

시작화면 UI 및 세부코드

```

5 export ID1P="1P LOGIN"
6 export ID2P="2P LOGIN"
7 JOIN="\e[44;1m JOIN \e[0m"
8 SIGN_IN="\e[44;1m SIGN IN \e[0m"
9 EXIT="\e[44;1m EXIT \e[0m"
10 SIGN_OUT="\e[44;1m SIGN OUT \e[0m"
11 LOG1P="\e[44;1m $ID1P \e[0m"
12 LOG2P="\e[44;1m $ID2P \e[0m"
13 SELECTED=0
14
15
16 makeUI(){
17     echo -e "\n\n\n\n\n\n\n\n\n\n"
18     echo "
19     echo "
20     echo "
21     echo "
22     echo "
23     echo "
24     echo "
25     echo "
26     echo "
27     echo "
28     echo "
29     echo "
30     echo "
31     echo "
32     echo -e "                                $LOG1P                                $SIGN_IN"
33     echo ""
34     echo -e "                                $LOG2P                                $SIGN_OUT"
35     echo ""
36     echo -e "                                $JOIN                                $EXIT"
37     echo -e "\n\n"
38 }

```



line 5,6 : ID와 PW를 export로 설정하여 다른 셸에서도 변수이용이 가능하게 설정하였다.

그 외의 변수나 UI는 project1에서 했던 것과 동일하게 구현하였다.

```

52 while true;
53 do
54     makeUI
55     input_key
56     if [ $SELECTED = "0" ];then
57         LOG1P="\e[41;1m $ID1P \e[0m";SELECTED=1
58     elif [ $SELECTED = "1" ];then
59         LOG1P="\e[44;1m $ID1P \e[0m"
60         if [ "$INPUT" = "^[A" ];then
61             JOIN="\e[41;1m JOIN \e[0m"; SELECTED=5
62         elif [ "$INPUT" = "^[B" ];then
63             LOG2P="\e[41;1m $ID2P \e[0m"; SELECTED=3
64         elif [ "$INPUT" = "^[C" ] || [ "$INPUT" = "^[D" ];then
65             SIGN_IN="\e[41;1m SIGN IN \e[0m";SELECTED=2
66         elif [ $INPUT = "" ];then
67             clear
68             chmod 755 LOG1P.sh
69             source ./LOG1P.sh

```

EXIT에서 엔터를 누르기 전까지 실행시키기 위해 while문 사용.

```

120     elif [ $SELECTED = "5" ];then
121         JOIN="\e[44;1m JOIN \e[0m"
122         if [ "$INPUT" = "^[[A" ];then
123             LOG2P="\e[41;1m $ID2P \e[0m"; SELECTED=3
124         elif [ "$INPUT" = "^[[B" ];then
125             LOG1P="\e[41;1m $ID1P \e[0m";SELECTED=1
126         elif [ "$INPUT" = "^[[C" ] || [ "$INPUT" = "^[[D" ];then
127             EXIT="\e[41;1m EXIT \e[0m";SELECTED=6
128         elif [ $INPUT = "" ];then
129             if [ "$ID1P" != "1P LOGIN" ] && [ "$ID2P" != "2P LOGIN" ];then
130                 if [ "$ID1P" != "로그인실패" ] && [ "$ID2P" != "로그인실패" ];then
131                     if [ "$ID2P" != "ID중복" ];then
132                         clear
133                         chmod 755 JOIN.sh
134                         source ./JOIN.sh
135                         SELECTED=0
136                     else
137                         break
138                     fi
139                 else
140                     break
141                 fi
142             else
143                 break
144             fi

```

1P또는 2P가 로그인이 되어있지 않다면 프로그램 종료.

1P의 ID와 2P의 ID가 같다면 프로그램 종료.

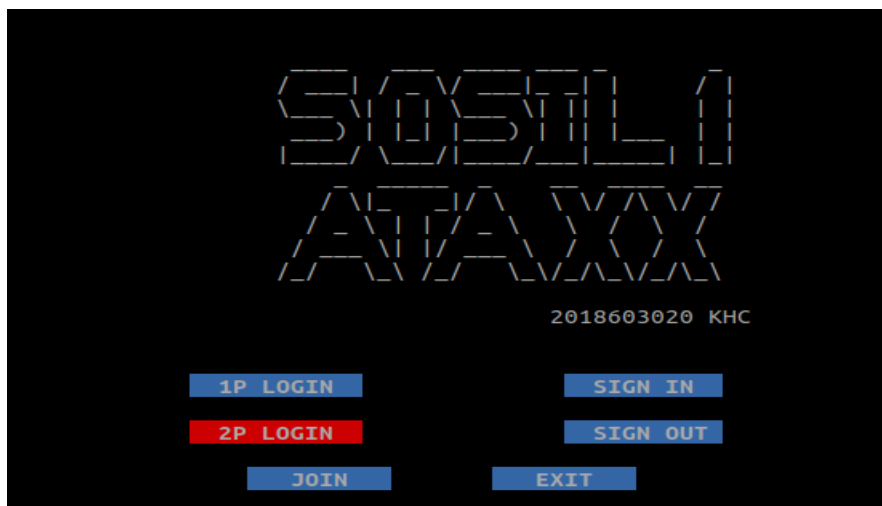
LINE 134 : 셸이 넘어가는 과정에서 source ./JOIN.sh란 명령을 쓰게 되는데

우선 source를 사용하지 않고 그냥 ./JOIN.sh로 셸을 변경했을 때

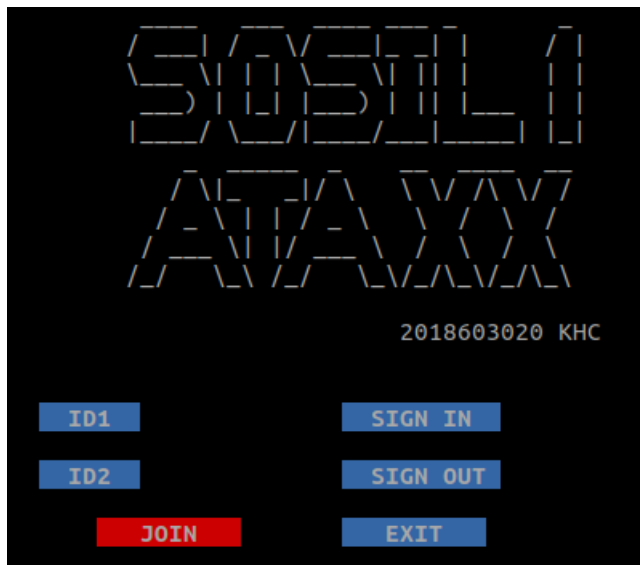
export로 설정한 ID값들이 넘어가지 않게 된다.

이러한 이유는 export로 설정한 값들은 현재 파일에서만 수정이 되었지 다른 파일에서는 적용이 안된다.

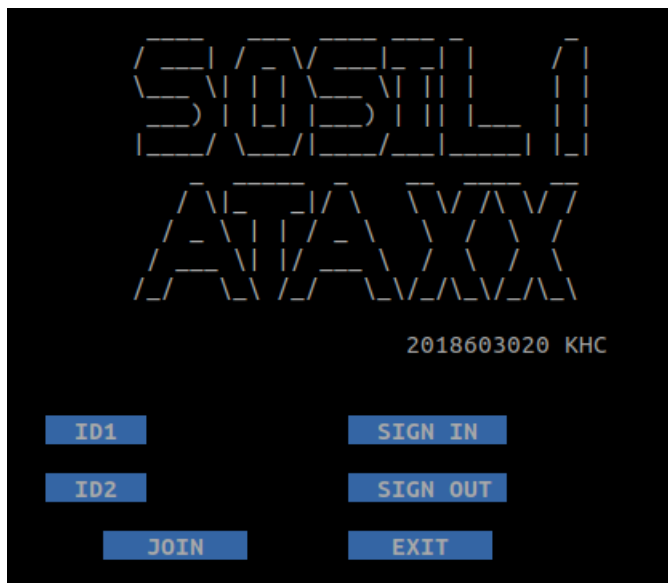
따라서 source 명령을 사용해 스크립트 파일에서 수정한 값을 바로 적용하기 위해 사용하였다.



1P LOGIN , 2P LOGIN , SIGN IN , SIGN OUT 은 저번 프로젝트와 동일하므로 생략하였다.



아이디 1과 아이디 2를 입력 후 나왔을 때의 모습.



JOIN을 눌렀을 때 나오는 UI

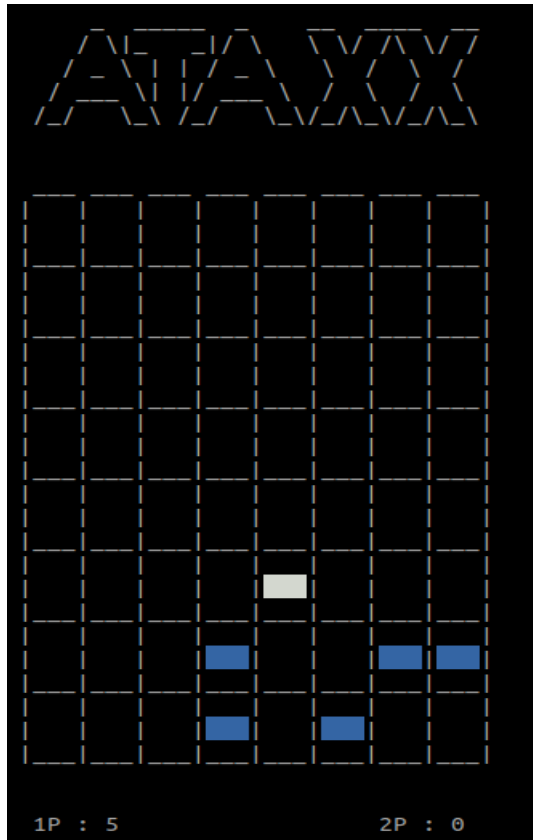


이때 유의해야 할 점은 백슬래시 3개를 써야 화면에 2개가 출력된다.

맵 선택화면



Pdf의 나온 그대로의 UI를 구현하지는 못했다. 그 외의 하이라이팅 기능은 작동한다.



맵1 의 구현

```
4 score1=0
5 score2=0
6 x=0;y=0
7
8 declare -A A
9 declare -A blue
10 for ((i=1; i<9; i++));do
11     for ((j=1; j<9; j++));do
12         A[$i,$j]="\e[0;1m  \e[0m"
13         blue[$i,$j]="0"
14     done
15 done
16
```

맵에서 하이라이팅되는 부분들을 사용하기 위해 배열을 사용하였다.

blue배열은 현재 자신이 가르키고 있는 부분이 blue인지 확인하기 위한 용도이다.

C 에서 bool과 같은 기능을 한다.

2중 for문으로 초기화를 시켜주었다.

```
echo -e "\n"
echo -e " "
echo -e " "
echo -e " "
echo -e " |${A[1,1]}|${A[1,2]}|${A[1,3]}|${A[1,4]}|${A[1,5]}|${A[1,6]}|${A[1,7]}|${A[1,8]}| "
echo -e " | "
echo -e " |${A[2,1]}|${A[2,2]}|${A[2,3]}|${A[2,4]}|${A[2,5]}|${A[2,6]}|${A[2,7]}|${A[2,8]}| "
echo -e " | "
echo -e " |${A[3,1]}|${A[3,2]}|${A[3,3]}|${A[3,4]}|${A[3,5]}|${A[3,6]}|${A[3,7]}|${A[3,8]}| "
echo -e " | "
echo -e " |${A[4,1]}|${A[4,2]}|${A[4,3]}|${A[4,4]}|${A[4,5]}|${A[4,6]}|${A[4,7]}|${A[4,8]}| "
echo -e " | "
echo -e " |${A[5,1]}|${A[5,2]}|${A[5,3]}|${A[5,4]}|${A[5,5]}|${A[5,6]}|${A[5,7]}|${A[5,8]}| "
echo -e " | "
echo -e " |${A[6,1]}|${A[6,2]}|${A[6,3]}|${A[6,4]}|${A[6,5]}|${A[6,6]}|${A[6,7]}|${A[6,8]}| "
echo -e " | "
echo -e " |${A[7,1]}|${A[7,2]}|${A[7,3]}|${A[7,4]}|${A[7,5]}|${A[7,6]}|${A[7,7]}|${A[7,8]}| "
echo -e " | "
echo -e " |${A[8,1]}|${A[8,2]}|${A[8,3]}|${A[8,4]}|${A[8,5]}|${A[8,6]}|${A[8,7]}|${A[8,8]}| "
echo -e " | "
echo -e " 1P : $score1 2P : $score2 "
```

각각의 칸들을 효율적인 방법으로 구현을 하고싶었지만 아이디어가 떠오르지 않아

2차원배열을 이용하였다.

```

66 while true;
67 do
68     clear
69     makeUI
70     input_key
71     if [ $x = "0" ];then
72         A[8,8]="\e[47;1m    \e[0m"
73         x=8;y=8;
74         continue
75     fi
76     if [ $y = "1" ] && [ $INPUT = "^[D" ];then
77         continue
78     fi
79     if [ $x = "1" ] && [ $INPUT = "^[A" ];then
80         continue
81     fi
82     if [ $y = "8" ] && [ $INPUT = "^[C" ];then
83         continue
84     fi
85     if [ $x = "8" ] && [ $INPUT = "^[B" ];then
86         continue
87     fi

```

맵 각각의 칸들을 행렬로 생각하여 x와 y를 설정 만약 아무것도 선택이 안된 경우 x와 y를 8로 초기화 시켜준 후 while문 뒤에 나오는 명령을 건너뛰기 위해 continue. 우선 기저사례로 각각의 칸들을 벗어나면 안되는 조건이 있으므로 각 상황에 맞게 Continue 옵션을 넣어서 아무 일도 일어나지 않도록 설정하였다.

```

88     if [ "$INPUT" = "^[A" ];then
89         if [ "${blue[$x,$y]}" = "1" ];then
90             A[$x,$y]="\e[44;1m    \e[0m"
91         else
92             A[$x,$y]="\e[0;1m    \e[0m"
93         fi
94         x=`expr $x - 1`
95         A[$x,$y]="\e[47;1m    \e[0m"

```

방향키가 상,하,좌,우 중 하나가 눌렸을 경우 만약 이동하기 전의 위치의 blue였을 경우 이동하기 전 배경색을 blue로 바꾸고 그게 아니라면 검정색으로 바꾼다. 그 후 x의 값을 변경해 행의 위치를 변경하고 변경된 위치의 배경색을 흰색으로 바꾼다. (나머지 하, 좌, 우 동일하게 구현하였다.)

```

120     elif [ $INPUT = "" ];then
121         A[$x,$y]="\e[44;1m    \e[0m"
122         if [ "${blue[$x,$y]}" = "1" ];then
123             continue
124         else
125             blue[$x,$y]=1
126             score1=`expr $score1 + 1`
127         fi
128     else
129         continue

```

enter를 입력 받았을 경우 현재 위치의 배경색을 파란색으로 바꾼다.

만약 기존에 이미 blue였을 경우 continue

기존에 blue가 아니었을 경우 blue를 true로 바꾸고 1P의 스코어를 1 증가시킨다.

맵2의 구현



Pdf의 나온 기능들 전부 잘 작동한다.

```
8 declare -A A
9 declare -A blue
10 declare -A yellow
11 for ((i=1; i<9; i++));do
12     for ((j=1; j<9; j++));do
13         blue[$i,$j]="0"
14         yellow[$i,$j]="0"
15         if [ $i -ge 2 ] && [ $i -le 7 ];then
16             sum=`expr $i + $j`
17             if [ $i = $j ] || [ "$sum" = 9 ];then
18                 yellow[$i,$j]="1"
19             fi
20         fi
21         if [ "${yellow[$i,$j]}" = "1" ];then
22             A[$i,$j]="\e[43;1m  \e[0m"
23         else
24             A[$i,$j]="\e[0;1m  \e[0m"
25         fi
26     done
27 done
```

맵1에서 했던것에서 추가로 yellow 배열을 넣어주었다.

(x,y) 에서 x가 2이상 7이하인 경우 :

X==Y 일 때 yellow true .

또는 X+Y=9 일 때 yellow는 true로 설정된다.

그 후 yellow가 true면 노란색 아니면 검은색으로 기본화면을 초기화하였다.


```

101     if [ "$INPUT" = "^[[A" ];then
102         if [ "${blue[$x,$y]}" = "1" ];then
103             A[$x,$y]="\e[44;1m  \e[0m"
104         elif [ "${yellow[$x,$y]}" = "1" ];then
105             A[$x,$y]="\e[43;1m  \e[0m"
106         else
107             A[$x,$y]="\e[0;1m  \e[0m"
108         fi
109         x=`expr $x - 1`
110         A[$x,$y]="\e[47;1m  \e[0m"

```

상하좌우 선택했을 때 yellow가 true인 경우만 추가로 고려하여

현재 위치의 배경색으로 어떤 것이 와야하는지 구현하였다.

그 외의 내용은 맵1에서의 구현과 같다.

```

141     elif [ $INPUT = "" ];then
142         if [ "${yellow[$x,$y]}" = "1" ];then
143             continue
144         else
145             A[$x,$y]="\e[43;1m  \e[0m"
146
147             A[$x,$y]="\e[44;1m  \e[0m"
148             if [ "${blue[$x,$y]}" = "1" ];then
149                 continue
150             else
151                 blue[$x,$y]=1
152                 score1=`expr $score1 + 1`
153             fi
154         fi
155     else
156         continue
157     fi

```

enter입력 시 만약 yellow가 true면 continue로 넘기고 그렇지 않다면

맵1에서의 blue와 동일하다.

느낀점 :

이번 과제에서 UI를 완벽하게 구현하지 못한 것이 제일 아쉽다 그 이외의 문제되는 것은 전혀

없었으므로 UI구현의 문제가 있었고 최대한 특수문자를 사용하지 않고 구현을 하려고 했기에

이번 과제에 만족감을 느낀다.