

[C언어 지렁이 게임 개발] 프로젝트 - 4월 진행 보고서

박지원, 손효림, 신하연, 한아림
2022.05.03

1. 목표
2. 진행상황
3. 스터디 내용1
4. 스터디 내용2
5. 앞으로의 계획
6. 어려웠던 점
7. 참고문헌

1. 목표

비주얼 스튜디오 프로그램을 사용하여 c언어로 지렁이 게임을 구현한다. 팀원들과 협력하여 프로그램을 개발함으로써 전공 과목에 대한 이해를 높이고 배운 지식을 활용함으로써 알고리즘 구현 능력을 함양한다.

2. 진행상황

팀원들끼리 진행상황을 공유하기 위해 디스코드 서버를 구축하였고 줌을 통해 향후 계획을 논의하였다. 처음에 팀원들과 생각했던 프로그램은 시중의 지렁이 게임이었는데 현실적으로 텍스트 기반 프롬프트로는 다소 어려움이 있다는 생각을 하게 되었다.

각종 자료 조사를 하던 중 다양한 종류의 지렁이 게임이 있다는 것을 알았다. 지렁이의 길이 조정 방식, 게임 종료 조건, 구현 화면 등 세부사항들을 의논하여 최종적으로 지렁이가 화면에 표시된 먹이를 먹으면 점수가 올라가는 프로그램을 구현하기로 결정하였다. 추가적으로 게임의 시간 제한을 두어 우리가 설정한 특정 시간이 되면 게임을 종료하는 방식으로 개발할 계획이다.

3. 스터디내용1-기본적인 함수

본격적으로 프로그램을 구현하기 전 필요한 지식을 공부하고 정리해보았다.

개발 시 필요한 헤더파일

<stdio.h> : Standard Input/Output library (표준입출력 라이브러리)의 약어로서, C 언어의 표준 라이브러리 함수의 매크로 정의, 상수, 여러 형의 입출력 함수가 포함된 헤더 파일이다.

<stdlib.h> : 문자열 변환, 의사 난수 생성, 동적 메모리 관리 등의 함수들을 포함하고 있다. 주로 프로그램 제어 관련 함수가 들어가 있다. 동적 메모리 할당 함수인 malloc, calloc 함수도 이 헤더에 포함되어 있으며, 시스템 명령어나 프로세스 제어 함수도 포함되어 있다. 원래는 itoa, atoi 함수도 있었으나 없어졌다.

<conio.h> : 콘솔 입출력 함수를 제공하는 헤더로 다양한 함수가 있다. MS-DOS 시절부터 사용되었다. 지금은 C 언어 표준도 아니고 POSIX 함수도 아니다. 따라서 Visual studio에서만 사용할 수 있고 리눅스나 OS X에서는 사용할 수 없다.

int kbhit() : keyboard hit의 약자이며 버퍼에 값이 있으면 1을, 없으면 0을 리턴한다.

int getch() : 값을 입력할 때까지 대기했다가 값이 들어오면 버퍼에 저장했다가 버퍼에서 꺼내서 리턴한다. 예외키를 누를 경우 두 개의 값이 생긴다. 확장키들을 구분하는 방법은 getch 함수를 다시 한 번 호출하면 알 수 있다. 단 에코 기능은 없다.

int getche() : getch 함수와 기능은 동일하되, 에코기능이 있다.

int ungetch(int _Ch) : 버퍼에 변수 _Ch를 push한다.

char* cgets(char *_Buffer) : Buffer[0]의 크기를 미리 입력하고 cgets 함수를 호출하면 Buffer[1]에 실제 입력받은 크기가 저장되고 Buffer[2]부터 실제로 받은 값들이 저장된다.

int cscanf(const char *_Format, ...) : scanf 함수와 동일하되 scanf 함수는 입력을 받고 엔터를 누를 때까지는 고칠 수 있지만 cscanf 함수는 문자의 경우 하나를 입력하는 즉시 cscanf 함수를 빠져나오고 정수의 경우 띄어쓰기, 엔터키 등을 입력하면 빠져나온다.

int cputs(const char *_Str) : 문자열 _Str 를 출력한다. 자동 줄 바꿈은 없다.

int cprintf(const char *_Format, ...) : printf 함수와 동일하다.

int putch(int _Ch) : 변수 _Ch의 아스키 값을 출력한다.

헤더파일 **<windows.h>** : 윈도우 개발자들이 필요한 모든 매크로들, 다양한 함수들과 서브시스템에서 사용되는 모든 데이터 타입들 그리고 윈도우 API의 함수들을 위한 정의를 포함하는 윈도우의 C 및 C++ 헤더파일이다. 이것은 C에서도 사용될 수 있는 윈도우용의 수많은 함수들을 정의한다.

헤더파일 **<time.h>** : C언어의 표준 라이브러리로, 시간과 날짜를 얻거나 조작하는 함수들을 포함하고 있다. 관련 함수에는 시간관련, 변환관련, 매크로, 타입이 있다.

시간 관련

clock : 시계 프로그램

difftime : 두 개 시간의 차이를 구한다.

mktime : tm 구조체를 time_t 로 바꾼다.

time : 현재 시간을 구한다.

변환 관련

asctime : tm 구조체를 문자열로 바꾼다.

ctime : time_t 값을 문자열로 바꾼다.

gmtime : time_t를 UTC 시간으로 하여 tm으로 바꾼다.

localtime : time_t를 현지 시간으로 하여 tm으로 바꾼다.

strftime : 특정한 형식(사용자가 지정한 형식 문자열)으로 시간을 출력한다.

매크로

CLOCKS_PER_SEC : 1초당 클럭 수

NULL : 널 포인터

타입

clock_t : 클럭 타입

size_t : 부호 없는 정수형

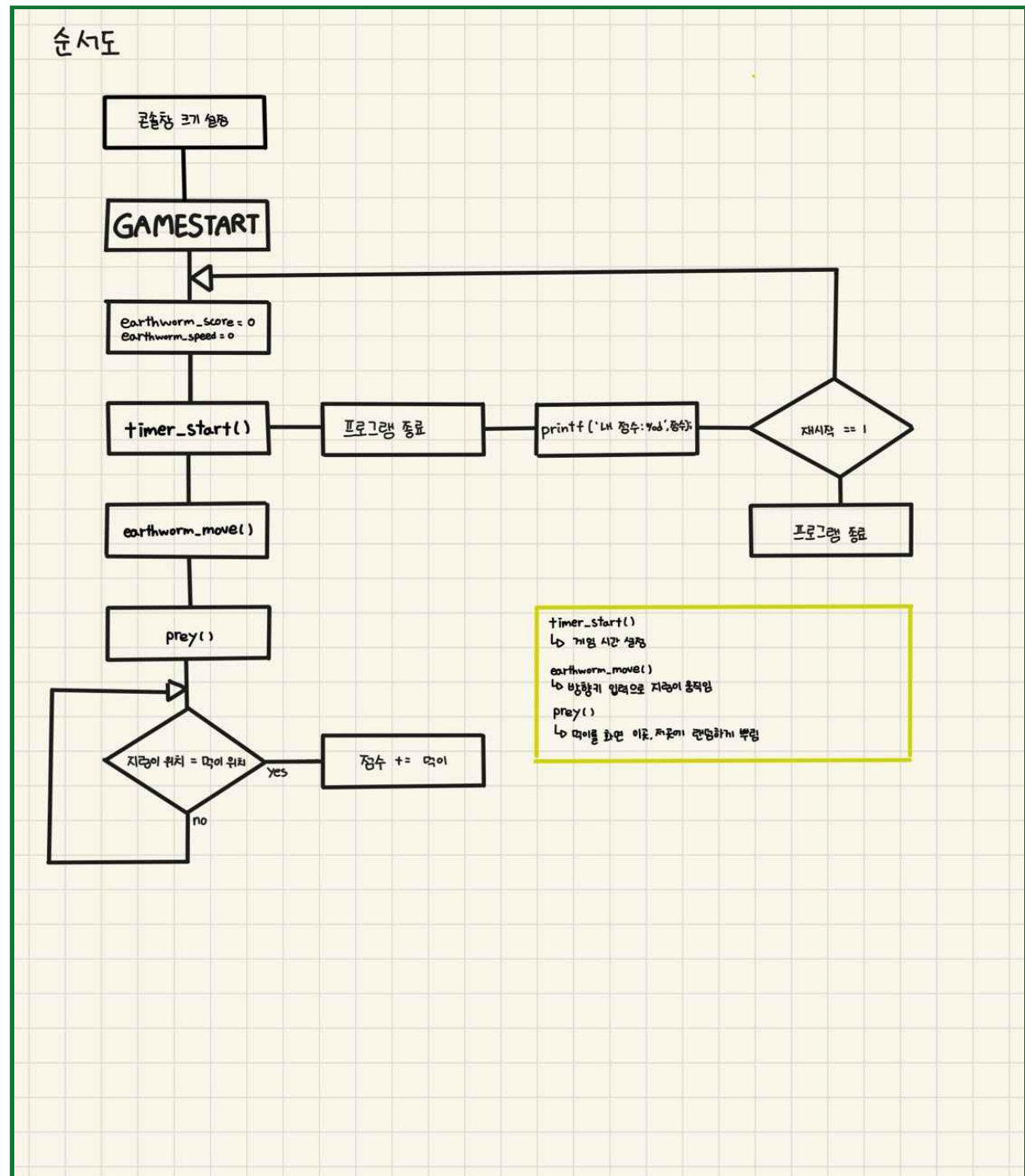
time_t : 시간 타입

struct TM : 시간 구조체

4. 스터디 내용2 - 게임 시스템 관련 스터디 & 회의

a.C 언어에서의 코드 빈출 함수 및 스터디

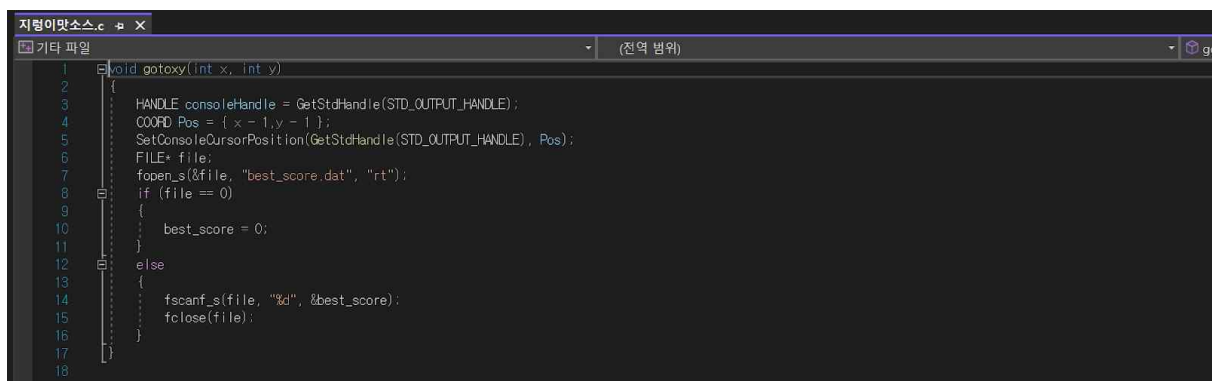
-게임 실행 알고리즘 및 게임 제작 순서도



-gotoxy 함수

: C언어나 C++같이 API나 MFC를 사용하지 않는 컴퓨터 언어의 경우 GUI(그래픽 인터페이스)가 불가능하다. 하지만 대신 모든 것이 문자로 표현된다는 특징을 가지고 있어 콘솔창 내부에서라면 어디에서든지 자신만의 좌표를 가지고 있다.

여기서 gotoxy 함수는 이 고유한 좌표를 다른 곳으로 옮겨주는 역할을 수행하는 함수이다. gotoxy 함수의 원형은 void gotoxy(int x,int y) 이고, 여기서 x는 가로를, y는 세로를 의미한다.

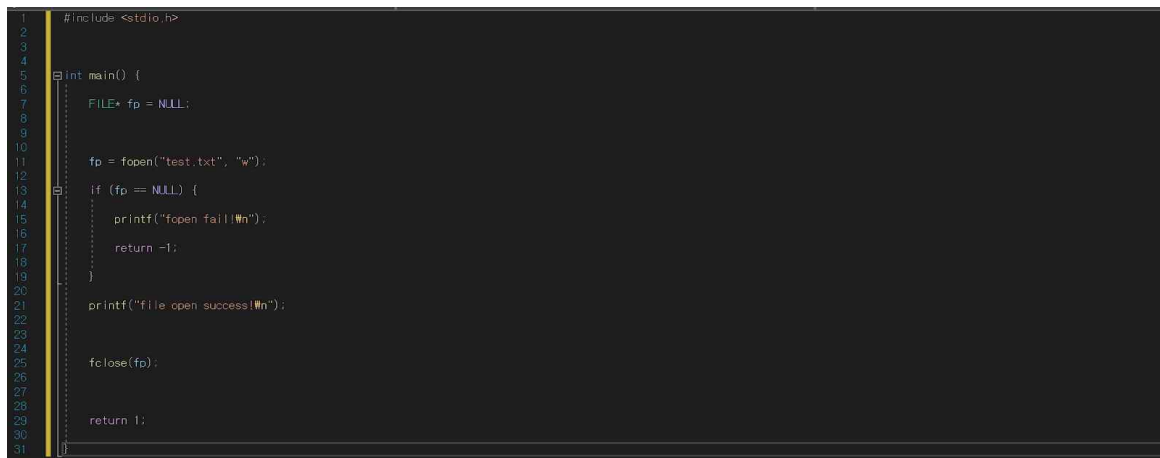


```
1 void gotoxy(int x, int y)
2 {
3     HANDLE consoleHandle = GetStdHandle(STD_OUTPUT_HANDLE);
4     COORD Pos = { x - 1, y - 1 };
5     SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), Pos);
6     FILE* file;
7     fopen_s(&file, "best_score.dat", "rt");
8     if (file == 0)
9     {
10         best_score = 0;
11     }
12     else
13     {
14         fscanf_s(file, "%d", &best_score);
15         fclose(file);
16     }
17 }
18
```

-fopen 함수와 fclose 함수

: fopen 함수는 지정된 파일을 모드에 맞게 열고, 스트림과 연결된 파일 포인터를 리턴시키는 역할을 수행한다. 이어 fclose 함수는 fopen 함수로 열려진 파일을 닫고, 스트림을 해제하는 역할을 한다.

fopen 함수와 fclose 함수는 헤더파일인 stdio.h 안에 정의되어 있다는 특징이 있다.



```
1 #include <stdio.h>
2
3
4
5 int main() {
6     FILE* fp = NULL;
7
8
9
10
11     fp = fopen("test.txt", "w");
12
13     if (fp == NULL) {
14         printf("fopen fail!\n");
15         return -1;
16     }
17
18     printf("file open success!\n");
19
20
21     fclose(fp);
22
23
24     return 1;
25 }
26
27
28
29
30
31
```

5. 앞으로의 계획

게임을 제작하는 데 자주 사용하는 함수들에 대한 추가적인 조사와 공부를 더 할 예정이다. 또한 점수의 계산 과정이나 게임의 상세 운용 과정 등 게임 시스템에 대한 상세한 설명을 추가하고, 매주 진행하는 C언어의 스터디도 지속할 계획이다. 그리고 다음 주부터 게임 개발에 착수할 예정이다.

6. 어려웠던 점

아무래도 게임 제작 프로젝트가 처음이다 보니 프로젝트를 시작할 때 방향성 설정 측면에서 어려움이 있었다. 프로젝트 최종 발표일에 맞추어 한정된 시간 내에 한정된 수의 보고서를 제출해야 하는데, 각 보고서에 기록할 내용과 매 회의에서의 진도를 조율하는 것 또한 어려운 과제 중 하나였다.

7. 참고문헌

1. https://github.com/Sehyeon-An/Earthworm_game/blob/master/%EC%88%9C%EC%84%9C%EB%8F%84.jpg
2. https://github.com/Sehyeon-An/Earthworm_game/blob/master/%EA%B2%8C%EC%9E%84%ED%94%84%EB%A1%9C%EC%A0%9D%ED%8A%B8%20%EB%B0%9C%ED%91%9C%EC%9E%90%EB%A3%8C.pptx
3. https://github.com/Sehyeon-An/Earthworm_game/blob/master/Earthworm_game%20%ED%9A%8C%EC%9D%98%EB%A1%9D.pdf
4. https://github.com/Sehyeon-An/Earthworm_game/blob/master/main2.cpp
5. https://github.com/Sehyeon-An/Earthworm_game/blob/master/Earthworm_game.cpp
6. [\[C언어\]conio.h 에 대해 알아보자 \(tistory.com\)](#)

7. [<conio.h>에 있는 함수의 기능 \(tistory.com\)](http://tistory.com)