

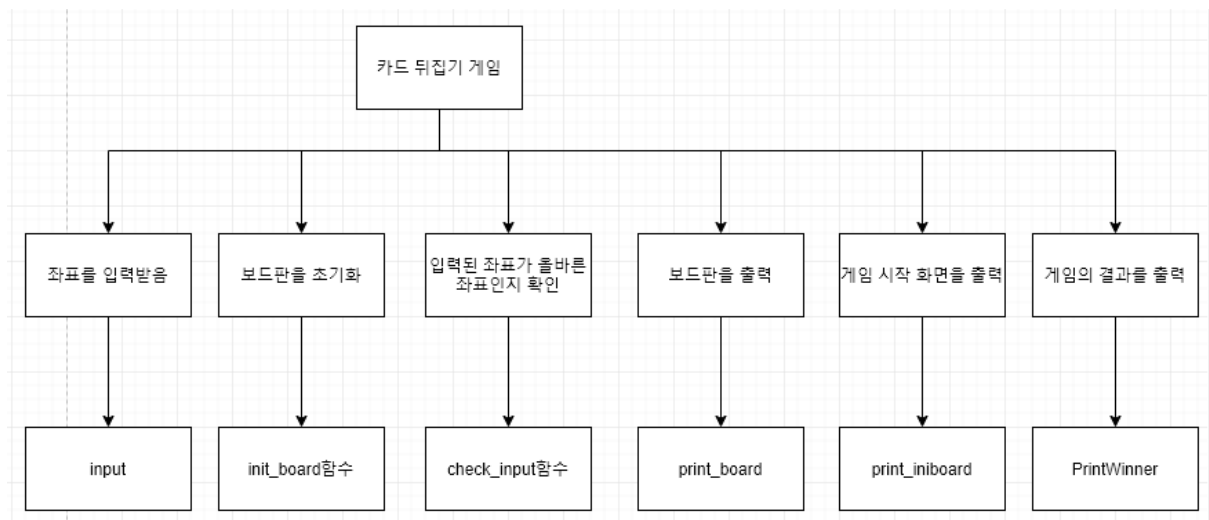
Problem2 : 카드 뒤집기 게임

1.문제의 개요

이 프로그램을 간략히 설명하자면 다음과 같다

- 프로그램을 실행하면 랜덤으로 숫자가 입력된 보드판이 출력된다. 3초 후 보드판이 지워지고 카드 뒷면만이 보이게 된다.
- 랜덤으로 순서가 정해진다. 자기 차례에 카드 2개를 선택한다. 카드 2개에 들어있는 수가 일치할 경우 점수를 한 점 얻고, 카드들을 오픈하고 기회를 한번 더 얻는다. 일치하지 않을 경우 선택된 카드에 들어있는 숫자를 3초간 보여준 후 다른 사람에게 순서가 돌아간다.
- 모든 카드가 오픈되면 게임이 종료되며 얻은 점수에 따라 승자가 정해진다.

이 내용을 구조도로 표현하자면 다음의 사진과 같다.



입력부1: 좌표를 입력받는다

처리부1: 보드판을 초기화시킨다.

처리부2: 입력받은 좌표가 올바른 좌표인지 확인한다.

출력부1: 보드판을 출력한다.

출력부2: 게임 시작 화면을 출력한다.

출력부3: 게임의 결과를 출력한다.

1. 알고리즘

본 프로그램 작성을 위한 알고리즘을 pseudo 코드 형태로 나타내면 다음과 같다.

```
init_board //보드판을 초기화하는 함수
print_board //보드판을 출력하는 함수
print_iniboard //게임 시작 시 화면( Memorize the board number! 와 보드)를 출력하는 함수
input //좌표를 입력받고, 조건을 검사하여 점수를 올리고, 플레이어의 순서를 반환하는 함수
check_input // 입력된 좌표가 올바른 값인지 확인하는 함수
reset //게임을 다시 시작할 때 보드와 보드 상태를 나타내는 배열들을 전부 0으로 초기화시키는 함수
PrintWinner //게임의 결과를 출력하는 함수
```

메인함수

1.변수와 배열 선언

2.시드 초기화

3.while이용, 재시작 문구에서 y를 입력받는 동안 보라색 문장들 반복수행

1.reset 함수 호출

2.init_board함수 호출

3.rand 함수 이용, 순서 결정

4.print_iniboard함수 호출

5.3초간 지연하기

6.화면 지우기

7.print_board함수 호출

8.while 이용, 총 획득된 점수가 8점 이하인 경우 아래 파랑 문장 반복수행

1.input함수 호출,

9.PrintWinner함수 호출

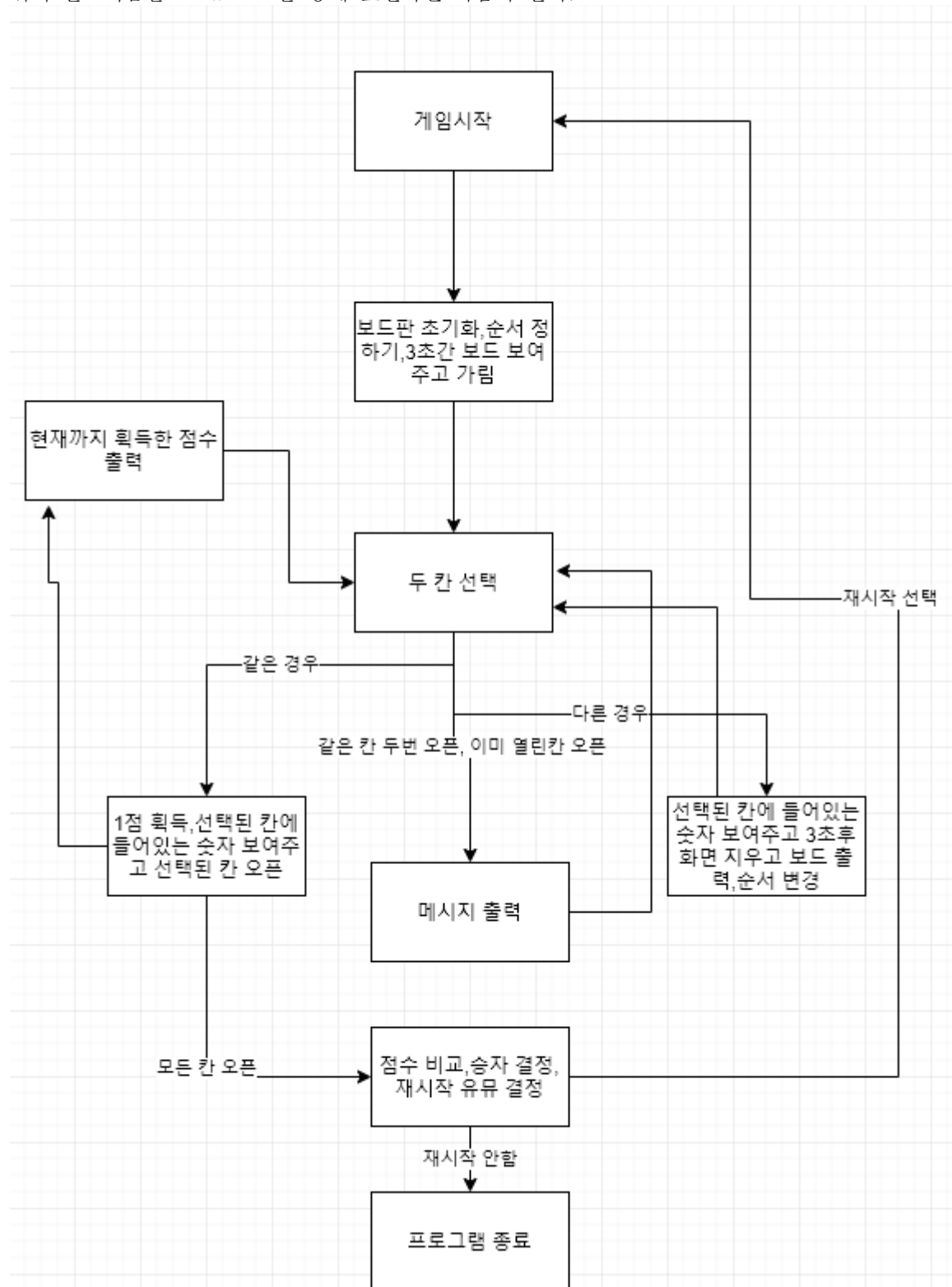
10.계속할것인지 말 것인지 입력받음

11.n을 입력받은 경우 break를 통해 while 탈출,

12.n을 입력받은 게 아니라면 플레이어들이 획득한 점수를 다시 0으로 초기화시킨다.

13.while이 끝나면 프로그램 종료

위의 알고리즘을 Flowchart을 통해 표현하면 다음과 같다.



2. 프로그램 구조 및 설명

a) 프로그램의 시작

-필요한 변수와 배열을 선언하고 srand함수를 이용해 시드를 초기화시킨다.

main함수 내의 while문에 의해 b~e 과정이 반복된다.

b) 보드 초기화와 순서결정

-reset 함수를 호출하여 우선 보드판과 보드판 상태를 나타내는 배열에 전부 0을 넣어준다. 보드판 상태 배열에 0이 있으면 현재 오픈되지 않은 상태라는 것이다. 보드판 배열에 0이 들어가 있으면 아직 초기화 되지 않은 배열이라는 뜻이다.

-init_board함수를 호출하여 보드를 초기화시켜준다. 이 과정은 다음과 같다. rand함수를 이용하여 x,y 좌표를 각각 2개씩 랜덤으로 선택한 후 그 좌표에 1을 넣어준다. 단, 선택된 두 좌표가 같거나, 선택된 좌표에 0이 아닌 다른 수가 들어있다면, 다시 좌표를 생성하도록 한다. 이 과정을 for를 이용해서 8을 넣어줄 때 까지 반복한다.

c) 게임의 시작

- print_iniboard를 호출한다: 게임 시작 시에는 보드를 모두 오픈시킨 채로 보드를 출력한다. 그러므로 우선 보드에 들어있는 수를 모두 보여준 후 sleep함수를 이용해서 3초간 지연한 후 화면을 지운다. 그 다음에는 print_board함수를 호출한다. 이 함수는 보드의 상태에 따라 오픈된 칸은 숫자를 보여주고 오픈되지 않은 칸은 *를 보여준다.

d) 좌표입력, 조건검사 (while문을 이용해서 player1과 player2가 얻은 점수의 합이 8이하인 경우 계속 d를 반복한다.)

-input함수를 호출한다: 좌표를 입력받는다. while문의 조건으로 checkinput==1 을 써넣는다. checkinput 함수는 입력받은 좌표가 겹치거나, 이미 오픈된 보드의 좌표라면 1을 반환한다. 즉 잘못된 입력을 받은 경우에 다음 문장을 반복한다: scanf를 이용하여 좌표를 입력받는다. 만약 옳은 좌표를 입력받은 경우 while이 끝나므로 다음 문장으로 넘어간다.

-입력받은 좌표의 보드를 오픈상태로 만들어 놓는다.

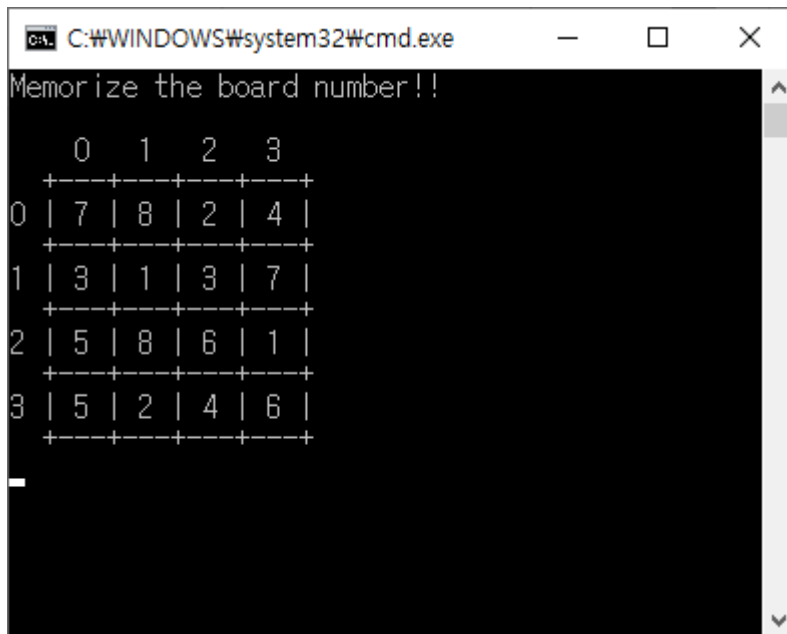
-print_board함수를 호출하여 보드 화면을 출력한다.

-입력받은 좌표에 들어있는 값이 같다면, 현재 순서를 나타내는 turn 변수(현재 순서를 나타내는 변수)에 들어있는 값에 따라 플레이어의 점수를 올리고 적절한 문구를 출력한다. 다르다면 입력받은 좌표의 보드를 오픈 상태에서 다시 닫힌 상태로 바꿔주고 turn변수의 값을 다른 값으로 바꿔준다. 그리고 turn변수에 들어있는 값을 반환시킨다.

e) 게임 결과 출력

- PrintWinner함수를 호출한다. 이 함수는 score1(player1이 얻은 점수를 저장하는 변수) 와 score2(player2가 얻은 점수를 저장하는 변수)의 값을 비교하여 누가 이겼는지, 혹은 비겼는지를 출력한다.
- 이후에 프로그램을 재실행할건지 말건지를 scanf를 통해 입력받고 n을 입력받은 경우에는 break를 통해 main 함수 내에 있는 while문을 탈출한다. Y를 입력받은 경우에는 main함수 내에 있는 while문에 의해 b에서 e까지의 과정이 반복된다.

4.프로그램 실행 방법 및 예제



```
Memorize the board number!!

  0  1  2  3
+---+---+---+---+
0 | 7 | 8 | 2 | 4 |
+---+---+---+---+
1 | 3 | 1 | 3 | 7 |
+---+---+---+---+
2 | 5 | 8 | 6 | 1 |
+---+---+---+---+
3 | 5 | 2 | 4 | 6 |
+---+---+---+---+
```

프로그램을 실행하면 랜덤으로 구성된 보드판이 출력되며 이 보드판은 3초 후에 지워진다. 3초동안 양 플레이어는 숫자들을 외운다.

```
선택 C:\WINDOWS\system32\cmd.exe

  0  1  2  3
+---+---+---+---+
0 | * | * | * | * |
+---+---+---+---+
1 | * | * | * | * |
+---+---+---+---+
2 | * | * | * | * |
+---+---+---+---+
3 | * | * | * | * |
+---+---+---+---+

[player 1]: _
```

3초후 화면이다. 랜덤으로 순서가 정해진다. 자기 차례인 사용자는 좌표값을 입력한다.

```
선택 C:\WINDOWS\system32\cmd.exe

  0  1  2  3
+---+---+---+---+
0 | 7 | * | * | * |
+---+---+---+---+
1 | * | * | * | 7 |
+---+---+---+---+
2 | * | * | * | * |
+---+---+---+---+
3 | * | * | * | * |
+---+---+---+---+

player 1 get score!
-----
[player 1] 1 : 0 [player 2]
-----

player 1 get one more chance!!
[player 1]: _
```

선택한 보드판에 들어있는 숫자가 같은 경우 1점을 획득하고 기회를 1번 더 얻게 된다.

```
C:\WINDOWS\system32\cmd.exe

0 | 7 | * | * | * |
+---+---+---+---+
1 | * | * | * | 7 |
+---+---+---+---+
2 | * | * | * | * |
+---+---+---+---+
3 | * | * | * | * |
+---+---+---+---+

player 1 get score!
-----
[player 1] 1 : 0 [player 2]
-----

player 1 get one more chance!!
[player 1]: 0 1 0 1

same card... try again..
[player 1]:
```

같은 카드를 선택한 경우에 오류가 발생하며 다시 값을 입력받는다.

```
C:\WINDOWS\system32\cmd.exe

  0  1  2  3
+---+---+---+---+
0 | 7 | * | * | * |
+---+---+---+---+
1 | * | * | * | 7 |
+---+---+---+---+
2 | * | * | * | * |
+---+---+---+---+
3 | * | * | * | * |
+---+---+---+---+

player 1 get score!
-----
[player 1] 1 : 0 [player 2]
-----

player 1 get one more chance!!
[player 1]: 0 1 0 1

same card... try again..
[player 1]: 0 0 0 3

already open... try again..
[player 1]:
```

선택한 카드 중에 이미 열린 보드판의 카드가 있는 경우에 오류가 발생하며 다시 값을 입력받는다.

```
C:\WINDOWS\system32\cmd.exe

-----
[player 1] 1 : 0 [player 2]
-----

player 1 get one more chance!!
[player 1]: 0 1 0 1

same card... try again..
[player 1]: 0 0 0 3

already open... try again..
[player 1]: 0 3 2 3

    0   1   2   3
  +---+---+---+---+
0 | 7 | * | * | 4 |
  +---+---+---+---+
1 | * | * | * | 7 |
  +---+---+---+---+
2 | * | * | * | 1 |
  +---+---+---+---+
3 | * | * | * | * |
  +---+---+---+---+

Mismatch!_
```

입력받은 보드판에 들어있는 숫자가 다른 경우에는 3초간 입력받은 칸에 들어있는 숫자들을 보여준 후 화면을 지운다.

```
C:\WINDOWS\system32\cmd.exe

    0   1   2   3
  +---+---+---+---+
0 | 7 | * | * | * |
  +---+---+---+---+
1 | * | * | * | 7 |
  +---+---+---+---+
2 | * | * | * | * |
  +---+---+---+---+
3 | * | * | * | * |
  +---+---+---+---+

[player 2]: _
```

그 후에 이렇게 순서가 바뀌게 된다.


```
C:\WINDOWS\system32\cmd.exe
[player 1] 7 : 0 [player 2]
-----
player 1 get one more chance!!
[player 1]: 2 2 3 3

  0  1  2  3
+---+---+---+---+
0 | 7 | 8 | 2 | 4 |
+---+---+---+---+
1 | 3 | 1 | 3 | 7 |
+---+---+---+---+
2 | 5 | 8 | 6 | 1 |
+---+---+---+---+
3 | 5 | 2 | 4 | 6 |
+---+---+---+---+

player 1 get score!
-----
[player 1] 8 : 0 [player 2]
-----

Game End! Player 1 wins!
Continue? (y/n) _
```

두 플레이어가 얻은 점수의 합이 8이 되면 게임이 종료되고 게임의 결과(누가 승리하였는지, 비겼는지)를 출력한다.

```
C:\WINDOWS\system32\cmd.exe
+---+---+---+---+

player 1 get score!
-----
[player 1] 8 : 0 [player 2]
-----

Game End! Player 1 wins!
Continue? (y/n) y

New Game!
Memorize the board number!!

  0  1  2  3
+---+---+---+---+
0 | 8 | 4 | 3 | 7 |
+---+---+---+---+
1 | 3 | 1 | 5 | 8 |
+---+---+---+---+
2 | 2 | 1 | 7 | 2 |
+---+---+---+---+
3 | 5 | 6 | 6 | 4 |
+---+---+---+---+
```

게임이 종료되고 y를 입력하면 게임이 새로 시작된다.

```
C:\WINDOWS\system32\cmd.exe

-----
player 2 get one more chance!!
[player 2]: 3 1 3 2

  0  1  2  3
+---+---+---+---+
0 | 8 | 4 | 3 | 7 |
+---+---+---+---+
1 | 3 | 1 | 5 | 8 |
+---+---+---+---+
2 | 2 | 1 | 7 | 2 |
+---+---+---+---+
3 | 5 | 6 | 6 | 4 |
+---+---+---+---+

player 2 get score!
-----
[player 1] 0 : 8 [player 2]
-----

Game End! Player 2 wins!
Continue? (y/n) n
Good bye계속하려면 아무 키나 누르십시오 . . .
```

게임이 종료되었을 때 n을 입력하면 프로그램이 종료된다.

5.토론

-맨 처음 보드판을 초기화하는방법이 엄청 까다로웠다. 랜덤으로 생성한 수를 (0,0)부터 (3,3)까지 차례대로 넣으려니, 중복되는 수가 있을 수도 있고 1~8까지의 숫자가 모두 2개씩 들어가지 않고 빠지는 수도 있을 수 있기 때문이었다.

-그래서 생각한 방법이 보드에 들어갈 수를 랜덤으로 정하는 것이 아니라, 반대로 수가 들어갈 보드칸 좌표를 랜덤으로 정하는 것이었다. 이 때 생성된 보드칸 좌표 2개가 같으면, 다시 좌표를 생성하도록 하고, 생성된 보드칸 좌표에 이미 수가 들어있으면 다시 좌표를 생성하도록 한다. 그리고 생성된 보드칸 좌표에 1부터 8까지 하나씩 넣어준다.

6.결론

assn2_2를 만들어 보는 과정에서 2차원 배열에 대해 익숙해 질 수 있었다. 그리고 배열을 사용자 정의 함수를 이용할 때 다루는 방법에 대해서도 익숙해 질 수 있었다.

7.개선 방향

- 각 플레이어에게 1번씩 힌트를 얻을 수 있는 기회를 주는 기능을 추가하면 좋을 것 같다.
- 보드판을 초기화할 때 (0,0)부터(3,3)까지 차례대로 숫자를 써넣은 후, 보드판을 섞어주는 방식으로 해도 좋을 것 같다.