

## [5 월 진행보고서]

- C 언어로 지렁이게임 개발 -

박지원 신하연 손효림

### 목표

- 지렁이 게임 주요 함수 구현

### 역할

- 게임 시간 설정 함수 : 신하연
- 지렁이 움직임 함수 : 손효림
- 지렁이 먹이 함수, 필드 구현 : 박지원

### 작성한 코드

- 먹이 함수, 필드 함수 구현

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

#define FIELD_WIDTH 50    // 게임 영역 - 가로
#define FIELD_HEIGHT 25  // 게임 영역 - 세로

void gotoXY(int x, int y);
int preyX(void);
int preyY(void);

int prey_m, prey_x, prey_y = 0;
```

```

// 콘솔 창 속 커서 이동
void gotoXY(int x, int y) {
    COORD pos; //COORD 구조체를 pos 로 선언

    pos.X = x + 20;
    pos.Y = y + 6;

    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), pos);
//지정 된 콘솔 화면 버퍼에서 커서 위치를 설정해주는 함수
    // GetStdHandle(STD_OUTPUT_HANDLE) : 콘솔 창 화면 버퍼에 대한
표준 출력 핸들
    // pos : X,Y 의 좌표 정보 저장
}

void field() { // 게임 영역

    gotoXY(18, -3);
    printf("지령이 게임");

    for (int i = 1; i < FIELD_WIDTH; i++)
    {
        gotoXY(i, -5);
        printf("—");
    }

    for (int i = 1; i < FIELD_WIDTH; i++)
    {
        gotoXY(i, 0);
        printf("—");
        gotoXY(i, FIELD_HEIGHT);
        printf("—");
    }
}

```

```

    }

    for (int i = 1; i < FIELD_HEIGHT; i++)
    {
        gotoXY(0, i);
        printf(" | ");
        gotoXY(FIELD_WIDTH, i);
        printf(" | ");
    }

    gotoXY(0, 0);
    printf(" ┌");

    gotoXY(0, FIELD_HEIGHT);
    printf(" └");

    gotoXY(FIELD_WIDTH, 0);
    printf("└ ");

    gotoXY(FIELD_WIDTH, FIELD_HEIGHT);
    printf("└ ");

    gotoXY(FIELD_WIDTH, 30);
    printf("Wn");
}

void prey() {
    if (prey_m == 0) { //초기값이 0 이므로 처음에는 무조건 출력된다
        //먹이 랜덤 x, y 좌표를 food_x, food_y 에 각각 저장.
        prey_x = preyX();
        prey_y = preyY();
    }
}

```

```
        gotoXY(pre_y, pre_x);
        printf("♣"); //위에서 저장했던 x, y 좌표에 먹이를 뿌려줌
    }
}

int preyX(void) {
    pre_x = (rand() % 20 + 2) * 2 - 1;

    return pre_x;
}

int preyY(void) {
    pre_y = rand() % 20 + 2;

    return pre_y;
}

int main() {

    system("mode con:cols=90 lines=50");

    field();
    prey();

}
```

- 결과 화면



- 지렁이 움직임 함수 구현

```

#include <stdio.h>
#include <windows.h>

#define LEFT 75 // 방향키 왼쪽
#define RIGHT 77 // 방향키 오른쪽
#define UP 72 // 방향키 위쪽
#define DOWN 80 // 방향키 아래쪽

```

```
unsigned char move; //방향키
```

```
int x = 1;
```

```
int y = 200;
```

```
void gotoxy(int x, int y)
```

```
{
```

```
    HANDLE consoleHandle = GetStdHandle(STD_OUTPUT_HANDLE);
```

```
    COORD Pos = {x,y};
```

```
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), Pos);
```

```
}
```

```
void earthworm_move(HDC hdc,int x,int y)
```

```
{
```

```
    if (move == UP) { //방향키 '상' 입력받을시
```

```
        y -= 10;
```

```
        gotoxy(x,y);
```

```
    }
```

```
    else if (move == DOWN) { //방향키 '하' 입력받을시
```

```
        y += 10;
```

```
        gotoxy(x,y);
```

```
    }
```

```
    else if (move == LEFT) { //방향키 '좌' 입력받을시
```

```
        x -= 10;
```

```
        gotoxy(x,y);
```

```
    }
```

```
    else if (move == RIGHT) { //방향키 '우' 입력받을시
```

```
        x += 10;
```

```
        gotoxy(x,y);
```

```
    }
```

```
}
```

```

int main(void) {

    HDC hdc = GetWindowDC(GetForegroundWindow());

    gotoxy(x,y);

    do {
        earthworm_move(move, x, y);
        system("cls");
        Ellipse(hdc, x, y, x + 10, y + 10);
    } while (x == 200);

    return 0;

}

```

- 게임 시간 함수 구현

```

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

void gotoxy(int x, int y)
{
    COORD CursorPosition = { x, y };
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE),
CursorPosition);
}

```

```

int main()
{
    //게임 함수 입력 이후에 좌표설정해서 위치 정할 것
    int hour, min, sec;
    hour = 0;
    min = 0;
    sec = 30;

    int before_start();

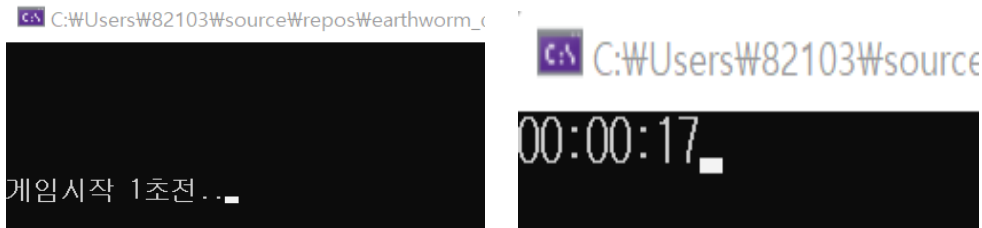
    int before_start(); {
        srand((int)time(NULL));
        for (int sec = 3; sec > 0; sec--)
        {
            gotoxy(0, 5);
            printf("게임시작 %d 초전..", sec);
            Sleep(1000);
        }
    }

    while (1) { //타이머와 게임화면 동시 진행
        system("cls");
        printf("%02d:%02d:%02d", hour, min, sec);
        sec--;
        if (sec == 60) { min++; sec = 0; }
        if (min == 60) { hour++; min = 0; }
        if (hour == 24) { hour = min = sec = 0; }
        if (sec == 0)
            break;
        Sleep(1000);
    }
    return 0;
}

```



- 결과 화면



## 코드 설명

### < 지렁이 먹이/필드 >

- **field()**

가로, 세로 값이 재정의 되지 않도록 #define 상수를 사용해 값을 선언해주었다.

const 를 사용해 지정할 수 있었지만, 이전에 써본 것보다는 새로 써보는 #define 을 사용해보고 싶었기 때문에 사용했다. field() 함수에서 gotoxy() 함수를 사용해 콘솔창 중간에 위치하도록 조정했다.

### **prey()**

rand()함수를 사용하여 좌표의 위치를 랜덤하게 받아온다.

받아온 랜덤 값을 gotoxy()함수에 적용하여 커서의 위치를 바꿔주고 그 위치에 먹이를 출력한다.

### < 지렁이 움직임 >

- \_getch() 함수로 화살표 키를 입력받는다. 방향키 별로 숫자는 정해져있다. 방향키 왼쪽은 75, 방향키 오른쪽은 77, 방향키 위쪽은 72, 방향키 아래쪽은 80 이다. 호출하기 쉽도록 define 으로 각각 LEFT, RIGHT, UP, DOWN 으로 지정해준다.

`system("cls");` 은 콘솔 화면 지우기 함수이다. 지령이의 위치가 이동할 때 그 전에 있던 도형을 지워줌으로 하여금 이동한 경로에 잔상이 남지 않도록 한다.

`Ellipse(hdc, x, y, x + 10, y + 10);` 로 지름이 10 인 원을 그린다.

## < 게임 시간 설정 >

- 메인 함수 내부에 `before_start()` 함수를 만들고 무한 반복 `while` 문을 사용하여 제한시간을 카운트하는 코드이다.

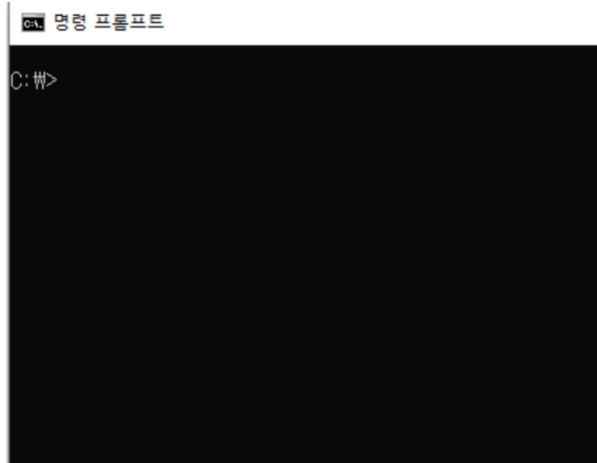
`Before_start()` 함수는 게임이 실행되기 전 게임시작 몇 초전인지 알려주고 그 초가 줄어드는 함수이다. `int` 형인 초를 `sec` 변수로 선언하였으며 3 초에서 줄어든다. 3 초가 지나면 `Sleep(1000)`이 실행되는데 이는 프로그램을 1 초 지연시키는 것이다. 이후 무한 반복 `while` 문이 실행되고 제한시간 30 초가 끝나면 `break` 문을 통해 프로그램이 자동 종료된다.

타이머가 구현될 때 '00:00:00' 형태로 구현되기 위해 "%02d:%02d:%02d"로 코드를 작성하였고 이것이 화면에 나타나기 전, `system("cls")` 코드를 사용하여 이전에 실행된 화면을 지워주었다.

## 스터디 내용

- `system()` : C 언어에서 `stdlib` 에 포함되어 있는 함수로 C 언어 프로그램 안에서 운영체제의 명령을 실행하기 위한 함수이다.

C 언어를 통해 소스코드를 작성하고 컴파일을 하면 운영체제에 따라 알맞게 하드웨어 자원을 컨트롤할 수 있도록 프로그램이 만들어지는데, 이를 운영체제의 명령 프롬프트 창 (cmd)이라는 프로그램 안에서 동작하고 있는 것 입니다. 이 명령 프롬프트 창(cmd)에 명령을 보내기 위한 함수가 바로 `system` 함수이다.



```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int input;
    printf("안녕하세요 프로그램 시작 화면
입니다.\n");
    printf("1.주문 2.나가기\n");
    printf("원하는 메뉴를 선택하세요 : ");
    scanf("%d",&input);
    if(input == 1)
    {
        system("cls");
        printf("안녕하세요 주문 화면 입니다.\n");
        system("pause");
    }
    else if(input ==2)
    {
        system("cls");
        printf("프로그램을 종료합니다.\n");
        system("pause");
    }
}
```

```

else
{
    system("cls");
    printf("잘못된 입력입니다.\n");
    system("pause");
}

return 0;
}

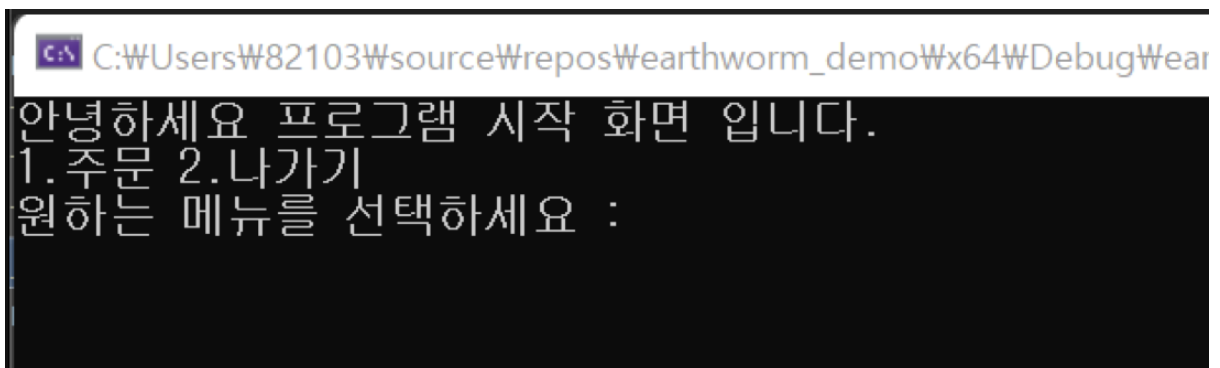
```

pause - 명령 프롬프트 창이 키 입력을 받을 때까지 멈춘다.(mac 환경에서는 read 를 사용하면 된다.)

cls - 명령 프롬프트 창의 내용을 모두 지운다.

이 외에도 cmd 창에서 사용할 수 있는 명령어들도 사용할 수 있다.

- 결과화면



```

C:\Users\W82103\source\repos\earthworm_demo\Wx64\Debug\ear
안녕하세요 프로그램 시작 화면 입니다.
1.주문 2.나가기
원하는 메뉴를 선택하세요 :

```



C:\Users\W82103\source\repos\earthworm\_demo\wx64\Debug

```
안녕하세요 주문 화면 입니다.  
계속하려면 아무 키나 누르십시오 . . .
```

보통 `system("cls")`를 사용하지 않으면 콘솔창에 출력된 값이 지워지지 않고 누적되는데,

`system("cls")`를 사용한다면, 이 전에 출력된 내용은 지워지고 새로운 값들이 출력된 것을 볼 수 있다.

`system()` 함수를 사용할 때 꼭 **"" 큰 따옴표**를 사용해야한다. 콘솔창에 문자열을 넘겨주는 것이기 때문에 **" 작은 따옴표**가 아닌 **"" 큰 따옴표**를 사용해야한다.

## 앞으로의 계획

작성한 함수들을 모아서 하나의 코드로 완성할 예정이며(`gotoxy()`함수 하나의 함수로 통일 예정), 주요 함수 뿐만 아니라 다른 세부적인 함수들도 필요시에 작성해서 게임을 완성할 예정이다. 현재는 주요 함수들이 따로 구현되어 있는 상황이기 때문에 앞으로 이 함수들을 어떠한 순서로 배치할지, 어떻게 묶어야 효과적이며 원활한 프로그램이 될지 고민해보고 팀원들과 함께 완성할 계획이다.

## 어려웠던 점

방향키로 지렁이를 움직이는 함수를 빌드할 때에는 문제가 검색되지 않았는데 막상 실행을 하면 콘솔 화면에 아무것도 출력되지 않는다는 점이 문제가 되었다.

## 참고문헌

도서, "누구나 쉽게 즐기는 C 언어 콘서트" -천인국 지음

연승욱, 박후린 교수님 PPT

[점수 기록, 게임 크기 조정, 먹이 생성, 지렁이 키우기 step\\_2 · Sehyeon-An/Earthworm\\_game@ff50b23 · GitHub](#)