

2019 C++ 프로그래밍 프로젝트 보고서

Ncurses library를 이용한 PushBox Game

11조

소프트웨어학부

20141733 조주현

20153227 정용훈

20153220 임지홍

▣ 목차 ▣

□ 1단계

ncurses library 함수를 사용하여 2차원 배열로 표시된 push box map 을 game 화면으로 표시하는 프로그램을 완성한다.

□ 2단계

1단계의 맵 위에 캐릭터를 표시하고 화살표를 입력 받아 캐릭터가 움직이도록 프로그램을 완성한다.

□ 3단계

2단계 프로그램에서 step 횟수 (캐릭터가 이동한 횟수) 와 push 횟수 (상자가 움직인 횟수) 를 화면에 보여준다. 모든 상자가 목적지에 도달하면 게임을 끝내고 다음 map으로 넘어가서 다시 게임을 시작하는 프로그램을 완성한다.

□ 1단계

levList() 함수에서 맵의 기본적인 틀을 배열로 그린 뒤, level() 함수에서 구현했습니다.

저희는 이번 push box 컨셉을 눈 마을 테마로, ASCII 코드를 사용하여 디자인했습니다.

배열에 저장된 숫자에 따라 switch문을 통해

0(이동 가능한 공간)과 4(외부)일 경우 -> ' '(빈칸) (ASCII 코드 => 32)

1(벽)의 경우 -> '#' (ASCII 코드 => 35)

3(골인 지점) -> 'x' (ASCII 코드 => 120)

2(상자) -> '@' (ASCII 코드 => 64)

을 사용했습니다.

◎ 캐릭터를 제외한 맵의 구현 코드 : levList() 함수

```
void pushBox::levList(int *h, int *w, int *array, int y, int x, int n)
```

```
if (n==0)
{
    *h=9;
    *w=7;
    int map0[9][7]={
        {1, 1, 1, 1, 4, 4, 4},
        {1, 3, 0, 1, 1, 4, 4},
        {1, 3, 5, 0, 1, 4, 4},
        {1, 3, 0, 2, 1, 4, 4},
        {1, 1, 2, 0, 1, 1, 1},
        {4, 1, 0, 2, 0, 0, 1},
        {4, 1, 0, 0, 0, 0, 1},
        {4, 1, 0, 0, 1, 1, 1},
        {4, 1, 1, 1, 1, 4, 4}};

    *array=map0[y][x];
}

else if(n==1)
{
    *h=7;
    *w=6;
    int map1[7][6]={
        {1,1,1,1,1,4},
        {1,0,0,0,1,4},
        {1,3,3,3,1,4},
        {1,2,2,2,1,1},
        {1,0,0,0,0,1},
        {1,0,5,0,0,1},
        {1,1,1,1,1,1}};

    *array=map1[y][x];
}

else if(n==2)
{
    *h=6;
    *w=8;
    int map2[6][8]={
        {1,1,1,1,1,1,1,1},
        {1,3,0,0,0,0,0,1},
        {1,0,3,2,2,2,5,1},
        {1,3,0,0,0,0,0,1},
        {1,1,1,1,1,0,0,1},
        {4,4,4,4,1,1,1,1}};

    *array=map2[y][x];
}

else if(n==3)
{
    *h=8;
    *w=7;
    int map3[8][7]={
        {1,1,1,1,1,1,1},
        {1,0,0,0,0,0,1},
        {1,0,3,2,3,0,1},
        {1,0,2,3,2,0,1},
        {1,0,3,2,3,0,1},
        {1,0,2,3,2,0,1},
        {1,0,0,5,0,0,1},
        {1,1,1,1,1,1,1}};

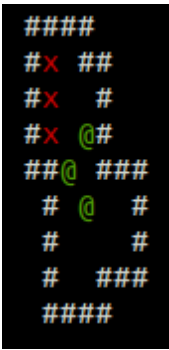
    *array=map3[y][x];
}

else if(n==4)
{
    *h=8;
    *w=10;
    int map4[8][10]={
        {4,1,1,1,1,4,4,4,4,4},
        {4,1,0,0,1,1,1,1,4,4},
        {4,1,0,0,0,0,0,1,1,4},
        {1,1,0,1,1,0,0,0,1,4},
        {1,3,0,3,1,0,5,2,1,1},
        {1,0,0,0,1,0,2,2,0,1},
        {1,0,0,3,1,0,0,0,0,1},
        {1,1,1,1,1,1,1,1,1,1}};

    *array=map4[y][x];
}
```

◎ 캐릭터를 제외한 맵의 실행 화면

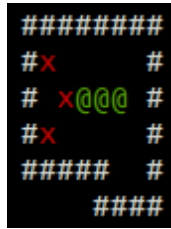
Lev 1



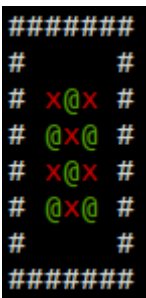
Lev 2



Lev 3



Lev 4



Lev 5



◎ 구현한 맵의 실행 코드 : level() 함수

```
void pushBox::Level(int n){
    int x=0,y=0, z=0,h=1,w=1,map;
    wbox=0;
    goal=0;

    for(y=0;y<h;y++){
        {
            for(x=0;x<w;x++){
                {
                    levList(&h,&w,&map,y,x,n);
                    switch(map)
                    {
                        case 0:
                            mvaddch(y+4,x+10,' '|COLOR_PAIR(2));
                            break;
                        case 4:
                            mvaddch(y+4,x+10,' '|COLOR_PAIR(1));
                            break;
                        case 1:
                            mvaddch(y+4,x+10,'#'|COLOR_PAIR(1));
                            break;
                        case 3:mvaddch(y+4,x+10,'x'|COLOR_PAIR(2));
                            g[goal].xP=x+10;
                            g[goal].yP=y+4;
                            goal++;
                            break;
                        case 2:
                            mvaddch(y+4,x+10,' '|COLOR_PAIR(2));
                            wbox +=1;
                            obj[wbox].ozn=mvinch(y+4,x+10);
                            obj[wbox].yP=y+4;
                            obj[wbox].xP=x+10;
                            obj[wbox].zn='@';
                            mvaddch(obj[wbox].yP,obj[wbox].xP,obj[wbox].zn|COLOR_PAIR(3));
                            break;
                    }
                }
            }
        }
    }
}
```

위 코드는 배열 map에 들어 있는 맵 정보를 이용하여 맵을 그리는 코드입니다.

□ 2단계

맵의 구현 코드 levList()에 캐릭터를 5로 설정해 맵 정보에 캐릭터의 시작 장소를 표시했습니다.

5(캐릭터) -> '8'(눈사람 모양) (ASCII 코드 => 56)

맵의 실행 코드에 아래의 코드를 추가해 캐릭터를 표시했습니다.

◎ 캐릭터의 구현 코드 : level() 함수

```
case 5:
    mvaddch(y+4,x+10,' ' | COLOR_PAIR(2));
    obj[0].ozn=mvinch(y+4,x+10);
    obj[0].yP=y+4;
    obj[0].xP = x+10;
    obj[0].zn='8';
    mvaddch(obj[0].yP,obj[0].xP,obj[0].zn | COLOR_PAIR(1));
    break;
}
}
}
move(obj[0].yP, obj[0].xP);
```

게임 내 움직임은 play() 함수에서 구현했습니다.

obj[0]를 통해 캐릭터를 나타냈고, obj[0]를 제외한 나머지 obj[i]에 맵 정보를 담았습니다.

여기서 캐릭터가 이동할 때와 상자가 움직일 때, count 값과 count_box 값을 추가해서 step과 push값을 저장할 수 있도록 했습니다.

캐릭터의 이동은 키패드(<, >, ↑, ↓)를 사용하여 움직이도록 했고 캐릭터의 이동과 상자의 이동은 아래와 같이 구현했습니다.

◎ 캐릭터의 구현 코드 : play() 함수

아래 코드는 이동 제한과 관련한 예외 처리를 위한 값들을 변수를 지정해 저장한 코드입니다.

```
void pushBox::Play(int input)
{
    bool restart = FALSE;
    clearStage=false;

    chtype up, lf, dw, rg, oup, olf, odw, org;

    up = (mvinch(obj[0].yP-1, obj[0].xP) & A_CHARTEXT);
    lf = (mvinch(obj[0].yP, obj[0].xP-1) & A_CHARTEXT);
    dw = (mvinch(obj[0].yP+1, obj[0].xP) & A_CHARTEXT);
    rg = (mvinch(obj[0].yP, obj[0].xP+1) & A_CHARTEXT);
    oup = (mvinch(obj[0].yP-2, obj[0].xP) & A_CHARTEXT);
    olf = (mvinch(obj[0].yP, obj[0].xP-2) & A_CHARTEXT);
    odw = (mvinch(obj[0].yP+2, obj[0].xP) & A_CHARTEXT);
    org = (mvinch(obj[0].yP, obj[0].xP+2) & A_CHARTEXT);

    for(int i=0; i<=wbox; i++)
    {
        mvaddch(obj[i].yP, obj[i].xP, obj[i].ozn);
    }
}
```

- keypad의 입력에 따른 switch 문

◦ ↑

```
case KEY_UP:
    if(up != 35)
    {
        if(up==64 && (oup==' ' || oup==120))
        {
            obj[0].yP -= 1;
            for(int i=1; i<=wbox; i++)
            {
                if((obj[0].yP==obj[i].yP) && (obj[0].xP==obj[i].xP))
                {
                    obj[i].yP -= 1;
                    count++;
                    count_box++;
                }
            }
        }
        else if(up!=64)
        {
            obj[0].yP -= 1;
            count++;
        }
    }
    isclear();
    printCount(count, count_box);
    break;
```

◦ ←

```
case KEY_LEFT:
    if(lf!=35)
    {
        if(lf==64 && (olf==' ' || olf==120))
        {
            obj[0].xP -= 1;
            for(int i=1; i<=wbox; i++)
            {
                if((obj[0].yP==obj[i].yP) && (obj[0].xP==obj[i].xP))
                {
                    obj[i].xP -= 1;
                    count++;
                    count_box++;
                }
            }
        }
        else if(lf!=64)
        {
            obj[0].xP -= 1;
            count++;
        }
    }
    isclear();
    printCount(count, count_box);
    break;
```

◦ ↓

```
case KEY_DOWN:
    if(dw!=35)
    {
        if(dw==64 && (odw==' ' || odw==120))
        {
            obj[0].yP += 1;
            for(int i=1; i<=wbox; i++)
            {
                if((obj[0].yP==obj[i].yP) && (obj[0].xP==obj[i].xP))
                {
                    obj[i].yP += 1;
                    count++;
                    count_box++;
                }
            }
        }
        else if(dw!=64)
        {
            obj[0].yP += 1;
            count++;
        }
    }
    isclear();
    printCount(count, count_box);
    break;
```

◦ →

```
case KEY_RIGHT:
    if (rg!=35)
    {
        if (rg==64 && (org == ' ' || org == 120))
        {
            obj[0].xP +=1;
            for (int i=1; i<=wbox; i++)
            {
                if ((obj[0].yP == obj[i].yP) && (obj[0].xP==obj[i].xP))
                {
                    obj[i].xP+=1;
                    count++;
                    count_box++;
                }
            }
        }
        else if (rg!=64)
        {
            obj[0].xP+=1;
            count++;
        }
    }
    isclear();
    printCount(count, count_box);
    break;
```

- 기타 기능의 switch 문

◦ restart 기능 'r'

```
case 'r':
case 'R':
    restart=TRUE;
    Level(lev);
    count=0;
    count_box=0;
    break;
```

◦ quit 기능 'q'

```
case 'Q':  
case 'q':  
    want_play=false; if (ch=='q' || ch=='Q')  
    count=0;        {  
    count_box=0;      clear();  
    break;           break;  
                    }
```

quit의 기능같은 경우 main 화면으로 돌아가야 하므로
main() 함수에 추가적으로 오른쪽 코드를 삽입했습니다.

□ 3단계

step값과 push값의 경우, play() 함수에서 저장된 count 값과 count_box 값을 printCount() 함수를 통해 화면에 나타
나게 했습니다.

◎ step, push 출력 코드 : printCount() 함수

```
void pushBox::printCount(int count, int count_box)  
{  
    mvprintw(1,21, "%d", count);  
    mvprintw(1,31, "%d", count_box);  
}
```

캐릭터가 움직일 때마다 isClear() 함수를 통해 현재 레벨을 클리어 했는지 확인했습니다.

◎ 레벨 클리어 여부 확인 코드 : isclear() 함수

```
void pushBox::isclear()  
{  
    chtype now[wbox]={};  
    chtype goal[wbox];  
  
    for (int i=0; i<wbox; i++)  
    {  
        goal[i]='x';  
    }  
    clearStage=false;  
  
    for (int i=0; i<wbox; i++)  
    {  
        now[i] = (mvinch(obj[i+1].yP, obj[i+1].xP) & A_CHARTEXT);  
    }  
    if (compare(now, goal, wbox))  
    {  
        clearStage=true;  
        lev++;  
    }  
}
```

박스의 위치가 골인 지점의 위치가 일치하는지를 비교해 클리어 여부를 판단했습니다.

레벨을 클리어하면 clear_lev() 함수를 통해 클리어했다는 새로운 창을 띄우도록 했습니다.

◎ 레벨 클리어 표현 코드 : clear_lev() 함수

```
void pushBox::clear_lev()
{
    win=newwin(5,14,9,13);
    wbkgd(win,COLOR_PAIR(4));
    wborder(win,'|', '|', '-', '-', '+', '+', '+', '+');
    watttrn(win,COLOR_PAIR(4));
    wattroff(win,COLOR_PAIR(4));
    int ch;

    if (lev==5)
    {
        mvwprintw(win,1,2,"ALL CLEAR!");
        mvwprintw(win,2,2,"SCORE = %d",count);
        mvwprintw(win,3,4,">quit");
        wmove(win,3,4);
        wrefresh(win);
        while ((ch=wgetch(win))!='\n') {}
        lev=0;
        want_play=false;
    }
    else
    {
        mvwprintw(win,1,4,"CLEAR!");
        mvwprintw(win,2,2,"SCORE = %d",count);
        mvwprintw(win,3,1,">next quit");
        keypad(win,TRUE);
        wmove(win,3,1);
        wrefresh(win);

        wrefresh(win);
    }

    while ((ch=wgetch(win))!='\n')
    {
        switch(ch)
        {
            case KEY_LEFT:
                mvwprintw(win,3,1,">next quit");
                wmove(win,3,1);
                break;
            case KEY_RIGHT:
                mvwprintw(win,3,1,"next >quit");
                wmove(win,3,8);
                break;
            default:
                break;
        }
    }

    int x, y;
    getyx(win,y,x);

    if (x==1)
    {
        clearStage=false;
        want_play=true;
    }
    else want_play=false;

    count=0;
    count_box=0;

    delwin(win);
    clear();
}
```

이 때 NEXT를 누르면 다음 레벨을 플레이할 수 있고, QUIT을 누르면 홈 화면으로 돌아갑니다.
홈 화면에서 플레이를 누를 경우 QUIT을 눌렀던 레벨부터 플레이할 수 있습니다.

만약 레벨 5까지 다 클리어했을 경우 ALL CLEAR!라는 문구와 함께 레벨 1로 다시 돌아갑니다.

□ 추가 구현

◎ 홈 화면 : HOME() 함수

```
void pushBox::HOME(int ch){

    border('|', '|', '-', '-', '+', '+', '+', '+');
    bkgd(COLOR_PAIR(4));
    attron(COLOR_PAIR(4));
    attroff(COLOR_PAIR(4));
    mvprintw(7,6," WELCOME TO THE SNOW WORLD!");

    mvprintw(9,6,"      *      *      *");
    mvprintw(10,6,"      *  /.\\      *");
    mvprintw(11,6,"      *  /..\\      *");
    mvprintw(12,6,"      *  /'.\\      *");
    mvprintw(13,6,"      *  /.'.'\\      *");
    mvprintw(14,6,"      *  /.'.'.'\\      *");
    mvprintw(15,6,"      *  /.'.'.'.'\\      *");
    mvprintw(16,6,"      *  ^^^[_]^^^      *");

    mvprintw(18,17,">PLAY");
    mvprintw(20,16," QUIT ");
    move( 18, 16);

    while((ch = getch()) != '\n')
    {
        switch(ch)
        {
            case KEY_UP:
                mvprintw(18,17,">PLAY");
                mvprintw(20,16," QUIT ");
                move( 18, 17);
                break;

            case KEY_DOWN:
                mvprintw(18,16," PLAY ");
                mvprintw(20,17,">QUIT");
                move( 20, 17);
                break;

            default:
                break;
        }
        int x, y;
        getyx(stdscr,y,x);

        if (y==18)
        {
            clearStage=false;
            want_play=true;
            delwin(home);
            clear();
        }
        else
        {
            want_play=false;
            delwin(home);
            endwin();
            exit(1);
        }
    }
}
```

◎ 홈 실행 화면



□ 완성된 pushBox 게임 실행 화면



Level 3 step = 0 push = 0
Move=keypad, Restart-R, Exit-Q.

```

*   #####   *   *
*   * #x   #   *   *
*   * # x@@@8#   *   *
*   * #x   #   *   *
*   * #####   *   *
*   *         #   *   *
*   *         #####
*   *
*   *
*   *

```

Level 3 step = 21 push = 10
Move=keypad, Restart-R, Exit-Q.

```

*   #####   *   *
*   * #@8   #   *   *
*   * # @   #   *   *
*   * # @   #   *   *
*   * #####   *   *
*   *
*   *
*   *

```

CLEAR!
SCORE = 21
>next quit

Level 4 step = 0 push = 0
Move=keypad, Restart-R, Exit-Q.

```

*   #####   *   *
*   * #   #   *   *
*   * # x@x #   *   *
*   * # @x@ #   *   *
*   * # x@x #   *   *
*   * # @x@ #   *   *
*   * # 8 #   *   *
*   * #####
*   *
*   *

```

Level 4 step = 70 push = 12
Move=keypad, Restart-R, Exit-Q.

```

*   #####   *   *
*   * #   #   *   *
*   * # @ @ #   *   *
*   * # @ #   *   *
*   * # @ @ #   *   *
*   * #
*   * #
*   * #####
*   *
*   *

```

CLEAR!
SCORE = 70
>next quit

Level 5 step = 0 push = 0
Move=keypad, Restart-R, Exit-Q.

```

*   #####   *   *
*   * # #####   *   *
*   * #   ##   *   *
*   * ## ## #   *   *
*   * #x x# 8@##   *   *
*   * # # @ @ #   *   *
*   * # x#   #   *   *
*   * #####
*   *
*   *

```

Level 5 step = 114 push = 41
Move=keypad, Restart-R, Exit-Q.

```

*   #####   *   *
*   * # #####   *   *
*   * #   ##   *   *
*   * ## ## #   *   *
*   * # @ @ #   *   *
*   * #
*   * # 8
*   * #####
*   *
*   *

```

ALL CLEAR!
SCORE = 114
>quit