



IoT 프로그래밍

최종 발표

7조 최유안 추희정 앵크졸

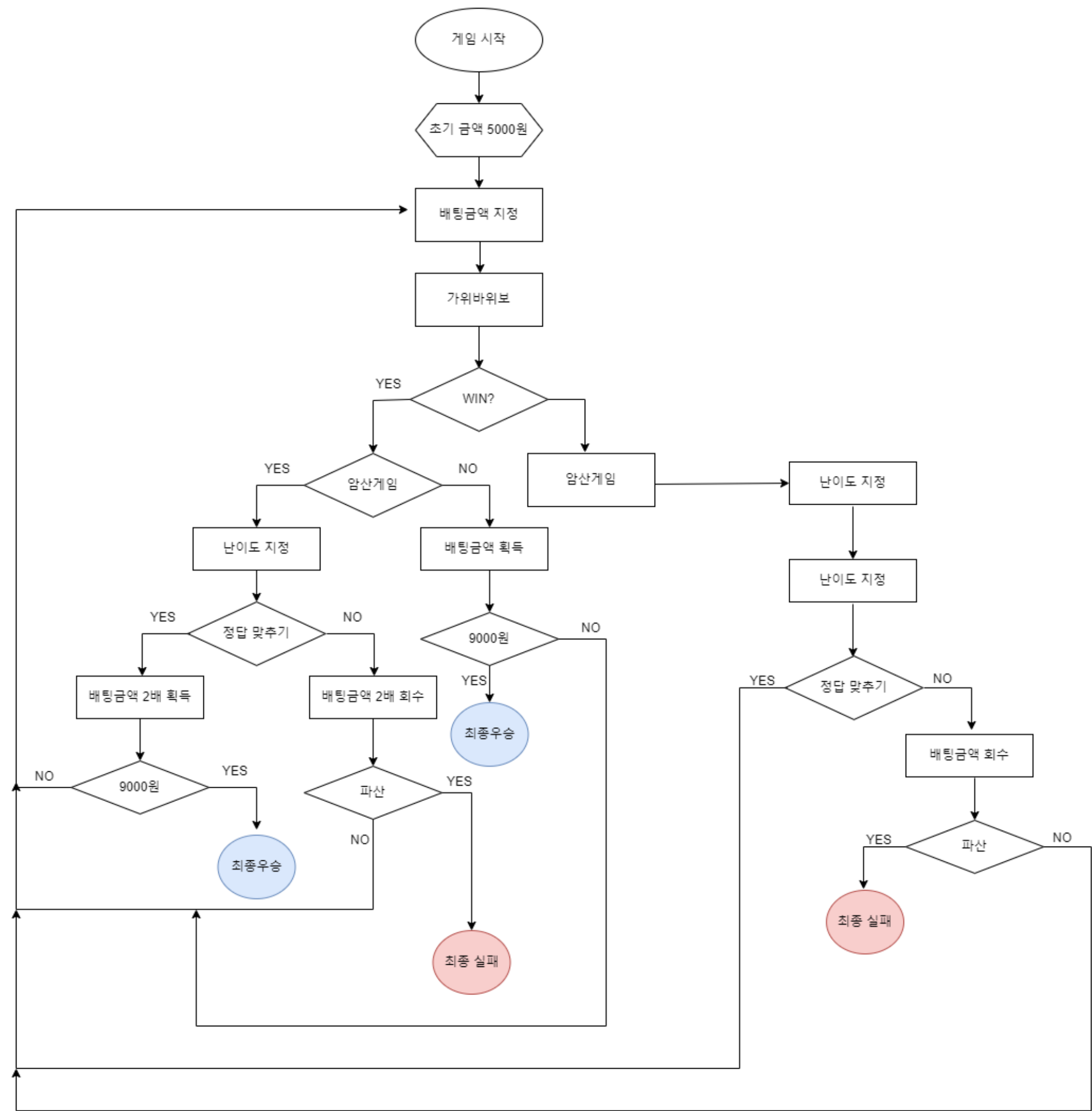
목차

- 1 게임 설명
- 2 알고리즘 및 코드
- 3 차별점

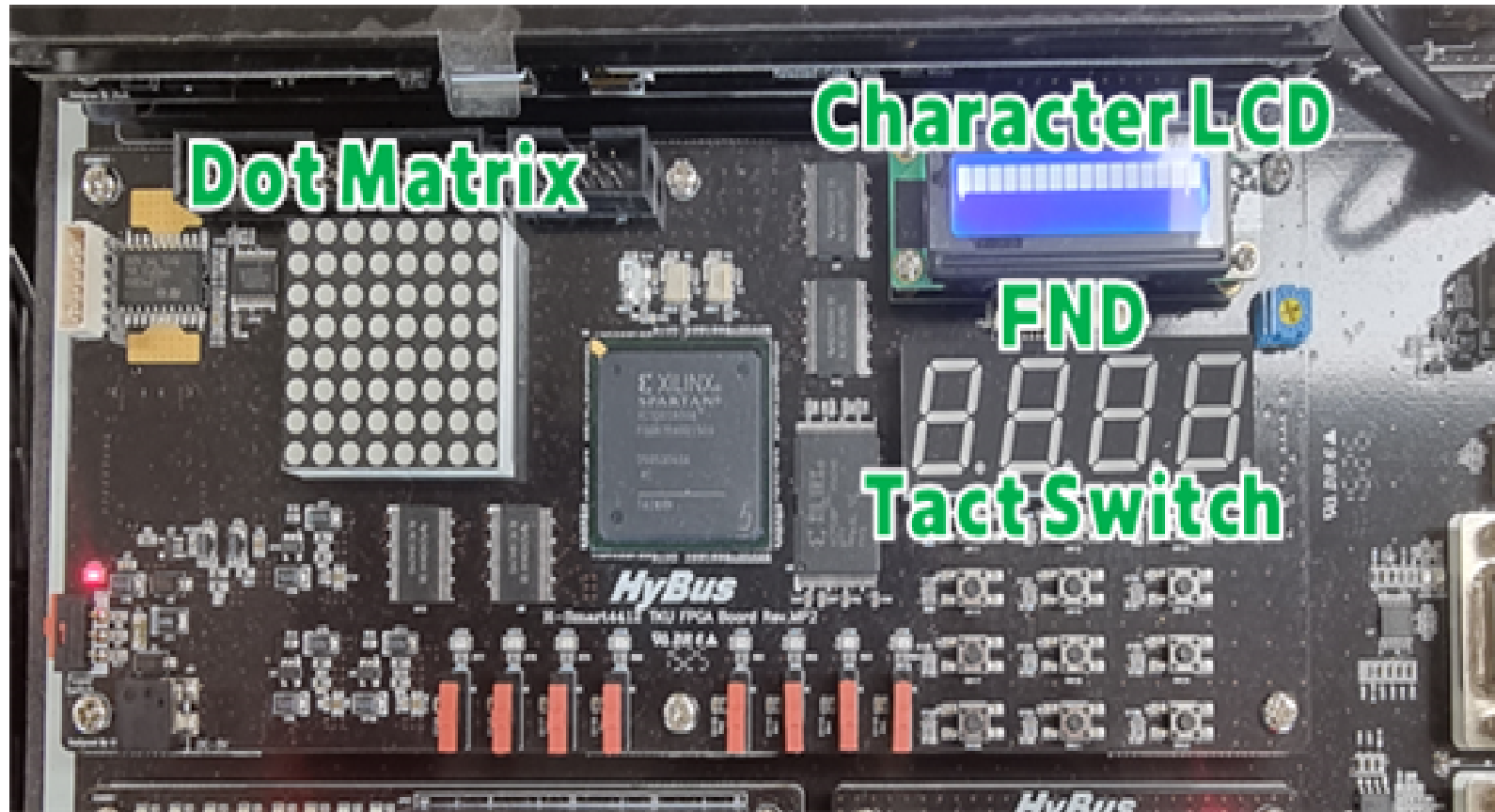
가위바위보 암산 배팅 게임

1. 가위바위보를 통해 암산게임의 여부를 결정
2. 배팅한 금액을 바탕으로 암산게임 진행
3. 사용자는 자신이 플레이할 암산 게임의 난이도를 조절 가능
4. 암산게임과 가위바위보의 승패 여부에 따라 배팅금액이 조절되어
획득/손실
5. 플레이어의 금액이 0원 또는 9000원이 될 때까지 플레이

게임 플로우차트



사용 장치



fnd(7-Segment)

: 현재 남은 배팅 금액을 조회하기 위해 사용

tact switch

: 베팅 금액 조절 및 가위바위보 플레이 조작을 위해 사용
숫자 값 입력용(구구단 계산)

dot matrix(8X8)

: “묵”, “찌”, “빠” 를 8 * 8 매트릭스를 이용해서 표현함
가위바위보와 암산 게임 승패 여부 출력

character lcd

: 현재 진행 상태를 알려주기 위해 사용(가위바위보 진행
중 이거나 암산 게임 넘어 갔을 때 숫자 출력용)

알고리즘 및 코드

1. 택트 스위치 입력받는 함수

```

unsigned char tactsw_get(int tmo)
{
    unsigned char b;
    if (tmo) {
        if (tmo < 0)
            tmo = ~tmo * 1000;
        else
            tmo *= 1000000;

        while (tmo > 0) {
            usleep(10000);
            read(tactswFd, &b, sizeof(b));
            if (b) return(b);
            tmo -= 10000;
        }
        return(-1);
    }
    else {
        read(tactswFd, &b, sizeof(b));
        return(b);
    }
}

```

2. 호출 시 택트 스위치 입력값 반환

```

int tact_switch_listener(){
    unsigned char c;
    int selected_tact = 0; // false 값 넣기

    if((tactswFd = open( tactswDev, O_RDONLY )) < 0){           // 예외처리
        perror("tact error");
        exit(-1);
    }

    // 택트스위치는 1,2,3,4,5,7,8번 스위치만 사용함
    while(1){
        c = tactsw_get(10);
        switch (c) {
            case 1: selected_tact = 1 ; break; // 묵
            case 2: selected_tact = 2 ; break; // 찌
            case 3: selected_tact = 3 ; break; // 빠
            case 4: selected_tact = 4 ; break; // 배팅금액 증가
            case 5: selected_tact = 5 ; break; // 배팅금액 확정
            case 7: selected_tact = 7 ; break; // 게임 계속하기
            case 8: selected_tact = 8 ; break; // 게임 나가기
            default: printf("press other key\n", c); break; // 기본값 메세지
        }
        return selected_tact; // 어떤 스위치가 눌렸는지 int 형으로 반환함
    }
}

```

알고리즘 및 코드

3. LCD 화면 문자열 표시 함수

```
void clcd_input(char clcd_text[]){
    int clcd_d;

    clcd_d = open(clcd , O_RDWR);
    if (clcd_d < 0) { printf("clcd error\n"); } // 예외처리

    write(clcd_d , clcd_text , strlen(clcd_text)); // 두번째부터 각각 문자열, 문자열 크기
    close(clcd_d);
}
```

4. FND 표시 함수

```
int FND_control(int money[], int time_sleep){
    unsigned char FND_DATA_TBL[]={
        0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90,0x88,
        0x83,0xC6,0xA1,0x86,0x8E,0xC0,0xF9,0xA4,0xB0,0x99,0x89
    };

    int fnd_fd = 0;
    unsigned char fnd_num[4];

    // money 배열의 원소들을 순서에 맞게 넣어주기
    fnd_num[0] = FND_DATA_TBL[money[0]];
    fnd_num[1] = FND_DATA_TBL[money[1]];
    fnd_num[2] = FND_DATA_TBL[money[2]];
    fnd_num[3] = FND_DATA_TBL[money[3]];

    fnd_fd = open(fnd_dev, O_RDWR);
    if(fnd_fd < 0){ printf("fnd error\n"); } // 예외처리

    write(fnd_fd, &fnd_num, sizeof(fnd_num)); // 출력
    sleep(time_sleep); // 점등시간 조절

    close(fnd_fd);
}
```

알고리즘 및 코드

5. 도트 매트릭스 함수

```
void smile()
{
    int dot_d,i;
    unsigned char smile[8] = {0x7E,0x81,0xA5,0xDB,0xA5,0x99,0x7E};
    dot_d = open(dot,O_RDWR);
    if(dot_d<0)
    {
        printf("Error\n");
        return;
    }
    write(dot_d,&smile,sizeof(smile));
    sleep(2);
    close(dot_d);
    return;
}
```

```
void sad()
{
    int dot_d,i;
    unsigned char sad[8] = {0x7E,0x81,0xA5,0xDB,0xA5,0x99,0x7E};
    dot_d = open(dot,O_RDWR);
    if(dot_d<0)
    {
        printf("Error\n");
        return;
    }
    write(dot_d,&sad,sizeof(sad));
    sleep(2);
    close(dot_d);
    return;
}
```


알고리즘 및 코드

6. 암산문제 생성 함수

// 암산 문제를 생성하는 함수

```
int generateMathProblem(int difficulty) {
    srand((unsigned int)time(0));

    int num1, num2, answer;
    char operator_ = '+';

    switch (difficulty) {
    case 1:
        num1 = rand() % 9 + 1;
        num2 = rand() % 9 + 1;
        break;
    case 2:
        num1 = rand() % 90 + 10;
        num2 = rand() % 90 + 10;
        break;
    case 3:
        num1 = rand() % 900 + 100;
        num2 = rand() % 900 + 100;
        break;
    }
```

```
switch (rand() % 4) {
case 0:
    operator_ = '+';
    answer = num1 + num2;
    break;
case 1:
    operator_ = '-';
    if (num1 < num2) { // num1이 항상 num2보다 크거나 같게 함
        int temp = num1;
        num1 = num2;
        num2 = temp;
    }
    answer = num1 - num2;
    break;
case 2:
    operator_ = '*';
    answer = num1 * num2;
    break;
case 3:
    operator_ = '/';
    num2 = (num2 == 0) ? 1 : num2; // 0으로 나누는 경우를 방지
    num1 = num1 * num2; // 결과가 정수가 되도록 num1을 조정
    answer = num1 / num2;
    break;
}

printf("문제: %d %c %d = ?\n", num1, operator_, num2);
return answer;
```

알고리즘 및 코드

7. 메인 함수

```
while (1) {
    write(clcd_d, "\nYour balance: %d\n", money, 12);
    do {
        write(clcd_d, "Input your bet (100): ", 12);
        scanf_s("%d", &betAmount);
        if (betAmount % 100 != 0)
            write(clcd_d, "only 100won each.\n", 12);
    } while (betAmount % 100 != 0);

    write(clcd_d, "Choose Rock(1), Scissor(2), Paper(3) \n\nRock Scissor Paper!\n ", 12);
    scanf_s("%d", &userChoice);

    compChoice = rand() % 3 + 1;

    if (userChoice == compChoice) {
        write(clcd_d, "draw! Rock Scissor Paper again.\n", 12);
        continue;
    }
}
```

```
else if ((userChoice == 1 && compChoice == 2) || (userChoice == 2 && compChoice == 3) || (userChoice == 3 && compChoice == 1)) {
    write(clcd_d, "You Lose!\n", 12);
    sad();
    write(clcd_d, "Your balance: %d\n", money, 12);

    write(clcd_d, "Choose Level (1: one digit, 2: two digits, 3: three digits): ", 12);
    scanf_s("%d", &difficulty);

    answer = generateMathProblem(difficulty);
    write(clcd_d, "Your answer : ", 12);
    scanf_s("%d", &userAnswer);

    if (userAnswer == answer) {
        write(clcd_d, "Correct.\n", 12);
    }
    else {
        write(clcd_d, "wrong\n", 12);
        sad();
        money -= betAmount;
    }
}
```

알고리즘 및 코드

7. 메인 함수

```
else {
    write(clcd_d, "You Win!\n", 12);
    smile();
    write(clcd_d, "Your balance: %d\n", money, 12);
    write(clcd_d, "1: calculate 2: RSC ", money, 12);
    scanf_s("%d", &userChoice);
    if (userChoice == 1)
    {
        write(clcd_d, "Choose Level (1: one digit, 2: two digits, 3: three digits): ", 12);
        scanf_s("%d", &difficulty);

        answer = generateMathProblem(difficulty);
        write(clcd_d, "Your answer : ", 12);
        scanf_s("%d", &userAnswer);

        if (userAnswer == answer) {
            write(clcd_d, "Correct. gain twice of bet.\n", 12);
            smile();
            money += betAmount*2;
        }
        else {
            write(clcd_d, "Wrong. lose bet.\n", 12);
            sad();
            money -= betAmount*2;
        }
    }
    else
    {
        money += betAmount;
    }
}
```

```
if (money <= 0) {
    write(clcd_d, "Loser!\n", 12);
    break;
}
if (money >= 9000) {
    write(clcd_d, "Winner!\n", 12);
    break;
}
```

```
return 0;
```

차별점

1. 하나의 게임에 두 개의 게임을 결합하여 사용자의 흥미를 유발
2. 사용자의 선택에 따라 배팅액을 다양하게 조절할 수 있도록 함
3. 기존의 프로젝트보다 추가적인 출력 장치를 사용하여
사용자가 게임을 플레이할 때 시각적 효과를 느낄 수 있도록 함

출처

- https://github.com/qkrejr00/2022_IoTProgramming_Team6/blob/master/%EB%B0%9C%ED%91%9C/1%EC%B0%A8%20%EC%A4%91%EA%B0%84%EB%B0%9C%ED%91%9C.md
- <https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=lyw94k&logNo=220840476325>
- <https://syki66.github.io/blog/2020/06/15/H-smart4412TKU.html#1-%EC%82%AC%EC%9A%A9%EB%90%98%EB%8A%94-%EC%84%BC%EC%84%9C-%EB%B0%8F-%EC%9E%A5%EC%B9%98>
- <https://cccding.tistory.com/77>
- <https://cccding.tistory.com/78>
- <https://cccding.tistory.com/67>
- <https://cccding.tistory.com/64>
- <https://cccding.tistory.com/69>
- <https://takethat.tistory.com/category/%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D%20%EA%B3%B5%EB%B6%80/%EB%A6%AC%EB%88%85%EC%8A%A4%20%26%20%EC%9E%84%EB%B2%A0%EB%94%94%EB%93%9C%20%EC%8B%9C%EC%8A%A4%ED%85%9C>



THANK YOU!