프로그래밍 과제 2 – 시계/빙고

20222663 권기영

1. 현재 시각 계산하기

<핵심 아이디어>

1분은 60초, 1시간은 60분, 1일은 24시간, 1년은 (365일 or 366일) 임을 이용해서 초를 변환한다. 윤년의 경우에 대한 예외처리를 하는 것이 중요하다.

<구현>

1) 년 계산

윤년인지를 판단한 뒤, 윤년이면 – 366 * 24 * 60 * 60, 윤년이 아니라면 -365 * 24 * 60 * 60을 하고 년도를 +1 한다. 이 과정을 전체 초가 음수가 되기 전 까지 반복한다.

2) 달 계산

12달 각각에 대한 일 수를 담은 month 배열이 있는데, 이를 활용한다.

윤년인지를 판단한 뒤, 윤년이면 2월의 일수를 29로 바꾼다.

이후 전체 초가 음수가 되기 전 까지 –일 수 * 24 * 60 * 60을 반복한다.

3) 일, 시간, 분, 초 계산

일, 시간, 분, 초는 윤년이란 조건에 따라서 달라지지 않기 때문에 몫과 나머지의 원리에 따라서 구해주었다.

예를 들어 일 = (년과 달을 계산하고 남은 초) / (1일의 초) + 1 로 계산하고, 일을 계산하고 남은 나머지는 시간, 분, 초를 계산하기 위한 초로 사용한다.

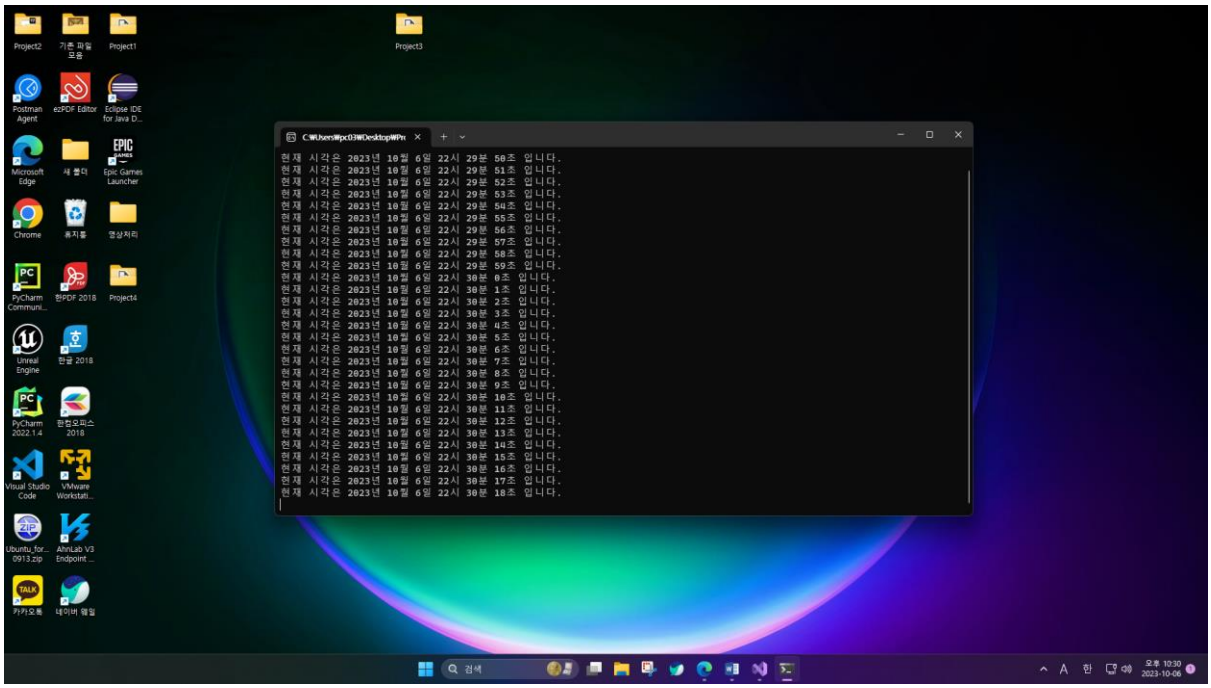
4) print_eval_time() 함수

현재 시각을 받고, 우리나라 시간으로 변환한 뒤, 년, 달, 일, 시간, 분, 초를 계산해서 출력해주는 함수이다.

5) main

Sleep(1000)을 활용해서 1초를 지연시켜주었고, 1초마다 한번씩 print_eval_time 함수를 실행시켜 주었다.

<실행 결과>



1초에 한 번씩 현재 시각이 잘 출력됨을 확인할 수 있다.

2. 빙고 프로그램 만들기

<핵심 아이디어>

프로그램을 똑똑하게 만들기 위한 규칙을 만들 때의 핵심 아이디어는 바로 greedy 하게 생각하기 이다.

그 이유는 1) 사람이 빙고를 할 때도 항상 최적의 수를 구하는 것이 아닌 greedy하게 선택하는 경향이 있기 때문이고, 2) 12줄 중 5줄만 빙고를 해도 되기 때문이다.

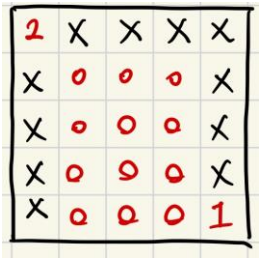
12줄에 가까워 질수록 미래를 생각해서 전체 빙고판에 대해 이득이 되는 수를 선택하는 것이 중요한데, 12줄에서 멀어질수록 미래보다는 눈 앞의 빙고 1줄에 대한 수를 선택하는 것이 중요하다.

<규칙>

총 3가지의 규칙이 있고, 규칙에는 우선순위가 존재한다. 즉, 앞의 규칙에서 판단이 덜 되었으면

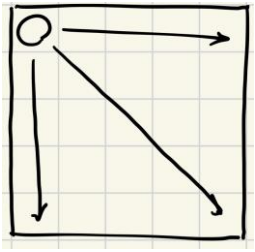
(동일한 가중치의 칸이 여러 개가 나온다면) 다음 규칙으로 넘어가는 식이다.

RULE 1 : 선택했을 때 만들 수 있는 빙고 줄의 개수가 가장 많은 칸을 선택한다.

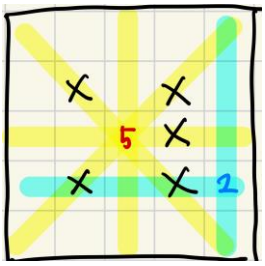


예를 들어 이 경우에, (1,1) 위치의 칸은 2줄의 빙고를 만들 수 있고, (5,5) 위치의 칸은 1줄의 빙고를 만들 수 있고, 나머지는 0줄의 빙고를 만들 수 있으므로 다음 수로 (1,1) 위치의 칸을 선택한다.

RULE 2: 동일한 라인에 있는 칸들 중, 이미 채워진 칸의 개수가 가장 많은 칸을 선택한다.



동일한 라인이라 함은, 위 그림처럼 서로서로 빙고에 영향을 주는 라인이다. (1,1)은 같은 행, 같은 열, 같은 대각선의 칸 들과 동일한 라인이다.



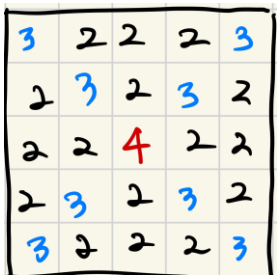
이런 상황에 대해서, RULE 1을 적용했을 때 모든 칸이 전부 0개의 빙고를 완성할 수 있으므로, RULE 2로 넘어온다.

RULE 2에서는 같은 라인의 칸들 중 이미 채워진 칸의 개수가 가장 많은 칸을 고르게 되는데, 위 그림과 같이 (3,3)은 가로, 세로, 대각선 총 5개의 채워진 칸이 있고, (4,5)는 가로 세로 총 2개의 채워진 칸이 있다. 이 경우에서 (3,3)만이 5개로 최대이므로, (3,3)을 다음 수로 선택한다.

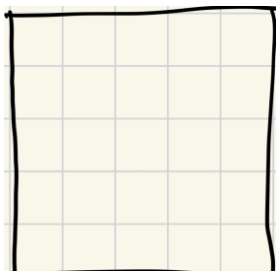
RULE 3: 같은 라인의 개수가 가장 많은 칸을 선택한다.

예를 들어, (1,2)의 경우 같은 라인이 가로, 세로로, 같은 라인의 개수는 2개이다. (3,3)의 경우, 중간이므로 가로, 세로, 대각선1, 대각선2가 같은 라인으로, 총 4개이다.

따라서 각 칸에 대한 같은 라인의 개수는 다음과 같다.



RULE 1, 2를 모두 적용했는데도 동일한 가중치의 칸이 여러 개가 나오게 된다면, 그 칸들 중 같은 라인의 개수가 가장 많은 칸을 선택하면 된다.



예를 들어 이러한 경우(시작 직후의 상태), RULE 1을 적용했을 때 가능한 빙고의 개수는 모두 0개이므로 RULE 2로 넘어간다. RULE 2를 적용했을 때 같은 라인에 채워진 칸의 개수가 모두 0개이므로 RULE 3으로 넘어간다. 즉 RULE 2까지 적용했는데, 모든 칸의 가중치가 동일한 것이다.

그럼 RULE 3을 적용하면, 중간(3,3)위치의 칸이 같은 라인의 개수가 4로 가장 많으므로 (3,3) 위치를 첫 번째 수로 선택하게 된다.

RULE 3에 의해 첫 번째 수를 중간으로 고정하지 않더라도, 규칙에 의해 자동적으로 중간을 고르게 된다!

<구현>

바보 컴퓨터와 똑똑한 컴퓨터 2가지의 버전이 있고, 프로그램 시작 시에 2가지 버전 중 하나를 선택할 수 있다.바보 컴퓨터는 남은 칸들 중 랜덤으로 칸을 선택하고, 똑똑한 컴퓨터는 위 3가지의 규칙을 적용시켜 가장 최적의 수를 선택한다.

<실행 결과>

```
대결할 컴퓨터를 선택하세요(바보 : 1 똑똑이 : 2) >> 1


      <my board>                    <his_board>
 24   10   13   21    8    |    -    -    -    -    -
  1    9   12   15   14    |    -    -    -    -    -
 17   11   19   16   20    |    -    -    -    -    -
 22    5   18    3    7    |    -    -    -    -    -
  6    4    2   23   25    |    -    -    -    -    -
 빙고의 수 : 0                빙고의 수 : 0

#1. 당신의 차례입니다. 몇 번을 선택할까요? >>
```

초기 빙고판이다. 바보 컴퓨터를 선택했고, 빙고판의 모습이 주어진다.

```
#19. 당신의 차례입니다. 몇 번을 선택할까요? >> 9
       <my board>                    <his_board>
(23) (22)  19   11  (17)   |  (24)  (9) (22)  (1)  (3)
(14)   8    7   (6)  (3)   |  (16)   -    -  (13) (15)
 (4) (21) (24) (18) (16)   |   (4)   -   (6) (21)   -
(13)  (5)  (1)  (9) (12)   |  (17)   -  (18) (25) (23)
(10) (15)  20    2  (25)   |  (12) (10)  (5) (14)   -
 빙고의 수 : 5                 빙고의 수 : 3

당신이 이겼습니다!!!!!!!!!!
```

나와 컴퓨터 중 하나가 빙고 5줄을 맞추면 게임이 종료된다. 바보 컴퓨터는 랜덤으로 선택하기 때문에 쉽게 이겼다.

```
#18. 컴퓨터의 차례입니다. 21번을 선택했습니다.
       <my board>                    <his_board>
 (1) (14) (17)  (7) (24)   |  (10)   -  (25)   -   (9)
 11    8  (13)  (9) (21)   |  (14)  (1) (21)  (4)  (6)
(25)  (5)  (2)  (6) (10)   |    -    -   (7) (24)   -
 23   19   (4) (20) (22)   |   (2) (22) (17)  (5)  (3)
 18   15   (3) (12)  16    |  (20)   -  (12)   -  (13)
 빙고의 수 : 4                 빙고의 수 : 5

컴퓨터가 이겼습니다!!!!!!!!!!
```

위 그림은 똑똑이 컴퓨터랑 한 결과인데, 나보다 잘하는 것 같다.