

<[C언어 게임 만들기] 뱀게임 (Snake Game) -스네이크 게임 소스 코드 보기>

```
#include<stdio.h>
#include<windows.h>
#include<conio.h>
#include<stdlib.h>
#include<time.h>

#define LEFT 75
#define RIGHT 77
#define UP 72
#define DOWN 80
#define PAUSE 112
#define ESC 27

#define MAP_ADJ_X 3
#define MAP_ADJ_Y 2
#define MAP_X 30
#define MAP_Y 20

int x[100], y[100]; //x,y 좌표값을 저장 총 100개
int food_x, food_y; //food의 좌표값을 저장
int length; //몸길이를 기억
int speed; //게임 속도
int score; //점수 저장 --reset함수에 의해 초기화됨
int best_score=0; //최고 점수 저장 --reset함수에 의해 초기화 되지 않음
int last_score=0; //마지막 점수 저장 --reset함수에 의해 초기화 되지 않음
int dir; //이동방향 저장
int key; //입력받은 키 저장
int status_on=0; // 개발자용 status 표시활성화 변수.. 게임중에 S키를 누르면 활성화

void gotoxy(int x,int y,char* s) { //x값을 2x로 변경, 좌표값에 바로 문자열을 입력할 수 있도록 printf함수 삽입
    COORD pos={2*x,y};
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE),pos);
    printf("%s",s);
}

void title(void); //게임 시작화면
void reset(void); //게임을 초기화
void draw_map(void); // 게임판 테두리를 그림
void move(int dir); //뱀머리를 이동
void pause(void); //일시정지
void game_over(void); //게임 오버를 확인
void food(void); // 음식 생성
void status(void); // 개발자용 status표시

////////////////////MAIN START////////////////////////////////////

int main(){
    title();

    while(1){
```

```

if(kbhit()) do{key=getch();} while(key==224); //키 입력받음
Sleep(speed);

switch(key){ //입력받은 키를 파악하고 실행
    case LEFT:
    case RIGHT:
    case UP:
    case DOWN:
        if((dir==LEFT&&key!=RIGHT)||((dir==RIGHT&&key!=LEFT)||((dir==UP&&key!=DOWN)||
(dir==DOWN&&key!=UP))))//180회전이동을 방지하기 위해 필요.
            dir=key;
        key=0; // 키값을 저장하는 함수를 reset
        break;
    case PAUSE: // P키를 누르면 일시정지
        pause();
        break;
    case 115: //S키를 누르면 개발자용 status를 활성화 시킴
        if(status_on==0) status_on=1;
        else status_on=0;
        key=0;
        break;
    case ESC: //ESC키를 누르면 프로그램 종료
        exit(0);
}
move(dir);

if(status_on==1) status(); // status표시
}
}

////////////////////MAIN END////////////////////////////////////
void title(void){
    int i,j;

    while (kbhit()) getch(); //버퍼에 있는 키값을 버림

    draw_map(); //맵 테두리를 그림
    for(i=MAP_ADJ_Y+1;i<MAP_ADJ_Y+MAP_Y-1;i++){ // 맵 테두리 안쪽을 빈칸으로 채움
        for(j=MAP_ADJ_X+1;j<MAP_ADJ_X+MAP_X-1;j++) gotoxy(j,i," ");
    }

    gotoxy(MAP_ADJ_X+(MAP_X/2)-7,MAP_ADJ_Y+5,"+-----+");
    gotoxy(MAP_ADJ_X+(MAP_X/2)-7,MAP_ADJ_Y+6,"|          S N A K E          |");
    gotoxy(MAP_ADJ_X+(MAP_X/2)-7,MAP_ADJ_Y+7,"+-----+");

    gotoxy(MAP_ADJ_X+(MAP_X/2)-7,MAP_ADJ_Y+9," < PRESS ANY KEY TO START > ");

    gotoxy(MAP_ADJ_X+(MAP_X/2)-7,MAP_ADJ_Y+11," ◇ ←,→,↑,↓ : Move ");
    gotoxy(MAP_ADJ_X+(MAP_X/2)-7,MAP_ADJ_Y+12," ◇ P : Pause ");
    gotoxy(MAP_ADJ_X+(MAP_X/2)-7,MAP_ADJ_Y+13," ◇ ESC : Quit ");
}

```

```

gotoxy(MAP_ADJ_X+(MAP_X/2)-7,MAP_ADJ_Y+17," BLOG.NAVER.COM/AZURE0777");

while(1){
    if(kbhit()){ //키입력받음
        key=getch();
        if(key==ESC) exit(0); // ESC키면 종료
        else break; //아니면 그냥 계속 진행
    }
    gotoxy(MAP_ADJ_X+(MAP_X/2)-7,MAP_ADJ_Y+9," < PRESS ANY KEY TO START > ");
    Sleep(400);
    gotoxy(MAP_ADJ_X+(MAP_X/2)-7,MAP_ADJ_Y+9,"
    Sleep(400);

}
reset(); // 게임을 초기화
}

void reset(void){
    int i;
    system("cls"); //화면을 지움
    draw_map(); //맵 테두리를 그림
    while (kbhit()) getch(); //버퍼에 있는 키값을 버림

    dir=LEFT; // 방향 초기화
    speed=100; // 속도 초기화
    length=5; //뱀 길이 초기화
    score=0; //점수 초기화
    for(i=0;i<length;i++){ //뱀 몸통값 입력
        x[i]=MAP_X/2+i;
        y[i]=MAP_Y/2;
        gotoxy(MAP_ADJ_X+x[i],MAP_ADJ_Y+y[i],"o");
    }
    gotoxy(MAP_ADJ_X+x[0],MAP_ADJ_Y+y[0],"h"); //뱀 머리 그림
    food(); // food 생성
}

void draw_map(void){ //맵 테두리 그리는 함수
    int i,j;
    for(i=0;i<MAP_X;i++){
        gotoxy(MAP_ADJ_X+i,MAP_ADJ_Y,"■");
    }
    for(i=MAP_ADJ_Y+1;i<MAP_ADJ_Y+MAP_Y-1;i++){
        gotoxy(MAP_ADJ_X,i,"■");
        gotoxy(MAP_ADJ_X+MAP_X-1,i,"■");
    }
    for(i=0;i<MAP_X;i++){
        gotoxy(MAP_ADJ_X+i,MAP_ADJ_Y+MAP_Y-1,"■");
    }
}

```

```

void move(int dir){
    int i;

    if(x[0]==food_x&&y[0]==food_y){ //food와 충돌했을 경우
        score+=10; //점수 증가
        food(); //새로운 food 추가
        length++; //길이증가
        x[length-1]=x[length-2]; //새로만든 몸통에 값 입력
        y[length-1]=y[length-2];
    }
    if(x[0]==0||x[0]==MAP_X-1||y[0]==0||y[0]==MAP_Y-1){ //벽과 충돌했을 경우
        game_over();
        return; //game_over에서 게임을 다시 시작하게 되면 여기서부터 반복되므로
        //return을 사용하여 move의 나머지 부분이 실행되지 않도록 합니다.
    }
    for(i=1;i<length;i++){ //자기몸과 충돌했는지 검사
        if(x[0]==x[i]&&y[0]==y[i]){
            game_over();
            return;
        }
    }

    gotoxy(MAP_ADJ_X+x[length-1],MAP_ADJ_Y+y[length-1]," "); //몸통 마지막을 지움
    for(i=length-1;i>0;i--){ //몸통좌표를 한칸씩 옮김
        x[i]=x[i-1];
        y[i]=y[i-1];
    }
    gotoxy(MAP_ADJ_X+x[0],MAP_ADJ_Y+y[0],"o"); //머리가 있던곳을 몸통으로 고침
    if(dir==LEFT) --x[0]; //방향에 따라 새로운 머리좌표(x[0],y[0])값을 변경
    if(dir==RIGHT) ++x[0];
    if(dir==UP) --y[0];
    if(dir==DOWN) ++y[0];
    gotoxy(MAP_ADJ_X+x[i],MAP_ADJ_Y+y[i],"ㅎ"); //새로운 머리좌표값에 머리를 그림
}

void pause(void){ // p키를 눌렀을 경우 게임을 일시 정지
    while(1){
        if(key==PAUSE){
            gotoxy(MAP_ADJ_X+(MAP_X/2)-9,MAP_ADJ_Y,"< PAUSE : PRESS ANY KEY TO RESUME > ");
            Sleep(400);
            gotoxy(MAP_ADJ_X+(MAP_X/2)-9,MAP_ADJ_Y,"");
            Sleep(400);
        }
        else{
            draw_map();
            return;
        }
    }
    if(kbhit()){

```

```

        do{
            key=getch();
        }
        while(key==224);
    }

}

}

void game_over(void){ //게임종료 함수
    gotoxy(MAP_ADJ_X+(MAP_X/2)-6,MAP_ADJ_Y+5,"+-----+");
    gotoxy(MAP_ADJ_X+(MAP_X/2)-6,MAP_ADJ_Y+6,"|      GAME OVER..      |");
    gotoxy(MAP_ADJ_X+(MAP_X/2)-6,MAP_ADJ_Y+7,"+-----+");
    gotoxy(MAP_ADJ_X+(MAP_X/2)-6,MAP_ADJ_Y+8," YOUR SCORE : ");
    printf("%d", last_score=score);

    gotoxy(MAP_ADJ_X+(MAP_X/2)-7,MAP_ADJ_Y+12," Press any keys to restart.. ");

    if(score>best_score) {
        best_score=score;
        gotoxy(MAP_ADJ_X+(MAP_X/2)-4,MAP_ADJ_Y+10,"☆ BEST SCORE ☆");
    }
    Sleep(500);
    while (kbhit()) getch();
    key=getch();
    title();
}

void food(void){
    int i;

    int food_crush_on=0;//food가 뱀 몸통좌표에 생길 경우 on
    int r=0; //난수 생성에 사용되는 변수
    gotoxy(MAP_ADJ_X,MAP_ADJ_Y+MAP_Y," YOUR SCORE: "); //점수표시
    printf("%3d, LAST SCORE: %3d, BEST SCORE: %3d", score, last_score, best_score);

    while(1){
        food_crush_on=0;
        srand((unsigned)time(NULL)+r); //난수표생성
        food_x=(rand()%(MAP_X-2))+1;    //난수를 좌표값에 넣음
        food_y=(rand()%(MAP_Y-2))+1;

        for(i=0;i<length;i++){ //food가 뱀 몸통과 겹치는지 확인
            if(food_x==x[i]&&food_y==y[i]){
                food_crush_on=1; //겹치면 food_crush_on 를 on
                r++;
            }
        }
        break;
    }
}

```

```

        if(food_crush_on==1) continue; //겹쳤을 경우 while문을 다시 시작

        gotoxy(MAP_ADJ_X+food_x,MAP_ADJ_Y+food_y,"♪"); //안겹쳤을 경우 좌표값에 food를 찍고
        speed-=3; //속도 증가
        break;
    }
}

void status(void){ //각종 스텟을 볼수 있는 함수
    gotoxy(MAP_ADJ_X+MAP_X+1,MAP_ADJ_Y,"head= ");
    printf("%2d,%2d",x[0],y[0]);
    gotoxy(MAP_ADJ_X+MAP_X+1,MAP_ADJ_Y+1,"food= ");
    printf("%2d,%2d",food_x,food_y);
    gotoxy(MAP_ADJ_X+MAP_X+1,MAP_ADJ_Y+2,"leng= ");
    printf("%2d",length);
    gotoxy(MAP_ADJ_X+MAP_X+1,MAP_ADJ_Y+3,"key= ");
    printf("%3d",key);
    gotoxy(MAP_ADJ_X+MAP_X+1,MAP_ADJ_Y+4,"spd= ");
    printf("%3d",speed);
    gotoxy(MAP_ADJ_X+MAP_X+1,MAP_ADJ_Y+6,"score= ");
    printf("%3d",score);
}

```