

# 프로젝트 최종 보고서

---

주제 : 리눅스로 오목 구현하기



과 목 명 : 자료구조실습  
제출일자 : 2021년 12월 6일  
학 과 : 소프트웨어학부  
학 번 : 2020203002  
이 름 : 박상천

# 목 차

I. 주제 및 프로그램 목적	1
II. 개발 환경	2
III. 개발 일정	3
IV. 프로그램 기능	4
V. 프로그램 동작	5
VI. 문제 발생 및 해결방법	6
VII. 과제 후기	7
VIII. 참고 자료	8
IX. 작성 코드	9

## I. 주제 및 프로그램 목적



리눅스로 오목을 구현하는 것을 이번 프로젝트의 주제로 정하였습니다.

하드웨어의 성능이 높아지며 여러 게임들이 출시하고, 게임 시장에 도전을 하고 있습니다. 그러는 와중에 리눅스는 무료와 활발한 오픈소스 환경, 패키지 매니저(yum, rpm, 등등) 등 개발 친화적이기 때문에 리눅스를 많이 이용하는 추세입니다. 그렇기 때문에 게임을 만드는 쪽으로 진로를 나가고 싶은 저도 리눅스로 게임을 만들어 보는 것이 좋은 경험이 될 수 있을 것 같아 리눅스로 오목 게임을 구현하기로 한 것입니다. 이번 프로젝트는 제가 게임 개발자가 되는 발판이 되어 줄 것이라고 생각하여 이번 프로그램을 만들기로 결심하였습니다.

## II. 개발 환경



기본 리눅스 운영체제에서 C언어를 이용하여 오목을 구현할 것입니다.

사용하는 헤더 파일의 경우는 기본적인 입출력을 위해서 `stdio.h`를 이용할 것입니다. 사용자가 게임이 끝났을 때나 메인화면에서 게임을 종료할 수 있도록 프로그램을 종료하는 함수인 `exit()`를 사용하기 위해서 `stdlib.h` 헤더파일을 이용하였습니다.

### Ⅲ. 개발 일정

---



- 12주 – 프로젝트 주제 선정 및 프로젝트 계획서 제출
- 13주 – 프로젝트의 주요 기능들을 구현( ex) 오목 기능 )
- 14주 – 오류 수정 및 추가적으로 필요하다고 생각하는 기능추가  
( ex) 인터페이스 )
- 15주 – 최종 테스트 및 프로젝트 최종 보고서 제출

## IV. 프로그램 기능

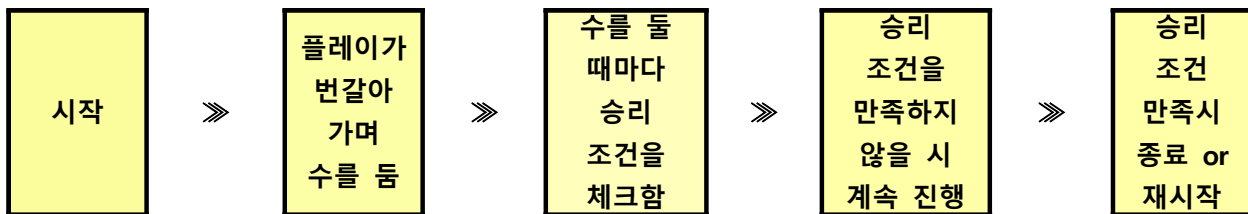


먼저 프로그램을 실행을 하면 처음 메인화면이 출력이 되고, 거기서 입력을 통하여 사용자가 다음 행동을 조작할 수 있습니다.

1을 입력하면 게임에 대한 간단한 설명을 볼 수 있는 페이지가 출력이 되고, 거기서 다시 1을 입력하면 메인 화면으로 돌아올 수 있습니다.

2를 입력하면 삼목이 실행이 됩니다. 또한 3을 입력하면 오목이 실행이 됩니다. 그렇게 삼목 또는 오목을 플레이하고 경기가 종료가 되면 프로그램에서 재경기를 할 것인지, 아니면 그냥 게임을 종료할 것인지를 묻습니다. 거기서 사용자가 각 기능에 맞는 입력을 해주면 그 기능들이 실행이 됩니다.

그리고 마지막 4를 입력하면 바로 게임이 종료가 됩니다.



< 오목 순서도 >

위는 오목(삼목) 프로그램이 동작하는 순서도입니다. 승리 조건을 만족하면 게임이 종료가 되고, 사용자로부터 입력을 받아 재경기를 할 것인지, 아니면 게임을 종료를 할 것인지 정해지게 됩니다. 만약 무승부가 나왔을 때에도 똑같이 프로그램이 입력을 받아 다음 기능을 선택할 수 있도록 동작하게 만들었습니다.

삼목은 3x3배열에 숫자를 지정하여 그 숫자에 O or X가 들어있지 않다면 현재 플레이어의 돌을 두게 하고, 그때마다 매번 똑같은 3개의 돌이 이어져 있는지 검사를 하는 방식을 game3라는 한 함수 안에 구현하였습니다.

오목과 같은 경우도 전체적인 흐름은 비슷하나, 삼목과 같이 한 함수 안에 구현을 하려고 하였으나 어려움(문제 발생 및 해결방법에서 설명)이 있어, 판은 그려주는 print함수, 승리 조건을 만족하는 체크하는 wincheck 함수, 사용자의 차례를 p1과 p2로 나누어 각 함수는 Player1, 2의 각 차례에서 시작할 때 실행하는 함수이다. 먼저 좌표를 입력받고, 그 좌표에 초기값(0)이면 돌을 놓고, 초기값이 아니면 다시 좌표를 받게하였습니다.



삼목 함수를 구현하고, 그 방식 그대로 오목함수를 구현하려고 하였더니 배열이 커져 그것을 불러오는 과정에서 char로 사용할 수 없었습니다. 그래서 string으로 각 배열에 영어를 저장해주어 그 영어가 들어오면 그 칸에 돌을 넣는 식으로 하려고 하였으나, 입력을 받는 과정에서 무한 루프가 도는 오류가 생겨 int 배열을 이용하여 처음부터 다시 오목 함수를 구상해야 했습니다.

처음 프로젝트를 기획하였을 때는 간단하게 배열에 넣고, 거기에 조건을 충족하면 배열에 돌을 넣고, 승리를 체크하는 식으로 한 함수 안에 다 구현하려고 생각하였습니다. 그리고 그대로 만들어 삼목 함수를 만들 수 있었습니다. 그렇지만 오목함수에서 배열을 사용하는 과정에서 오류가 생겼고, 그를 해결하기 위해서 먼저 int배열에 초기값을 설정해놓고, 좌표를 입력받았을 때 그 좌표에 초기값이 들어있으면 초기값을 Player에 맞는 값으로 넣어주었습니다. 그리고 그것을 판을 그리는 함수에서 체크하여 그 좌표에 판을 그리는 것 대신 돌을 그리게 하는 방식으로 해결하였습니다.



## VII. 과제 후기



과제를 계획하는 단계에서 게임을 리눅스로 구현하는 것이 미래에 게임을 만드는 업계에서 일할 때 분명 도움이 될 것 같아서 과제를 계획하였습니다. 중간중간에 오랜 시간을 투자하여 제대로 프로그래밍을 하려고 생각하였는데 오류가 뜰 때도 있었습니다. 그때마다 포기를 하고 싶기도 하였지만, 다시 처음부터 천천히 코드를 보고 오류가 어디서 났는지 알아내, 고치기를 반복하였습니다. 그러면서 점점 제 형태를 갖춰가는 프로그램을 보고 흥미가 붙어 더욱 몰입을 할 수 있었습니다. 처음에는 간단히 판을 출력하는 것에서 시작하여, 기능을 추가하여 게임이 되고, 거기에 인터페이스를 만들어서 프로그램을 멋지게 만드는 과정들을 거치며 비로소 프로그램을 만들 수 있었습니다. 저는 이번 과제를 통해서 사소한 오류를 찾아내어 고치며 자신이 어느 부분에 약해서 오류를 만들게 했는지 알 수 있었습니다, 또한 완성되어 제대로 작동하는 게임을 보고 미래에 게임을 만들고 싶다는 목표가 한층 더 확고해졌습니다. 이번 과제를 통한 경험은 앞으로 제가 나아갈 길에 있어서 큰 도움이 될것이라고 확신할 수 있었습니다.

## VIII. 참고 자료

---



<https://kldp.org/node/645> - 리눅스 터미널 지우기

<https://jybaek.tistory.com/339> - 인터페이스 색상 지정

<https://sonseungha.tistory.com/299> - 게임 종료

## IX. 작성 코드



```
#include <stdio.h>
#include <stdlib.h>

int explain(); //게임 설명화면
int intro(); //메인 화면
int game3(); //삼목 메인함수
////////////////////// *아래는 오목 함수*
int game5(); //오목 메인함수
int wincheck(); //승리조건 체크함수
void print(); //판 그려주는 함수
void p1(); //player1의 차례
void p2(); //player2의 차례
int x, y;
int map[10][10];

int main(void){
    int menu = -1;

    do
    {
        menu = intro();

        if (menu == 1) //게임설명
        {
            explain();
        }
        else if (menu == 2) //3x3
        {
            if(game3() == 0){
                exit(0);
            }
            else{
                game3();
            }
        }
        else if (menu == 3) //5x5
        {
            if(game5() == 0){
                exit(0);
            }
            else{
                game5();
            }
        }
    }
}
```

}

$$\}$$

[illegible]

```

scanf("%d", &num);

num--;

c = num % 3;
r = num / 3;
}while(num < 0 || num > 9 || map[r][c] > '9');

map[r][c] = (player == 1) ? 'X' : 'O';

if(((map[0][0] == map[1][1]) && (map[0][0] == map[2][2])) ||
    ((map[2][0] == map[1][1]) && (map[2][0] == map[0][2]))) {
    winner = player;
}
else {
    for(int i = 0; i <= 2; i++){
        if(((map[i][0] == map[i][1]) && (map[i][0] == map[i][2])) || ((map[0][i] ==
map[1][i]) && (map[0][i] == map[2][i]))) {
            winner = player;
        }
    }
}

system("clear");
printf("WnWn");
printf("+ ---+ ---+ ---+ Wn");
printf("| %c | %c | %c | Wn", map[0][0], map[0][1], map[0][2]);
printf("+ ---+ ---+ ---+ Wn");
printf("| %c | %c | %c | Wn", map[1][0], map[1][1], map[1][2]);
printf("+ ---+ ---+ ---+ Wn");
printf("| %c | %c | %c | Wn", map[2][0], map[2][1], map[2][2]);
printf("+ ---+ ---+ ---+ Wn");

if(winner == 0){
    printf("Wn이번 게임은 무승부입니다! Wn");
}
else {
    printf("Wn축하드립니다! 승자는 Player %d(%c)입니다!WnWnWn", winner, (player == 1) ? 'X'
: 'O');
}
printf("게임 종료 : 0 | 재시작 : 1 을 입력해주세요. => ");

int re = 9;
scanf("%d", &re);

```

```

if(re == 0){
    return 0;
}else if(re == 1){
    return 1;
}

return -1;
}

```

```

////////////////////////////////////

```

```

int wincheck() {
    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10; j++) {
            if (map[i][j] == 1 && map[i + 1][j + 1] == 1 && map[i + 2][j + 2] == 1 &&
map[i + 3][j + 3] == 1 && map[i + 4][j + 4] == 1) {
                return 1;
            }
            else if (map[i][j] == 1 && map[i][j + 1] == 1 && map[i][j + 2] == 1 && map[i][j
+ 3] == 1 && map[i][j + 4] == 1) {
                return 1;
            }
            else if (map[j][i] == 1 && map[j + 1][i] == 1 && map[j + 2][i] == 1 && map[j +
3][i] == 1 && map[j + 4][i] == 1) {
                return 1;
            }
            else if (map[i][j] == 1 && map[i - 1][j + 1] == 1 && map[i - 2][j + 2] == 1 &&
map[i - 3][j + 3] == 1 && map[i - 4][j + 4] == 1) {
                return 1;
            }
        }
    }
}

```

```

    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10; j++) {
            if (map[i][j] == 2 && map[i + 1][j + 1] == 2 && map[i + 2][j + 2] == 2 &&
map[i + 3][j + 3] == 2 && map[i + 4][j + 4] == 2) {
                return 2;
            }
            else if (map[i][j] == 2 && map[i][j + 1] == 2 && map[i][j + 2] == 2 && map[i][j
+ 3] == 2 && map[i][j + 4] == 2) {
                return 2;
            }
            else if (map[j][i] == 2 && map[j + 1][i] == 2 && map[j + 2][i] == 2 && map[j +
3][i] == 2 && map[j + 4][i] == 2) {
                return 2;
            }
        }
    }
}

```

```

    }
    else if (map[i][j] == 2 && map[i - 1][j + 1] == 2 && map[i - 2][j + 2] == 2 &&
map[i - 3][j + 3] == 2 && map[i - 4][j + 4] == 2) {
        return 2;
    }
}
}
}
}

```

```

void print() {
    system("clear");
    printf(" 0 1 2 3 4 5 6 7 8 9 \n");

    for (int i = 0; i < 10; i++) {
        printf("%d",i);
        for (int j = 0; j < 10; j++) {
            if (map[i][j] == 0) {
                printf(" + ");
            }
            else if (map[i][j] == 1) {
                printf("● ");
            }
            else if (map[i][j] == 2) {
                printf("○ ");
            }
        }
        printf("\n");
    }
}

```

```

void p1() {
    printf("\n");
    printf("● 돌의 차례입니다. X Y 좌표를 입력주세요( ex: 3 2 ) : ");
    scanf("%d %d",&x ,&y);
    if (map[y][x] == 0) {
        map[y][x] = 1;
    }
    else if (map[y][x] != 0) {
        printf("좌표를 다시 입력해주세요!\n");
        p1();
    }
}

```

```

void p2() {
    printf("\n");
    printf("○ 돌의 차례입니다. X Y 좌표를 입력주세요( ex: 3 2 ) : ");
}

```



```

scanf("%d %d", &x, &y);
if (map[y][x] == 0) {
    map[y][x] = 2;
}
else if (map[y][x] != 0) {
    printf("좌표를 다시 입력해주세요!\n");
    p2();
}
}

int game5() {
    int turn = 1;

    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10; j++) {
            map[i][j] = 0;
        }
    }
    x = 0;
    y = 0;

    print();
    while (1) {
        if (turn == 1) {
            p1();
            print();
            ++turn;
        }
        else if (turn == 2) {
            p2();
            print();
            --turn;
        }
        int a = wincheck();
        if (a == 1) {
            printf("\n");
            printf("축하합니다! ● 돌이 승리하였습니다!\n");
            break;
        }
        else if (a == 2) {
            printf("\n");
            printf("축하합니다! ○ 돌이 승리하였습니다!\n");
            break;
        }
    }
}

```

```
printf("Wn");
printf("게임 종료 : 0 | 재시작 : 1 을 입력해주세요! => ");

int ans = 9;
scanf("%d", &ans);

if (ans == 0) {
    return 0;
}
else if (ans == 1) {
    return 1;
}
return -1;
}
```