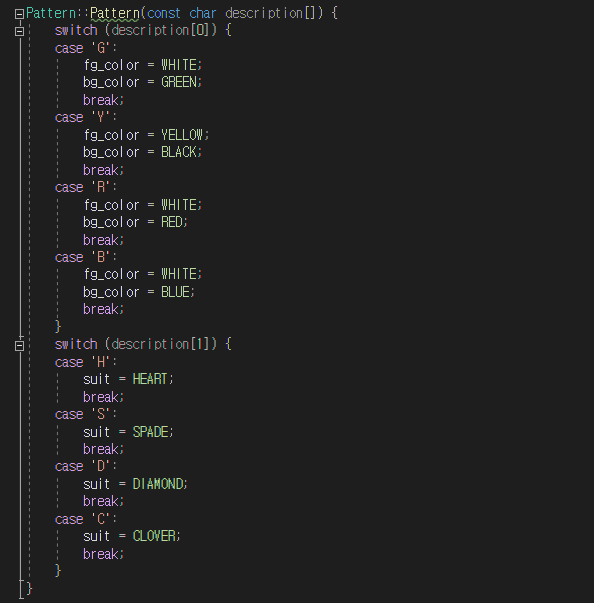
**프로그래밍 방법론 Project 2 – Tile Game Report**

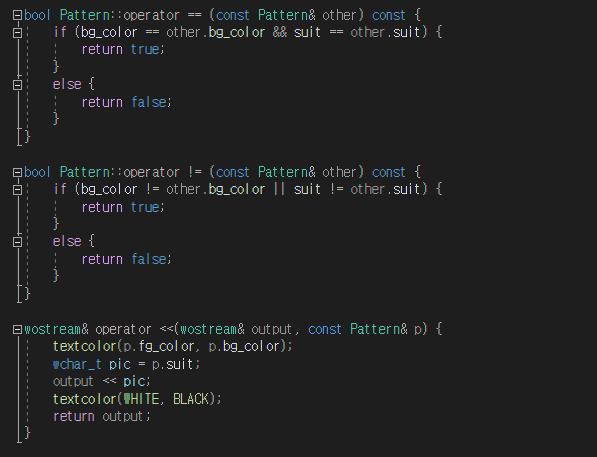
2018-12554 장승현

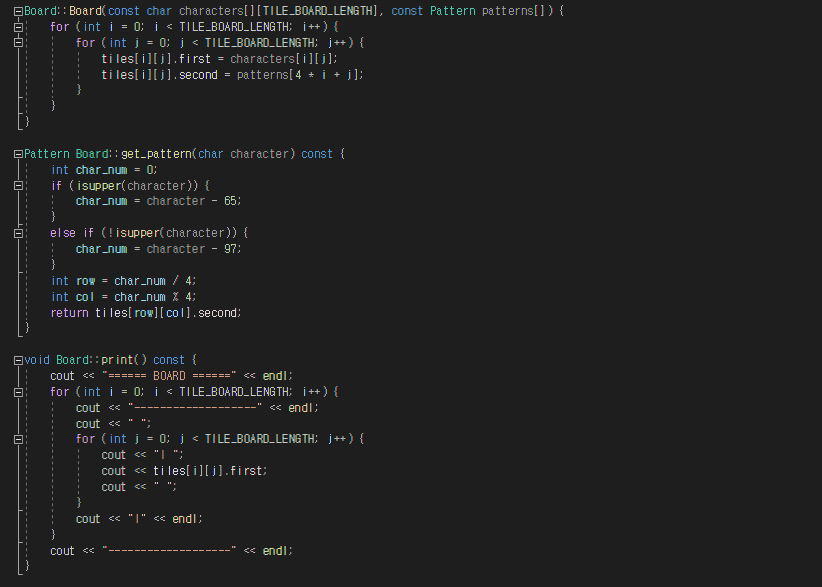
1. Class Pattern

Pattern의 Default 생성자는 배경색과 문양 색을 BLACK으로 지정하고 문양을 SPADE로 지정하였다.

Pattern의 생성자가 description을 받았을 때는 switch-case 구문을 이용하여 description의 index 별로 구분하여 배경색과 문양색, 문양을 지정하였다.

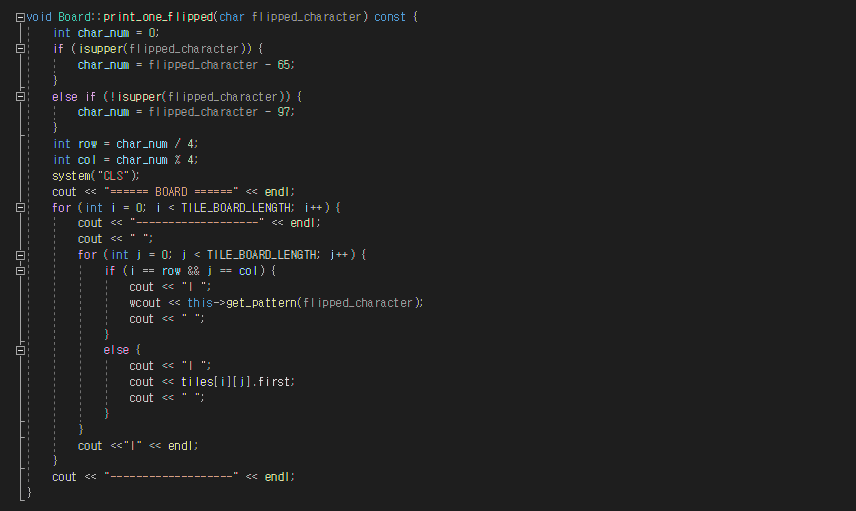
Pattern 클래스가 서로 같고 다름을 판단하기 위해서 operator ‘==’, ‘!=’를 overloading 하였다. Pattern은 배경색과 문양으로 구분할 수 있으므로 배경색과 문양이 같으면 Pattern이 같은 것으로 배경색이 다르거나 문양이 다르면 Pattern이 다른 것으로 구성하였다.

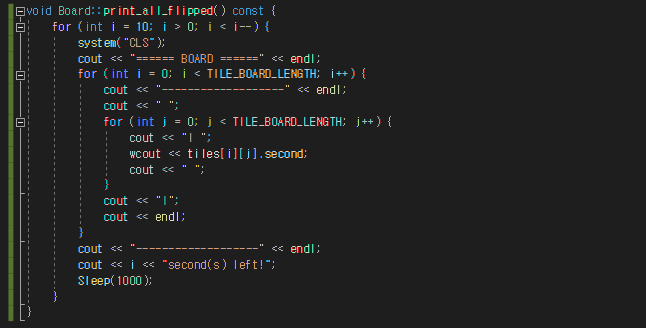
Pattern을 출력하기 위해서 ‘<<’ operator를 overloading 하였다. 배경색과 문양색을 textcolor 함수를 이용하여 설정하였고 wchar\_t 타입의 pic에 문양을 할당하여 출력하고 다시 textcolor를 기본설정으로 변경한 후 wostream을 리턴하였다.

1. Class Board

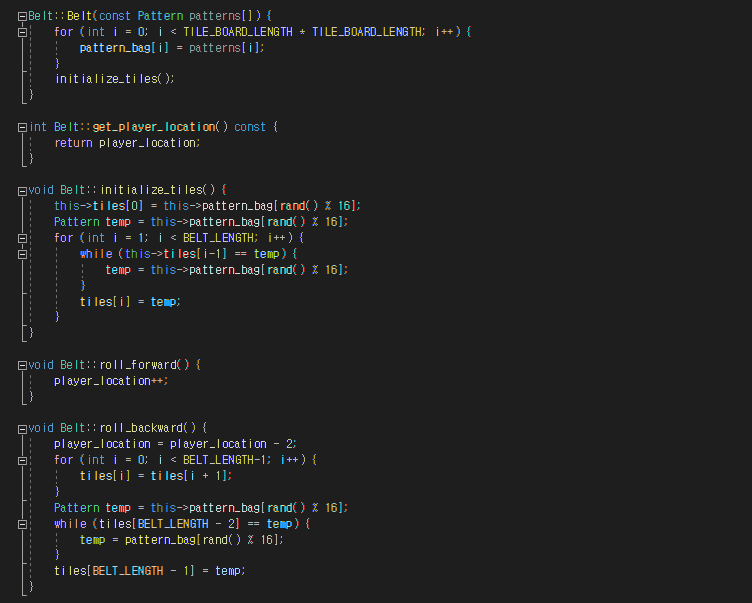
Board 클래스의 생성자에서는 문자열 characters와 Pattern배열 patterns를 받아서 Board의 Pair 타입의 tiles배열을 초기화하였다.

또한 Board에서 character를 받아 해당 Pattern을 리턴하는 get\_pattern 함수를 정의하였다. isupper() 함수를 이용하여 대문자와 소문자를 구분하여 character가 몇 번째인지 구분하기 위해 A부터 P까지 0부터 15까지 char\_num에 대응시켜 4로 나눈 몫과 나머지를 이용하여 row와 col을 계산하고 이를 이용하여 해당 위치의 pattern을 리턴한다.

print() 함수에서는 Board의 기본상태를 출력하는 함수로 Board의 tiles에서 first에 해당하는 char부분이 Board에 출력한다.

Print\_one\_flipped 함수는 flipped\_character를 받아 이에 해당하는 tile의 pattern만을 Board에서 보여준 채로 다른 tile들은 알파벳을 보여주는 함수이다. 이는 player가 알파벳을 입력했을 때 해당 tile만을 뒤집어 출력하도록 구성하였다.

print\_all\_flipped 함수는 Board에 모든 pattern을 출력하는 함수로 chance1을 player가 사용할 때 출력되는 함수이다. 따라서 pattern에 해당하는 모든 tiles의 second를 출력한다. Chance 1을 사용하면 10초부터 1초까지 몇 초가 남았는지 출력해주어야 하므로 for문과 Sleep함수를 이용하여 이를 구현하였다.

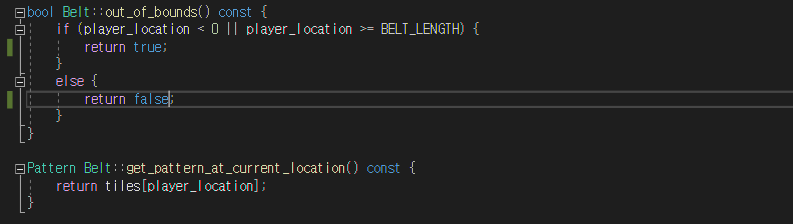
1. Class Belt

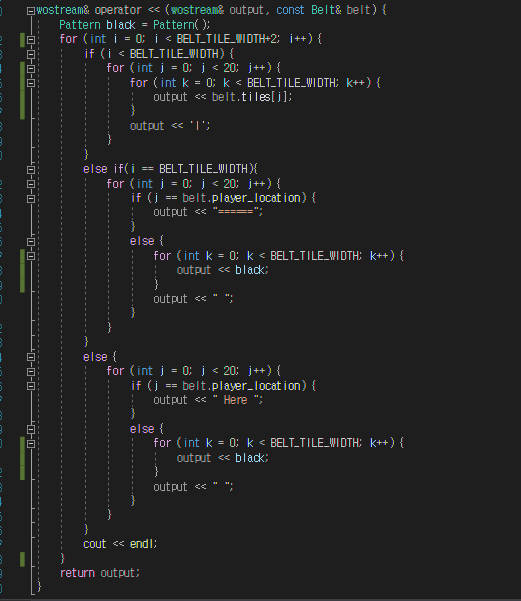
Belt의 생성자는 Pattern배열 patterns를 받아 이를 Pattern의 가능한 모든 경우인 pattern\_bag을 구성하고 initialize\_tiles 함수를 이용하여 Belt의 구성을 초기화한다.

get\_player\_location 함수에서는 player\_location을 리턴하여 멤버변수 player\_location의 get 함수 역할을 한다. 이는 새로 정의한 기능이다.

Initialize\_tiles 함수에서는 Belt의 Pattern배열 멤버변수 tiles를 초기화하는데 이웃한 tile이 중복되면 안되고 랜덤하게 구성되어야 하므로 rand 함수를 이용하고 인덱스가 하나 작은 tile의 pattern과 겹치지 않도록 temp변수에 pattern을 할당하고 while문을 이용하여 다를 때까지 temp를 재할당하고 다르면 이를 tiles[i]에 할당하는 작업을 for문을 이용하여 구성하였다.

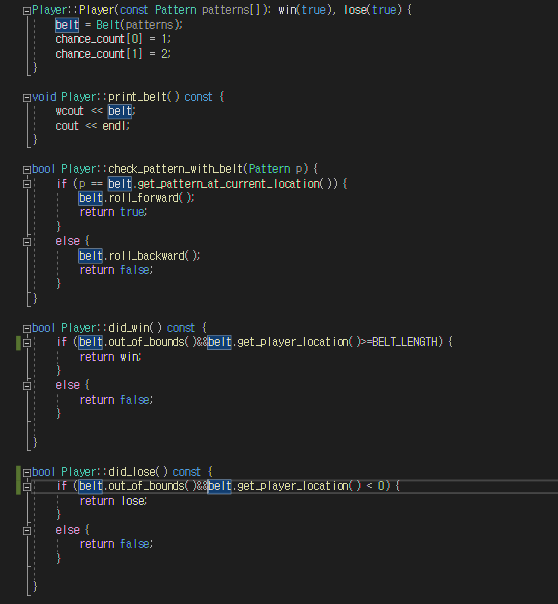
roll\_forward 함수는 player가 알파벳을 이용하여 Board에서 지목한 char의 pattern과 현재 tile의 pattern이 같을 때 호출될 함수로 player\_location을 1증가시키는 역할을 한다.

roll\_backward 함수는 함수는 player가 알파벳을 이용하여 Board에서 지목한 char의 pattern과 현재 tile의 pattern이 다를 때 호출될 함수로 player\_location을 2감소시키고 for문을 이용하여 belt의 tile들을 한 칸씩 왼쪽으로 이동시킨 후, 맨 오른쪽에는 그 왼쪽의 tile의 pattern과 겹치지 않는 pattern의 tile이 추가되도록 initialize\_tiles함수와 유사한 방식으로 tile을 추가한다.

out\_of\_bounds 함수는 if-else문을 이용하여 player\_location이 Belt의 내부에 있는지 외부에 있는지 0과 BELT\_LENGTH와 대소비교를 통해 판단하고 내부에 있으면 true를, 외부에 있으면 false를 리턴한다.

get\_pattern\_at\_current\_location 함수는 player\_location의 pattern을 리턴하도록 구성하였다.

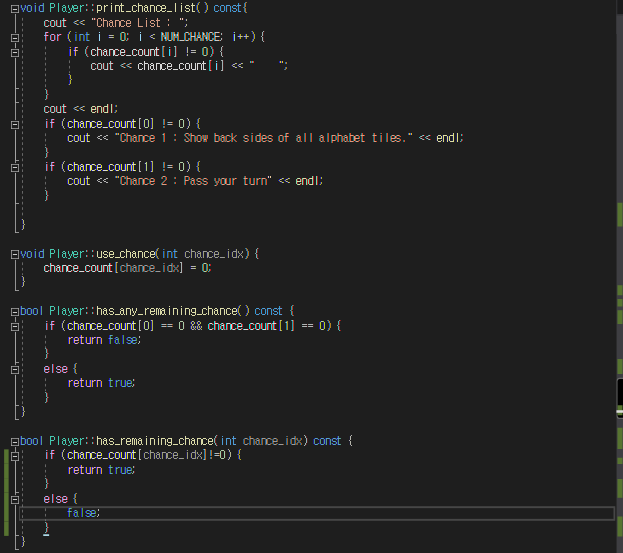
Belt class의 operator ‘<<’는 Belt type을 적절한 형태로 출력할 수 있도록 다시 overloading 되었다. for문은 BELT\_TILE\_WIDTH+2보다 i가 0부터 시작하여 작을동안 반복되는데 2를 더해준 이유는 “======”와 “ HERE “를 출력하여 player\_location을 표현하기 위함이다. tile 부분을 출력할 때 i는 줄수를 의미하고 j는 tiles의 인덱스를 나타내어 각 tile을 k는 각 tile별로 가로 tile 출력 칸 수를 의미한다.

1. Player Class 

Player 클래스의 생성자에서는 Pattern배열 patterns를 받아 멤버변수인 belt를 생성하여 player의 belt를 만든다. 또한 각 player의 chance 사용여부를 확인할 수 있는 chance\_cont 배열에서 각 chance를 0인 상태를 사용한 상태로 보고 각각 1과 2로 초기화한다. 또한 win과 lose를 true로 초기화한다.

print\_belt는 player의 belt를 출력할 때 호출된다. 이 때의 operator “<<”는 Belt 클래스에서 overloading 하였다.

check\_parttern\_with\_belt는 Pattern p를 받아 p와 belt의 get\_pattern\_at\_current\_location에서 리턴된 pattern을 비교하여 같으면 roll\_forward를 호출하여 player\_location을 하나 오른쪽으로 옮기고 true를 리턴한다. 만약 다르면 roll\_backward를 호출하여 belt를 한 칸씩 왼칸으로 옮기고 player\_location은 두 칸 왼쪽으로 옮긴다. 또한 false를 리턴한다.

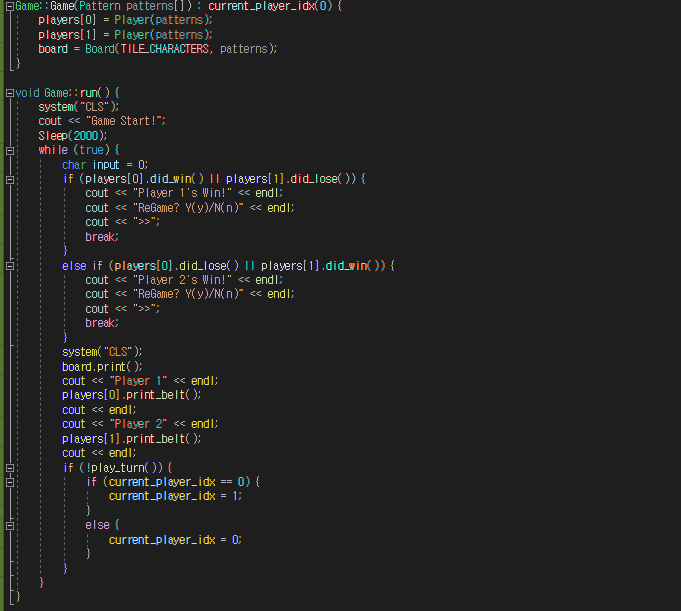
did\_win과 did\_lose 함수는 out\_of\_bounds와 get\_player\_location을 이용하여 belt위에 player\_location이 있는지 판단하고 없을 경우 player\_location이 0보다 작으면 true로 초기화된 lose를 BELT\_LENGTH보다 크면 true로 초기화된 win을 호출하여 승패 상황을 리턴한다.

print\_chance\_list는 현재 사용가능한 chance를 호출하는 함수로 for문과 if문을 이용하여 chance\_count가 0이 아닌 chance를 모두 출력하고 그에 대한 내용을 출력하게 구성하였다.

use\_chance는 chance\_idx를 받아 해당 chance\_count를 0으로 바뀌게 구성하였다.

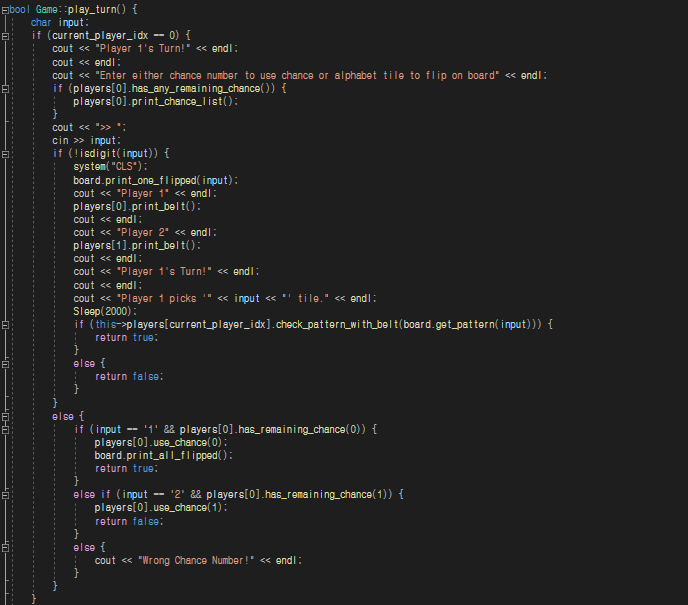
has\_any\_remaining\_chance는 chance\_count가 모두 0이면 false를 그렇지 않으면 1을 리턴하게 구성하였다. 이는 chance list를 출력할 때 사용가능한 chance가 있는지 없는지 구분하여 출력할 수 있게 해준다.

Has\_remaining\_chance는 chance\_idx를 받아 해당하는 chance가 남아있는지 판단하고 남아있으면 true를 리턴하고 그렇지 않으면 0을 리턴하게 구현하였다. 이 함수는 chance의 중복 사용을 방지한다.

1. Class Game

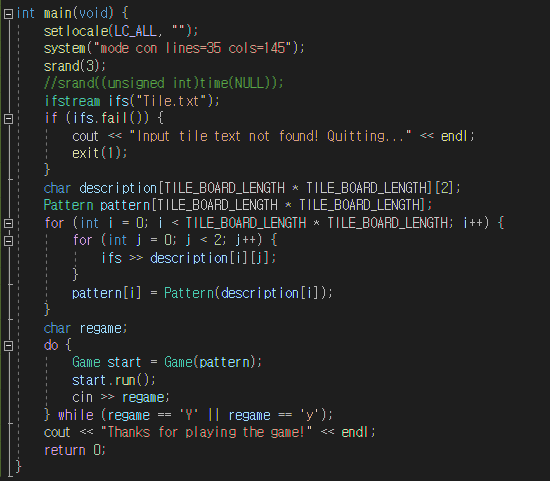
Game 클래스의 생성자는 Pattern배열 patterns를 받아 Player 클래스를 생성하여 players 배열의 각각에 할당하고 Board클래스를 생성하여 board에 할당한다. 그리고 current\_player\_idx를 0으로 초기화하여 player1부터 진행하도록 한다.

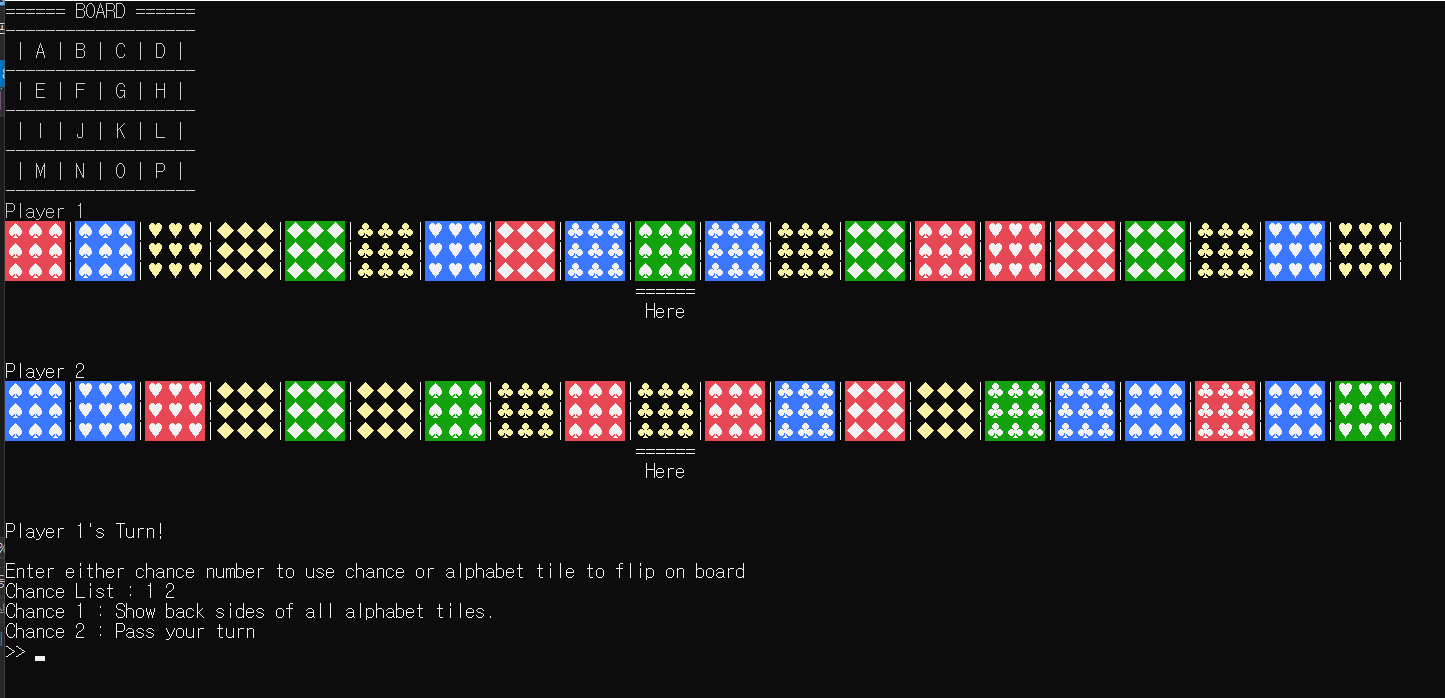
Run 함수는 호출되면 콘솔을 clear하고 “Game Start!”를 출력하고 2초를 기다리도록 한다. 이후 무한루프 while 구문을 사용하여 player중 한명이 이기거나 질 때까지 루프가 돌도록 한다. 루프의 한바퀴는 하나의 턴을 의미하므로 board와 player의 belt를 출력하고 한 번의 turn을 제어하는 play\_turn을 호출하고 여기서 리턴된 값을 이용하여 play하는 player를 전환 여부를 결정한다.

play\_turn은 한 턴의 진행을 제어하는 함수이다. 위의 사진은 player1의 턴을 제어하는 내용이다. Player 2의 턴을 제어하는 부분도 유사한 구조로 구성되었다.

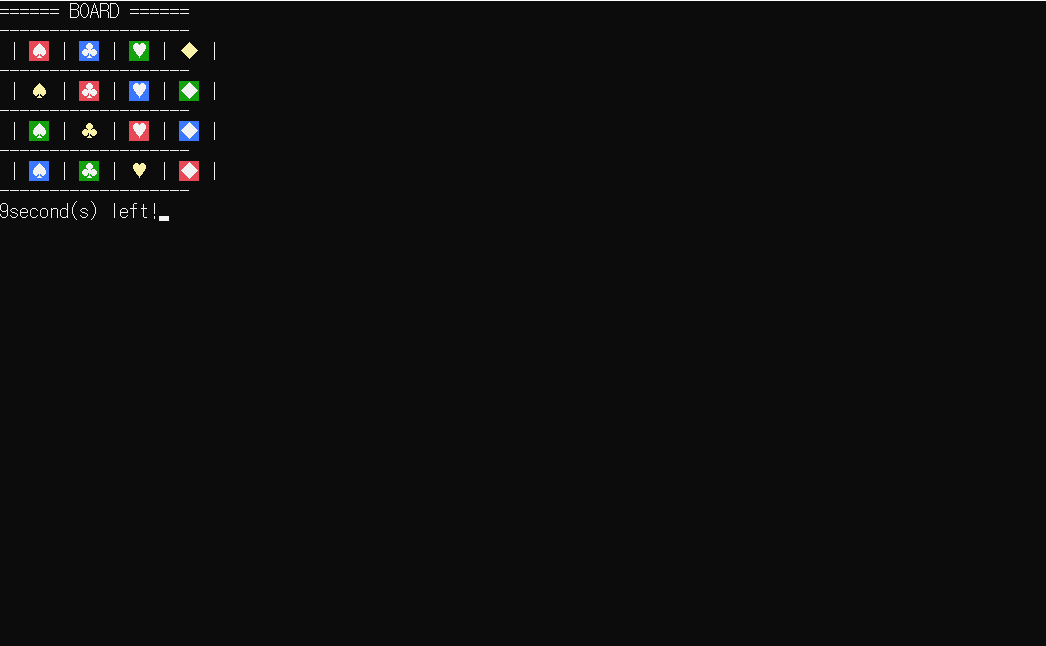
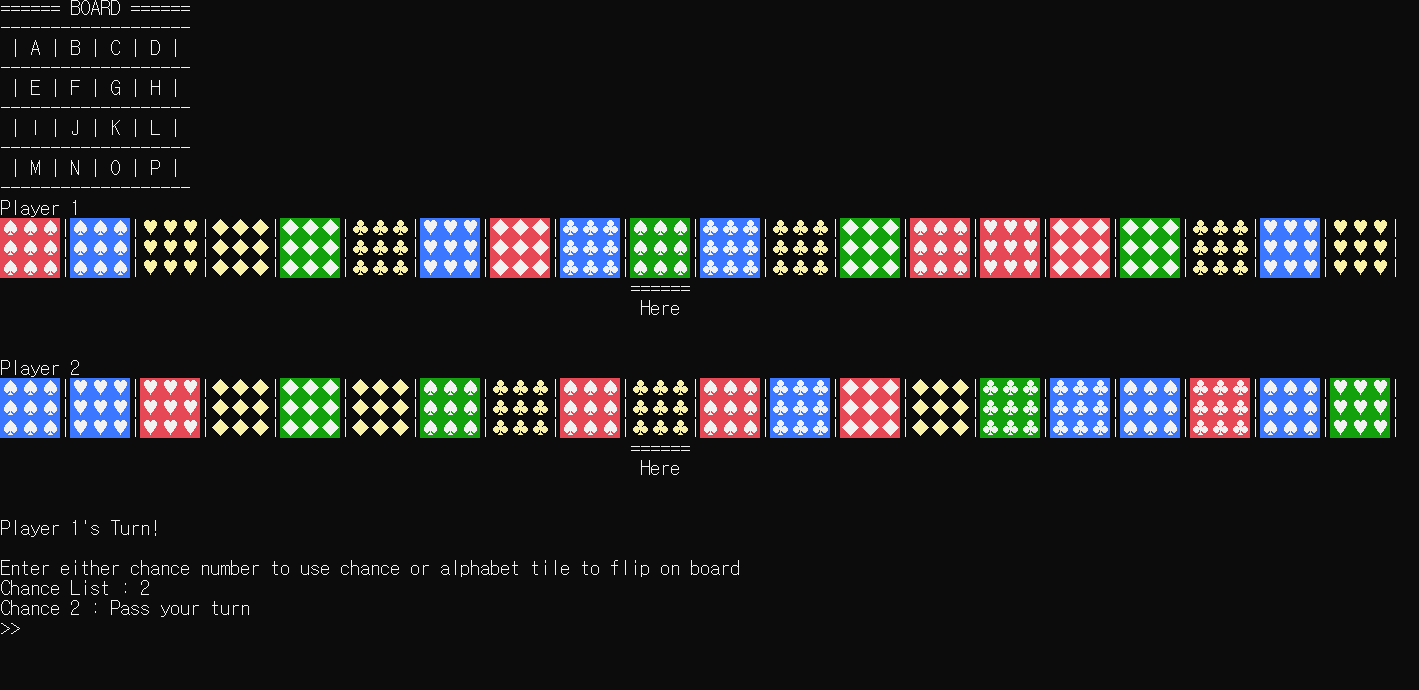
1. main 함수

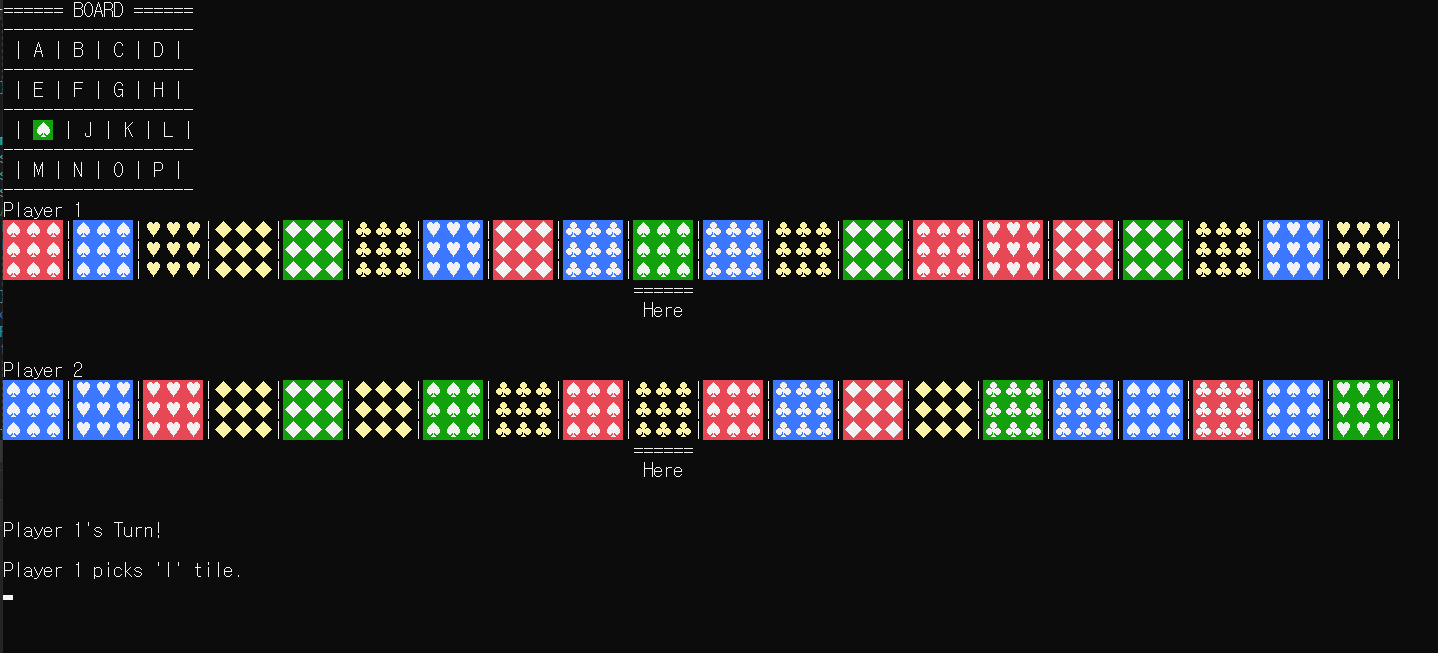
main 함수는 description 2차원 배열을 이용하여 Tile.txt의 내용을 받아 Pattern으로 변환하여 Pattern배열 pattern에 저장한다. 이후 do-while 구문을 이용하여 한 번 실행하여 한번의 게임을 진행하고 regame에 Y나 N을 받아 한번 더 게임을 진행할지를 판단하여 진행하도록 한다.

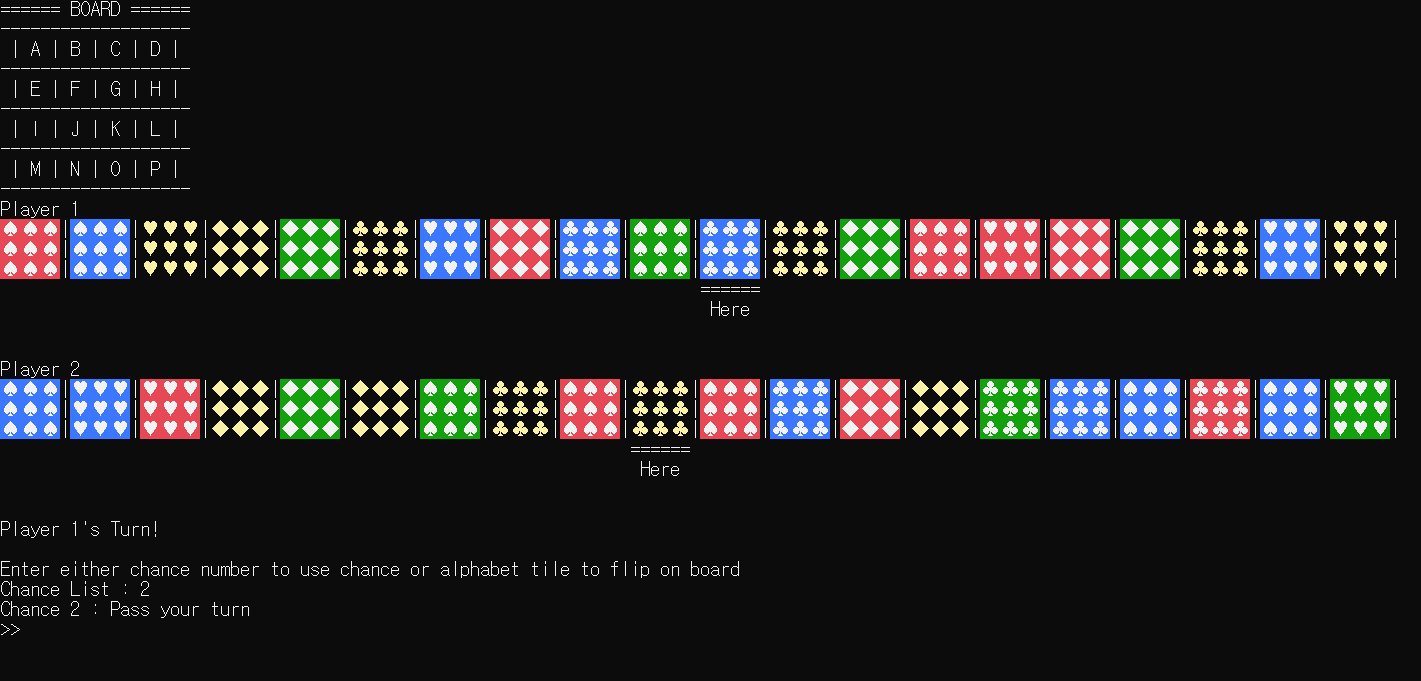
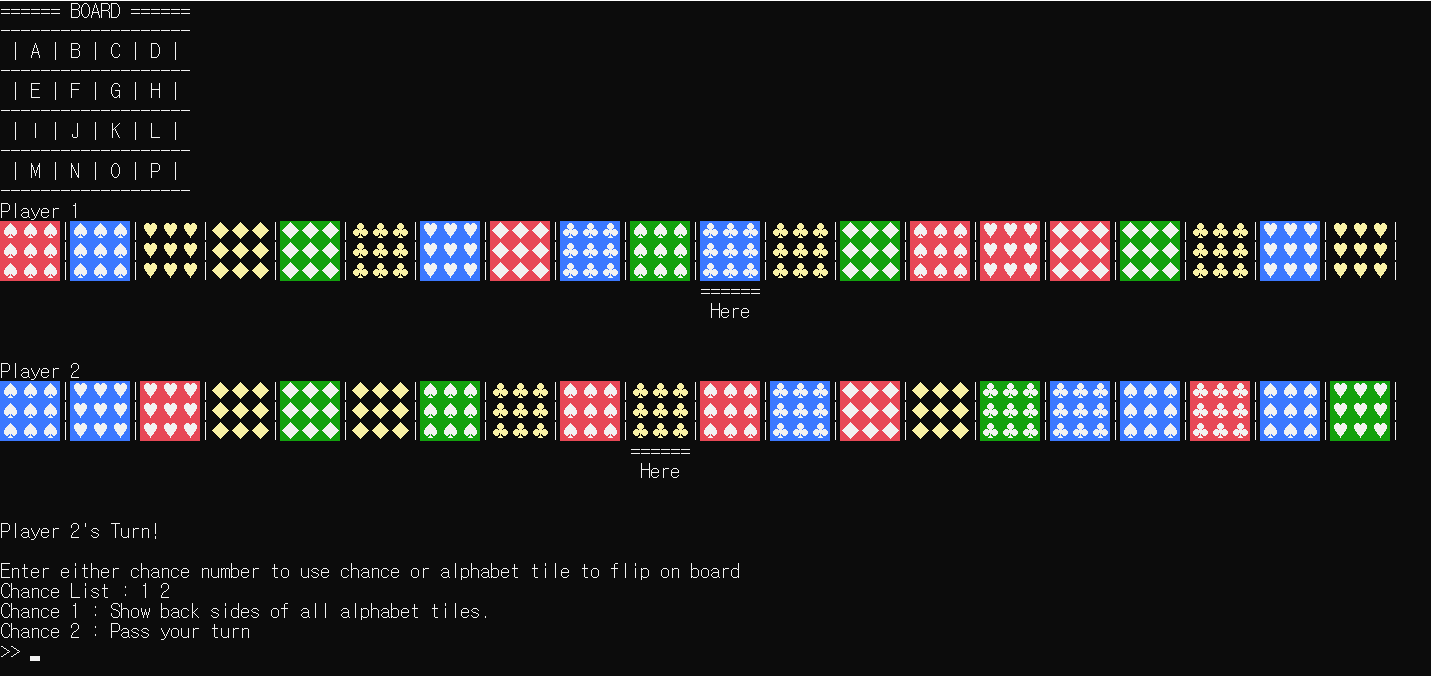
1. Result

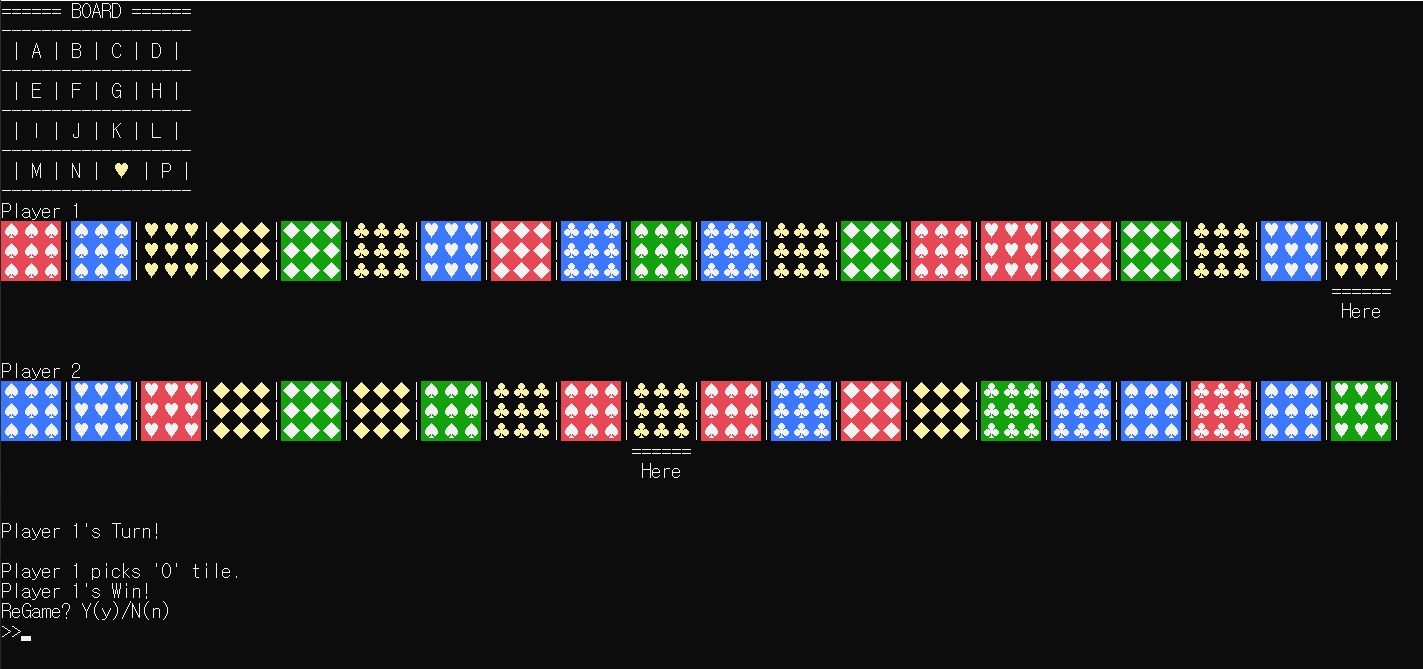
7-1 초기화면

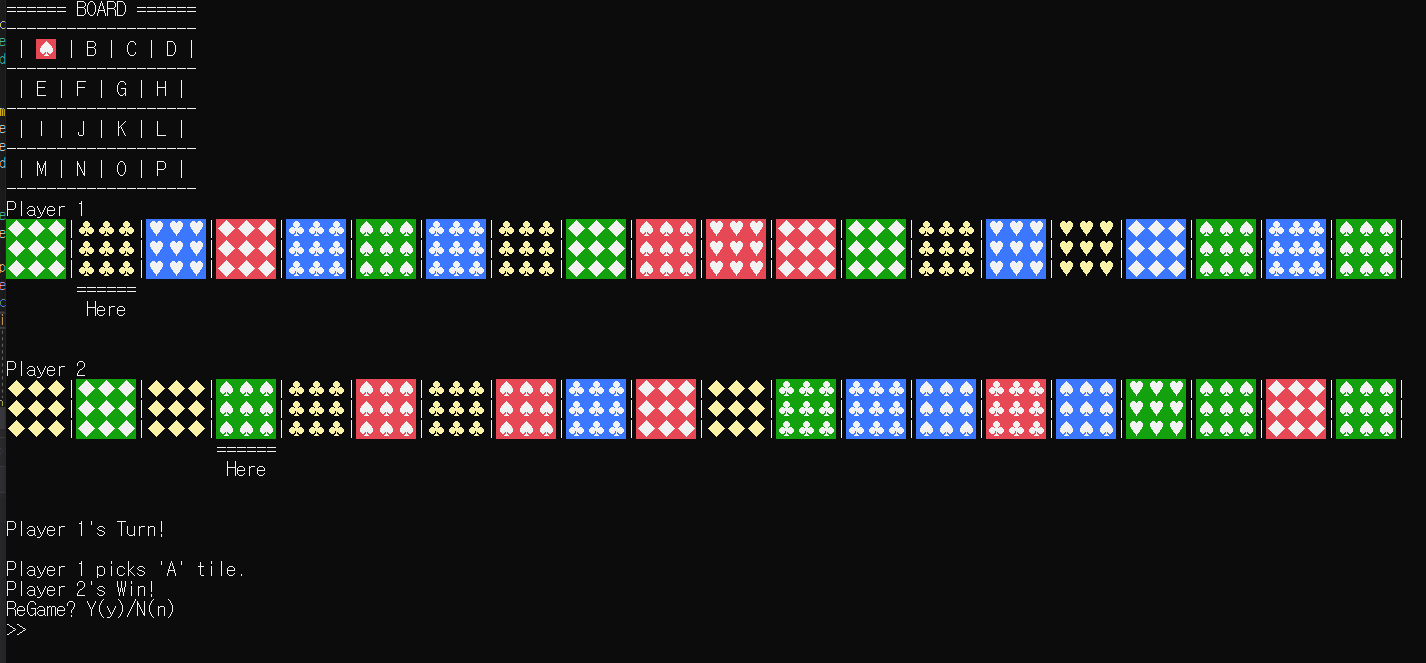
7-2 찬스 1 사용

7-3 찬스 1 사용 후

7-4 알파벳 입력 일치 경우

7-5 player1 turn에서 찬스 2 사용

7-6 승리 상황

7-7 player 1 패배 상황