

4조

C언어 프로젝트

숫자야구 / 장애물 피하기 게임

목차

1

프로젝트 개요

2

팀 구성원 역할소개

3

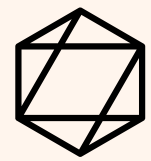
프로젝트 수행절차

4

프로젝트 수행결과

5

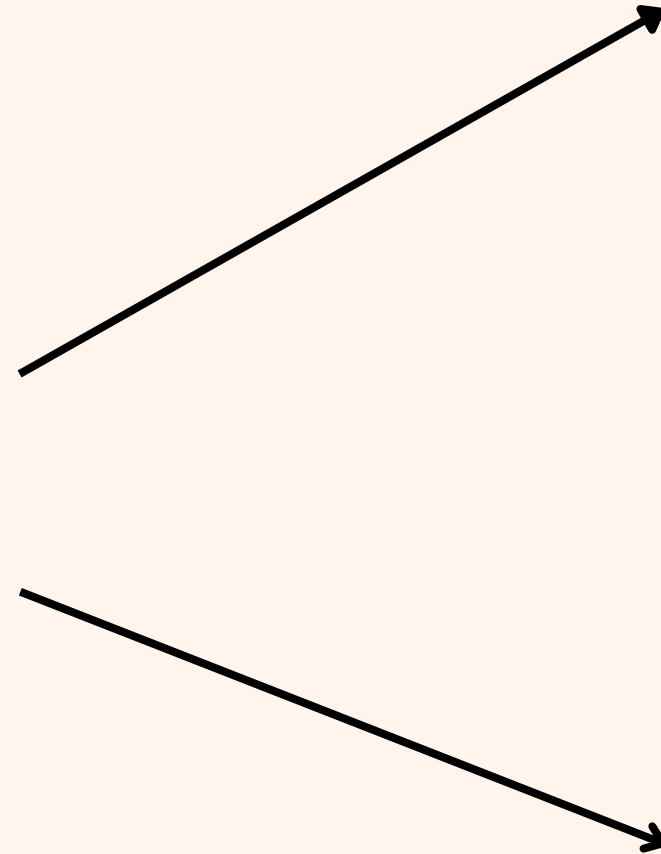
자체 평가 의견



프로젝트 개요

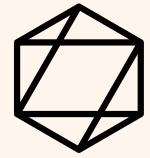
c언어로 2가지
게임 만들기

MAIN MENU



숫자야구 : 3자리 숫자를 전부 맞추는 게임
전부 맞으면 strike, 숫자는 맞지만 자리가 틀리면
ball, 하나라도 맞는게 없다면 out
제한 횟수내에 클리어해야한다

장애물 피하기 게임: 공룡을 움직여 장애물에 닿이지
않게 피해야 한다



프로젝트 구성절차

프로젝트 주제 토론 및
확정



파트 분배 및 프로젝트
구성 시작



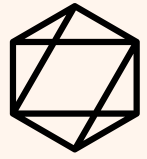
프로젝트 구성 완료



ppt 작성 시작



모든 절차 완료



팀 구성원 소개



김희수

팀의 리더
메인 화면 및 랭킹시
스템



이진곤

장애물 피하기
게임 제작



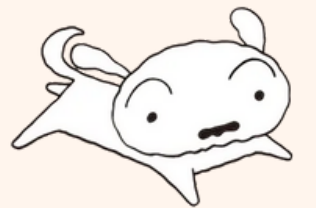
정선호

숫자야구 게임 제작



임나현

숫자야구 게임 제작
및 ppt 제작



GPT

프로젝트의
중심 담당
지대한 도움을
주심

프로그램 수행 절차 - 메인 화면

```
// txt파일 출력을 위한 코드
void printTxtFile(const char* fileName) {
    FILE* file;
    if (fopen_s(&file, fileName, "r") != 0) {
        printf("Failed to open file %s.\n", fileName);
        return;
    }

    char buffer[500];
    while (fgets(buffer, sizeof(buffer), file) != NULL) {
        printf("%s", buffer);
        fflush(stdout); // 출력 버퍼를 비웁니다.
    }
    printf("\n");
    printf("\n");
    printf("\n");
    fclose(file);
}
```

←
텍스트 파일 출력

```

// wav파일 재생을 위한 문자열 변환 함수
wchar_t* charToWChar(const char* str)
{
    const size_t size = strlen(str) + 1;
    wchar_t* wstr = (wchar_t*)calloc(size, sizeof(wchar_t));
    mbstowcs_s(NULL, wstr, size, str, _TRUNCATE);

    return wstr;
}

// wav파일 재생함수
void playMusic(const char* fileName)
{
    wchar_t* wfileName = charToWChar(fileName);
    PlaySound(wfileName, NULL, SND_FILENAME | SND_ASYNC);
    free(wfileName);
}

```



wav 파일 재생

```

// 숫자야구 랭킹 나열
void baseballscore()
{
    read_and_sort_baseball_data();

    FILE* fp_in;
    if (fopen_s(&fp_in, "sorted_baseball.txt", "r") != 0) {
        printf("파일이 열리지 않습니다.");
        return;
    }

    char name[MAX_LEN];
    int sec;
    printf("<< 명예의 전당 >>\n");
    printf("Name\ttime\n");
    printf("-----\n");
    while (fscanf_s(fp_in, "%s %d", name, MAX_LEN, &sec) == 2) {
        printf("%s\t%d\n", name, sec);
    }

    fclose(fp_in);
}

```



숫자야구 랭킹 출력 시스템


```

//숫자야구 점수 정렬
void read_and_sort_baseball_data()
{
    // 파일에서 데이터 읽기
    FILE* fp_in;
    if (fopen_s(&fp_in, "baseball.txt", "r") != 0) {
        printf("파일이 열리지 않습니다.");
        return;
    }

    int i, score[NUM_SCORES];
    char arr[NUM_SCORES][MAX_LEN];
    int count = 0;

    for (i = 0; i < NUM_SCORES; i++, count++)
    {
        fscanf_s(fp_in, "%s %d", arr[i], MAX_LEN, &score[i]);
        if (score[i] < 0)
            break;
    }

    fclose(fp_in);
}

```



숫자야구 랭킹
 저장 정보 정렬(1)

```

// 오름차순으로 정렬
int j, temp;
char temp_str[MAX_LEN];
for (i = 0; i < count; i++) {
    for (j = 0; j < count - i - 1; j++) {
        if (score[j] > score[j + 1]) {
            temp = score[j];
            score[j] = score[j + 1];
            score[j + 1] = temp;

            strcpy_s(temp_str, MAX_LEN, arr[j]);
            strcpy_s(arr[j], MAX_LEN, arr[j + 1]);
            strcpy_s(arr[j + 1], MAX_LEN, temp_str);
        }
    }
}

// 정렬된 데이터 파일에 쓰기
FILE* fp_out;
if (fopen_s(&fp_out, "sorted_baseball.txt", "w") != 0) {
    printf("파일이 열리지 않습니다.");
    return;
}

for (i = 0; i < NUM_SCORES; i++) {
    fprintf(fp_out, "%s %d\n", arr[i], score[i]);
}

fclose(fp_out);
}

```

← 숫자야구 랭킹
저장 정보 정렬(2)

```

//장애물피하기 점수 나열
void dinoscore()
{
    read_and_sort_dino_data();

    FILE* fp_in;
    if (fopen_s(&fp_in, "sorted_dino.txt", "r") != 0) {
        printf("파일이 열리지 않습니다.");
        return;
    }

    char name[MAX_LEN];
    int score;
    printf("<< 명예의 전당 >>\n");
    printf("Name\tScore\n");
    printf("-----\n");
    while (fscanf_s(fp_in, "%s %d", name, MAX_LEN, &score) == 2) {
        printf("%s\t%d\n", name, score);
    }

    fclose(fp_in);
}

```

← 장애물 피하기 랭킹 출력 시스템

```

//장애물피하기 랭킹 정렬
void read_and_sort_dino_data()
{
    // 파일에서 데이터 읽기
    FILE* fp_in;
    if (fopen_s(&fp_in, "dino.txt", "r") != 0) {
        printf("파일이 열리지 않습니다.");
        return;
    }

    int i, score[NUM_SCORES];
    char arr[NUM_SCORES][MAX_LEN];

    for (i = 0; i < NUM_SCORES; i++)
        fscanf_s(fp_in, "%s %d", arr[i], MAX_LEN, &score[i]);

    fclose(fp_in);
}

```



장애물 피하기
랭킹 저장
정보 정렬(1)

```

// 내림차순으로 정렬
int j, temp;
char temp_str[MAX_LEN];
for (i = 0; i < NUM_SCORES - 1; i++) {
    for (j = 0; j < NUM_SCORES - i - 1; j++) {
        if (score[j] < score[j + 1]) {
            temp = score[j];
            score[j] = score[j + 1];
            score[j + 1] = temp;

            strcpy_s(temp_str, MAX_LEN, arr[j]);
            strcpy_s(arr[j], MAX_LEN, arr[j + 1]);
            strcpy_s(arr[j + 1], MAX_LEN, temp_str);
        }
    }
}

// 정렬된 데이터 파일에 쓰기
FILE* fp_out;
if (fopen_s(&fp_out, "sorted_dino.txt", "w") != 0) {
    printf("파일이 열리지 않습니다.");
    return;
}

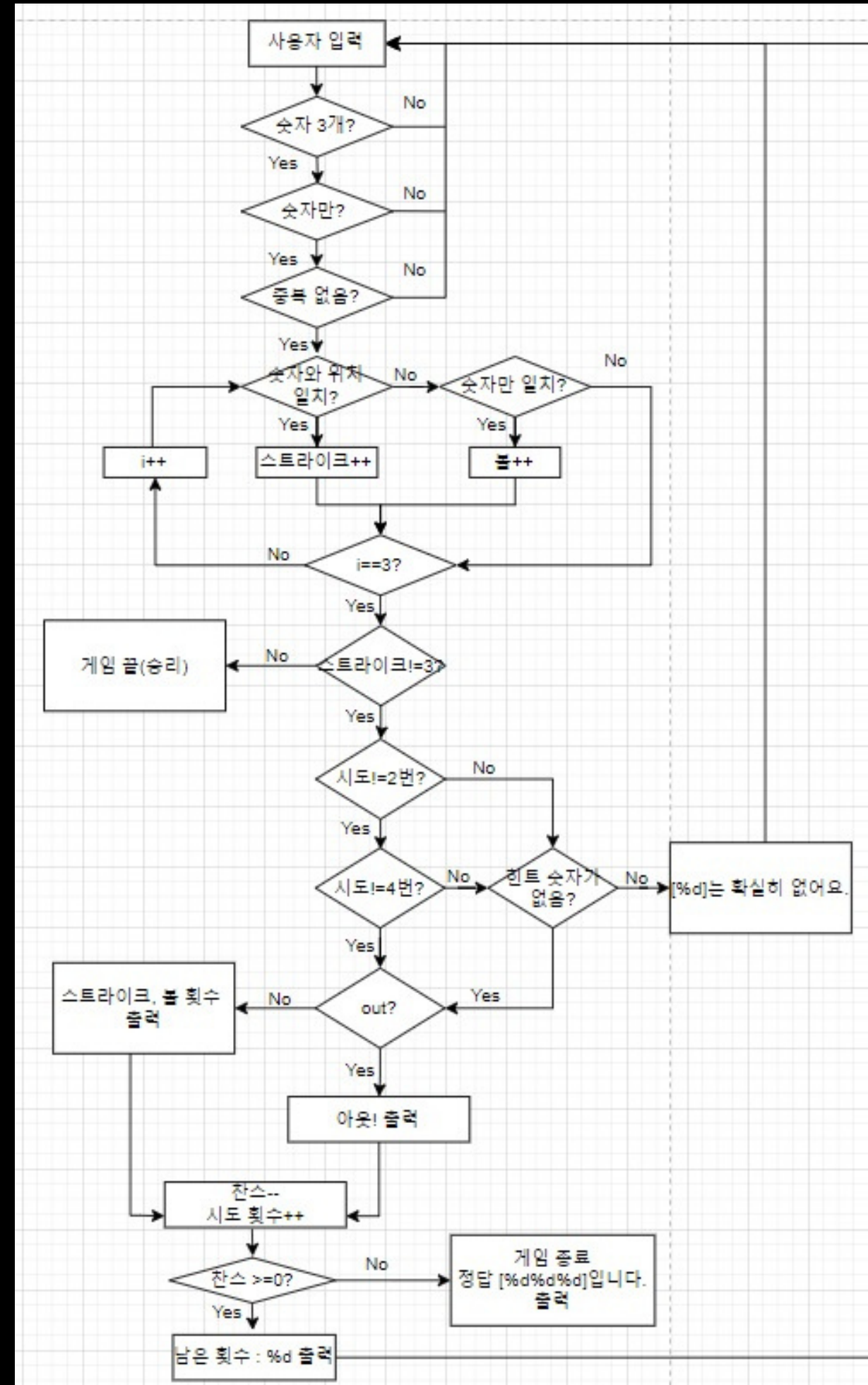
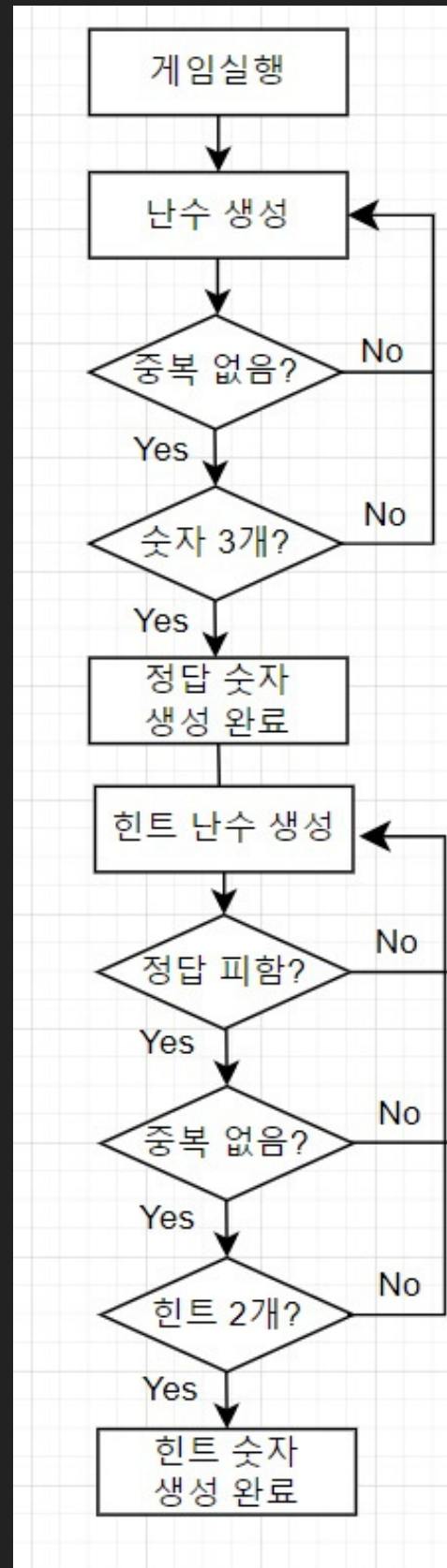
for (i = 0; i < NUM_SCORES; i++) {
    fprintf(fp_out, "%s %d\n", arr[i], score[i]);
}

fclose(fp_out);
}

```

← 장애물 피하기 랭킹 저장 정보 정렬(2)

숫자야구 게임- 순서도



숫자야구- 코드 설명

```
#pragma once
#define MAX_CHANCE 10
#define DIGIT 3
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void random_numbers(int random[], int hint[]);

void get_user(int user[], char usernumber[]);

void start_game(int random[], int hint[], int user[], int* chance, int* try_count, int* game_over);

void gameover();

void play_baseball();
```

← 헤더함수

time 함수: 시간을 설정할때 쓰이는 함수

stdlib 함수: 참/거짓을 판별해주는 함수

```
#include "game.h"
```

```
void random_numbers(int random[], int hint[])  
{  
    srand(time(NULL));  
    int i = 0;  
    while (i < 3)  
    {  
        random[i] = rand() % 10;  
        for (int j = 0; j < i; j++)  
        {  
            if (random[j] == random[i])  
            {  
                i--;  
                break;  
            }  
        }  
        i++;  
    }  
    for (int index = 0; index < 2; index++)  
    {  
        hint[index] = rand() % 10;  
        if (hint[index] == random[0]  
            || hint[index] == random[1]  
            || hint[index] == random[2])  
        {  
            index--;  
        }  
        if (index == 1 && hint[0] == hint[1])  
            index--;  
    }  
}
```



정답 생성



힌트 생성


```

int get_user(int user[], char usernumber[]) // 숫자입력받음
{
    printf("숫자 3개를 입력하세요. (중복 안됨)\n");

    while (1)
    {
        gets(usernumber);
        int digitCount = 0;
        for (; digitCount < 4; digitCount++)
        {
            if (usernumber[digitCount] == NULL)
            {
                break;
            }
        }
        if (digitCount != 3)
        {
            printf("3자리 입력\n");
            continue;
        }
        if (usernumber[0] < '0' || usernumber[0] > '9' || usernumber[1] < '0' || usernumber[1] > '9' || usernumber[2] < '0' || usernumber[2] > '9')
        {
            printf("잘못된 값.\n");
            continue;
        }
        for (int i = 0; i < 3; i++)
        {
            user[i] = usernumber[i] - '0';
        }
        if (user[0] == user[1] || user[0] == user[2] || user[1] == user[2])
        {
            printf("중복, 다시 입력해주세요.\n");
            continue;
        }

        break;
    }
}

```

☐ (char [36])"숫자 3개를 입력하세요. (중복 안됨)\n"
 온라인 검색

← 사용자의 입력
 오류 체크

```
void play_game(int random[], int hint[], int user[], int* chance, int* try_count, int* game_over)
{
    int strike = 0;
    int ball = 0;

    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            if (user[i] == random[j])
            {
                if (i == j)
                {
                    strike++;
                }

                else
                {
                    ball++;
                }
            }
        }
    }
}
```



스트라이크와 볼
카운트

```
if (strike == 3)
{
    printf("%d번째 시도에 성공했습니다.\n", *try_count);
    *game_over = 1;
    return;
}
if (*try_count == 2)
{
    for (int i = 0; i < 3; i++)
    {
        if (user[i] == hint[0])
        {
            printf("[ %d ]는 확실히 없어요\n", user[i]);
            return;
        }
    }
}
if (*try_count == 4)
{
    for (int i = 0; i < 3; i++)
    {
        if (user[i] == hint[0] || user[i] == hint[1])
        {
            printf("[ %d ]는 확실히 없어요\n", user[i]);
            return;
        }
    }
}
```



스트라이크인 경우 성공표시



힌트 주는 타이밍 설정

```
if (strike == 0 && ball == 0)
{
    printf("아웃!%t");
}

else
{
    printf("%d 스트라이크, %d 볼입니다.%t", strike, ball);
}

(*chance)--;
(*try_count)++;
```



아웃 및 스트라이크/볼 표시

```
#include "game.h"

int main()
{
    int random[3];
    int hint[2];
    random_numbers(random, hint);

    int chance = MAX_CHANCE;
    int try_count = 1;
    int game_over = 0;
    int user[3];
    char usernumber[1024];

    while (!game_over && chance >= 0)
    {
        printf("남은 횟수: %d\n", chance);
        get_user(user, usernumber);
        play_game(random, hint, user, &chance, &try_count, &game_over);
    }

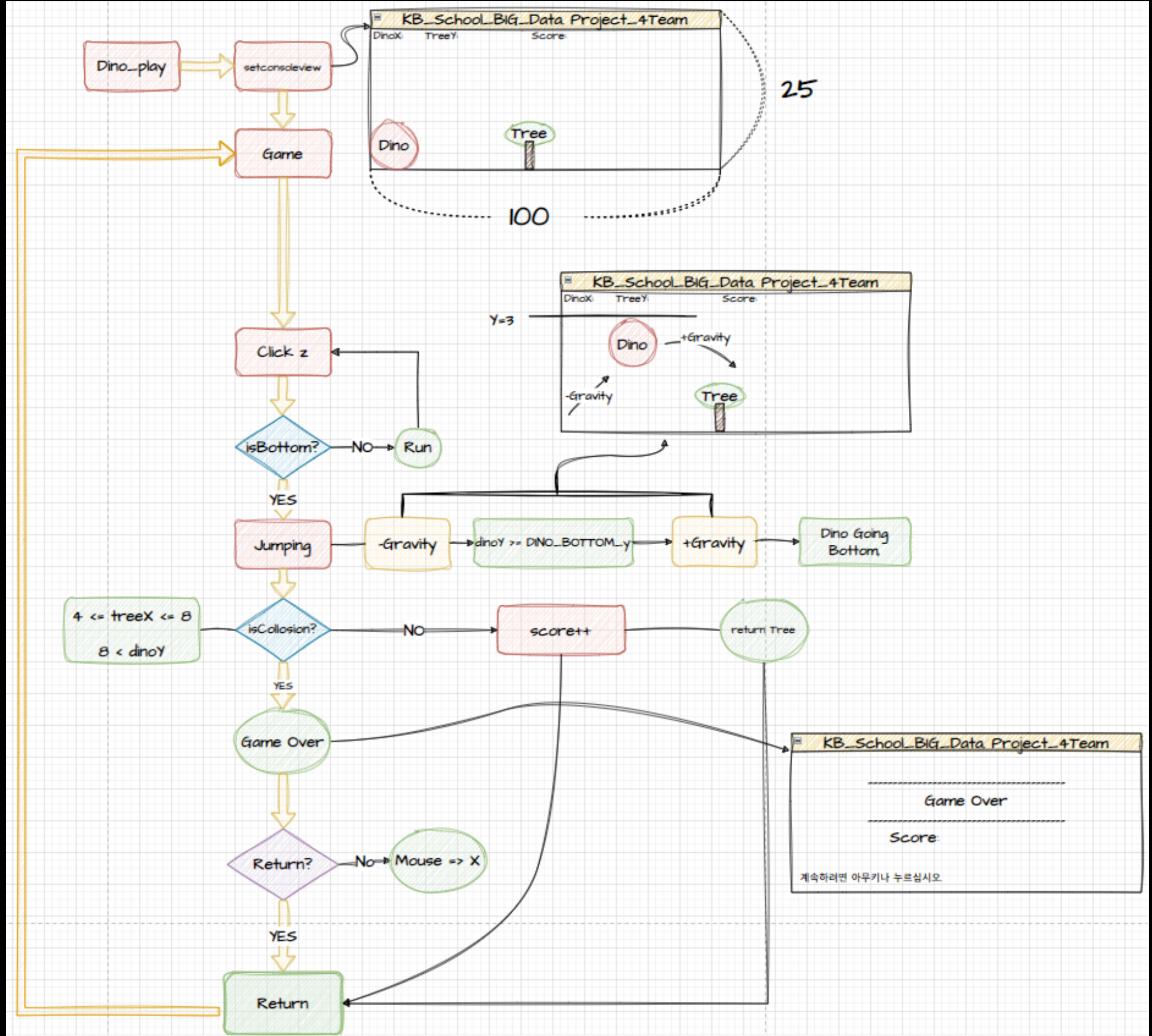
    if (chance == -1)
    {
        printf("\n게임 종료! %d번의 시도 내에 정답을 맞추지 못했습니다. 정답은 %d%d%d입니다.\n", MAX_CHANCE, random[0], random[1], random[2]);
    }

    return 0;
}
```



main

장애물 피하기- 순서도



장애물 피하기 게임- 코드 설명

```
1  #pragma once
2
3  #include<stdio.h>
4  #include<windows.h>
5  #include<conio.h>
6  #include<time.h>
7  #include<stdbool.h>
8  #define DINO_BOTTOM_Y 12
9  #define TREE_BOTTOM_Y 20
10 #define TREE_BOTTOM_X 45
11
12 //1
13 void SetConsoleView();
14 //2
15 void GotoXY(int x, int y);
16 //3
17 int GetKeyDown();
18 //4
19 void DrawDino(int dinoY);
20 //5
21 void DrawTree(int treeX);
22 //6
23 void DrawGameOver(const int score);
24 //7
25 bool isCollision(const int treeX, const int dinoY);
26 //8
27 void Dino_play();
```

헤더파일 5개
매크로 3개
함수 8개




```
1      #include "Dino_head.h"
2
3      int main()
4      {
5          Dino_play();
6          return 0;
7      }
```

← MAIN


```
1  #include "Dino_head.h"
2
3
4  void Dino_play()
5  {
6      SetConsoleView();
7
8      while (true)
9      {
10
11         bool isJumping = false;
12         bool isBottom = true;
13         const int gravity = 3;
14
15         int dinoY = DINO_BOTTOM_Y;
16         int treeX = TREE_BOTTOM_X;
17
18         int score = 0;
19         clock_t start, curr;
20         start = clock();
```

← 전체코드 함수
GAME LOOP

```
22 while (true)
23 {
24     if (isCollision(treeX, dinoY))
25         break;
26
27     if (GetKeyDown() == 'z' && isBottom)
28     {
29         isJumping = true;
30         isBottom = false;
31     }
32
33     if (isJumping)
34     {
35         dinoY -= gravity;
36     }
37     else
38     {
39         dinoY += gravity;
40     }
41
42     if (dinoY >= DINO_BOTTOM_Y)
43     {
44         dinoY = DINO_BOTTOM_Y;
45         isBottom = true;
46     }
```

← 한판에 대한
GAME LOOP

← 한판에 대한 GAME LOOP2

```
48     treeX -= 2;
49     if (treeX <= 0)
50     {
51         treeX = TREE_BOTTOM_X;
52     }
53
54     if (dinoY <= 3)
55     {
56         isJumping = false;
57     }
58
59     DrawDino(dinoY);
60     DrawTree(treeX);
61
62     curr = clock();
63     if (((curr - start) / CLOCKS_PER_SEC) >= 1)
64     {
65         score++;
66         start = clock();
67     }
68
69     Sleep(60);
70     system("cls");
71
72
73     GotoXY(22, 0);
74     printf("Score : %d ", score);
75 }
```

```

77         DrawGameOver(score);
78     }
79     return 0;
80
81
82 }

```

```

3  #pragma region 콘솔 창의 크기와 제목을 지정하는 함수
4  void SetConsoleView()
5  {
6      system("mode con:cols=100 lines=25");
7      system("title KB_School_BIG_Data. project_4Team");
8  }
9  #pragma endregion

```

```

11 #pragma region 커서의 위치를 x, y로 이동하는 함수
12 void GotoXY(int x, int y)
13 {
14     COORD Pos;
15     Pos.X = 2 * x;
16     Pos.Y = y;
17     SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), Pos);
18 }
19 #pragma endregion

```

```

23 #pragma region 키보드의 입력을 받고, 입력된 키의 값을 반환하는 함수
24
25 int GetKeyDown()
26 {
27     if (_kbhit() != 0)
28     {
29         return _getch();
30     }
31     return 0;
32 }
33 #pragma endregion

```



맨위 는 GAME OVER 표현

콘솔함수

커서이동 함수

키입력함수

```

66  #pragma region 나무를 그리는 함수
67  void DrawTree(int treeX)
68  {
69      GotoXY(treeX, TREE_BOTTOM_Y);
70      printf("$$$");
71      GotoXY(treeX, TREE_BOTTOM_Y + 1);
72      printf(" $$ ");
73      GotoXY(treeX, TREE_BOTTOM_Y + 2);
74      printf(" $$ ");
75      GotoXY(treeX, TREE_BOTTOM_Y + 3);
76      printf(" $$ ");
77      GotoXY(treeX, TREE_BOTTOM_Y + 4);
78      printf(" $$ ");
79  }
80  #pragma endregion

```

```

35  #pragma region 공룡을 그리는 함수
36  void DrawDino(int dinoY)
37  {
38      GotoXY(0, dinoY);
39      static bool legFlag = true;
40      printf("      $$$$$$ \n");
41      printf("    $ $$$$$$ \n");
42      printf("    $$$$$$$$ \n");
43      printf("$    $$$ \n");
44      printf("$ $ $$$$$$ \n");
45      printf("$ $ $$$$ \n");
46      printf("$ $ $$$$$$$$ \n");
47      printf("$ $$$$$$$$ \n");
48      printf("$ $$$$$$$$ \n");
49      printf(" $$$$$$ \n");
50      printf(" $$$$ \n");
51      if (legFlag)
52      {
53          printf("    $    $$$ \n");
54          printf("    $ $ ");
55          legFlag = false;
56      }
57      else
58      {
59          printf("    $$$ $ \n");
60          printf("    $ $ ");
61          legFlag = true;
62      }
63  }
64  #pragma endregion

```

←

함수
(나무, 공룡)

```

82  #pragma region 충돌 했을때 게임오버
83  void DrawGameOver(const int score)
84  {
85      system("cls");
86      int x = 18;
87      int y = 8;
88      GotoXY(x, y);
89      printf("=====");
90      GotoXY(x, y + 1);
91      printf("====G A M E O V E R====");
92      GotoXY(x, y + 2);
93      printf("=====");
94      GotoXY(x, y + 5);
95      printf("SCORE : %d", score);
96
97      printf("\n\n\n\n\n\n\n\n\n\n");
98      system("pause");
99  }
100 #pragma endregion

```



GAME OVER 함수

```

102  #pragma region 충돌했으면 true, 아니면 false
103  bool isCollision(const int treeX, const int dinoY)
104  {
105      GotoXY(0, 0);
106      printf("treeX : %d, dinoY : %d", treeX, dinoY);
107      if (treeX <= 8 && treeX >= 4 &&
108          dinoY > 8)
109      {
110          return true;
111      }
112      return false;
113  }
114  #pragma endregion

```

← 충돌함수

게임파일에 구현된 부가기능 - 메뉴입력

```
printf("\n\n\n\n\n\n\n\n");
printf("원하는 메뉴를 입력해주세요 (r = 재시작, e = 메뉴, w = 종료) : ");

char input;
do
{
    input = getchar();
} while (input != 'r' && input != 'e' && input != 'w');

if (input == 'r')
{
    Dino_play();
}
else if (input == 'e')
{
    system("cls");
    system("mode con cols=100 lines=40"); // 콘솔창 크기 조정
    playMusic("C:\\Users\\KB\\Desktop\\music.wav");
    printTxtFile("text.txt");
    menu();
}

else if (input == 'w')
{
    system("mode con cols=100 lines=40"); // 콘솔창 크기 조정
    printTxtFile("text.txt");
    printf("\n                게임이 종료되었습니다.\n");
    exit(0);
}
```

코드 실행 후
입력에 따른
메뉴 실행



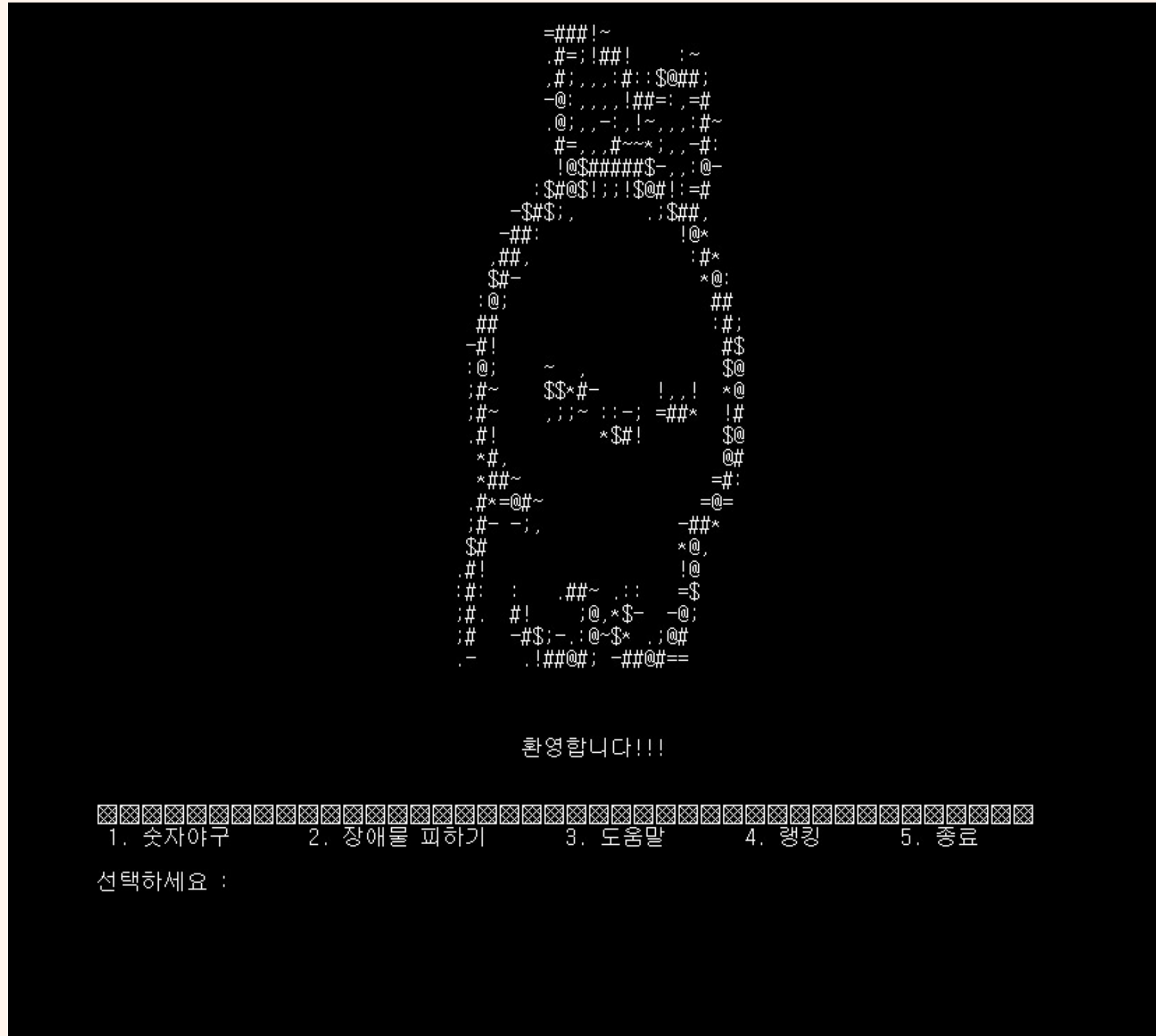

```

20 {
21     system("mode con:cols=100 lines=25");
22     printf("\n");
23     printf("\n");
24     printf("\n");
25     printf("\n");
26     printf("\n");
27     printf("\n");
28     printf("\n");
29     printf("\n");
30     printf("\n");
31     printf("\n");
32     printf("\n");
33     printf("\n");
34     printf("\t      3초후 게임을 시작합니다.\n");
35     Sleep(1000);
36     system("cls");
37     printf("\n");
38     printf("\n");
39     printf("\n");
40     printf("\n");
41     printf("\n");
42     printf("\n");
43     printf("\n");
44     printf("\n");
45     printf("\n");
46     printf("\n");
47     printf("\n");
48     printf("\n");
49     printf("\t      2초후 게임을 시작합니다.\n");
50     Sleep(1000);
51     system("cls");
52     printf("\n");
53     printf("\n");
54     printf("\n");
55     printf("\n");
56     printf("\n");
57     printf("\n");
58     printf("\n");
59     printf("\n");
60     printf("\n");
61     printf("\n");
62     printf("\n");
63     printf("\n");
64     printf("\t      1초후 게임을 시작합니다.\n");
65     Sleep(1000);
66 }

```

← 게임파일에 구현된 부가기능 카운트다운

프로젝트 수행 결과- main과 랭킹 시스템



<< 명예의 전당 >>

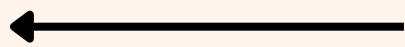
Name	time
sun	8
ldj	17
what	18



숫자야구 랭킹

<< 명예의 전당 >>

Name	Score
test1	29
test2	19
wow	1



장애물 피하기 랭킹

프로젝트 수행 결과- 숫자 야구

점화 횟수: 10

숫자 3개를 입력하세요. (중복 안됨)

123

0 스트라이크, 1 볼입니다.

남은 횟수: 9

숫자 3개를 입력하세요. (중복 안됨)

456

[6] 7. 없어요

수: 9
일: 70

숫자 3개를 입력하세요. (중복 안됨)

132

1 스트라이크, 0 볼입니다.

남은 횟수: 8

숫자 3개를 입력하세요. (중복 안됨)

123

0 스트라이크, 1 볼입니다.

남은 횟수: 7

숫자 3개를 입력하세요. (중복 안됨)

123

[1] 는 없어요

회 회 수 : 7

숫자 3개를 입력하세요. (중복 안됨)

456

[6] ㄴ 없어요

남은 횟수: 7

숫자 3개를 입력하세요. (중복 안됨)

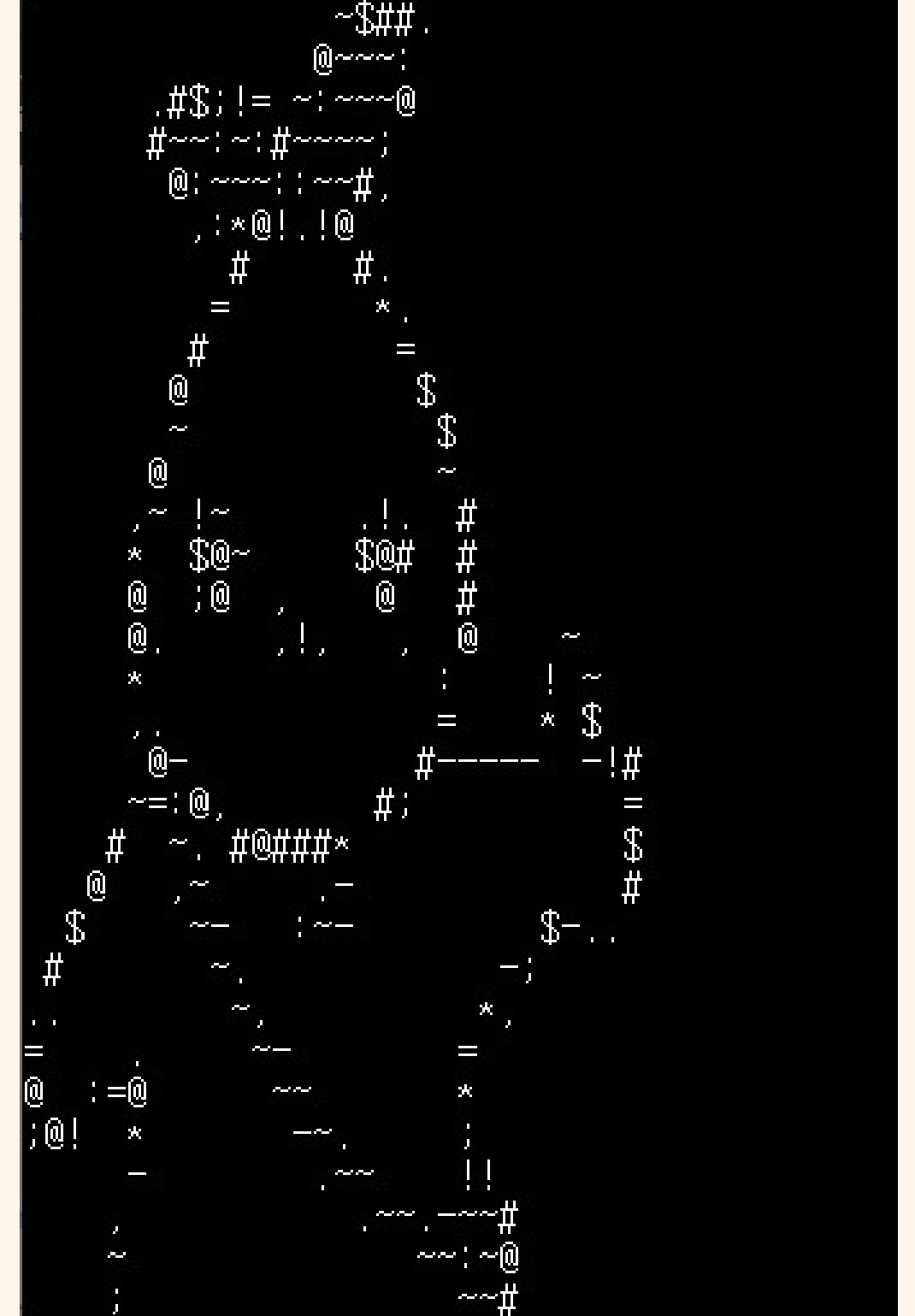
798

0 스트라이크, 1 볼입니다.

남은 횟수: 6



게임 종료! 10번의 시도 내에 정답을 맞추지 못했습니다. 정답은 902입니다.

원하는 메뉴를 입력해주세요 (r = 재시작, e = 메뉴, w = 종료) : 

4번째 시도에 성공했습니다.

소요된 시간은 8초입니다.

레이어 이름에 일괄해주세요(한글제외) :

프로젝트 수행 결과-장애물 피하기

```
treeX : 21, dinoY : 9           Score : 0

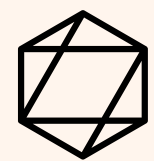
          $$$$$$
         $$ $$$$$$
        $$$$$$$$$$
       $$$
      $$$
     $$$$$$
    $$$$
   $$$$$$
  $$$$$$$$$$
 $$$$$$$$$$$$
$$$$$$$$$$$$
 $$$$$$$$$$
  $$$$$$
   $   $$$
    $$

          $$$$
          $$
          $$
          $$
          $$
```

```
=====
====G A M E O V E R====
=====

SCORE : 3

플레이어 이름을 입력해주세요(영어로만) :
```



자체 평가 의견



살라살라



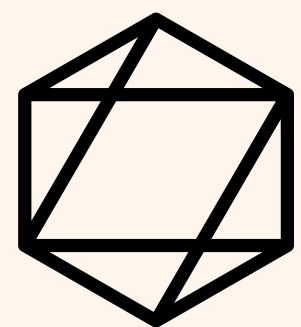
살라살라



살라살라



살라살라



THANK YOU!