# 聪明的小蛇游戏设计指导

**一、简介**

有一条聪明的小蛇正在草地上玩耍。草地上有美味的食物、有智慧草，小蛇玩得很开心。可是同时草地上也危机四伏，有地雷和会突然同时出现多个的毒草。可是小蛇非常聪明，每次它们都能逢凶化吉、遇难成祥。

1. **设计要求**
2. 在打开程序后，出现选择界面，里面包含难度（关卡）选择和退出功能，其中难度选择能够是程序出现相应的游戏难度，要求至少有三种。难度与以下因素相关：
3. 蛇的运动速度：难度越高，速度越快；
4. 地图不同：地图的边界可以不规则，难度越高，边界的变化越多；
5. 地雷和毒草的个数不同：难度越高，个数越多；
6. 开始游戏后，上下左右方向键控制小蛇的移动方向，小蛇的运动方式有两种选择。
7. 小蛇可以向上下左右移动，但是当小蛇向一个方向移动时，不能立刻向反方向移动。如小蛇正在向右移动，则必须向下或上移动一段后才能向左移动。
8. \*小蛇可以360度转动。
9. 小蛇撞到不同的东西会有不同的命运。
10. 美味的食物：吃到一个食物后，身体随之增加一个单位长度；
11. \*智慧草：智慧草每次出现一根，一段时间后消失。吃到智慧草后，小蛇无需控制可自动运动一段时间；
12. 地雷：小蛇踩到地雷会被炸掉一半，如果它当时身体的长度只有一节则会被炸死；
13. 毒草：毒草每次出现多根，\*保持闪烁状，一段时间后全部消失。小蛇吃到毒草后，身体随之减少一个单位长度。
14. 小蛇撞到墙时会死。
15. 当小蛇身体长度达到一定时，游戏胜利可进入下一关卡，如果通过了最后一关则游戏胜利。
16. 需要对游戏情况进行计分，计分规则可自行拟定。
17. 提供排行榜功能，将游戏得分记载到排行榜文件中。
18. 提供进度保存功能，即中断游戏时可保存当前游戏状态，且要求将状态保存到文件中。至少要保存进入到哪一关，得分多少。\*另外可保存当时的游戏场景。
19. \*可以将地图保存在文件中，并且让用户选择地图开始游戏。
20. 界面美观，使用方便。\*最好用图形界面。

说明：以上标\*的部分为选做部分。

1. **所需数据结构分析**

**要求蛇必须采取数据结构1，功能键选择2中所示键，其余数据结构仅供参考。**

1. **蛇数据结构**

struct snake{/\*\*\*\*\*\*\*\*\*\*\*\*\*蛇结构体\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int x;

int y;

struct snake \*previous;/\*前\*/

struct snake \*next; }\*head,\*tail;

1. **功能键的键值**

#define LEFT 0x4B00

#define RIGHT 0x4D00

#define UP 0x4800

#define DOWN 0x5000

#define ESC 0x011B

#define ENTER 0x1C0D

1. 游戏取界面数据

Char map[30][30] /\*游戏界面为30\*30\*/

#define STARTX 50 /\*表示游戏区在屏幕上的起始X坐标\*/

#define STARTY50 /\*表示游戏区在屏幕上的起始Y坐标\*/

#define SIZEX 20 /\*表示一个方块的长度\*/

#define SIZEY 20 /\*表示一个方块的宽度\*/

1. 游戏状态数据

Int gameRes /\*记录游戏的结果状态，值为0表示按ESC键退出；-1表示游戏失败；1表示游戏胜利\*/

1. **游戏流程图**

开始

图形方式初始化

初始化新游戏

从键盘读入操作信息

各类消息判断并作相应的处理

重新游戏?

结束

是

否

对“各类消息判断并作相应的处理”步骤的流程图细化：

Switch(key){

case ENTER:

case ESC:

gameRes=0;

case LEFT:

moveleft(); break;

case RIGHT:

moveright(); break;

case UP:

moveup(); break;

case DOWN:

movedown; break;

default: beeak;

}

1. **所需算法分析**
   1. 写出主程序的基本框架，确定函数各个内部模块的函数名和参数

int main()

{

initGraph(); /\*图形显示方式初始化\*/

{

Newgame(); /\*开局\*/

Int gameRes=0;

do

{

int key=getkey(); /\*读入操作信息\*/

if(key==ESC)

{

gameRes=0;break;

}

Switch(key)

{

对其他key值作相应的处理

}

If （checkwin()） /\*判断游戏是否胜利\*/

gameRes=1;

}while(!gameRes);

}while(!confirm(gameRes));

Return 0;

}

* 1. 图形初始化

void s\_detectgraph() {/\*自定义图形检测\*/

int gdriver,gmode,errorcode;

gdriver=VGA;

gmode=VGAHI;

registerbgidriver(EGAVGA\_driver);

initgraph(&gdriver,&gmode,"");

errorcode = graphresult();

if (errorcode !=0){

printf("\n\t\t\tGame : Snake Game\n");

printf("\n\terror:\t%s\n", grapherrormsg(errorcode));

getch();

exit(1);

}

}

* 1. 读入操作信息的处理

Switch(key){

case ENTER:

case ESC:

gameRes=0;

case LEFT:

moveleft(); break;

case RIGHT:

moveright(); break;

case UP:

moveup(); break;

case DOWN:

movedown; break;

default: beeak;

1. **关键函数介绍**
   1. 蛇链表

用链表的节点表示蛇，移动的时候增加一个头节点去掉一个尾节点，吃食物只需增加一个头节点。

* 1. barbox函数

void barbox(int x,int y,int color,int width,int height){/\*填充一定范围的函数\*/

setfillstyle(SOLID\_FILL,color); bar(x,y,x+width-1,y+height-1);

}

* 1. bioskey()

功 能: 直接使用BIOS服务的键盘接口

　　 函数原型：int bioskey (int cmd)

　　 说明：bioskey()的函数原型在bios.h中

　　 bioskey()完成直接键盘操作，cmd的值决定执行什么操作。

　　 cmd = 0：

　　 当cmd是0，bioskey()返回下一个在键盘键入的值（它将等待到按下一个键）。它返回一个16位的二进制数，包括两个不同的值。当按下一个普通键时，它的低8位数存放该字符的ASCII码；对于特殊键（如方向键、F1～F12等等），低8位为0，高8位字节存放该键的扫描码。

　　 cmd = 1：

　　 当cmd是1，bioskey()查询是否按下一个键，若按下一个键则返回非零值，否则返回0。

* 1. setviewport()

功 能: 为图形输出设置当前视口

　 用 法: void far setviewport(int left, int top, int right, int bottom, int clipflag);

5、settextstyle()

功 能: 为图形输出设置当前的文本属性

　　 用 法: void far settextstyle (int font, int direction, char size);

　　fornt 为字体：DEFAULT\_FONT， TRIPLEX\_FONT， SMALL\_FONT， SANSSERIF\_FONT， GOTHIC\_FONT，也可以用0 ~ 4代替。

　 　direction 为字符的排列方向：横向和竖向，0为横向排列，1为竖向排列。

　 　size 为字体大小：可用interger做参数。

6、outtextxy（）

　　 功 能: 在指定位置显示一字符串

　　 用 法: void far outtextxy(int x, int y, char \*textstring);