



# 게임프로그래밍

2019775009

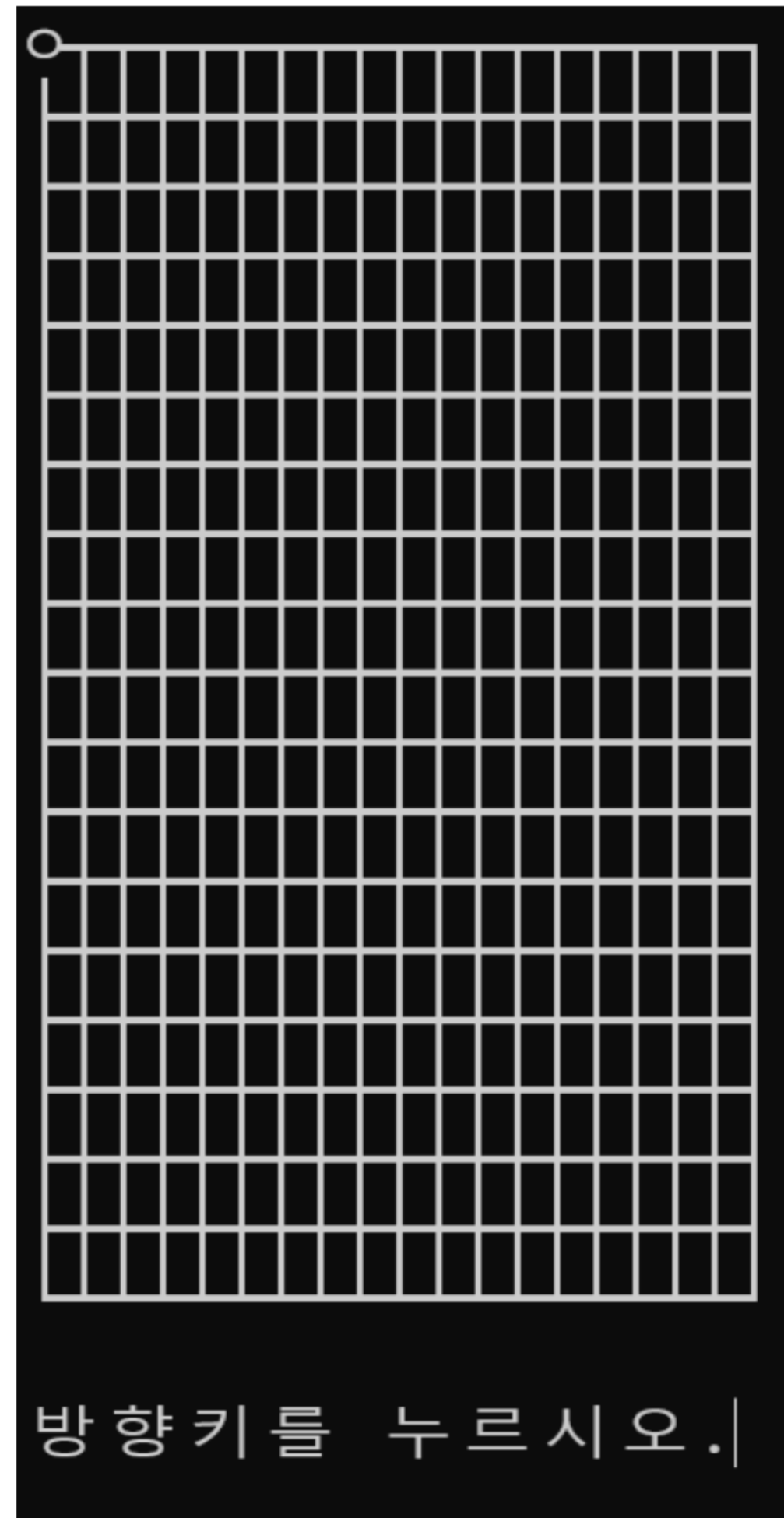
소프트웨어학과

김수민

# 게임 소개

## 오목

오목에 필요한 바둑판과 바둑돌



## 4가지의 핵심 함수

```
void move_arrow_key(char chr, int *x, int *y, int x_b, int y_b); // 방향키 입력  
void gotoxy(int x, int y); // 커서의 좌표  
void draw_check01(int c, int r); // 모눈종이의 형태의 격자 생성  
void game_control(void); // 게임 제어
```

# game\_control 함수

```
int main(void)
{
    game_control();
    return 0;
}
```

```
void game_control(void)
{
    int x=1, y=1; // 현재 점의 위치
    char key;
    do
    {
        gotoxy(1,1);
        draw_check01(18, 18); // 가로 세로
        // 커서좌표를 초기화후 격자를 그림

        gotoxy(x, y);
        printf("o ");
        // 사용자가 지정한 위치로 이동후 점을 그림니다

        gotoxy(1, 21);
        printf("방향키를 누르시오.");
        // 게임영역 하단에 문자열 표시
        key=getch(); // 사용자의 입력

        if (key==27)
            exit(0);
        else if (key>=72)
            move_arrow_key(key, &x, &y, 37, 19);
    }while(1);
}
```

# move\_arrow 함수

```
void move_arrow_key(char key, int *x1, int *y1, int x_b, int y_b)
{
    switch(key)
    {
        case 72: //위쪽(상) 방향의 화살표 키 입력
            *y1=*y1-1;
            if (*y1<1) *y1=1; //y좌표의 최소값
            break;
        case 75: //왼쪽(좌) 방향의 화살표 키 입력
            *x1=*x1-2; //좌측방향으로 2씩 감소
            if (*x1<1) *x1=1; //x좌표의 최소값
            break;
        case 77: //오른쪽(우) 방향의 화살표 키 입력;
            *x1=*x1+2; //우측방향으로 2씩 증가
            if (*x1>x_b) *x1=x_b; //x좌표의 최대값(경계)
            break;
        case 80: //아래쪽(하) 방향의 화살표 키 입력;
            *y1=*y1+1;
            if (*y1>y_b) *y1=y_b; //y좌표의 최대값(경계)
            break;
        default:
            return;
    }
}
```

## draw\_check 함수(1)

0xa6 => 16진수  
10진수로 표현하면 166

```
void draw_check01(int c, int r)
{
    int i, j;
    unsigned char a=0xa6;
    unsigned char b[12];

    for(i=1;i<12;i++)
        b[i]=0xa0+i;
    // 바둑판을 그리기 위한 문자를 저장:
```

## draw\_check 함수(2)

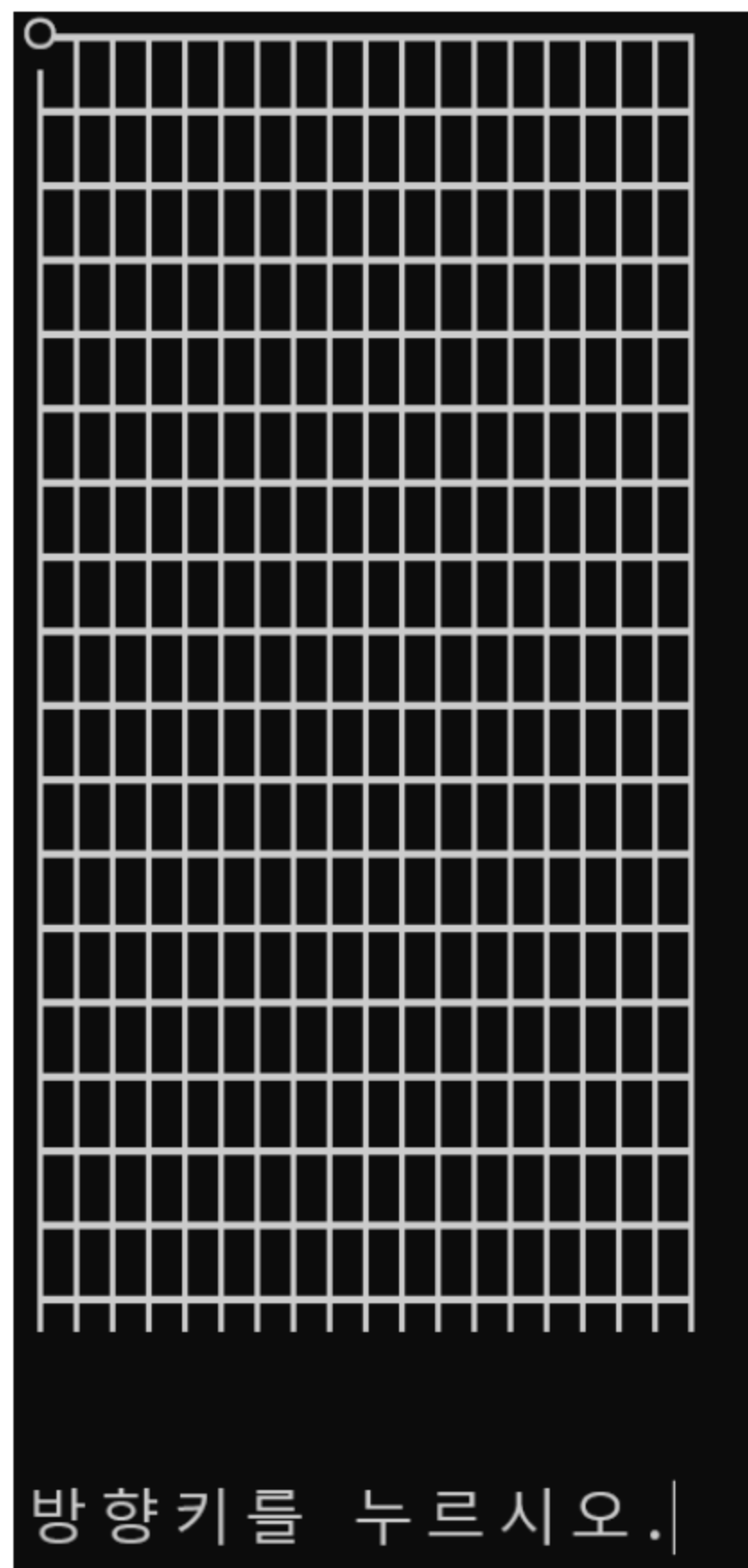
```
printf("%c%c", a, b[3]); // ㄱ |  
for(i=0; i<c-1; i++)  
... printf("%c%c", a, b[8]); // ㅍ  
printf("%c%c", a, b[4]); // ㄷ  
printf("\n");  
// 첫 줄의 가로 줄  
...
```



## draw\_check 함수(3)

3

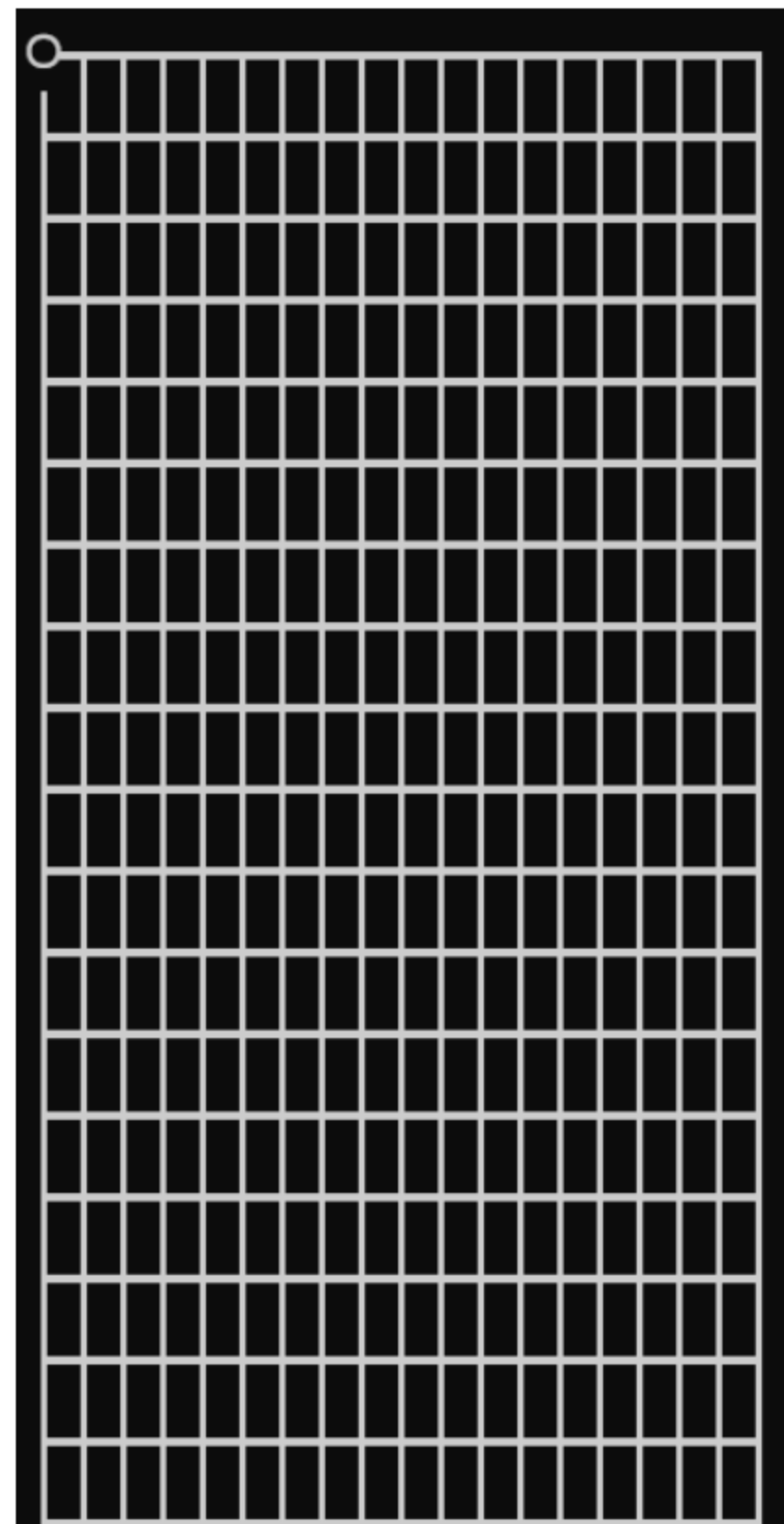
```
... for(i=0;i<r-1;i++)  
{  
printf("%c%c", a, b[7]); // |  
for(j=0;j<c-1;j++)  
... printf("%c%c", a, b[11]); // +  
printf("%c%c",a, b[9]); // |  
printf("\n");  
}  
// 중간 의 가로, 세로 줄;
```





## draw\_check 함수(4)

```
printf("%c%c", a, b[6]); // L
for(i=0;i<c-1;i++)
printf("%c%c", a, b[10]); // ⊥
printf("%c%c", a, b[5]); // J
printf("\n");
```



방향키를 누르시오.

# 사용된 특수문자 정리

<code>printf("%c%c", a, b[3]); -&gt;</code>	$\ulcorner$
<code>printf("%c%c", a, b[8]); -&gt;</code>	$\top$
<code>printf("%c%c", a, b[4]); -&gt;</code>	$\neg$
<code>printf("%c%c", a, b[7]); -&gt;</code>	$\vdash$
<code>printf("%c%c", a, b[11]); -&gt;</code>	$\vdash$
<code>printf("%c%c", a, b[9]); -&gt;</code>	$\dashv$
<code>printf("%c%c", a, b[6]); -&gt;</code>	$\perp$
<code>printf("%c%c", a, b[10]); -&gt;</code>	$\perp$
<code>printf("%c%c", a, b[5]); -&gt;</code>	$\lrcorner$

# gotoxy 함수

```
void gotoxy(int x, int y)
{
    COORD Pos = {x - 1, y - 1};
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), Pos);
}
```

**감사합니다**