VERT,0,30,FALSE);
            SetScrollPos(hWnd,SB_VERT,nVscrollPos,TRUE);
            hInst = ((LPCREATESTRUCT)lParam)->hInstance;
            return (0);

        case WM_COMMAND ;
            hMenu = GetMenu(hWnd);
            switch (wParam)
            {
                case IDM_INPUT ;
                    lpParaDlgProc = MakeProcInstance((FARPROC)ParaDlgProc,hInst);

                    DialogBox(hInst,"ParaDLG",hWnd,lpParaDlgProc);

                    FreeProcInstance(lpParaDlgProc);
                    InvalidateRect(hWnd, NULL, TRUE);
                    UpdateWindow(hWnd) ;
                    break;
                case IDM_EXIT ;
                    PostQuitMessage(0) ;
                    break ;
            }
    }
}

```

```

    }
    return (0);

case WM_SIZE :
    Client.x = LOWORD(lParam);
    Client.y = HIWORD(lParam);
    return (0);

case WM_PAINT :
    hDC = BeginPaint(hWnd, &ps);

    DrawGraph(hDC);

    EndPaint(hWnd, &ps);
    return (0);

case WM_VSCROLL :
    switch(wParam)
    {
        case SB_LINEUP :
            nVscrollPos -= 1 ;
            break ;
        case SB_LINEDOWN :
            nVscrollPos += 1 ;
            break ;
        case SB_PAGEUP :
            nVscrollPos -= Client.y/yChar ;
            break ;
        case SB_PAGEDOWN :
            nVscrollPos += (Client.y/yChar) ;
            break ;
        case SB_THUMBPOSITION :
            nVscrollPos = LOWORD(lParam) ;
            break ;
        default :
            break ;
    }

    nVscrollPos = max(0,min(nVscrollPos,30)) ;
    if(nVscrollPos != GetScrollPos(hWnd,SB_VERT))
    {
        SetScrollPos(hWnd,SB_VERT,nVscrollPos,TRUE) ;
        InvalidateRect(hWnd,NULL,TRUE) ;
    }

```

```
    }
    break ;

case WM_DESTROY :
    PostQuitMessage(0);
    return (0);
default :
    return(DefWindowProc(hWnd, message, wParam, lParam));
}
}

BOOL FAR PASCAL ParaDlgProc(HWND hDlg, unsigned msg,
                             WORD wParam, LONG lParam)
{
    BOOL btran1, btran2, btran3, btran4;
    int x1, y1, x2, y2;

    switch (msg)
    {
        case WM_INITDIALOG :
            SetDlgItemInt(hDlg, DI_LLEFT, Year, TRUE);
            SetDlgItemInt(hDlg, DI_LRIGHT, Month, TRUE);
            return (TRUE);

        case WM_COMMAND :
            switch (wParam)
            {
                // 获得输入的年代和月份
                case DI_LOK :
                    x1 = GetDlgItemInt(hDlg, DI_LLEFT, &btran1, TRUE);
                    x2 = GetDlgItemInt(hDlg, DI_LRIGHT, &btran3, TRUE);

                    if (btran1 && btran2 && btran3 && btran4)
                    {
                        Year = x1 ;
                        Month = x2 ;
                    }

                    EndDialog(hDlg, TRUE);
                    return (TRUE);

                case DI_LCANCEL :
```

```

        EndDialog(hDlg, FALSE);
        return (TRUE);
    }

    return (TRUE);

default :
    return (FALSE);
}

}

void DrawGraph(HDC hDC)
{
    short nLength ;

    for(m = 0 ; m < 12 ; m++)
        for(j = 0 ; j < 6 ; j++)
            for(i = 0 ; i < 7 ; i++)
                md[m][j][i] = 0 ;

    // 求出该年代的第一天为星期几
    t = Year * 365 + Year/4 - Year/100 + Year/400 ;
    w = t/7 - t/7 ;
    // 算出该年代是否为闰年
    // 能被 4 整除且不为 100 整除即为闰年
    // 能被 400 整除的为闰年
    i = fabs(w * 7.) + .5 + 1 ;
    if(Year/4. == Year/4 && Year/100. != Year/100)
        dd[1] = 29 ;
    if(Year/100. == Year/100 && Year/400. == Year/400)
        dd[1] = 29 ;
    m = 0 ; j = 0 ; date = 1 ;
    if(i == 7) i = 0 ;
    lab1 :
        md[m][j][i] = date ;
        i++ ;
        if(i >= 7)
        {
            i = 0 ;
            j++ ;
        }
        date++ ;
        if(date > dd[m])
        {

```

```

    date = 1;
    m++;
    j = 0;
}
if(m<12) goto lab1;

if(Month!=0)
{
    SetTextColor(hDC,RGB(255,0,0));
    TextOut(hDC,xChar*32,yChar*2,szBuffer,sprintf(szBuffer,"%d",Year));
    SetTextColor(hDC,RGB(125,0,125));
    TextOut(hDC,xChar*40,yChar*2,szBuffer,sprintf(szBuffer,"%s",mm[Month-1]));
    SetTextColor(hDC,RGB(0,0,255));
    TextOut(hDC,xChar*29,yChar*4,szBuffer,sprintf(szBuffer,"%s",MSG2));
    MoveTo(hDC,xChar*24,yChar*6);
    LineTo(hDC,xChar*60,yChar*6);
    SetTextColor(hDC,RGB(0,0,0));
    for(i=0;i<=5;i++)
    {
        nLength = sprintf(szBuffer,"%29c",' ');
        for(j=0;j<=6;j++)
            if(md[Month-1][i][j]==0)
                TextOut(hDC,xChar*(nLength+4*j),yChar*(7+i),szBuffer,sprintf(szBuffer,"  "));
            else
                TextOut(hDC,xChar*(nLength+4*j),yChar*(7+i),szBuffer,sprintf(szBuffer,
                    "%3d",md[Month-1][i][j]));
    }
}

if(Month==0)
{
    for(m=0;m<=11;m+=3)
    {
        SetTextColor(hDC,RGB(255,0,0));
        TextOut(hDC,xChar,yChar*(4*m-nVscrollPos),szBuffer,sprintf(szBuffer,"%d",Year));
        SetTextColor(hDC,RGB(125,0,125));
        TextOut(hDC,xChar*10,yChar*(4*m-nVscrollPos),szBuffer,sprintf(szBuffer,"%s",
            mm[m]));
        TextOut(hDC,xChar*40,yChar*(4*m-nVscrollPos),szBuffer,sprintf(szBuffer,"%s",
            mm[m+1]));
        TextOut(hDC,xChar*70,yChar*(4*m-nVscrollPos),szBuffer,sprintf(szBuffer,"%s",
            mm[m+2]));
        SetTextColor(hDC,RGB(0,0,255));
    }
}

```

```

TextOut(hDC,xChar,yChar*(4*m-nVscrollPos+2),szBuffer,sprintf(szBuffer,"%s",
MSG2));
TextOut(hDC,xChar*30,yChar*(4*m-nVscrollPos+2),szBuffer,sprintf(szBuffer,"%s",
MSG2));
TextOut(hDC,xChar*60,yChar*(4*m-nVscrollPos+2),szBuffer,sprintf(szBuffer,"%s",
MSG2));

MoveTo(hDC,xChar,yChar*(4*m-nVscrollPos+3));
LineTo(hDC,Client.x,yChar*(4*m-nVscrollPos+3));

SetTextColor(hDC,RGB(0,0,0));
for(i=0;i<=5;i++)
{
    for(j=0;j<=6;j++)
    {
        if(md[m][i][j] == 0)
            TextOut(hDC,xChar*4*j,yChar*(4*m-nVscrollPos+4+i)," ",3);
        else
            TextOut(hDC,xChar*4*j,yChar*(4*m-nVscrollPos+4+i),szBuffer,
                sprintf(szBuffer,"%3d",md[m][i][j]));
    }

    for(j = 0 ; j<=6 ; j++)
        if(md[m+1][i][j] == 0)
            TextOut(hDC,xChar*(4*j+30),yChar*(4*m-nVscrollPos+4+i)," ",3);
        else
            TextOut(hDC,xChar*(4*j+30),yChar*(4*m-nVscrollPos+4+i),szBuffer,
                sprintf(szBuffer,"%3d",md[m+1][i][j]));

    for(j=0;j<=6;j++)
        if(md[m+2][i][j] == 0)
            TextOut(hDC,xChar*(4*j+60),yChar*(4*m-nVscrollPos+4+i)," ",3);
        else
            TextOut(hDC,xChar*(4*j+60),yChar*(4*m-nVscrollPos+4+i),szBuffer,
                sprintf(szBuffer,"%3d",md[m+2][i][j]));

    // end for i
} // end for m
} // end for if month

return;
}

```

1.2 关键路径

关键路径是一个比较典型的图论问题。

这里随便举两个应用的例子。

第一,在一个偏僻地区,有一批盛产不同土特产的村镇,某公司承担开发任务,先召集一批汽车司机研究如何修路,使这些村镇通过公路互相连通又使总路程最短。

设有 n 个村镇 $t_1, t_2, t_3, \dots$, 任意两村镇间距离为 $A[i][j]$, 如果打算修 $n-1$ 条道路, 将这 n 个村镇沟通, 问怎样修才能使道路的总长度最短?

第二, 可以通过 AOE 网来估算整个工程完成的时间。可以通过该网知道:

1) 完成整项工程至少需要多少时间。

2) 哪些活动是影响工程进度的关键。

本程序通过把所有的结点的数据放入数据文件 AOE.DAT 中, 其中, 输入的第一组数据, start 为起点, sink 为终点, 以后的每一组数据即为起点、终点、长度。在 AOE.DAT 中, 我们共输入了 9 个结点。

程序利用这些数据求出 1)~9) 的所有路径, 将所有路径以 9×9 的矩阵表示出来, 再寻找所有路径中长度最大者, 即为关键路径(至少一条, 可能不止一条)。

项目文件 AOE.PRJ 包括文件

AOE.DEF AOE.C

经编译连接运行后, 屏幕即可显示出 9×9 矩阵和关键路径。

用户可自行修改 AOE.DAT 文件。

下面是程序运行结果(图 1.2)。

CRITICAL PATH									
to from\	1	2	3	4	5	6	7	8	9
1	0	6	4	5					
2		6			1				
3			4		1				
4				5		2			
5					7		9	7	
6						7		4	
7							16		2
8								14	4
9									18

Critical Path[s]

Path 1: [1] -6-> [2] -1-> [5] -9-> [7] -2-> [9]

Path 2: [1] -6-> [2] -1-> [5] -7-> [8] -4-> [9]

There are 2 critical paths in the input file

That is all ! I can not find any more Bye

图 1.2