万水 课程设计丛书

C语言

课程设计案例精编

郭翠英 等编著

中国水利水电出版社
www.waterpub.com.cn

万水课程设计丛书

C 语言课程设计案例精编

郭翠英　等编著

中国水利水电出版社

内 容 提 要

C 语言由于具有灵活、高效、可移植性好等诸多优点，成为软件开发中常用的计算机编程语言之一。全书介绍了贪吃蛇游戏、计算器、黑白棋游戏、迷宫问题、扫地雷游戏、速算 24、数据结构 CAI 演示、进程调度、存储管理等十四个案例，各个案例独具特色。覆盖了 C 语言的基本知识点和各种数据结构，如堆栈、队列、链表等，综合应用了光带菜单、下拉菜单、图形设计、鼠标应用等知识。本书应用性极强，读者可以根据这些案例进行研究、修改和扩展。

本书适合作为高等院校、高职高专各专业学生进行 C 语言、数据结构、课程设计的参考用书，也可供在校教师以及相关工程技术人员参考使用。

图书在版编目（CIP）数据

C 语言课程设计案例精编 / 郭翠英等编著. —北京：中国水利水电出版社，2004.3

（万水课程设计丛书）

ISBN 7-5084-2032-2

Ⅰ. C… Ⅱ. 郭… Ⅲ. C 语言－程序设计－高等学校－教材 Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2004）第 012753 号

书　　名	C 语言课程设计案例精编
作　　者	郭翠英　等编著
出版　发行	中国水利水电出版社（北京市三里河路 6 号　100044） 网址：www.waterpub.com.cn E-mail：mchannel@263.net（万水） sales@waterpub.com.cn 电话：（010）63202266（总机）　68331835（营销中心）　82562819（万水）
经　　售	全国各地新华书店和相关出版物销售网点
排　　版	北京万水电子信息有限公司
印　　刷	北京市天竺颖华印刷厂
规　　格	787mm×1092mm　16 开本　17.5 印张　399 千字
版　　次	2004 年 3 月第 1 版　2004 年 3 月第 1 次印刷
印　　数	0001—5000 册
定　　价	25.00 元

前　　言

C 语言由于其强大的功能，丰富的表达能力，高效的代码，良好的移植性和灵活性，自 1972 年问世以来，经久不衰，即使现在出现了面向对象的程序设计方法和开发工具，但它仍然是人们学习程序设计的首选语言，用它训练和提高编程技术，以及开发应用程序。

正如一句行话所说“C 语言入门容易得道难”，对于多数学生来说，在学习了 C 语言后，除能应付全国计算机等级考试以外，别无他用，丰富的 C 函数所知甚少，加之 DOS 环境所限，因而对 C 语言兴趣不大，为此作者集多年从事 C 语言教学和研究的工作经验，特别是在日益注重实用的今天，编制了本书，从算法、数据结构、C 语言丰富的函数以及程序设计等方面精心考虑和组织，追求典型性、完整性、实用性和趣味性，希望不仅能提高读者学习 C 语言的兴趣，更为开发程序打下坚实的基础。

每个案例的组织都考虑到 C 语言的知识点以及数据结构的关键知识点，本着由浅入深的原则，安排各个案例。14 个案例包括了数组、指针、函数、结构体、文件等 C 语言基础知识，涉及了堆栈、递归、队列、链表、排序、查找、二叉树等数据结构，让读者充分体会算法+数据结构=程序的思想。程序功能有游戏程序、数据结构演示程序，又有数据库管理程序，另外还设计了两个与操作系统知识有关的案例：进程调度和存储管理，涵盖的知识面很广。每个案例后均有留给读者完成的扩充功能。案例中使用了许多 C 函数，均在附录中加以说明举例，但这些函数也仅仅是 C 函数中的一部分，更多的 C 函数，还需读者进一步的学习，正所谓“师傅领进门，修行在个人”。

本书所有程序均在 Turbo C 2.0 环境下测试通过。

本书主要由郭翠英编写，另外参加编写的还有朱昀、路军、米丽萍。本书个别内容借鉴互联网上程序的思想，在此对 www.vcok.com 网站的作者表示感谢。

非常感谢中国水利水电出版社的石永峰先生以及出版社计算机编辑室的全体人员，他们对本书的出版给予了很好的指导和大力的支持。

由于时间仓促及作者水平有限，书中难免有错误和不妥之处，恳请广大读者批评指正。如有意见和建议，请与作者联系。

联系方式：

电话：0351-4170927，0351-4720883

E-mail：tygcying@public.ty.sx.cn

作者

2003 年 12 月

目　　录

前言

案例一　贪吃蛇游戏 1

1.1　程序功能 1

1.2　程序设计目的 1

1.3　程序设计 1

1.3.1　游戏界面 1

1.3.2　设计思路 1

1.3.3　源程序 3

1.4　小结 7

1.4.1　知识点 7

1.4.2　功能扩充 8

案例二　计算器 9

2.1　程序功能 9

2.2　程序设计目的 9

2.3　程序设计 9

2.3.1　主界面 9

2.3.2　设计思路 9

2.3.3　源程序 12

2.4　小结 19

2.4.1　知识点 19

2.4.2　功能扩充 20

案例三　黑白棋游戏 21

3.1　程序功能 21

3.2　程序设计目的 21

3.3　程序设计 21

3.3.1　游戏规则 21

3.3.2　游戏界面 21

3.3.3　设计思路 21

3.3.4　源程序 23

3.4　小结 31

3.4.1　知识点 31

3.4.2　功能扩充 32

案例四　迷宫问题 33

4.1　程序功能 33

4.2 程序设计目的.....33
4.3 程序设计.....33
4.3.1 设计界面.....33
4.3.2 设计思路.....33
4.3.3 源程序.....35
4.4 小结.....42
4.4.1 知识点.....42
4.4.2 功能扩充.....42
案例五 扫地雷游戏.....43
5.1 程序功能.....43
5.2 程序设计目的.....43
5.3 程序设计.....43
5.3.1 游戏规则.....43
5.3.2 游戏界面.....43
5.3.3 设计思路.....44
5.3.4 源程序.....46
5.4 小结.....56
5.4.1 知识点.....56
5.4.2 功能扩充.....56
案例六 速算 24.....57
6.1 程序功能.....57
6.2 程序设计目的.....57
6.3 程序设计.....57
6.3.1 数据结构.....57
6.3.2 程序运行界面.....58
6.3.3 设计思路.....58
6.3.4 源程序.....64
6.4 小结.....73
6.4.1 知识点.....73
6.4.2 功能扩充.....73
案例七 数据结构 CAI 系统.....74
7.1 程序功能.....74
7.2 程序设计目的.....74
7.3 程序设计.....74
7.3.1 栈的应用——递归算法（汉诺塔）演示.....74
7.3.2 双链表创建演示.....78
7.3.3 冒泡排序演示.....83

7.3.4 基数排序演示 86
7.3.5 二分查找演示 92
7.3.6 二叉树遍历演示 96
7.3.7 演示程序的总体设计 102
7.4 小结 109
7.4.1 知识点 109
7.4.2 功能扩充 109
案例八 进程调度 110
8.1 程序功能 110
8.2 程序设计目的 110
8.3 程序设计 110
8.3.1 设计思路 110
8.3.2 源程序 113
8.4 小结 119
8.4.1 知识点 119
8.4.2 功能扩充 119
案例九 存储管理分区分配算法 120
9.1 程序功能 120
9.2 程序设计目的 120
9.3 程序设计 120
9.3.1 设计思路 120
9.3.2 源程序 125
9.4 小结 131
9.4.1 知识点 131
9.4.2 功能扩充 132
案例十 通讯录 133
10.1 程序功能 133
10.2 程序设计目的 133
10.3 程序设计 133
10.3.1 设计思路 133
10.3.2 源程序 138
10.4 小结 146
10.4.1 知识点 146
10.4.2 功能扩充 147
案例十一 学生成绩管理 148
11.1 程序功能 148
11.2 程序设计目的 148

11.3 程序设计 148
11.3.1 设计思路 148
11.3.2 源程序 155
11.4 小结 168
11.4.1 知识点 168
11.4.2 功能扩充 169
案例十二 工资管理 170
12.1 程序功能 170
12.2 程序设计目的 170
12.3 程序设计 170
12.3.1 设计思路 170
12.3.2 源程序 181
12.4 小结 201
12.4.1 知识点 201
12.4.2 功能扩充 201
案例十三 图书借阅管理 203
13.1 程序功能 203
13.2 程序设计目的 203
13.3 程序设计 203
13.3.1 book.c 文件 203
13.3.2 bookfunction.c 文件 216
13.4 小结 228
13.4.1 知识点 228
13.4.2 功能扩充 228
案例十四 教师工作量计算 229
14.1 程序功能 229
14.2 程序设计目的 229
14.3 程序设计 229
14.3.1 数据结构 229
14.3.2 teacher.c 文件 230
14.3.3 mouse.c 鼠标文件 239
14.3.4 teacherfun.c 文件 240
14.4 小结 248
14.4.1 知识点 248
14.4.2 功能扩充 249
附录 本书所用函数说明 250
参考文献 272

案例一　贪吃蛇游戏

1.1　程序功能

贪吃蛇游戏是一个深受人们喜爱的游戏，一条蛇在密闭的围墙内，在围墙内随机出现一个食物，通过按键盘上的四个光标键控制蛇向上下左右四个方向移动，蛇头撞到食物，则表示食物被蛇吃掉，这时蛇的身体长一节，同时计 10 分，接着又出现食物，等待被蛇吃掉，如果蛇在移动过程中，撞到墙壁或身体交叉蛇头撞到自己的身体游戏结束。

1.2　程序设计目的

本程序实现的主要技巧在二维数组的应用上。目的是通过游戏程序增加编程的兴趣，提高编程水平。

1.3　程序设计

1.3.1　游戏界面

程序运行时的游戏界面如图 1-1 所示。边框表示围墙，红色矩形块代表蛇，绿色小方块代表食物。

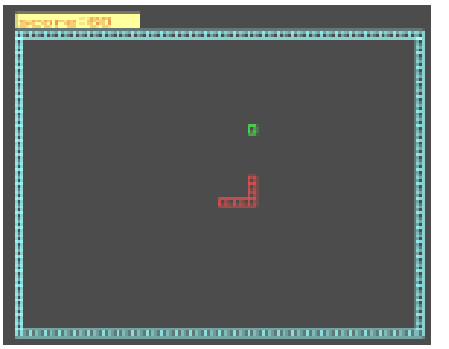

图 1-1　运行中的游戏界面

1.3.2　设计思路

这个程序的关键点是表示蛇的图形以及蛇的移动。用一个小矩形块表示蛇的一节身体，

身体每长一节，增加一个矩形块，蛇头用两节表示。移动时必须从蛇头开始，所以蛇不能向相反方向移动，也就是蛇尾不能改作蛇头。如果不按任何键，蛇自行在当前方向上前移，当游戏者按了有效的方向键后，蛇头朝着指定的方向移动，一步移动一节身体，所以当按了有效的方向键后，先确定蛇头的位置，然后蛇身体随着蛇头移动，图形的实现是从蛇头的新位置开始画出蛇，这时，由于没有清屏的原因，原来蛇的位置和新蛇的位置差一个单位，所以看起来蛇会多一节身体，所以将蛇的最后一节用背景色覆盖。食物的出现和消失也是画矩形块和覆盖矩形块。为了便于理解，定义了两个结构体：食物和蛇。

1. 数据结构

表示食物和蛇的矩形块都设计为 10×10 个像素单位，食物的基本数据域为它所出现的位置，用 x 和 y 坐标表示，则矩形块用函数 rectangle(x,y,x+10,y+10)或 rectangle(x,y,x+10,y-10)可以画出。由于每次只出现一个食物，而食物被吃掉后，才出现下一个食物，所以设定 yes 表示是否要出现食物的变量。蛇的一节身体为一个矩形块，这样表示每个矩形块只需起点坐标 x 和 y。身体是不断增长的，所以用数组存放每一节的坐标，最大设定为 N=200，node 表示当前节数。另外还需要保存蛇移动方向的变量 direction 和表示生命的变量 life，一但 life 为 1，则蛇死，游戏结束。所以程序功能的实现就是数组的操作。

```
#define N 200
struct Food
{
   int x;     /*食物的横坐标*/
   int y;     /*食物的纵坐标*/
   int yes;  /*判断是否要出现食物的变量*/
}food;      /*食物的结构体*/
struct Snake
{
   int x[N]; /*蛇的横坐标*/
   int y[N]; /*蛇的纵坐标*/
   int node;/*蛇的节数*/
   int direction;/*蛇的移动方向*/
   int life;/*蛇生命，0 活着，1 死亡*/
}snake;
```

2. main()主函数

主函数是程序的主流程，首先定义使用到的常数、全局变量及函数原型说明，然后初始化图形系统，调用函数 DrawK()画出开始画面，调用函数 GamePlay()，即玩游戏的具体过程，游戏结束后调用 Close()关闭图形系统，结束程序。

3. 画界面函数 DrawK()

主界面就是一个密封的围墙，用两个循环语句分别在水平方向和垂直方向输出连续的宽度和高度均为 10 个单位的矩形小方块，围成密闭图形，表示围墙，为了醒目，设置为淡青颜色，用函数 setlinestyle(SOLID_LINE,0,THICK_WIDTH)设置线型宽度为 3 个像素。设置 3 个像素的围墙线，蛇在贴墙走的时候，会擦掉部分围墙线，使线变细，图形变得不好看，如果不想这种情况出现，则将线型宽度设置为 1 个像素。

4. 游戏具体过程函数 GamePlay()

这个函数是游戏的主要部分，难点在表示蛇的新位置并消除前一次的图形。采用的方法是每次移动的时候从最后一节开始到倒数第二节（因蛇头为两节），将前一节的坐标赋值给后一节的坐标，移动后只要把最后一节用背景色去除即可，因为新位置 0 到 n-1 节还是要出现在画面上的。然后蛇头按照方向键来更改位置。

另外，食物的随机出现要确保它的位置在 10 的倍数位置上，因为蛇的坐标都是以 10 为模的，这样的话就可以让蛇吃到，蛇吃到食物的判断是蛇头的坐标和食物的坐标相等。

其算法过程为：

（1）设置初始值。为防止食物出现在一个位置上，要设置随机数发生器，真正产生随机数。初始时，蛇只有蛇头，设定一个开始的方向。

（2）循环执行，直到按 Esc 键退出。

1）没有按键的情况下，循环执行。

如果没有食物，随机出现食物；如果有食物，则显示食物，蛇移动身体，根据蛇的方向改变坐标值，并判断蛇是否撞到了墙或自己吃了自己，如果出现这两种情况之一，则蛇死，调用游戏结束函数 GameOver()，结束本次游戏，重新开始。

如果蛇吃到了食物，蛇身体长一节，数组元素增加一个，身体节数、分数都进行相应的改变。

在新位置画出蛇。

2）如果有按键，则识别键值。如果按键为 Esc 键则结束游戏，程序运行结束；如果所按键为方向键，则根据该键改变代表蛇方向的变量 direction 的值，要考虑相反方向键无效。

5. 游戏结束函数 GameOver()

游戏结束，清除屏幕，输出分数，显示游戏结束信息。

6. PrScore()输出分数

在指定位置利用 sprintf()将整数转换为字符串，用 outtextxy()输出，bar()函数的应用是为了覆盖原来的值。

7. Close()图形结束

在显示游戏结束信息的画面时，按任意键关闭图形系统，程序结束。

1.3.3　源程序

```
#define N 200
#include <graphics.h>
#include <stdlib.h>
#include <dos.h>
#define LEFT 0x4b00
#define RIGHT 0x4d00
#define DOWN 0x5000
#define UP 0x4800
#define Esc 0x011b
int i,key;
```

```
int score=0;/*得分*/
int gamespeed=50000;/*游戏速度可以自己调整*/
struct Food
{
   int x;/*食物的横坐标*/
   int y;/*食物的纵坐标*/
   int yes;/*判断是否要出现食物的变量*/
}food;/*食物的结构体*/
struct Snake
{
   int x[N];
   int y[N];
   int node;/*蛇的节数*/
   int direction;/*蛇的移动方向*/
   int life;/* 蛇的生命，0 活着，1 死亡*/
}snake;
void Init(void);/*图形驱动*/
void Close(void);/*图形结束*/
void DrawK(void);/*开始画面*/
void GameOver(void);/*结束游戏*/
void GamePlay(void);/*玩游戏的具体过程*/
void PrScore(void);/*输出成绩*/
/*主函数*/
void main(void)
{
   Init();/*图形驱动*/
   DrawK();/*开始画面*/
   GamePlay();/*玩游戏的具体过程*/
   Close();/*图形结束*/
}
/*图形驱动*/
void Init(void)
{
   int gd=DETECT,gm;
   initgraph(&gd,&gm,"c:\\tc");
   cleardevice();
}
/*开始画面，左上角坐标为（50，40），右下角坐标为（610，460）的围墙*/
void DrawK(void)
{
/*setbkcolor(LIGHTGREEN);*/
   setcolor(11);
   setlinestyle(SOLID_LINE,0,THICK_WIDTH);/*设置线型*/
   for(i=50;i<=600;i+=10)/*画围墙*/
   {
      rectangle(i,40,i+10,49); /*上边*/
      rectangle(i,451,i+10,460);/*下边*/
```

```
   }
  for(i=40;i<=450;i+=10)
  {
    rectangle(50,i,59,i+10); /*左边*/
    rectangle(601,i,610,i+10);/*右边*/
  }
}
/*玩游戏的具体过程*/
void GamePlay(void)
{
   randomize();/*随机数发生器*/
   food.yes=1;/*1 表示需要出现新食物，0 表示已经存在食物*/
   snake.life=0;/*活着*/
   snake.direction=1;/*方向往右*/
   snake.x[0]=100;snake.y[0]=100;/*蛇头*/
   snake.x[1]=110;snake.y[1]=100;
   snake.node=2;/*节数*/
   PrScore();/*输出得分*/
   while(1)/*可以重复玩游戏，按 Esc 键结束*/
   {
      while(!kbhit())/*在没有按键的情况下，蛇自己移动身体*/
      {
            if(food.yes==1)/*需要出现新食物*/
            {
              food.x=rand()%400+60;
              food.y=rand()%350+60;
              while(food.x%10!=0)/*食物随机出现后必须让食物能够在整格内，这样才可
                                   以让蛇吃到*/
                  food.x++;
              while(food.y%10!=0)
                  food.y++;
               food.yes=0;/*画面上有食物了*/
            }
          if(food.yes==0)/*画面上有食物了就要显示*/
          {
         setcolor(GREEN);
         rectangle(food.x,food.y,food.x+10,food.y-10);
           }
       for(i=snake.node-1;i>0;i--)
/*蛇的每个环节往前移动，也就是贪吃蛇的关键算法*/
       {
          snake.x[i]=snake.x[i-1];
              snake.y[i]=snake.y[i-1];
          }
      /*1,2,3,4 表示右、左、上、下四个方向、通过这个判断来移动蛇头*/
     switch(snake.direction)
       {
```

```
            case 1:snake.x[0]+=10;break;
            case 2: snake.x[0]-=10;break;
            case 3: snake.y[0]-=10;break;
            case 4: snake.y[0]+=10;break;
          }
         for(i=3;i<snake.node;i++)/*从蛇的第四节开始判断是否撞到自己了，因为蛇头为
                                    两节，第三节不可能拐过来*/
         {
          if(snake.x[i]==snake.x[0]&&snake.y[i]==snake.y[0])
           {
             GameOver();/*显示失败*/
             snake.life=1;
             break;
           }
      }
          if(snake.x[0]<55||snake.x[0]>595||snake.y[0]<55||
         snake.y[0]>455)/*蛇是否撞到墙壁*/
          {
            GameOver();/*本次游戏结束*/
            snake.life=1; /*蛇死*/
         }
         if(snake.life==1)/*以上两种判断以后，如果蛇死就跳出内循环，重新开始*/
          break;
         if(snake.x[0]==food.x&&snake.y[0]==food.y)/*吃到食物以后*/
     {
          setcolor(0);/*把画面上的食物去掉*/
          rectangle(food.x,food.y,food.x+10,food.y-10);
          snake.x[snake.node]=-20;
          snake.y[snake.node]=-20;
           /*新的一节先放在看不见的位置，下次循环就取前一节的位置*/
             snake.node++;/*蛇的身体长一节*/
           food.yes=1;/*画面上需要出现新的食物*/
           score+=10;
           PrScore();/*输出新得分*/
          }
        setcolor(4);/*画出蛇*/
        for(i=0;i<snake.node;i++)
          rectangle(snake.x[i],snake.y[i],snake.x[i]+10,snake.y[i]-10);
        delay(gamespeed);
        setcolor(0);/*用黑色去除蛇的最后一节*/
        rectangle(snake.x[snake.node-1],snake.y[snake.node-1],
        snake.x[snake.node-1]+10,snake.y[snake.node-1]-10);
} /*endwhile（! kbhit）*/
if(snake.life==1)/*如果蛇死就跳出循环*/
    break;
key=bioskey(0);/*接收按键*/
if(key==Esc)/*按 Esc 键退出*/
```

```
      break;
  else  if(key==UP&&snake.direction!=4)
/*判断是否往相反的方向移动*/
         snake.direction=3;
  else  if(key==RIGHT&&snake.direction!=2)
        snake.direction=1;
      else  if(key==LEFT&&snake.direction!=1)
         snake.direction=2;
      else if(key==DOWN&&snake.direction!=3)
           snake.direction=4;
   }/*endwhile(1)*/
}
/*游戏结束*/
void GameOver(void)
{
    cleardevice();
    PrScore();
    setcolor(RED);
    settextstyle(0,0,4);
    outtextxy(200,200,"GAME OVER");
    getch();
}
/*输出成绩*/
void PrScore(void)
{
   char str[10];
   setfillstyle(SOLID_FILL,YELLOW);
   bar(50,15,220,35);
   setcolor(6);
   settextstyle(0,0,2);
   sprintf(str,"score:%d",score);
   outtextxy(55,20,str);
}
/*图形结束*/
void Close(void)
{
    getch();
    closegraph();
}
```

1.4　小结

1.4.1　知识点

（1）**数组元素为结构体的数组应用**。

（2）全局变量应用。

（3）按键的处理。

（4）数组元素与蛇、食物的对应关系。

（5）图形方式。

1.4.2 功能扩充

（1）本程序利用 delay()控制蛇移动的速度。请将程序修改为：一但分数达到某个分值，则加快速度，提高游戏者的等级。

（2）编制程序实现弹球游戏，随机出现小球，移动一根棒子，棒子击中小球加分，并记录积分最高者的信息。

案例二　计算器

2.1　程序功能

计算器是 Windows 操作系统提供的一个附件功能，许多人用 Visual Basic、Visual C++ 等编制计算器，由于这些程序提供了控件，所以实现相对容易。Turbo C 没有控件，但我们可以利用它所具有的函数模仿画出其界面，实现计算器的基本功能，进行浮点数加、减、乘、除、乘方和求模运算。

2.2　程序设计目的

通过本程序训练读者程序设计的基本技能，掌握字符串的表示方法和字符串函数的功能、Turbo C 图形操作的基本知识、键盘上特殊键的获取以及图形方式下光标的显示。

2.3　程序设计

实现计算器功能，首先是输出计算器图形样式，然后是通过按键的方式实现数值运算。所以程序主要由两大部分实现，一部分功能是显示计算器，一部分是实现计算功能。显示计算器是在屏幕上显示图形，Turbo C 提供了 PC 系统环境下扩充的屏幕和图形支持系统，利用此系统提供的字符屏幕处理函数和图形系统的有关信息及函数即可很好地实现。计算功能主要要解决的是接收按键信息的处理，要进行识别，如果按键是数字符号，要将其转变为操作数，如果是运算操作符，则进行相应的处理。

2.3.1　主界面

首先是画一个带标题（calculator）的窗口，窗口为一白色边框，窗口中最上面是一个绿色光条，此光条上显示数字和运算结果，光条下有 20 个红色边框的灰色矩形块模拟命令按钮，按钮上面显示有数字和运算符号，并且在按钮上有图形光标，通过移动光标键移动到所需位置，按回车键即选择了相应的符号。主界面如图 2-1 所示。光标移动到字符 Q 上，按回车键结束运算。

2.3.2　设计思路

1. main()主函数

设置了程序的流程，首先初始化图形系统，然后调用计算器 computer()函数进行计算，