



게임 (가위바위보 + 숫자게임)

전진원 (1560050)

김민성 (1560006)

09

목차

1. 브레인 스토밍
2. 주제 선정 및 프로그램 선택 동기
3. 역할분담
4. 알고리즘
5. 개발목표
7. 개발과정
8. 개발일정
9. 프로그램 소스
10. 프로그램 테스트
11. TERMPROJECT 소감

브레인 스토밍

1. 성적조회

2. 계산기
(학점 체 질량)

3. 길 찾기
(학과 찾기)

4. 가위바위보

5. 숫자게임

6. 퀴즈

주제 선정 및 프로그램 선택동기 [1/2]



GAME

- 숫자 게임
- 가위바위보



주제 선정 및 프로그램 선택동기 [2/2]

...게임산업 규모는 연간 130조원에 달한다.

출처 : 이코노믹리뷰 2015.03.09.

- 현재 게임산업의 규모가 점점 더 커지고 있기 때문에 브레인스토밍 한 것들 중에 게임을 선택하였다
-



역할분담

김민성

- 아이디어 도출
- '가위 바위 보' 프로그램 작성
- PPT작성
- Printf문과 scanf문 작성

전진원

- 아이디어 도출
- PPT작성
- '숫자 게임' 프로그램 작성
- '가위 바위 보'작성
- 구조체작성
- 파일처리 작성
- 프로그램 전체적인 틀 작성

프로그램 소개 -> Player이름 작성 -> 가위바위보 -> 숫자게임

프로그램 제목 -> 프로그램 소개 ->

Player이름 작성(TXT파일로 저장) -> 가위바위보 게임 소개 ->

가위바위보게임(스코어 보드 입력 키) ->

[가위바위보 반복 (1~~3번을 입력 시)
	가위바위보게임 재 설명 (잘못 입력 시)

가위바위보 게임 반복 중지 (4번 입력 시) -> 숫자 게임 소개 ->

숫자게임 크기 선택($N \times N$) -> 숫자게임 난이도 선택 ->

숫자게임 설명 및 게임 플레이 -> 게임의 끝을 알림 -> 숫자 게임 반복



가위 바위 보 알고리즘

- 가위 바위 보를 1~3으로 입력 받는다.
(#define으로 가위바위보를 숫자로 입력 받는다.)
- 4를 누르면 다음 게임으로 이동한다.
- 1~4번이 아닌 다른 숫자를 누를 시 잘못된 값을 출력한다.
(if문사용 if(number<=0 || number > 4))
- 컴퓨터의 가위 바위 보를 입력 받는다.
(rand()함수를 이용한다.)
- 사용자와 컴퓨터를 비교하여 승 무 패 결정한다.
- 스코어 보드에 승 무 패 결과를 출력한다.
- 위의 과정을 4번이 입력되기 전까지 반복한다.



숫자 게임 알고리즘

- 배열을 이용하여 게임의 자료를 저장한다.
- 퍼즐의 값을 섞는 함수를 만든다.
(rand()함수 이용)
- 숫자 이동을 시키는 함수를 만든다.
(#define와 case문 사용)
- 게임의 크기를 정하는 함수를 만든다.
- 정렬 상태를 확인한다.
- 위의 과정을 반복한다



파일 처리 함수 알고리즘

- `char a[100] ;` → `a`를 문자 형으로 선언
- `FILE *fstream;` → `fstream`을 파일로 선언
- `fstream = fopen("C:\\\\gametxt.txt", "a+");` → c드라이브에 txt 파일을 저장
- `scanf("%s",&a);` → `scanf()`로 `a`값을 입력 받는다.
- `fwrite(a, strlen(a), 1,fstream);` → `fwrite`를 이용한다.
(주소, 내용의 크기, 배열의 수, 공간정보)



개발목표

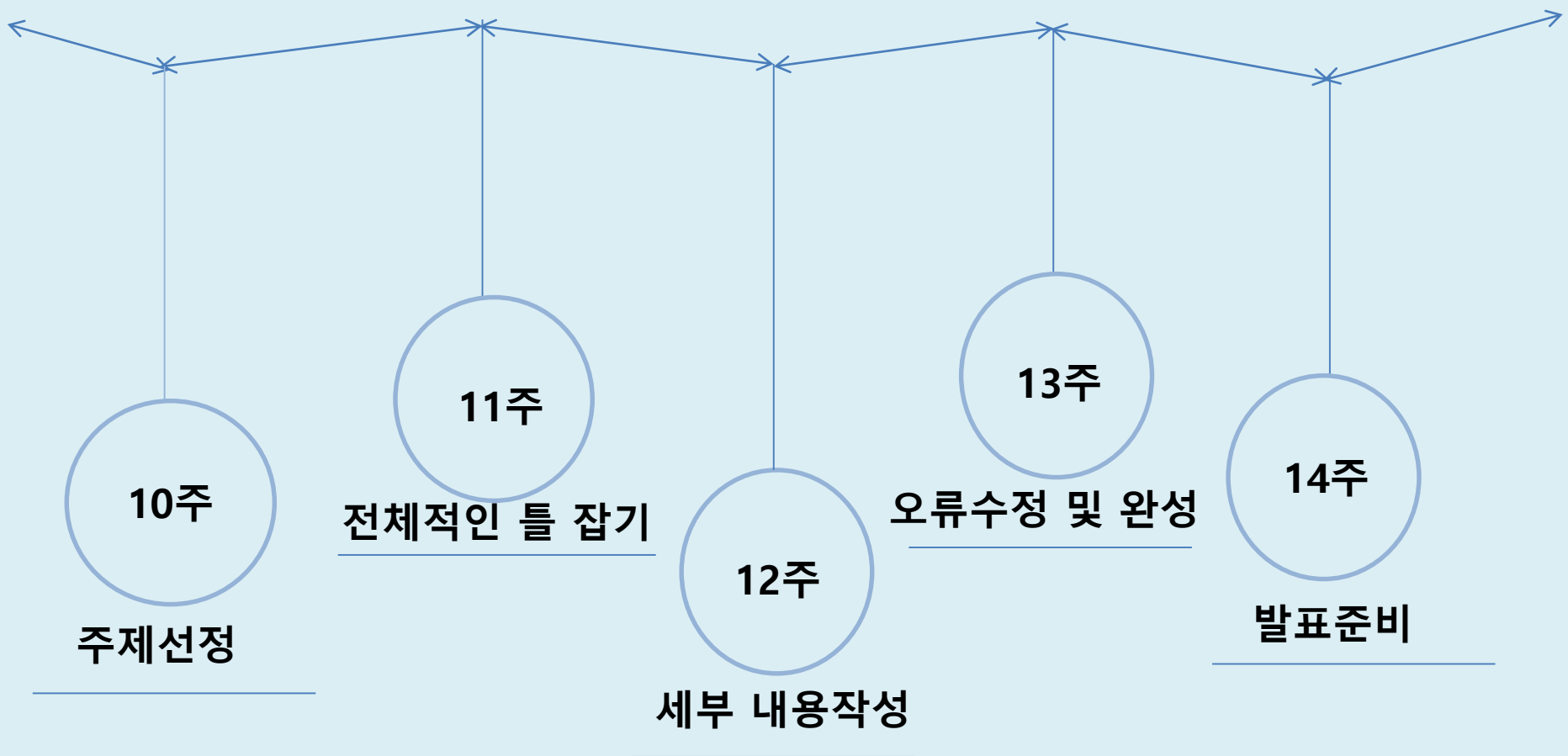
- **최종목표** 현재까지 배운 c언어 프로그래밍을 바탕으로 직접 프로그램을 개발하면서 c언어 프로그램 개발능력을 향상 시킨다
- **개발목표** 가위 바위 보 게임과 숫자게임 프로그램을 만들면서 1학년 동안 배운 것들을 최대한 활용한다.



개발과정

- 함수명과 전체적인 틀 잡기
- 파일처리로 Player 이름을 txt파일에 저장
- 구조체로 개발자 소개
- 가위 바위 보 프로그램 작성
- 숫자 퍼즐 게임 작성
- 두 게임을 연결 해주는 프로그램 작성
- 전체오류 수정

개발일정



전처리기 지시자

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

#define MAX 10
#define count 10
#define RIGHT 77 //오른쪽의 값
#define LEFT 75 //왼쪽의 값
#define DOWN 80 //아래의 값
#define UP 72 //위의 값
#define END 27 //ESC의 값

#define SCISSOR 1 //가위
#define ROCK 2 //바위
#define PAPER 3 //보
```

Main함수

```
void main(void)
{
    name1(); //프로그램 소개하는 함수
    producer_title(); //규조체로 개발자의 정보를 보여줌
    menu(); //파일 처리로 접속을 저장함
    finger_game(); //가위 바위 보 게임
    number_game(); //숫자 게임
}
```

가위바위보를 키로 입력하는 함수

```

void finger(void)
{
    int inputNum = 0;
    //< 승리 패배 무승부
    int winNumber = 0;
    int loseNumber = 0;
    int drawNumber = 0;
    //< 승패 여부
    int result = 0;
    //< 시드 재정렬

    while( inputNum != 4 )
    {
        printMenu( winNumber, loseNumber, drawNumber );
        inputNum = inputNumber();

        if ( inputNum == 4 ) continue;

        result = finger_computer ( inputNum );

        switch ( result )
        {
            case 1:
            {
                winNumber++;
            }
            break;
            case 2:
            {
                loseNumber++;
            }
            break;
            case 3:
            {
                drawNumber++;
            }
            break;
        }
    }

    return;
}

//가위바위보 게임 다시하는 함수
int inputNumber ( void )
{
    int number = 0;
    while ( number <= 0 || number > 4 )
    {
        printf("\n 입력 해주세요. : ");
        scanf("%d", &number );

        system("cls");

        if (number <= 0 || number > 4 )
        {
            printf("\n");
            printf("\n");
            printf("      잘못 입력 하셨습니다.\n");
            printf("\n");
            printf("\n");
            printf("      게임을 다시 설명합니다.\n");
            printf("\n");
            printf("\n");
            printf("-----\n");
            printf("      | 게임 설명\n");
            printf("      | 1번을 누를시 가위입니다.\n");
            printf("      | 2번을 누를시 바위입니다.\n");
            printf("      | 3번을 누를시 보입니다.\n");
            printf("      | 4번을 누를시 다음 게임으로 넘어 갑니다.\n");
            printf("-----\n");
            printf("\n");
            printf("\n");
        }
    }

    return number;
}

```

컴퓨터의 가위바위보 입력 받는 함수

```
int finger_computer ( int playerNumber )
{
    int computer = (rand() % 3) + 1;

    switch ( playerNumber )
    {
        case SCISSOR:
        {
            printf("\n");
            printf("\n");
            printf("   게임 상황\n");
            printf("   \n");
            printf("   사용자 : 가위\n");
            switch ( computer )
            {
                case SCISSOR:
                {
                    printf("   컴퓨터 : 가위\n");
                    printf("   무승부\n");
                    printf("   \n");
                    return 3;
                }
                break;
                case ROCK:
                {
                    printf("   컴퓨터 : 바위\n");
                    printf("   패배\n");
                    printf("   \n");
                    return 2;
                }
                break;
                case PAPER:
                {
                    printf("   컴퓨터 : 보\n");
                    printf("   승리\n");
                    printf("   \n");
                    return 1;
                }
                break;
            }
        }
        case ROCK:
        {
            printf("\n");
            printf("\n");
            printf("   게임 상황\n");
            printf("   \n");
            printf("   사용자 : 바위\n");
            switch ( computer )
            {
                case SCISSOR:
                {
                    printf("   컴퓨터 : 가위\n");
                    printf("   패배\n");
                    printf("   \n");
                    return 2;
                }
                break;
                case ROCK:
                {
                    printf("   컴퓨터 : 바위\n");
                    printf("   승리\n");
                    printf("   \n");
                    return 1;
                }
                break;
                case PAPER:
                {
                    printf("   컴퓨터 : 보\n");
                    printf("   무승부\n");
                    printf("   \n");
                    return 3;
                }
                break;
            }
        }
        case PAPER:
        {
            printf("\n");
            printf("\n");
            printf("   게임 상황\n");
            printf("   \n");
            printf("   사용자 : 보\n");
            switch ( computer )
            {
                case SCISSOR:
                {
                    printf("   컴퓨터 : 가위\n");
                    printf("   승리\n");
                    printf("   \n");
                    return 1;
                }
                break;
                case ROCK:
                {
                    printf("   컴퓨터 : 바위\n");
                    printf("   무승부\n");
                    printf("   \n");
                    return 3;
                }
                break;
                case PAPER:
                {
                    printf("   컴퓨터 : 보\n");
                    printf("   패배\n");
                    printf("   \n");
                    return 2;
                }
                break;
            }
        }
    }
}
```

```
case SCISSOR:
{
    printf("   컴퓨터 : 가위\n");
    printf("   승리\n");
    printf("   \n");
    printf("   \n");
    return 1;
}
break;
case ROCK:
{
    printf("   컴퓨터 : 바위\n");
    printf("   무승부\n");
    printf("   \n");
    printf("   \n");
    return 3;
}
break;
case PAPER:
{
    printf("   컴퓨터 : 보\n");
    printf("   패배\n");
    printf("   \n");
    printf("   \n");
    return 2;
}
break;
}
break;
case PAPER:
{
    printf("\n");
    printf("\n");
    printf("   게임 상황\n");
    printf("   \n");
    printf("   사용자 : 보\n");
    switch ( computer )
    {
        case SCISSOR:
        {
            printf("   컴퓨터 : 가위\n");
            printf("   패배\n");
            printf("   \n");
            printf("   \n");
            return 2;
        }
        break;
        case ROCK:
        {
            printf("   컴퓨터 : 바위\n");
            printf("   승리\n");
            printf("   \n");
            printf("   \n");
            return 1;
        }
        break;
        case PAPER:
        {
            printf("   컴퓨터 : 보\n");
            printf("   무승부\n");
            printf("   \n");
            printf("   \n");
            return 3;
        }
        break;
    }
}
break;
}
```


숫자 게임 초기 모양을 만드는 함수

□ //숫자게임의 초기 모양을 주는 함수

L

□ void number_first(void)

```
{
    int i, j; // 순환을 위한 변수
    int x1, y1, x2, y2; // 두 개의 숫자를 섞기위한 변수
    int temp; // 변수를 섞을 때 사용되는 임시 변수

    //퍼즐배열에 숫자 초기값 넣기

    current_x = 0;
    current_y = 0;

    // for문을 두 번 돌리면 정방형 정렬에 0부터 차례대로

    for (i=0; i<length; i++)
        for (j=0; j < length; j++)
            puzzle[i][j] = j + i * length;

    □ // rand()함수를 이용하여 x1,y1,x2,y2를 임의의 숫자가 발생
    // 두 수의 좌표를 length로 나눈 나머지를 이용하여 난수의

    //퍼즐 배열 섞기

    for (i=0; i < difficult; i++)
    {

    //랜덤함수로 난수 사용
    |
        x1 = rand() % length;
        y1 = rand() % length;
        x2 = rand() % length;
        y2 = rand() % length;
```

```
temp = puzzle[y1][x1];
puzzle[y1][x1] = puzzle[y2][x2];
puzzle[y2][x2] = temp;
```

```
// 0의 값을 갖는 좌표를 항상 모니터링하기 위하여 만약 이동 시킨 값이 0 이면 current_x와 current_y값을 바꾼다
if (puzzle[y1][x1]==0)
{
    current_y = y1;
    current_x = x1;
}
else
if (puzzle[y2][x2]==0)
{
    current_y = y2;
    current_x = x2;
}
}
```

// 퍼즐배열의 초기값 출력

```
for (i=0; i<length; i++)
{
    for (j=0; j < length; j++)
    {
        printf("%2d ",puzzle[i][j]);
    }
    printf("\n");
}
numgameplay();
}
```

방향키를 입력 받는 함수

```

void number_key(int key)
{
    int i, j;

    switch(key)
    {
        case LEFT :
            if ( current_x < length-1 )
            {
                puzzle[current_y][current_x] =
                puzzle[current_y][current_x + 1];
                current_x++;
            }
            break;

        case RIGHT :
            if ( current_x > 0 )
            {
                puzzle[current_y][current_x] =
                puzzle[current_y][current_x - 1];
                current_x--;
            }
            break;

        case UP :
            if ( current_y < length-1 )
            {
                puzzle[current_y][current_x] =
                puzzle[current_y + 1][current_x];
                current_y++;
            }
            break;

        case DOWN :
            if ( current_y > 0 )
            {
                puzzle[current_y][current_x] =
                puzzle[current_y - 1][current_x];
                current_y--;
            }
            break;

        case END :
            printf("\n");
            printf("\n");
            printf("숫자 퍼즐 게임을 그만 합니다.");
            printf("\n");

            number_finish();

    }

    puzzle[current_y][current_x] = 0;
}

```

```

// 퍼즐배열 출력
for (i=0; i<length; i++)
{
    for (j=0; j < length; j++)
    {
        printf("%2d ",puzzle[i][j]);
    }
    printf("\n");
}
printf("\n");

int number_check(void)
{
    int i, j;

    //퍼즐배열의 값들이 정렬되었는지 체크

    for (i=0; i < length; i++)

        for (j=0; j < length; j++)

            if (puzzle[i][j] != j+i*length)

                return 1;

    return 0;
}

```

Player의 이름이 txt파일에 저장되는 함수(파일처리)

```

void menu (void)
{
    char a[100] ;
    FILE *fstream;
    fstream = fopen("C:\gametxt.txt", "a+" );

    if (fstream)
    {
        printf("\n");
        printf("\n");
        printf("\n");
        printf("          FBI warning\n");
        printf("\n");
        printf("          게임을 플레이 하시면 접속 기록이 남습니다. \n");
        printf("          접속 기록이 팬찮으시면 이름을 입력하세요. \n");
        printf("\n");
        printf("\n");
        printf("\n");
        printf("          앞에서 부터 3글자 입력합니다.\n");
        printf("\n");
        printf("          Player의 이름을 입력하세요 : ");
        scanf("%s",&a);

        system("pause");
        system("cls");

        fwrite( a, strlen(a), 1,fstream);

        printf("\n");
        printf("\n");
        printf("          FBI warning\n");
        printf("\n");
        printf("\n");
        printf("          c드라이브 gametxt.txt 파일에 접속 기록이 남았습니다. \n");
        printf("          현재 player의 이름은 %s입니다. \n",&a);
        printf("\n");
        printf("\n");
        printf("\n");
        system("pause");
        system("cls");
    }

    fclose(fstream);
}
    
```

개발자 정보를 받는 함수 (구조체)

```
//구조체 사용
struct student
{
    char producer[100];
    char producename1[100];
    int producernumber1;
    char producename2[100];
    int producernumber2;
};
```

```
//-----구조체를 이용한 개발자 소개-----

int producer_title(void)
{
    struct student s1;

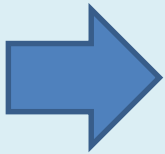
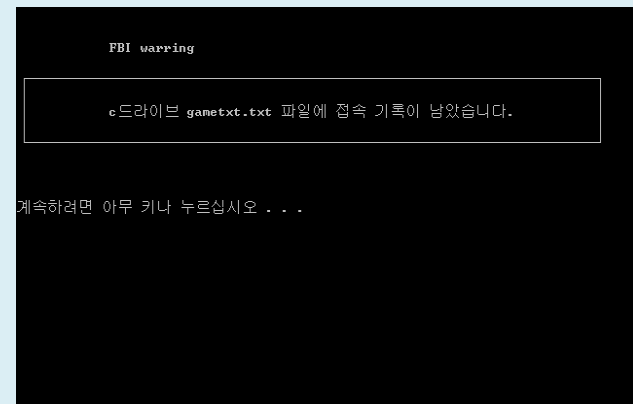
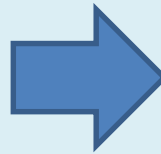
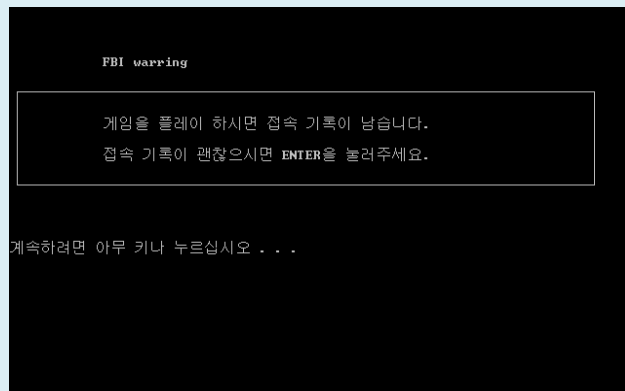
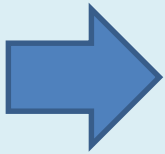
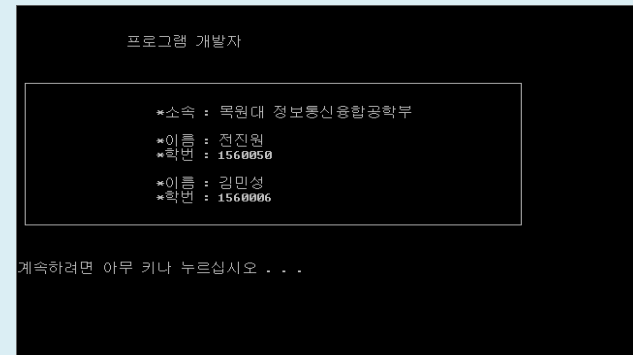
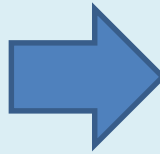
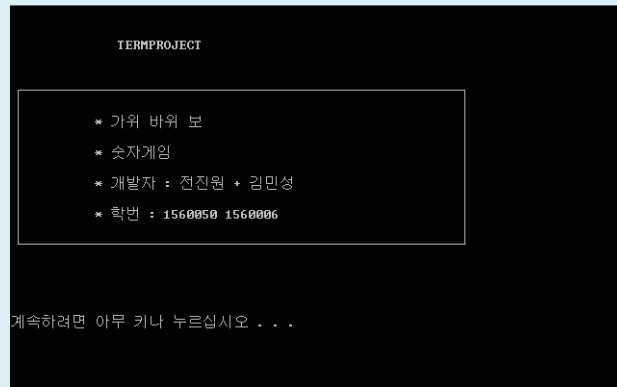
    strcpy(s1.producer, "목원대 정보통신융합공학부");
    strcpy(s1.producename1, "전진원");
    s1.producernumber1 = 1560050;
    strcpy(s1.producename2, "김민성");
    s1.producernumber2 = 1560006;

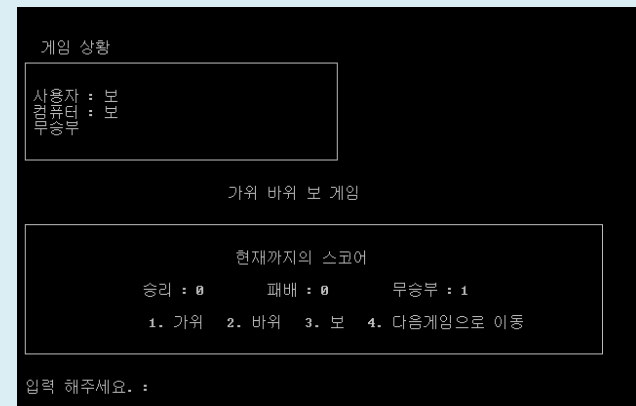
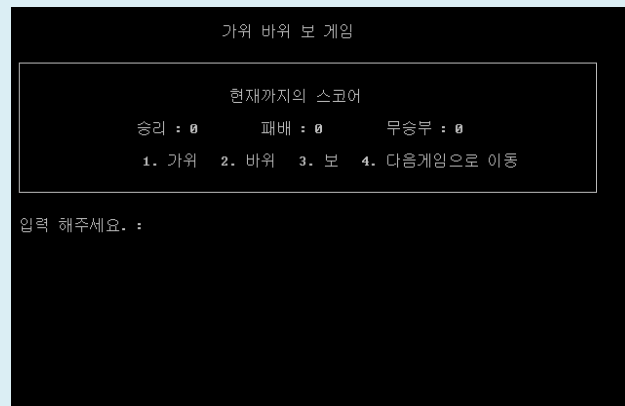
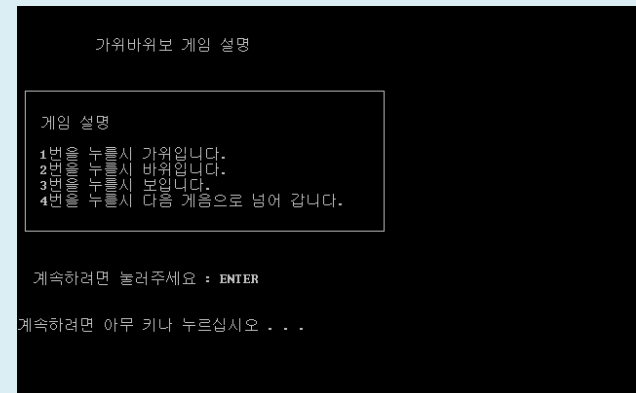
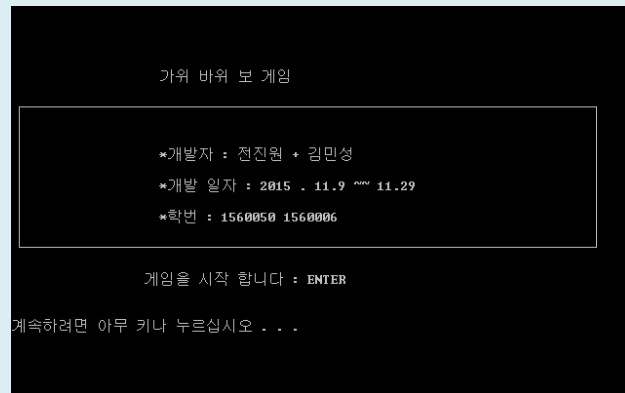
    printf("\n");
    printf("\n");
    printf("        프로그램 개발자 \n");
    printf("\n");
    printf("\n");
    printf("        \n");
    printf("        +소속 : %s          | \n", s1.producer);
    printf("        +이름 : %s          | \n", s1.producename1);
    printf("        +학번 : %d          | \n", s1.producernumber1);
    printf("        \n");
    printf("        +이름 : %s          | \n", s1.producename2);
    printf("        +학번 : %d          | \n", s1.producernumber2);
    printf("        \n");
    printf("\n");
    printf("\n");

    system("pause");
    system("cls");

    return 0;
}
```

프로그램 테스트 [1/5]





잘못 입력 하셨습니다.

게임을 다시 설명합니다.

게임 설명

- 1번을 누를시 가워입니다.
- 2번을 누를시 바워입니다.
- 3번을 누를시 보워입니다.
- 4번을 누를시 다음 계층으로 넘어 갑니다.

입력 해주세요. :

*숫자 퍼즐 게임

- * 개발자 : 전진원 + 김민성
- * 개발 일자 : 2015 . 11.9 ~ 11.29
- * 학번 : 1560050 1560006

게임을 시작 합니다 : ENTER

계속하려면 아무 키나 누르십시오 . . .

숫자게임

- *1 부터 나열하는 게임입니다.
- *3x3 부터 9x9 까지 크기를 선택할 수 있습니다.

ex> 3x3

1	2	3
4	5	6
7	8	0

- *1 부터 9까지 난이도를 고를 수 있습니다.

게임을 시작 합니다 : ENTER

게임의 크기를 고르세요.

n x n 퍼즐입니다.

크기를 선택하세요. <몇 x 몇 퍼즐>

단 3 x 3 ~ 9 x 9 중에 가능합니다.

3 ~ 9 사이의 숫자를 입력하세요 :

게임의 난이도를 입력하세요.

난이도를 입력하세요.

단 1 ~ 9 중에 선택 가능 합니다.

1 ~ 9 사이의 숫자를 입력하세요 :

```
0 3 2
1 4 8
6 5 7
```

게임 설명

1부터 숫자를 순서대로 나열하는 게임입니다.

게임의 시작은 위의 모양과 같습니다
0은 빈칸을 나타냅니다
방향키를 사용하여 플레이 합니다
게임을 종료하고 싶으시면 **esc**를 눌러주세요.
게임 종료후 바로 재게임이 실행됩니다.

게임이 끝나셨습니다.

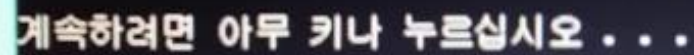
수고하셨습니다.

게임을 다시 하시려면 아무키나 누르세요.

개발자 : 전진원 * 김민성

계속하려면 아무 키나 누르십시오 . . .

테스트 영상



김민성



C언어가 많이 미흡해 하기에 많이 벅찬 과제라 생각했지만 조원이랑 같이하면서 모르는 것도 배워나가고 조금 더 알아가는 시간이라 너무 좋았습니다. 하나의 좋은 추억이라 생각하고 열심히 했습니다.

전진원



과제를 하면서 1년간 배운 것들을 다시 한번 공부하는 계기가 되어서 아직 미흡한 것들이 어느 부분인지 알게 되었습니다. 그리고 처음에는 프로그램을 짜면서 오류가 많이 나오는데 그 오류들을 찾아 보고 들어가면서 오류의 수를 점점 줄여 오류가 '0'이 되고 성공이 됐을 때 기분이 좋았습니다.

Q & A

THANK