



”

# 기초프로그래밍 프로젝트

- I. C언어 기초문법을 활용한 홀 맨 점프 게임
- II. C언어 기초문법을 활용한 추억의 2048게임

[발표시작 >](#)

보기만해도  
흐뭇한

# 2조

조원소개 >



도광현

Main함수 제작  
함수수정 및 함수화



박지호

새로운 숫자 생성 함수제작  
gameovercheck함수 제작



노민영

draw함수 디자인 제작



장혜정

방향키 조작 함수 제작



# 발표 순서

2조의 발표 순서는 말이죠!

”

01

## 프로젝트 개요

- 프로젝트 개발 일정 및 과정
- 프로젝트 관련 이론
- 게임 소개

02

## 프로젝트 수행절차

- 알고리즘
- 코드작성

03

## 프로젝트 수행결과

- 게임실행 화면

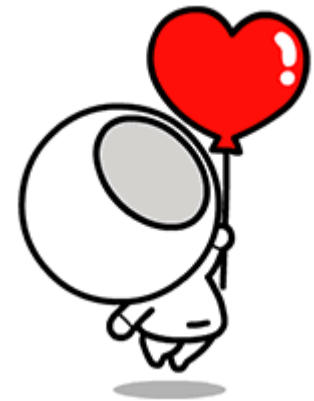
04

## 자체 평가의견

● ● ○

# I. 홀 맨 점프 게임

## II. 2048 게임





”

# 프로젝트 개요

# 01

- 프로젝트 개발 일정 및 과정
- 프로젝트 관련 이론
- 홀 맨 점프 게임소개



# 프로젝트 개발 일정

콘솔창으로 홀 맨 점프 게임 구현하기



**8 . 9일**

개발방향 및 의견 수립

**8 . 10일**

알고리즘 작성  
코딩작성 및 구현하기

**8 . 11일**

통합테스트

# 콘솔 창을 활용한 홀 맨 점프 게임

● ● ○ 공통 게임을 참고하여 홀 맨의 이미지를 printf로 구현

장애물인 나무를 점프로  
뛰어 넘으며 점수를 얻어가는 게임이야.  
단계가 높아질 수록 홀 맨의  
달리는 속도도 빨리 지기 때문에  
온전히 게임에 집중해야 성공할 수 있어😊





”

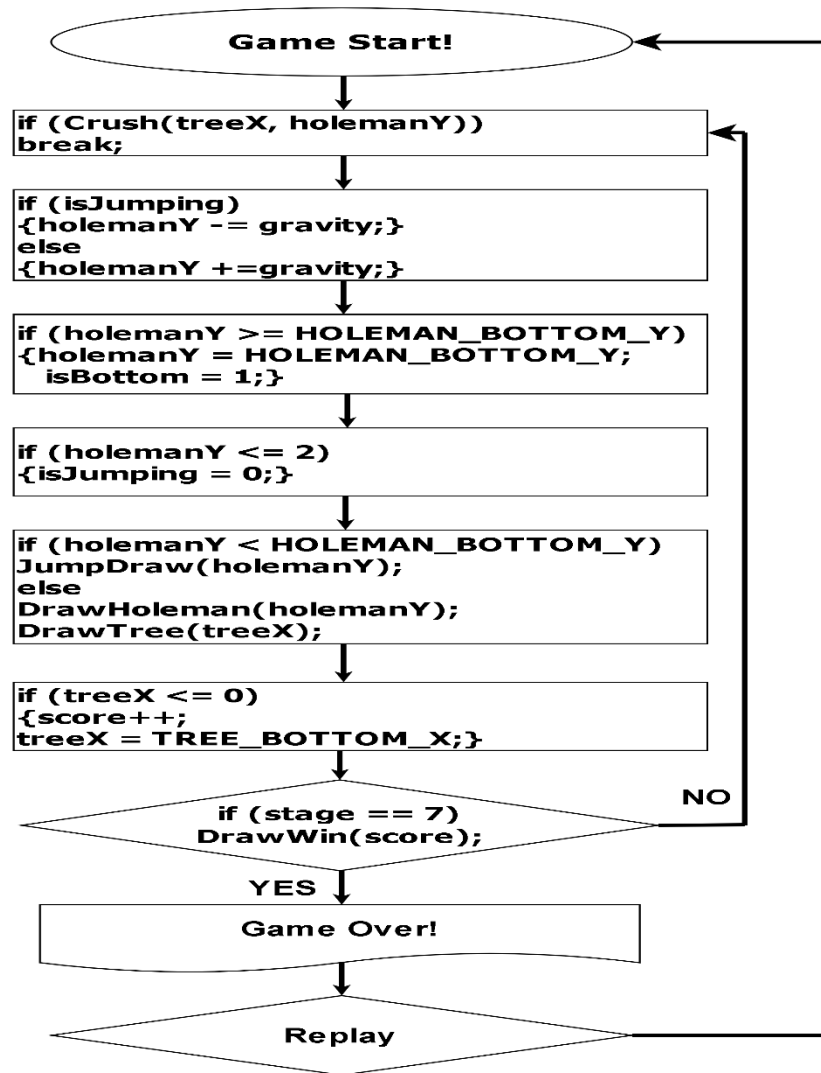
# 프로젝트 수행절차

# 02

- 홀 맨 점프 게임 알고리즘
- 홀 맨 점프 게임 코드작성



# 홀 맨 점프 게임 코드 알고리즘



## GetKeyDown 함수 코드 & Crush 함수 코드

```
void GetKeyDown()
{
    if (_kbhit() != 0)
    {
        int nkey;
        nkey = _getch();
        if (nkey == 'z' && isBottom)
        {
            isJumping = 1;
            isBottom = 0;
        }
        else if (nkey == ESC)
        {
            system("cls");
            exit(0);
        }
    }
}
```

**GetKeyDown 함수**

```
boolean Crush(const int treeX, const int holemanY)
{
    GotoXY(0, 0);

    if (treeX <= 10 && treeX >= 6 && holemanY > 8)
    {
        return 1;
    }

    return 0;
}
```

**Crush 함수**

# 홀 맨 점프 게임 Main 코드

```
while (1)
{
    const int gravity = 3;
    int speed = 60;
    int holemanY = HOLEMAN_BOTTOM_Y;
    int treeX = TREE_BOTTOM_X;

    int score = 0;

    while (1)
    {
        if (Crush(treeX, holemanY))//
            break;

        GetKeyDown();

        if (isJumping)
        {
            holemanY -= gravity;
        }
        else
        {
            holemanY += gravity;
        }

        if (holemanY >= HOLEMAN_BOTTOM_Y)
        {
            holemanY = HOLEMAN_BOTTOM_Y;
            isBottom = 1;
        }
    }
}
```

01

**Main 코드**

홀 맨 점프 코드

```
treeX -= 2;
if (treeX <= 0)
{
    score++;
    treeX = TREE_BOTTOM_X;
}

if (holemanY <= 2)
{
    isJumping = 0;
}
if (holemanY < HOLEMAN_BOTTOM_Y)
    JumpDraw(holemanY);
else
    DrawHoleman(holemanY);
DrawTree(treeX);

Sleep(speed - (stage - 1) * 10);
system("cls");
stage = score / 10+1;

if (stage == 7)
    DrawWin(score);

GotoXY(22, 0);
printf("Score : %d \n", score);
GotoXY(23, 1);
printf("%d 단계", stage);

}

DrawGameOver();
}
return 0;
```

02

**Main 코드**

장애물(나무) 코드  
단계 별 속도 코드



”

# 프로젝트 수행결과

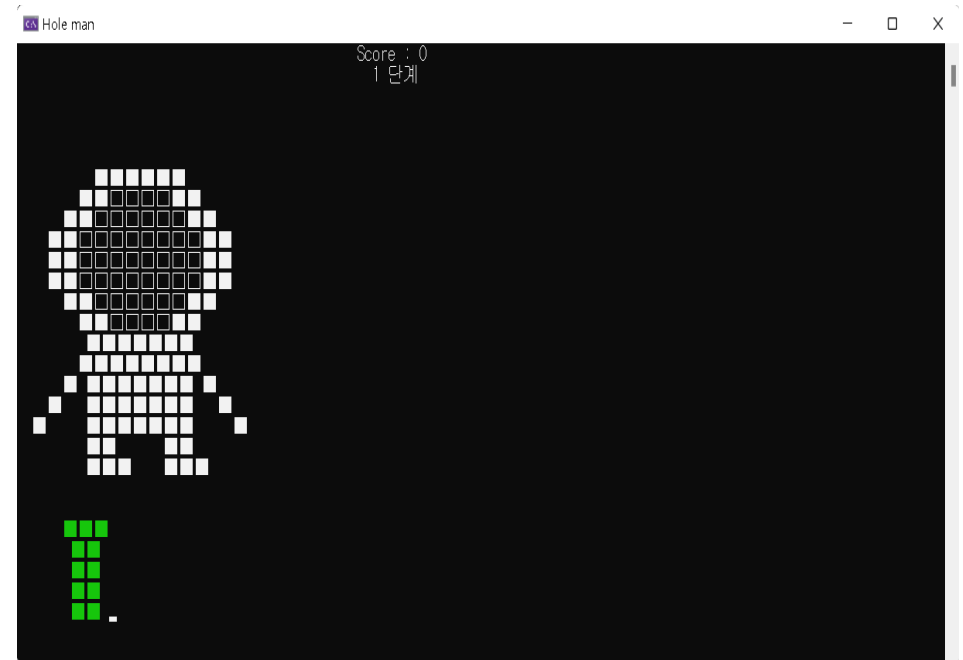
# 03

- 홀 맨 점프 게임 실행화면

# 홀 맨 점프 게임 코드 실행화면



걷는 모습



점프하는 모습

홀 맨 점프 게임 코드 실행화면



**작동화면은  
직접 보여드리겠습니다:)**

**잠시만 기다려주세요!**

● ● ○

# I. 홀 맨 점프 게임

## II. 2048 게임





”

# 프로젝트 개요

# 01

- 프로젝트 개발 일정 및 과정
- 프로젝트 관련 이론
- 2048게임방법 소개



# 프로젝트 개발 일정

콘솔창으로 2048 게임 구현하기



7.29일, 8.1일

개발방향및의견수립

8.2일 - 8.4일

알고리즘작성  
코딩작성및구현하기

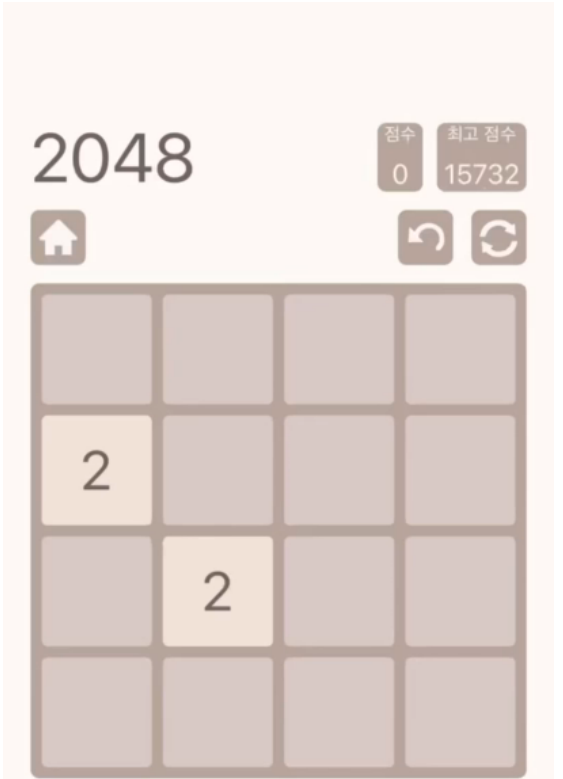
8.3일 - 8.5일

통합테스트

## 프로젝트 관련 이론

● ● ○	
☑ 조건문	조건문을 사용하여 게임에 필요한 동작을 구현할 수 있음
☑ <b>Play sound</b> 함수	Play sound함수를 사용하여 원하는 시간대에 음악을 출력할 수 있음
☑ 포인터	메모장을 이용하여 내가 원하는 데이터를 저장하고 내려 받을 수 있음
☑ <b>구조체</b>	콘솔 창 커서 위치를 원하는 곳으로 위치시키고 내용을 입력할 수 있음

# 추억의 2048게임 방법 소개

	<p>1</p> <p>4 X 4의 판 위에 2또는 4 중 랜덤으로 숫자 2개가 생성되면서 게임이 시작된다.</p>	<p>2</p> <p>위, 아래, 오른쪽, 왼쪽 중 한 방향으로 몰면서 같은 숫자가 나오면 그 두 숫자는 합해진다.</p>
	<p>3</p> <p>숫자가 움직일 때 마다 랜덤으로 빈 자리에는 2 또는 4가 생성되면서 게임이 계속 진행된다.</p>	<p>4</p> <p>빈 공간이나 더 이상 같은 숫자가 없어 움직일 수 없으면 게임은 끝이 난다.</p>



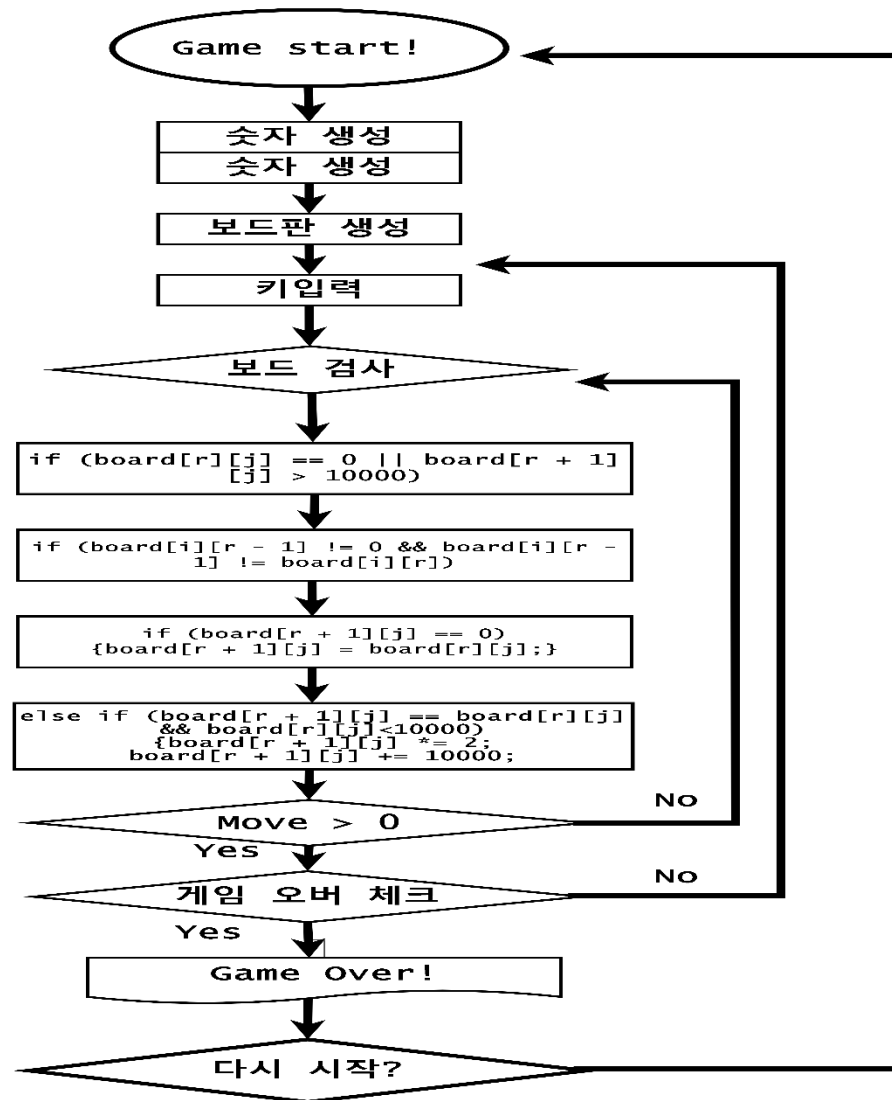
”

# 프로젝트 수행절차

# 02

- 2048 알고리즘
- 2048 코드작성

# 2048 코드 알고리즘



# Draw함수 코드



```
1  for (i = 0; i < 4; i++)
1  {
1      for (j = 0; j < 4; j++)
1      {
1          if (board[i][j] == 0)
1          {
1              gotoxy(MAP_ADJ_X + j*5+25, MAP_ADJ_Y + i*2-1);
1              printf("  .");
1          }
1          else
1          {
1              if (board[i][j] > 10000)
1              {
1                  gotoxy(MAP_ADJ_X + j*5+25, MAP_ADJ_Y + i*2-1);
1                  printf("%5d", board[i][j] - 10000);
1              }
1              else
1              {
1                  gotoxy(MAP_ADJ_X + j*5+25, MAP_ADJ_Y + i*2-1);
1                  printf("%5d", board[i][j]);
1              }
1          }
1      }
1      printf("\n");
1  }
1  printf("\n");
1  gotoxy(MAP_ADJ_X + 80, MAP_ADJ_Y );
1  printf("Score = %d\n", score);
1  gotoxy(MAP_ADJ_X + 80, MAP_ADJ_Y+1 );
1  printf("Max Score = %d", max);
1  }
```

## 오른쪽 버튼 코드 ➡

```
void rightbutton()
{
    for (int i = 0 ; i < 4; i++)
    {
        for (int j = 2; j >= 0; j--)
        {
            for (int r = j; r < 3; r++)
            {
                if (board[i][r] == 0)
                {
                    break;
                }
                if (board[i][r + 1] != 0 && board[i][r] != board[i][r + 1])
                {
                    continue;
                }
                if (board[i][r + 1] == 0)
                {
                    board[i][r + 1] = board[i][r];
                    board[i][r] = 0;
                    move++;
                }
                else if (board[i][r + 1] == board[i][r] && board[i][r] < 10000)
                {
                    board[i][r + 1] = board[i][r] * 2;
                    score += board[i][r]*2;
                    board[i][r + 1] += 10000;
                    board[i][r] = 0;
                    move++;
                }
            }
        }
    }
}
```

## 아래쪽 버튼 코드



```
void underbutton()
{
    for (int i = 2; i >= 0; i--)
    {
        for (int j = 0; j < 4; j++)
        {
            for (int r = i; r < 3; r++)
            {
                if (board[r][j] == 0)
                {
                    continue;
                }
                if (board[r + 1][j] != 0 && board[r][j] != board[r + 1][j]) //
                {
                    continue;
                }
                if (board[r + 1][j] == 0)
                {
                    board[r + 1][j] = board[r][j];
                    board[r][j] = 0;
                    move++;
                }
                else if (board[r + 1][j] == board[r][j] && board[r][j] < 10000)
                {
                    board[r + 1][j] = board[r][j] * 2;
                    score += board[r][j]*2;
                    board[r + 1][j] += 10000;
                    board[r][j] = 0;
                    move++;
                }
            }
        }
    }
    // ...
}
```



# 2048 게임 Main 코드

```
int main()
{
    //system("color F0");
    curserdelete();

    FILE* fp;
    fp = fopen("score.txt", "r");
    fscanf(fp, "%d", &temp_max);
    if (temp_max > max)
        max = temp_max;
    fclose(fp);

    score = 0;
    int key;
A:
    PlaySound(TEXT("tylenol.wav"), 0, SND_FILENAME | SND_ASYNC | SND_LOOP);
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            board[i][j] = 0;
        }
    }
}
```

01

**Main 코드**  
메모장 점수 저장 코드  
배경음악 코드

```
new_num();
new_num();
draw();

while(1)
{
    key = _getch();
    if (key == 0 || key == 0xE0)
    {
        key = _getch();
    }
    switch (key)
    {
    case LEFT:
        leftbutton();
        break;
    case RIGHT:
        rightbutton();
        break;
    case UP:
        upbutton();
        break;
    case UNDER:
        underbutton();
        break;
    }
}
```

02

**Main 코드**  
버튼입력 코드

```
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 4; j++)
    {
        if (board[i][j] > 10000)
            board[i][j] -= 10000;
    }
}
if (move > 0)
{
    new_num();
    draw();
    checkgameover();
    checkwin();
}
move = 0;

if (win == 1)
{
    gotoxy(MAP_ADJ_X + 80, MAP_ADJ_Y + 2);
    printf("Clear");
    break;
}
if (over == 1)
{
    PlaySound(NULL, 0, 0);
    PlaySound(TEXT("debuff.wav"), 0, SND_FILENAME | SND_ASYNC | SND_LOOP);
    char re;
    if (max < score)
        max = score;

    if (max > temp_max)
    {
        fp = fopen("score.txt", "w");
        fprintf(fp, "%d", max);
        fclose(fp);
    }
}
```

03

**Main 코드**  
Game over코드  
점수저장 코드



”

# 프로젝트 수행결과 03

- 2048 게임 실행화면

## 2048 게임 코드 실행화면

```
C:\Users\WKB\Desktop\Project7\#x64\Debug\Project7.exe

iBB      □□□ □□□ □ □ □□□      iBM
PgBs      □ □ □ □ □ □ □ □      BBQi
Bv g      □□ □ □ □ □ □ □ □ □      iq Bi
Bb Z      □ □ □ □ □ □ □ □      Q Bi
QR B      □□□ □□□ □ □ □ □      qv iB
dB i B    □□□□□□□□□□□□      gr BDi
D B Pv    □ □ □ □ □ □ □ □      B SZ gi
B Q B     □ . . . . . □      Xu uv vU
Bi BJ iB  □ . . . . . □      vB Bi Bi
iBi iB vB □ . . . . . □      UB RB BR
kr Zk BB Ps □ . . . . . □      izi KBi B B
UB Zr BBU iRv □ . 2 . . .      iDli iBBi dv iB
SdR iBv PBq Bv vPY □ . . . . . □      iKl BiBQv Bd vBj
B iB rBX BQ iBr iU □ . . . . . □      r BK iBr rBP sQ Pp
B iB rBX BQ iBr iUKv □ . . . . . □      r BK iBr rBP sQ Pp
sv vB iBQ jUi Lj i □□□□□□□□□□      r Sv il iiQ vBl Ss
iu kQL EBvQBv iQUi YPi uXi vDK DbugBv iZM Q
M iQS g Sv LXEBl dv Mr vuQQaki iXv jU rBu Ei
BJ vMivDi iXu iBr Qi Mi vBiiqv sD IP igb
LiBi iSPBBk JkkBQi Ki R sBKKUi iBBEKL dbii
ri sBi iBiiirvLrKR iVv VU vBrvvrrr Mu MQ U
gQ KQuBri irvgii Bi Bi iiXXrr iiQddgr iB
qKi iUBBBDLvJPBv Pkr ePv BBJjvv iBBBDr vMi
Lri vBBLi iVBjir vb iD viQMv iqBzi Pqi
VB irjaBB i Qi Ri qBQuLii iB
iBvsgi YXQBBR Pi bi BBBDui iUbrPZ
Ur s UBBqli ur iji uUBBdi rv Eaqwea
iMPRV ri XBPPviiiUvli irjZdBi l iBKDS
vUvB Q u u iJ uQrUK
iiiYvii Yilrviii
```



**작동화면은  
직접 보여드리겠습니다:)**

**잠시만 기다려주세요!**



”

자체 평가의견

04



# 자체 평가의견

기초프로그래밍 프로젝트

“



도 광 현

팀프로젝트를 하면서  
새로운 경험을 해본 것 같아서  
좋았다.

“



박 지 호

팀프로젝트를 하고 게임을  
직접 만들어보면서  
즐거웠고 C언어에 몰랐던  
부분들을 많이 알게 되었다.

“



노 민 영

팀프로젝트를 하며 깊이 알지 못했던  
함수를 다시 공부할 수 있었고,  
팀원들과 게임을 만드는 경험을 하여  
좋았다.

“



장 혜 정

팀 프로젝트를 하면서 수업시간에  
제대로 알지 못했던 부분들을  
보충할 수 있게 되었고 학창시절에  
하던 게임을 C언어로 직접 만들어  
봄으로써 더욱 뜻 깊었다.



”

**지금까지 상큼한**  
**2조의 발표였습니다 !**

**발표 끝 !**