　上一篇博客里，我介绍了如何用数组存储坐标，不断全屏刷新以实现动态效果，这几乎是最显而易见的思路，但带来的副作用也十分明显，那就是始终伴随着游戏的闪烁现象，之所以会造成闪烁现象，原因在于频繁的清空与打印，其实贪吃蛇游戏里我们并不需要重打印整个界面，要实现蛇的移动，我们只要打印出新的蛇头，清除原来的蛇尾就好了。食物只有在被吃掉时才需要重新打印，边界更是只用打印一次。好了，既然我们看到了可提升的地方，就开始动手优化吧。显然，我们将需要一个可以自由移动光标的函数，这样我们才能做到在需要的地方打印。TC上有一个叫gotoxy()的很方便的函数，该函数，顾名思义，就是将光标移动到(x,y)位置。然而现在估计很少有人还在使用TC，笔者使用的是vs2015，自然无法使用该函数。那么我们就着手自己编写一个gotoxy()函数。

　　先上代码：

[复制代码](javascript:void(0);)

void gotoxy(unsigned char x, unsigned char y)

{

COORD cor;

HANDLE hout;

cor.X = x;

cor.Y = y;

hout = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleCursorPosition(hout, cor);

}

[复制代码](javascript:void(0);)

如果是没有接触过windows编程的同学看到这段代码可能会有些懵逼，不过没关系。整段代码的逻辑其实十分简单。

COORD是windows API定义的结构，其声明如下：

typedef struct \_COORD {

SHORT X;

SHORT Y;

} COORD, \*PCOORD;

正如其名字coordinate（坐标）一样，这是一个存储二维坐标的结构体。

HANDLE（句柄）在windows编程中是一个十分重要的概念。在window编程中，对于一个Object(对象)我们只能通过Handle来访问它。觉得不好理解的同学把句柄当作指针来看待就好了。

GetStdHandle函数获取一个指向特定标准设备的句柄，包括标准输入，标准输出和标准错误。STD\_OUTPUT\_HANDLE正是代表标准输出（也就是显示屏）的宏。

SetConsoleCursorPosition函数用于设置控制台光标的位置。

以上的函数若有疑问都可以参阅MSDN。

有了gotoxy函数，适当地修改上一篇的代码就可以解决闪屏的问题。

[复制代码](javascript:void(0);)

int main()　　　　　　　　　　　　　　　　　//去掉print\_game函数，加入init\_game函数

{

int dir=UP;

init\_game();

while (1){

dir = get\_dir(dir);

move\_snake(dir);

Sleep(100);

if (!isalive())

break;

}

system("cls");

printf("Game Over\n");

}

[复制代码](javascript:void(0);)

[复制代码](javascript:void(0);)

void init\_game(void)

{

int i, j;

unsigned char x, y,fx,fy;

tran(snake[0], &x, &y);

tran(food, &fx, &fy);

for (i = 0; i<17; i++) {

for (j = 0; j<17; j++) {

if (i == 0 || i == 16 || j == 0 || j == 16)

putchar('#');

else if (i == y&&j == x)

putchar('\*');

else if (i == fy&&j == fx)

putchar('!');

else

putchar(' ');

}

putchar('\n');

}

}

[复制代码](javascript:void(0);)

主要修改的是snake\_move函数，代码如下：

[复制代码](javascript:void(0);)

void move\_snake(int dir)

{

int last = snake[0],current;

int i;

int grow=0;

unsigned char x, y,fx,fy;

tran(food, &fx, &fy);

tran(snake[0], &x, &y);

switch (dir){

case UP:

y--;

break;

case DOWN:

y++;

break;

case LEFT:

x--;

break;

case RIGHT:

x++;

break;

}

snake[0] = ((x ^ 0) << 4) ^ y;

if (snake[0] == food) {　　　　　　　　　　　　　　　//如果吃到食物

grow = 1;

food = generate\_food();　　　　　　　　　　　　//生成食物

}

for (i = 0; i<len; i++) {

if (i == 0)

continue;

current = snake[i];

snake[i] = last;

last = current;

}

gotoxy(x, y);

putchar('\*');　　　　　　　　　　　　　　　　　//打印新的蛇头

if (grow) {　　　　　　　　　　　　　　　　　　//如果要长节的话就不去除旧的蛇尾

snake[len] = last;

len++;  
        tran(food, &fx, &fy)；　　　　　　　　　　// 打印食物  
 gotoxy(fx,fy);  
 putchar('!');　  
 }else {　　　　　　　　　　　　　　　　　　　　　　//去除旧的蛇尾

tran(last, &x, &y);

gotoxy(x, y);

putchar(' ');

}

}

[复制代码](javascript:void(0);)

至此，我们就完成了贪吃蛇游戏局部刷新的优化，消除了闪烁现象，游戏性大大提升。

https://bbs.csdn.net/topics/391949781