

## 贪吃蛇小游戏笔记

笔记本: C语言小项目—Snack

创建时间: 2023/1/3/星期二 下午 9:14

更新时间: 2023/2/10/星期五 下午 4:57

作者: 152xpkbw578

URL: <https://www.runoob.com/cprogramming/c-standard-library-time-h.html>

---

### 1.C 头文件

只能被引用一次

头文件就是扩展名为.h的文件, 定义常量、全局变量、声明函数

两种类型:

#### 1) 自己编写的头文件

```
#include "fileName"
```

#### 2) 编译器自带的头文件

```
#include <fileName>
```

### 2.C 预处理器

```
#define PI 3.14 //定义一个常量

//如果一个宏没有定义, 返回一个真
//包装器
#ifndef HEADER_FILE
#define HEADER_FILE
    header file
#endif
```

### 3.time.h头文件

定义了四个变量类型、两个宏和各种操作日期和时间的函数。

```
struct tm
这是一个用来保存时间和日期的结构。

tm 结构的定义如下:
struct tm {
    int tm_sec;        /* 秒, 范围从 0 到 59 */
    int tm_min;        /* 分, 范围从 0 到 59 */
    int tm_hour;       /* 小时, 范围从 0 到 23 */
    int tm_mday;       /* 一月中的第几天, 范围从 1 到 31 */
    int tm_mon;        /* 月, 范围从 0 到 11 */
    int tm_year;       /* 自 1900 年起的年数 */
    int tm_wday;       /* 一周中的第几天, 范围从 0 到 6 */
    int tm_yday;       /* 一年中的第几天, 范围从 0 到 365 */
    int tm_isdst;      /* 夏令时 */
};
```

## 4.引入的头文件

```
#define _CRT_SECURE_NO_WARNINGS 1
//stdio .h 头文件定义了三个变量类型、一些宏和各种函数来
执行输入和输出。
#include<stdio.h>

//time.h 头文件定义了四个变量类型、两个宏和各种操作日期
和时间的函数。
#include<time.h>
#include<Windows.h>

//stdlib .h 头文件定义了四个变量类型、一些宏和各种通用工
具函数。
#include<stdlib.h>

/**conio是Console Input/Output(控制台输入输出)的简写，
其中定义了通过控制台进行数据输入和数据输出的函数，
主要是一些用户通过按键盘产生的对应操作，比如getch()函数
等等。*/
#include<conio.h>  //接收键盘输入输出
```

## 5、函数实现

```
/**
*1、设置光标位置
*/

/**
*2、更改文字颜色
*/

/**
*3、开始界面
*/

/**
*4、绘制地图
*/

/**
*5、游戏界面右侧的得分和小提示
*/
```

/\*\*

\*6、等级展示

\*/

/\*\*

\*7、初始化蛇身

\*/

/\*\*

\*8、创建并随机出现食物

\*/

/\*\*

\*9、判断时候咬到自己

\*/

/\*\*

\*10、设置蛇撞墙的情况

\*/

/\*\*

\*11、按F1加速

\*/

/\*\*

\*12、按F2减速

\*/

/\*\*

\*13、控制蛇前进方向

\*/

/\*\*

\*14、控制键盘按键

\*/

/\*\*

\*15、绘制游戏失败界面

\*/

/\*\*

\*16、游戏结束

\*/

/\*\*

```
*17、游戏结束之后的选择
```

```
*/
```

```
/**
```

```
*18、游戏说明
```

```
*/
```

## 6、句柄

Windows系统中有许多内核对象。比如打开的文件，创建的线程，程序的窗口，等等。这些重要的对象肯定不是4个字节或者8个字节足以完全描述的，他们拥有大量的属性。为了保存这样一个"对象"的状态，往往需要上百甚至上千字节的内存空间，那么怎么在程序间或程序内部的子过程（函数）之间传递这些数据呢？

简单理解在Windows系统中，可以通过"句柄"区操作这些对象，使用它们的属性。

Handle本身是一个32位的无符号整数，它用来代表一个内核对象。它并不指向实际的内核对象，用户模式下的程序永远不可能获得一个内核对象的实际地址（一般情况下）。那么Handle的意义何在？它实际上是作为一个索引在一个表中查找对应的内核对象的实际地址。那么这个表在哪里呢？每个进程都有这样的一个表，叫句柄表。该表的第一项就是进程自己的句柄，这也是为什么你调用GetCurrentProcess()总是返回0x7FFFFFFF原因。

简单地说，Handle就是一种用来"间接"代表一个内核对象的整数值。你可以在程序中使用handle来代表你想要操作的内核对象。这里的内核对象包括：事件（Event）、线程、进程、Mutex等等。我们最常见的就是文件句柄（file handle）。

## 7、示例程序

```
#include <stdio.h>
#include <windows.h>
int main(void)
{
    HANDLE consolehwnd;//创建句柄，详细句柄知识，请百度一下或查MSDN
    consolehwnd = GetStdHandle(STD_OUTPUT_HANDLE);//实例化句柄
    SetConsoleTextAttribute(consolehwnd, FOREGROUND_RED);//设置字体颜色
    printf("hello");
```

```

        SetConsoleTextAttribute(consolehwnd, FOREGROUND_INTENSITY
| FOREGROUND_GREEN);
        printf("world!\n");
        getchar();
        SetConsoleTextAttribute(consolehwnd, BACKGROUND_INTENSITY
| BACKGROUND_BLUE);
        printf("It is really beautiful!\n");
        return 0;
}

```

## 8.打印类函数

### 4.绘制地图

```

void createMap()
{
    int i, j;
    for (i = 0; i < 58; i += 2) //打印上下边框
    {
        gotoxy(i, 0);
        color(5);
        printf("□");
        gotoxy(i, 26);
        printf("□");
    }
    for (i = 1; i < 26; i++) //打印左右边框
    {
        gotoxy(0, i);
        printf("□");
        gotoxy(56, i);
        printf("□");
    }
    for (i = 2; i < 56; i += 2)
    {
        for (j = 1; j < 26; j++)
        {
            gotoxy(i, j);
            color(3);
            printf("■");
        }
    }
}

```

### 5.小提示

```

void scoreAndTips()
{
    gotoxy(74, 2);
}

```

```

    color(12);
    printf("得分: %d", score);
    gotoxy(74, 4);
    color(12);
    levelChange();

    gotoxy(74, 8);
    color(14);
    printf("每个食物得分: %d分!", add);
    gotoxy(74, 10);
    printf("不能碰墙, 不能咬到自己!");
    gotoxy(74, 12);
    printf("用↑↓↔分别控制蛇的移动!");
    gotoxy(74, 14);
    printf("F1为加速, F2为减速!");
    gotoxy(74, 16);
    printf("space: 暂停游戏!");

    gotoxy(74, 18);
    printf("ESC: 退出游戏!");

    time_t timep;
    struct tm *p;
    time (&timep);
    p=gmtime(&timep);

    gotoxy(74, 22);
    color(15);
    printf("Date: %d/%d/%d\n", 1900+p->tm_year, 1+p-
>tm_mon, p->tm_mday);
    gotoxy(74, 24);
    printf("Time :%d:%d:%d", 8+p->tm_hour, p-
>tm_min, p->tm_sec);

    gotoxy(74, 26);
    printf("由学以致知提供技术支持!");
}

```

## 6. 等级展示

```

void levelChange()
{
    if(score >= 0 && score < 50) level = 1;
    else if(score >= 50 && score < 200) level = 2;
}

```

```

else if(score >= 200 && score < 500) level = 3;
else if(score >= 500 && score < 1000) level = 4;
else level = 5;

switch(level)
{
    case 1 : printf("等级: 坚韧黑铁"); break;
    case 2 : printf("等级: 英勇黄铜"); break;
    case 3 : printf("等级: 不屈白银"); break;
    case 4 : printf("等级: 荣耀黄金"); break;
    case 5 : printf("等级: 华贵铂金"); break;
    default : break;
}
}

```

### 15. 绘制游戏失败界面

```

void lostDraw()
{
    system("cls");
    int i;
    gotoxy(46, 2);
    color(6);
    printf("\\\\\\\\\\|///");
    gotoxy(43, 3);
    printf("\\\\\\");
    gotoxy(47, 3);
    color(15);
    printf(".-.-");
    gotoxy(54, 3);
    color(6);
    printf("//");

    gotoxy(44, 4);
    color(14);
    printf("(");

    gotoxy(47, 4);
    color(15);
    printf(".@.@");

    gotoxy(54, 4);
    color(14);

```

```
printf(")");

gotoxy(17, 5);
color(11);
printf("+-----");

gotoxy(35, 5);
color(14);
printf("o00o");

gotoxy(39, 5);
color(11);
printf("-----");

gotoxy(48, 5);
color(14);
printf("(_)");

gotoxy(51, 5);
color(11);
printf("-----");

gotoxy(61, 5);
color(14);
printf("o00o");

gotoxy(65, 5);
color(11);
printf("-----+");

for (i = 6; i <= 19; i++)
{
    gotoxy(17, i);
    printf("|");
    gotoxy(82, i);
    printf("|");
}
```



```

gotoxy(17, 20);
printf("+-----");

gotoxy(47, 20);
color(14);
printf("☆ ☆ ☆");

gotoxy(56, 20);
color(11);
printf("-----+");
}

18.游戏说明
void exlination()
{
    system("cls");
    int i,j=1;

    color(13);
    gotoxy(44,3);
    printf("游戏说明");
    color(2);
    for (i = 6; i <= 25; i++) //输出上下边框
    {
        for (j = 20; j <= 75; j++) //输出左右边框
        {
            gotoxy(j, i);
            if (i == 6 || i == 25) printf("=");
            else if (j == 20 || j == 75)
printf("||");
        }
    }
    color(3);
    gotoxy(30, 8);
    printf("1.不能穿墙，不能咬到自己");
    color(10);
    gotoxy(30, 11);
    printf("2.分别用↑↓↔控制蛇的移动");
    color(14);
    gotoxy(30, 14);
    printf("3.小惊喜 F1可以加速\tF2可以减速");
    color(11);
    gotoxy(30, 17);

```

```

printf("4.按空格键暂停，再按继续");

color(15);
gotoxy(30, 20);
printf("5.由一蓑烟雨任平生提供技术支持");

color(4);
gotoxy(30, 23);
printf("6.按任意键返回开始界面");
_getch(); //任意键返回开始界面
system("cls");
welcomeToGame();
}

```

## 9.示例程序

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int i, n;
    time_t t;
    n = 5;

    /* 初始化随机数发生器 */
    srand((unsigned) time(&t));

    /* 输出 0 到 50 之间的 5 个随机数 */
    for( i = 0 ; i < n ; i++ )
    {
        printf("%d\n", rand() % 50);
    }
    return(0);
}

```

