오목 트레이닝 프로그램

팀 명 : 으라차차
팀 원 : 최승진
오상헌
양회욱

목차

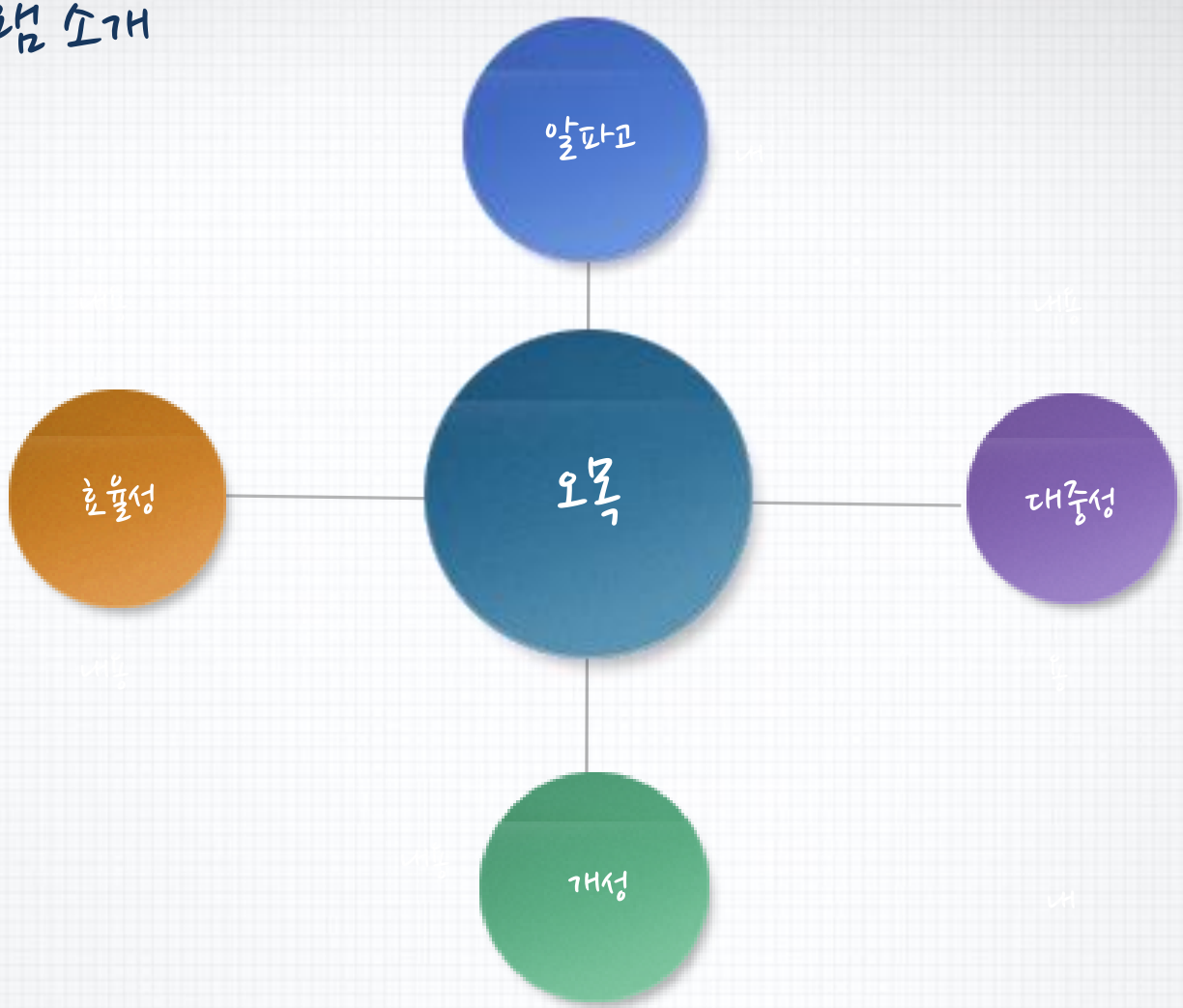
목차

팀 소개

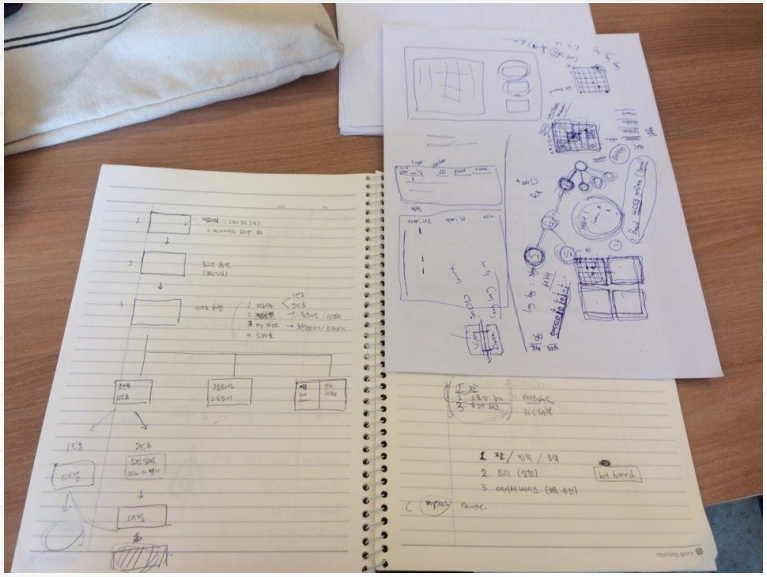
팀 이름 : 으라차차

팀　원 : 최승진, 양회욱, 오상헌

프로그램 소개

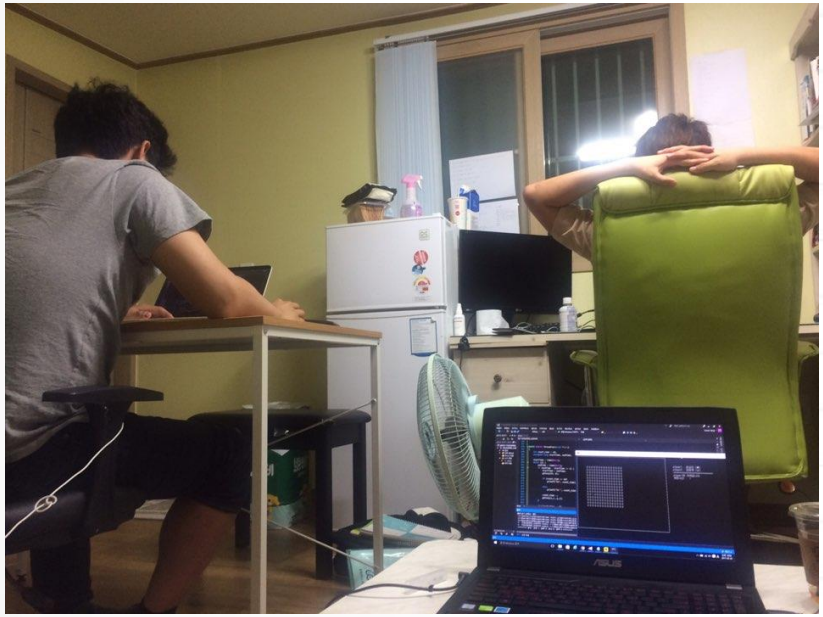


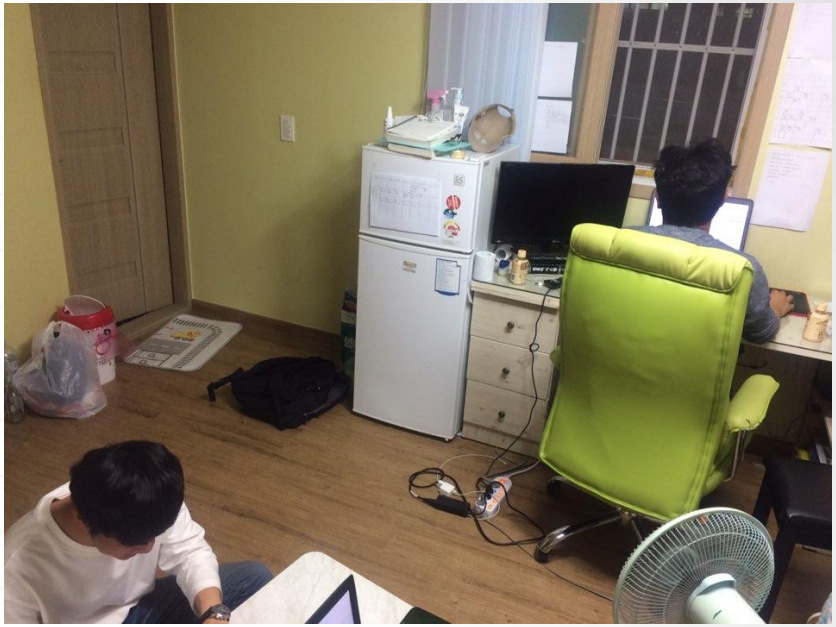
프로그래밍 과정



프로그래밍 과정

프로그래밍 과정

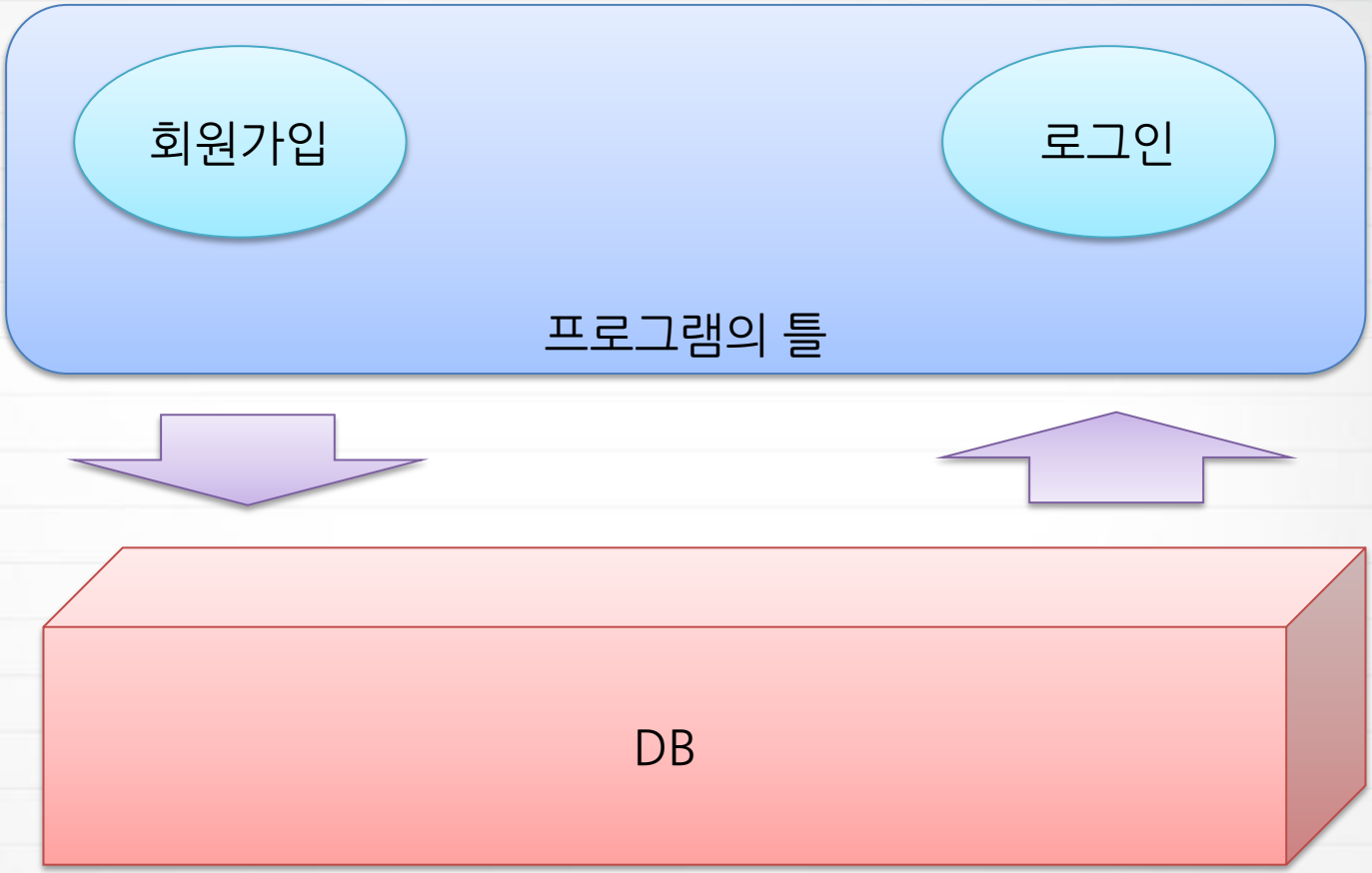




프로그래밍 과정

주차 내 용	1	2	3	4	5	6	7	8	9	10
계획서	■									
요구사항 분석		■								
구조 설계			■							
기능 설계				■						
중간고사				■						
기술 개발 및 구현					■	■	■	■		
테스트 및 완료									■	■

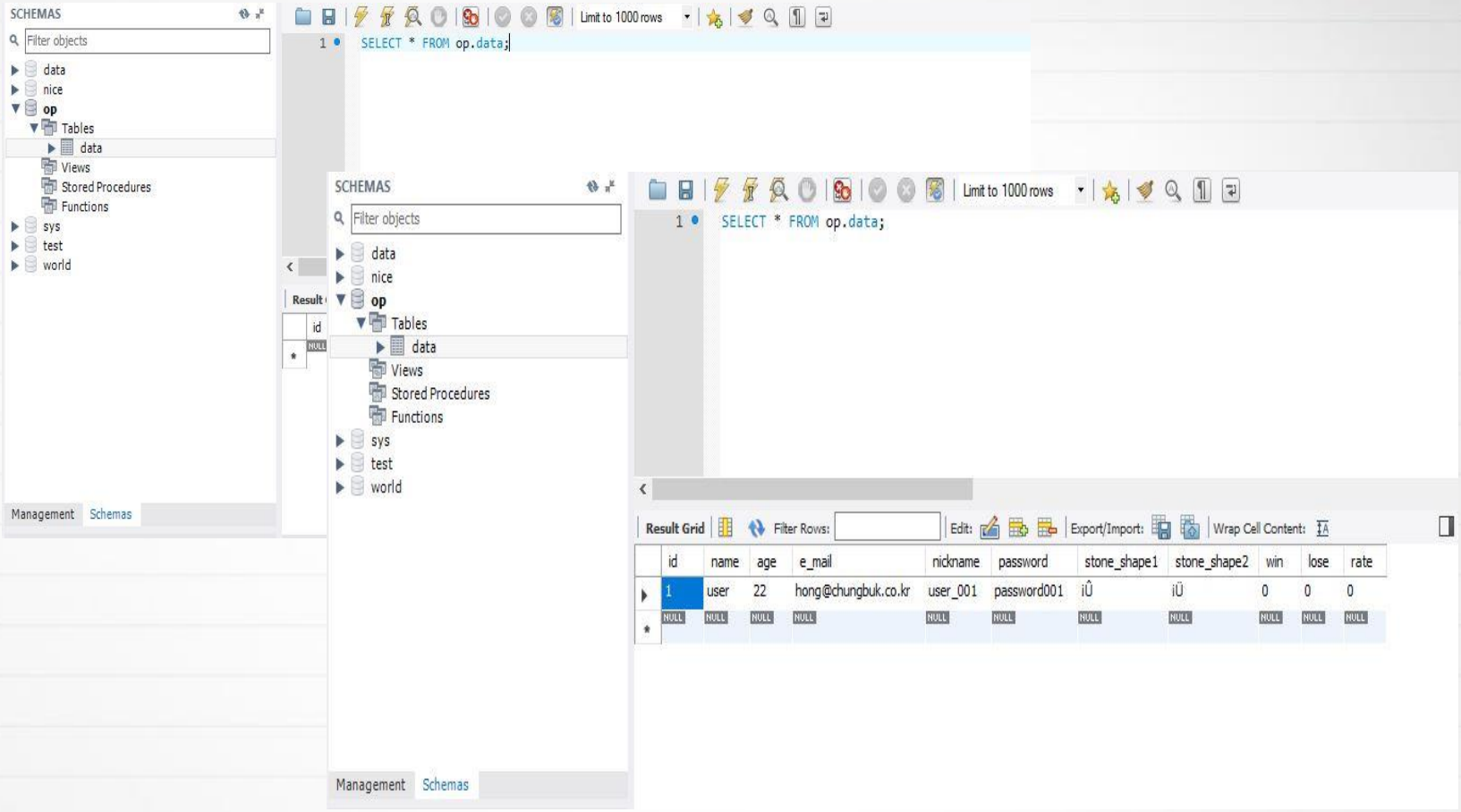
프로그램 설명(최승진)



프로그램 설명(최승진)


오 목 프 로 그 램
by으라차차
1 번 째 메 뉴 >> 프로그램 설명
2 번 째 메 뉴 >> 개인 정보 출력
3 번 째 메 뉴 >> 1인용 / 2인용 게임플레이
4 번 째 메 뉴 >> 로 그 아 웃
메뉴를 입력해주십시오 >>
오 목 프 로 그 램
by으라차차
고객님의 이름 : user
고객님의 나이 : 22
고객님의 아이디 : user_001
고객님의 이메일 : hong@chungbuk.co.kr
고객님의 오목알 모양 : ○ / ●
고객님의 승 : 0 / 패 : 0 / 승률 : 0

프로그램 설명(최승진)


SCHEMAS
Filter objects
data
nice
op
Tables
data
Views
Stored Procedures
Functions
sys
test
world
Management
Schemas
Limit to 1000 rows
1
SELECT * FROM op.data;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
id
name
age
e_mail
nickname
password
stone_shape1
stone_shape2
win
lose
rate
1
user
22
hong@chungbuk.co.kr
user_001
password001
iÛ
iÜ
0
0
0
NULL

프로그램 설명(최승진)


아 이 디를 입력하시오 >> user_001
비밀번호를 입력하시오 >> password12345
오 목 프 로 그 램
by으라차차
오 목 프 로 그 램
by으라차차
1 번 째 메 뉴 >> 회 원 가 입
2 번 째 메 뉴 >> 로 그 인
3 번 째 메 뉴 >> 종 료
메뉴를 입력해주십시오 >>

프로그램 설명(양회욱)

1. Gomouk Rule 적용
2. 튜토리얼
3. 2인용 오목게임 실행(게임 테이블 형성)

강의시간에 배운 내용 최대한 활용하기

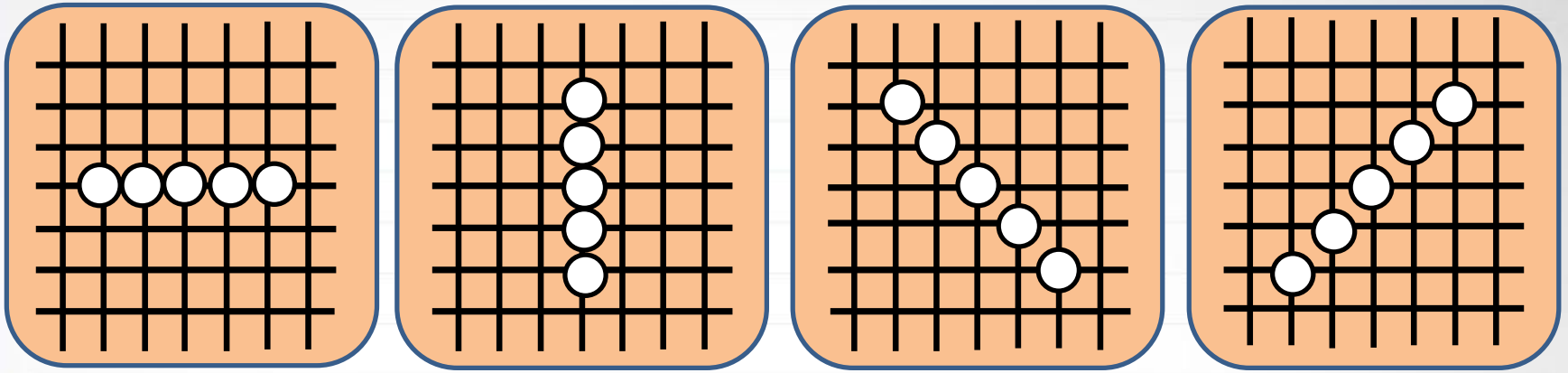
프로그램 설명(양회욱)

<Gomoku Rule>

1. 오목알이 5개 놓이면 승리한다.
   다른 제한사항은 없다.

2. 한 턴당 30초로 제한한다.
   시간 내에 놓지 못하면 실격처리된다.

프로그램 설명(양회욱)



TIME

승패 여부 판별

프로그램 설명(양회욱)

```
bool check_board_complete(Board* board)
{
    int i, j;
    bool isComplete = false;

    Board* boardHorizontal = makeBoard((ll)0, (ll)2181431069507584, (ll)0, (ll)0);  //가로 직선
    Board* boardReverseSlash = makeBoard((ll)0, (ll)281479271743488, (ll)65537, (ll)0);  //역슬래시
    Board* boardVertical = makeBoard((ll)0, (ll)9223653520421683200, (ll)16384, (ll)0);   //세로 직선
    Board* boardSlash = makeBoard((ll)0, (ll)4611967510585016320, (ll)4096, (ll)0);   //슬래시

    isComplete = check_board_mask(board, boardHorizontal);
    if (isComplete) return true;

    isComplete = check_board_mask(board, boardReverseSlash);
    if (isComplete) return true;

    isComplete = check_board_mask(board, boardVertical);
    if (isComplete) return true;

    isComplete = check_board_mask(board, boardSlash);
    if (isComplete) return true;

    return isComplete;
}
```

프로그램 설명(양회욱)

```
bool check_board_mask(Board* board, Board* mask)
{
    int i;
    int max = MAX_LINE * MAX_LINE;
    int mid = max / 2;
    bool result;
    Board *checker;
    Board *target = board_copy(board);

    for (i = 0; i < max; i++) {
        if (i < mid) {
            checker = board_shift_left(mask, mid - i);
            result = board_isEqual(board_shift_right(board_and(target, checker), mid - i), mask);
        }
        else if (i == mid) {
            result = board_isEqual(board_and(target, mask), mask);
        }
        else {
            checker = board_shift_right(mask, i - mid);
            result = board_isEqual(board_shift_left(board_and(target, checker), i - mid), mask);

        }

        if (result == true) return true;
    }
```

프로그램 설명(양회욱)

튜토리얼

게임을 처음 접하는 사람에게 게임 플레이 방법 소개 및 간단한 연습

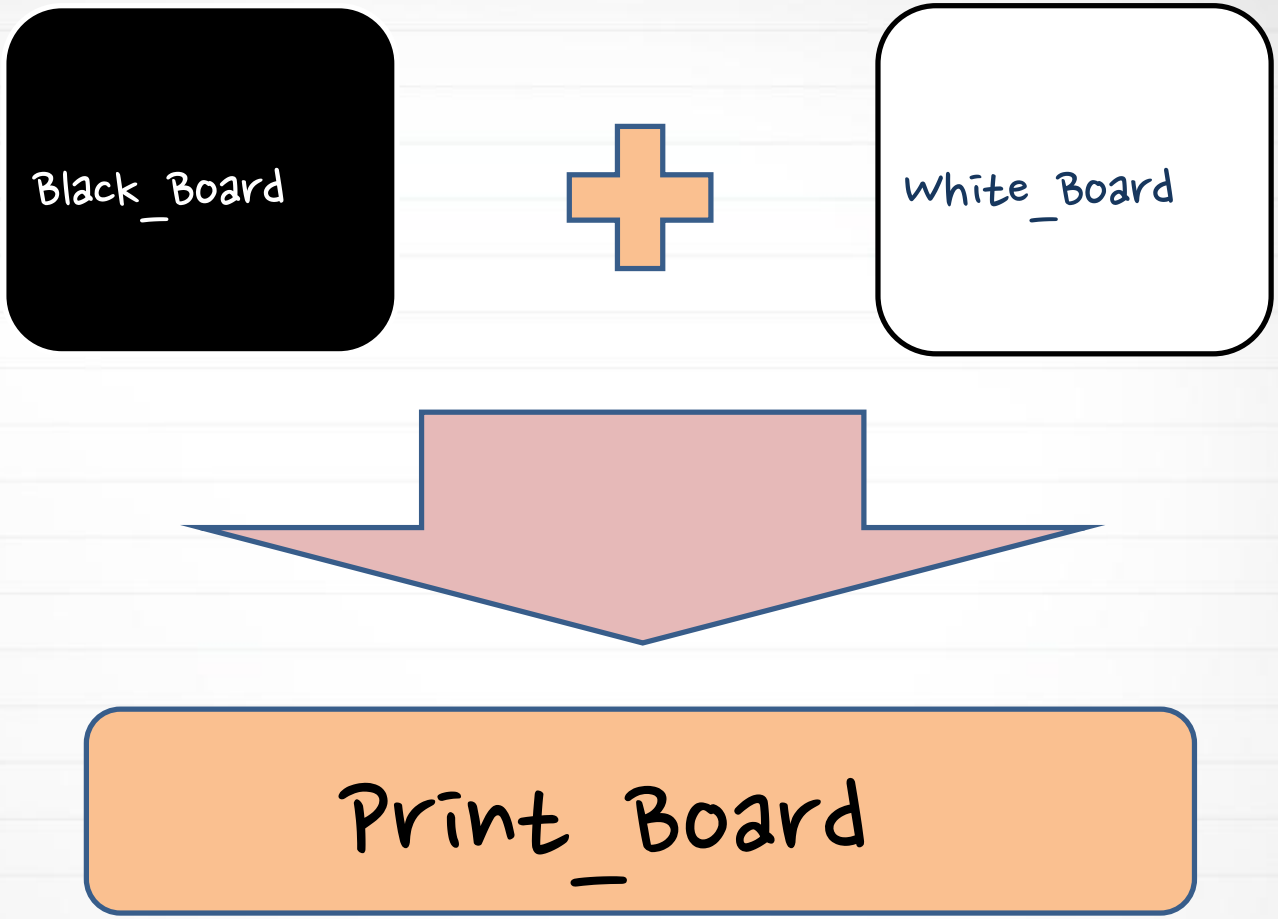
```
void game_table_tutorial(Board *board) { ... }

void Game_Tutorial(Board* board) { ... }
```

프로그램 설명(양회욱)

<게임 테이블>

1. 오목판
2. 흑과 백 설정
3. 오목알 이동 및 놓기
4. 시간

프로그램 설명(양희욱)



프로그램 설명(양회욱)

오목판 -> 교재 3.5.3 바둑판 그리기 draw_check의 응용

```
void draw_omok()
{
    int i, j;
    unsigned char a = 0xa6;
    unsigned char b[13];
    for (i = 1; i < 12; i++)
        b[i] = 0xa0 + i;
    gotoxy(22, 7);
    printf("%c%c", a, b[3]);
    for (i = 0; i < 13; i++)
    {
        gotoxy(i * 2 + 24, 7);
        printf("%c%c", a, b[8]);

    }
    printf("%c%c", a, b[4]);
    printf("\n");
    for (i = 0; i < 13; i++)
    {
        gotoxy(22, 8 + i);
        printf("%c%c", a, b[7]);
        for (j = 0; j < 13; j++)
        {
            gotoxy(j * 2 + 24, 8 + i);
            printf("%c%c", a, b[11]);
        }

        gotoxy(50, 8 + i);
        printf("%c%c", a, b[9]);
        printf("\n");
    }
    gotoxy(22, 21);
    printf("%c%c", a, b[6]);

    for (i = 0; i < 13; i++)
    {
        gotoxy(i * 2 + 24, 21);
        printf("%c%c", a, b[10]);
    }
    gotoxy(50, 21);
    printf("%c%c", a, b[5]);
    printf("\n");
}
```

프로그램 설명(양회욱)

```
void print_board(Board* board_1, Board* board_2, int n)
{
    draw_omok();

    int i, j;

    for (i = 0; i < MAX_LINE; i++)
    {
        for (j = 0; j < MAX_LINE; j++)
        {
            gotoxy(i * 2 + 22, j + 7);

            if (n % 2 == 1)
            {
                if (board_isSet(board_1, i, j))
                    printf("%s", user.stone_shape2); // user오목알

                if (board_isSet(board_2, i, j))
                    printf("%s", user2.stone_shape1); // user2오목알
            }

             else
             {
                 if (board_isSet(board_1, i, j))
                     printf("%s", user.stone_shape1); // user오목알

                 if (board_isSet(board_2, i, j))
                     printf("%s", user2.stone_shape2); // user2오목알
             }
        }
    }
}
```

프로그램 설명(양회욱)

흑과 백 설정

```
srand(time(NULL));
n = rand() % 2;
first_n = n;
```

난수를 이용하여 누가 먼저 둘지 정한다

-> 원카드 프로그램 응용

```
gotoxy(80, 10);
printf("player1의 차례입니다. ");
gotoxy(80, 11);
printf("제한시간 : ");
move_control(board_1, board_2);
n++;
```

First_n = n;

홀수로 시작했는지 짝수로 시작했는지 기억하기 위해 (바둑돌 색깔 지정)

```
int turn_two_players(int n)
{
    if (n % 2 == 1) // 홀수
        return 1;

    else          // 짝수
        return 0;
}
```

플레이어가 둘 때마다 n++;

프로그램 설명(양회욱)

오목알 이동 및 놓기

교재 함수 3.2.1 참고

```
void move_control(Board* board_A, Board* board_B)
{
    char key;
    int put = 0;

    do
    {
        key = getch();

        if (count_time < 1)
        {
            break;
        }

        switch (key)
        {
        case 32:
            //printf("%d %d\n", x_c, y_c);
            if (board_isSet(board_A, (x_c - 22) / 2, y_c - 7) || board_isSet(board_B, (x_c - 22) / 2, y_c - 7))
                continue;
            *board_A = *(board_set(board_A, (x_c - 22) / 2, y_c - 7)); /*수를 놓았을때 표현*/
            put = 1;
            break;
```

프로그램 설명(양회욱)

오목알 이동 및 놓기

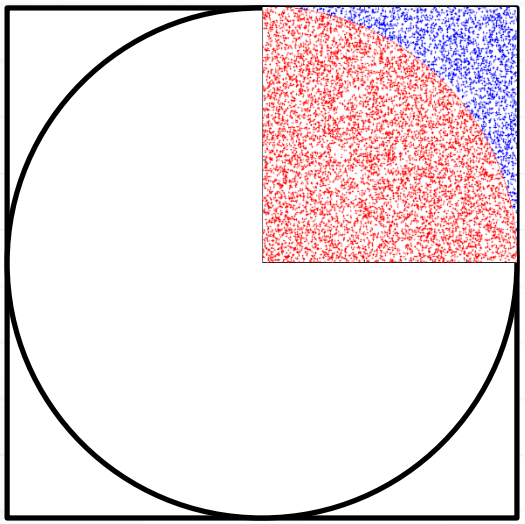
```
        case 72:
            y_c--;
            if (y_c < 1)
                y_c = 1;
            break;
        case 75:
            x_c -= 2;
            if (x_c < 1)
                x_c = 1;
            break;
        case 77:
            x_c += 2;
            break;
        case 80:
            y_c++;
            break;
        }

        gotoxy(x_c, y_c);
    } while (put != 1);

    if (key == 32)
    {
        TerminateThread(thread, 0);
    }
```

프로그램 설명(양회욱)

시간

```
DWORD WINAPI ThreadFunc(void *data)
{
    count_time = 30;
    unsigned long startTime, endTime;

    startTime = time(NULL);
    while (1)
    {
        endTime = time(NULL);
        if (count_time < 0) {
            gotoxy(92, 11);
            printf("시간초과!!!\n");

            timeover++;

            return 0;
        }

        if (endTime - startTime >= 1) {
            startTime = endTime;
            gotoxy(92, 11);

            if (count_time >= 10)
                printf("%d", count_time);
            else
                printf("%d ", count_time);

            count_time--;
            gotoxy(x_c, y_c);
        }
    }

    return 0;
}
```

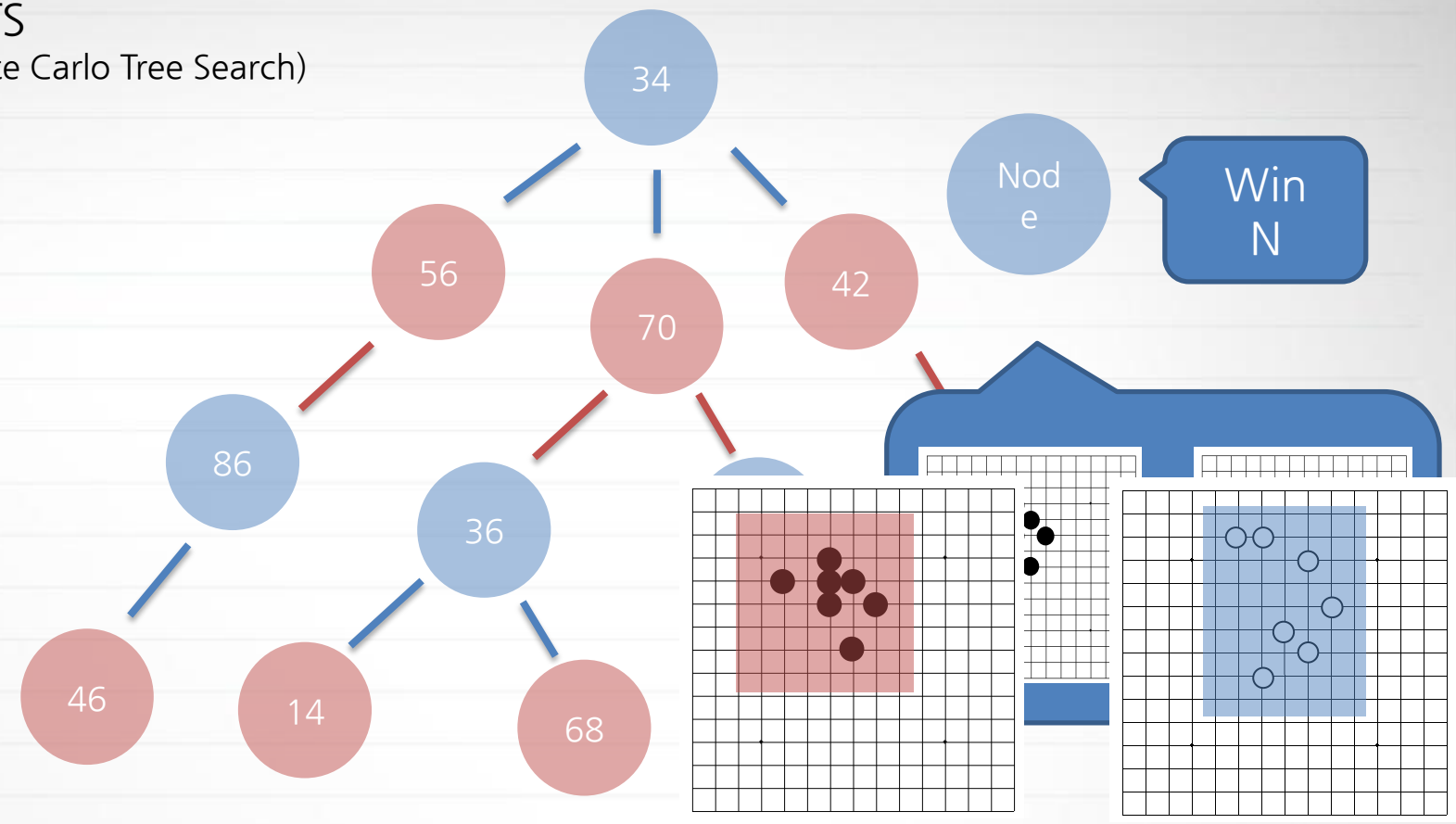
Monte Carlo Simulation



$$\frac{\text{원의 넓이}}{\text{정사각형의 넓이}} = \frac{\text{빨간 점의 개수}}{\text{전체 점의 개수}}$$

$$\text{원의 넓이} = \text{정사각형의 넓이} * \frac{\text{빨간 점의 개수}}{\text{전체 점의 개수}}$$

MCTS
(Monte Carlo Tree Search)



프로그램 실행

Q & A

감사합니다