

---

# 프로젝트 중간 발표 3

---

8팀

# 목차

**1** 기존의 두더지 잡기 코드

**2** 추가 및 개선과정

**두더지 생성 1**

DotMatrix에  
두더지를 랜덤하게  
생성하도록 함

**두더지 생성 2**

2초를 간격으로  
한 마리 씩  
나오도록 함

**Dot Matrix <-> tact sw 1**

tact sw와 Dot Matrix가  
연동되도록 코드를 작성

**Dot Matrix <-> tact sw 3**

12번 버튼을 누르면 종료

**두더지 생성 3**

1~9번 버튼을 누르면 Dot  
Matrix의 위치에 맞는  
두더지가 잡히고  
다음으로 넘어가게 됨

**Dot Matrix <-> tact sw 2**

1~9번 버튼을 누르면  
Dot Matrix의 위치에  
맞는 두더지가 잡히고  
다음으로 넘어가게 함

**Dot Matrix <-> tact sw 4**

10번 버튼을 통해  
아이템을 사용

# Part 1, 기존의 두더지 잡기 코드

Mole\_Game\_main.c

```
void clcd_input2(int score_player, int score_mole)
{
    int clcd;
    char player_score[5];
    char mole_score[5];
    char clcd_text1[30] = "Player_Score: ";
    char clcd_text2[30] = "Mole_Score: ";
```

```
//다음 단계로 넘어간다면
if (isNext == 1)
{
    nextmole = 0;
    isNext = 0;
    startmole = 1;
    stage++;
    //1stage에는 한마리, 2stage에는 2마리...
```

## void clcd\_input2

CLCD 장치를 연 후, 점수를 문자열로 변환 및 조합함. 이후 CLCD에 출력함.

## startmole

1일 시, 현재 스테이지의 두더지 위치를 도트 매트릭스에 표시함.

```
//스테이지 시작시 매트릭스에 두더지 표시
if (startmole == 1)
{
    startmole = 0;
    for (q = 0; q < stage; q++)
```

## isNext

변수가 1이면 새로운 스테이지의 두더지 위치와 시간을 설정함.

## nextmole

1일 시, 새 두더지 위치를 추가함.

```
if (nextmole == 1)
{
    nextmole = 0;
    for (p2 = 0; p2 < 8; p2++)
```

## 메인 게임 루프

도트 매트릭스의 전술 스위치를 읽고, 두더지를 잡았는지 검사함.  
이후 두더지를 잡으면 점수를 증가 시키고, 두더지를 재배치함.  
스테이지 완료 시 다음 스테이지로 넘어가며,  
게임 종료 조건인 점수 50이상을 검사하고 종료함.

# Part 1, 기존의 두더지 잡기 코드

Mole\_CLCD\_Item.c, Mole\_Stage.c

```
if (count == 10)    //10번 맞혔을때
{
    item ++;
    if(item >= 4)
    {
        item = 3;
    }
    count = 0;
}
```

```
if(item == 1)
{
    dev = open(led, O_RDWR);
    if(dev < 0) {printf("Can't Open\n"); exit(0);}
    data = 0x80;
    write(dev, &data, sizeof(unsigned char));
    usleep(3000000);
    if (num == 10)    // 10번 스위치 마이템 사용
    {
        close(dev);
        item--;
        num == 0;
        int x = 1;
        for(x;x<11;x++)    // 두더지 10마리 자동으로 잡기
        {
```

## Mole\_CLCD\_Item.c

두더지를 10번 맞출 시, 10번 스위치를 눌러 두더지를 10번 자동으로 잡을 수 있는 아이템 개수를 증가시킴.

## Mole\_Stage.c

스테이지가 올라갈 수록 한 마리 씩 두더지의 수가 올라가며 난이도가 조정됨.

```
while(1)
{
    //다음 단계로 넘어간다면
    if (isNext == 1)
    {
        nextmole = 0;
        isNext = 0;
        startmole = 1;
        stage++;
        //1stage에는 한마리, 2stage에는 2마리...동적할당

        for (k2 = 0; k2 < 8; k2++)
        {
            printmatrix[k2] = 0;
        }

        for (j = 0; j < stage; j++)
        {
            renumber = (rand() % 9) + 1;
            timeleft[j] = (rand() % 10);
            for (k = 1; k < stage + 1; k++)
            {
                if (renumber == random[k - 1])
                {
                    renumber = (rand() % 9) + 1;
                    k--;
                }
            }
            random[j] = renumber;
            renumber = 0;
        }
    }
}
```



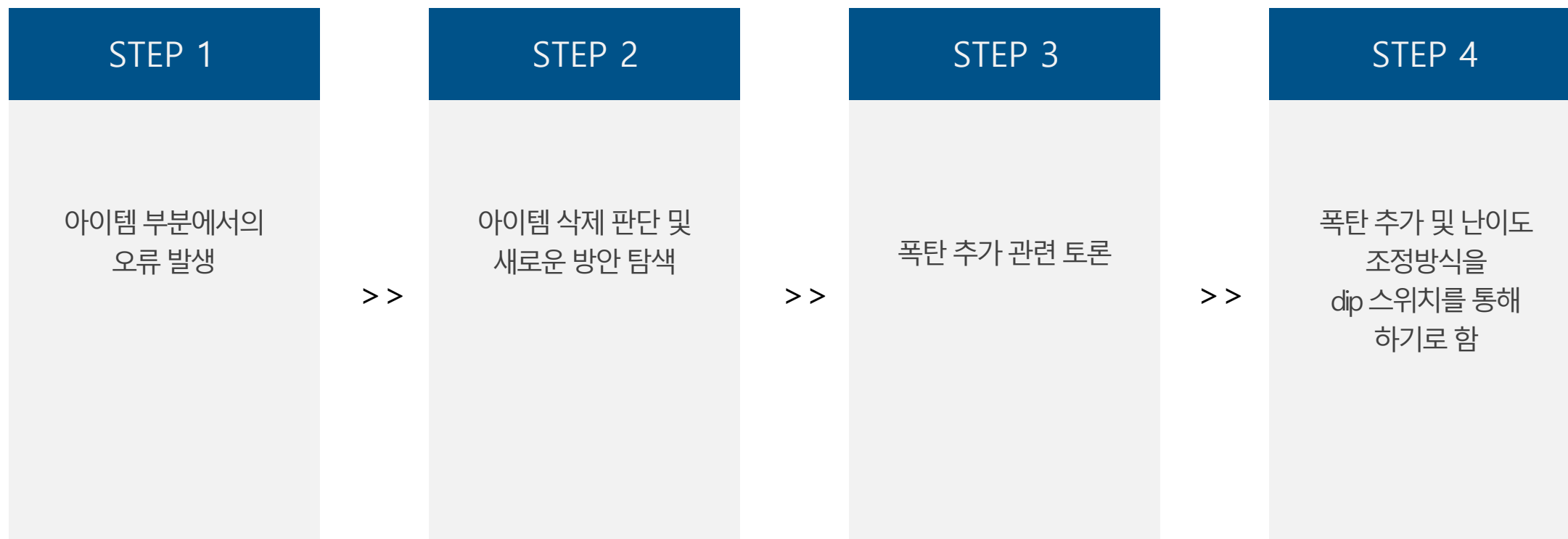
# Q.

무엇을, 어떻게 바꿀 수 있을까?

# A.

폭탄과 같은 새로운  
기능을 추가하면 어떨까?

## 개선 과정





```
// 폭탄을 맞춘 경우
if (random[z] > 10) {
    if (num == random[z] - 10)
    {
        count++;
        score_player -= 2;
        printf("점수 : %d\n", score_player);
        clcd_input2(score_player, score_mole);

        for (x1 = 0; x1 < 8; x1++)
        {
            printmatrix[x1] = printmatrix[x1] - bomb[num-1][x1];
        }

        timeleft[z] = rand() % 10;
        num = 0;
        renumber = (rand() % 11) + 1;
        if (renumber > 9) {
            isBomb = 1;
            renumber = (rand() % 9) + 1;
        }
    }
}
```

## 폭탄을 맞춘 경우

폭탄을 맞춘 경우, 플레이어의 점수를 2점 감소 시킴. 이후 해당 폭탄의 시간을 재설정 한 후, 다음에 생성될 폭탄과 두더지를 랜덤하게 선택함.

## isBomb

해당 변수는 폭탄의 여부를 나타냄  
1인 경우 폭탄을 클릭한 것이고,  
0인 경우 폭탄 아닌 두더지를 클릭.

폭탄 클릭 시, 해당 두더지의 번호를 표시하기 위해  
10을 더한 값으로 표현됨.

전체적으로 폭탄 클릭의 여부를 추적하는데 사용됨.

## Part 2, 추가 및 개선과정

Mole\_Game\_main3.c

```
if (dipval != 0) {  
    // DIP 스위치 값에 따라 난이도 설정  
    if (dipval & 0x01) {  
        stagetime = 40; // DIP 스위치 1번 ON: 난이도 1 (가장 쉬움)  
    } else if (dipval & 0x02) {  
        stagetime = 35; // DIP 스위치 2번 ON: 난이도 2  
    } else if (dipval & 0x04) {  
        stagetime = 30; // DIP 스위치 3번 ON: 난이도 3  
    } else if (dipval & 0x08) {  
        stagetime = 25; // DIP 스위치 4번 ON: 난이도 4  
    } else if (dipval & 0x10) {  
        stagetime = 20; // DIP 스위치 5번 ON: 난이도 5  
    } else if (dipval & 0x20) {  
        stagetime = 15; // DIP 스위치 6번 ON: 난이도 6  
    } else if (dipval & 0x40) {  
        stagetime = 10; // DIP 스위치 7번 ON: 난이도 7  
    } else if (dipval & 0x80) {  
        stagetime = 5; // DIP 스위치 8번 ON: 난이도 8 (가장 어려움)  
    }  
}
```

### DIP 스위치를 통한 난이도 조절

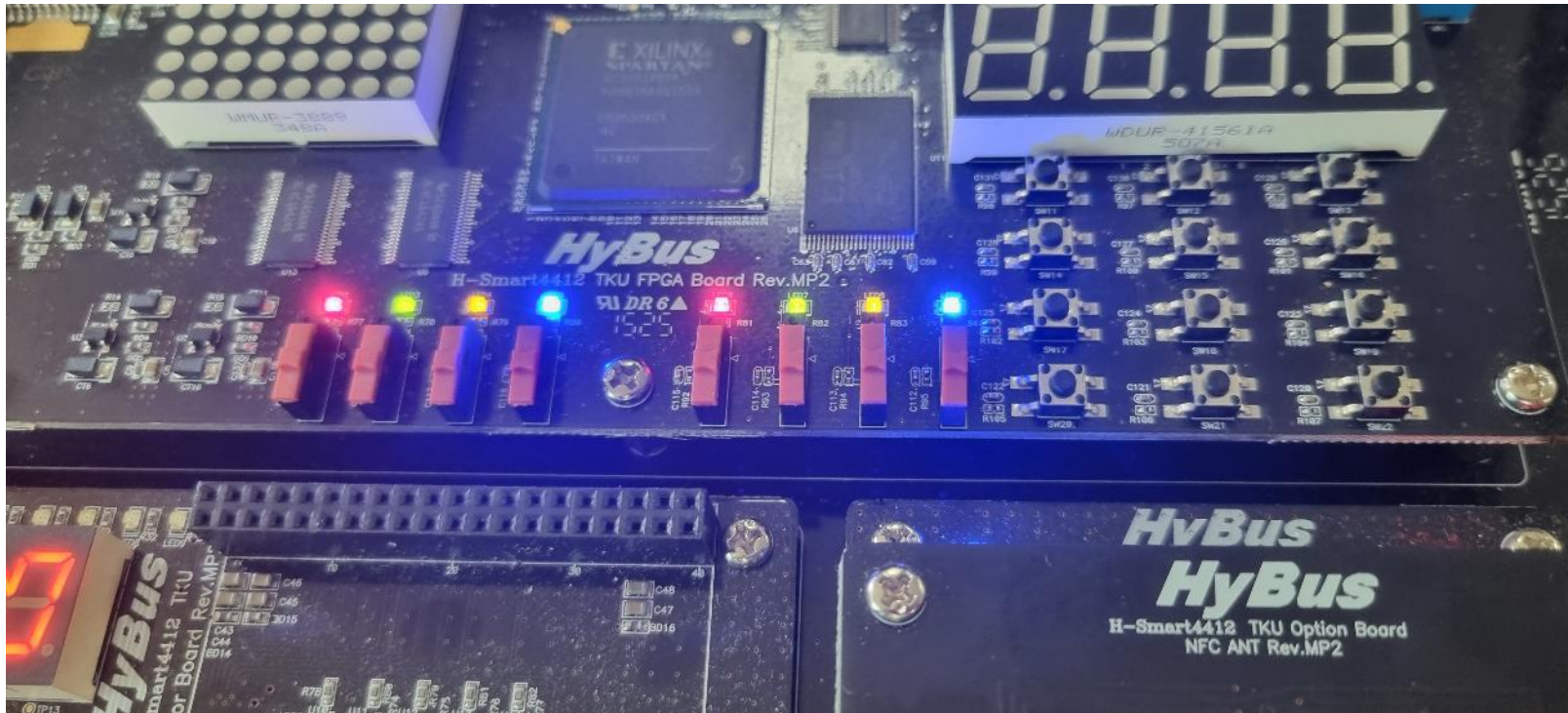
첫 번째 스위치가 켜져 있으면  
난이도가 1, 두 번째 스위치가  
켜져 있으면 난이도가 2가 됨.

### dipval

DIP 스위치의 값을 저장하는 변수임.  
각각의 스위치가 ON 또는 OFF 상태를  
나타내면, 난이도가 조정됨.

## Part 2, 추가 및 개선과정

### DIP LED



### DIP LED

DIP LED가 모두 켜져 있는 상태에서 DIP 스위치를 올리면 LED가 꺼지는 방식

12번 스위치를 누르면 종료

“

감사합니다

”