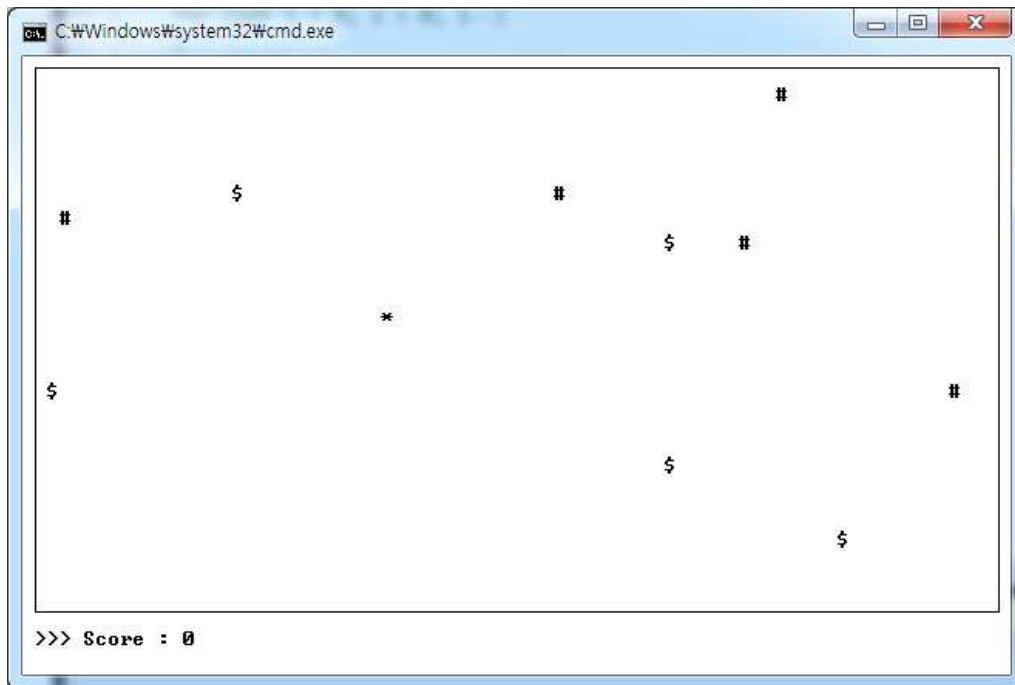


<과제 #7 : 덜 자란 Snake 게임>

- 원래 목표로 하는 게임은 Snake 게임이다. 그러나 본 과제에서는 이를 단순화하여 ‘*’ 문자 1개로 표현되는 “덜 자란 Snake 게임”을 완성하면 된다. 이 게임에서는 ‘*’ 문자 1개로 표현되는 Snake가 있고 ‘\$’ 문자로 표현되는 먹이가 있으며 ‘#’ 문자로 표현되는 뱀이 있다. Snake가 먹이를 먹으면 점수가 올라가고(먹이는 사라진다) 뱀에 걸리면 게임은 바로 종료된다. 먹이를 모두 먹게 되는 경우에도 게임은 종료된다. Snake는 주기적으로 이동하며 방향키에 따라 방향이 전환된다.



- 덜 자란 Snake 게임에서의 준수 사항 (점수 관련 사항)
- Snake, 먹이, 뱀의 위치는 모두 구조체(예, struct Point)로 표현되어야 한다.
 - 먹이와 뱀은 각각 5개이고 무작위 위치(x, y)에 나타나게 된다.
 - Snake는 자동으로 이동이 가능해야 하고, 키보드 입력에 따라 방향이 변경되어야 한다.
 - Snake의 위치가 먹이의 위치와 같아지면(먹이를 먹으면) 점수가 증가한다.
 - Snake의 위치가 뱀의 위치와 같아지면(뱀에 걸리면) 프로그램은 종료된다.
- 점수와 무관하지만 도전해 볼 만한 내용
- Snake가 벽에 부딪히면 프로그램은 종료한다.
 - Snake의 위치가 먹이의 위치와 같아지면 점수가 증가할 뿐만 아니라 해당 먹이는 사라지게 된다.
 - Snake가 모든 먹이를 먹게 되면 다음 Stage로 넘어간다. 다음 Stage에서는 먹이와 뱀의 개수가 늘어나고 Snake의 속도도 빨라진다.
 - Snake가 먹이를 먹게 되면 Snake의 길이가 길어진다.

1. 자체 평가 결과

기능	구현 여부	완성도	배점(3)	검토 결과
기본 화면 출력 (사각 박스, Snake, Score)	○	완성	0.5	
먹이, 뱀 화면 출력 (무작위 위치)	○	완성	0.5	
Snake 이동, 방향 전환	○	완성	0.5	
구조체 사용 (위치 : Snake, 먹이, 뱀)	○	완성	0.5	
Snake의 먹이 먹기 처리 (점수 증가)	○	완성	0.5	
Snake의 뱀 걸림 처리 (프로그램 종료)	○	완성	0.5	
합계			3	

2. 소스 코드

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

#define KEY_UP (256 + 72)
#define KEY_DOWN (256 + 80)
#define KEY_LEFT (256 + 75)
#define KEY_RIGHT (256 + 77)

int Score = 0;
int width = 58;
int length = 20;
double speed = 0.5;
int count = 0;
int GetKey(void);
double GetElapsedTime(clock_t initial_clock);
void GotoXY(int x, int y);
void Erase(int x, int y);    // (x, y)로 이동하여 공백 문자 출력
void Draw(int x, int y);    // (x, y)로 이동하여 '*' 문자 출력

#define LEFT 0
#define RIGHT 1
```

```

#define UP 2
#define DOWN 3

typedef struct Point
{
    int x;
    int y;
} Point;

void initScreen(void);

int GetKey(void)
{
    int ch = _getch();
    if (ch == 0 || ch == 224)
        ch = 256 + _getch();
    return ch;
}

double GetElapsedTime(clock_t initial_clock)
{
    clock_t current_clock = clock();
    return (double)(current_clock - initial_clock) / CLOCKS_PER_SEC;
}

void GotoXY(int x, int y)
{
    // COORD 구조체 변수를 통해 이동할 위치 설정
    COORD pos = { x, y };
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), pos);
}

void Erase(int x, int y)
{
    GotoXY(x, y);
    printf(" ");
}

void Draw(int x, int y)
{
    GotoXY(x, y);
    printf("*");
}

void SetCursorVisible(int visible)

```

```

{
    CONSOLE_CURSOR_INFO ci = { 100, visible };
    SetConsoleCursorInfo(GetStdHandle(STD_OUTPUT_HANDLE), &ci);
}

void DrawFood(int fx, int fy)
{
    GotoXY(fx, fy);
    printf("$");
}

void DrawTrap(int tx, int ty)
{
    GotoXY(tx, ty);
    printf("#");
}

void initScreen(void)
{
    printf("# : Trap      $ : Food\n");
    printf("┌");
    for (int i = 0; i < width; i++)
        printf("—");
    printf("┐ \n");
    for (int j = 0; j < length; j++)
    {
        printf(" | ");
        for (int i = 0; i < (width / 2); i++)
            printf(" ");
        printf(" | \n");
    }
    printf("└");
    for (int i = 0; i < 58; i++)
        printf("—");
    printf("┘ \n\n");

    GotoXY(0, 25);
    printf(">>> Score : 0\n");

}

void ScoreSet()

```

```

{
    Score += 1000;
    GotoXY(0, 25);
    printf(">>> Score : %d\n", Score);
}

int main(void)
{
    int Gamesign = 1;
    SetCursorVisible(0);
    Point food[5];
    Point trap[5];
    int sign[5] = { 0 };
    initScreen();

    int direction = DOWN;
    srand(time(NULL));
    Point snake = { rand() % (width - 2) + 3 , rand() % (length - 2) + 3 };
    for (int i = 0; i < 5; i++)
    {
        food[i].x = rand() % (width - 2) + 3;
        food[i].y = rand() % (length - 3) + 3;
        DrawFood(food[i].x, food[i].y);
    }
    for (int i = 0; i < 5; i++)
    {
        trap[i].x = rand() % (width - 2) + 3;
        trap[i].y = rand() % (length - 3) + 3 ;
        DrawTrap(trap[i].x, trap[i].y);
    }
    Draw(snake.x, snake.y);
    clock_t initial_clock = clock();

    while (Gamesign)
    {
        for (int i = 0; i < 5; i++)
        {
            if (snake.x == food[i].x && snake.y == food[i].y && sign[i] != 1)
            {
                ScoreSet();
                speed -= 0.1;
                count += 1;
                sign[i] = 1;
                if (Score == 5000)

```

```

        {
            GotoXY(0, 27);
            printf("Game Clear!\n\n");
            Gamesign = 0;
        }
    }
}
for (int i = 0; i < 5; i++)
{
    if (snake.x == trap[i].x && snake.y == trap[i].y)
    {
        GotoXY(snake.x, snake.y);
        printf("※");
        GotoXY(0, 27);
        printf("Game Over!\n\n");
        Gamesign = 0;
    }
}

if (GetElapsedTime(initial_clock) > speed)
{
    // 0.5초 경과
    Erase(snake.x, snake.y);
    if (direction == LEFT)
        snake.x--;
    else if (direction == RIGHT)
        snake.x++;
    else if (direction == UP)
        snake.y--;
    else if (direction == DOWN)
        snake.y++;
    Draw(snake.x, snake.y);
    if (snake.y == length + 2 || snake.x == width + 1 || snake.y == 1 ||
snake.x == 1)
    {
        GotoXY(0, 27);
        printf("Game Over!\n\n");
        Gamesign = 0;
    }
    initial_clock = clock(); // 기준 시각 재설정
}
if (_kbhit())
{
    int key = GetKey();

```

```

        if (key == KEY_LEFT)
            direction = LEFT;
        else if (key == KEY_RIGHT)
            direction = RIGHT;
        else if (key == KEY_UP)
            direction = UP;
        else if (key == KEY_DOWN)
            direction = DOWN;
    }

}
}

```

3. 실행화면



게임이 시작되면 무작위로 배치된 덫과 먹이가 있다. 먹이를 먹게되면 점수가 오르고 속도 또한 점차 증가한다.

