* 메인

- 수도코드 짜기 필요

- 1> 인트로 출력

< ------- loop ------- >

- 2> 체스판 출력

- 3> 체크상태와 체크메이트 상태를 구현하여, 체크메이트일 시 게임 종료

- 4> 사용자 1 에게 입력을 받음

~ 어떤 말을 움직일 것인가?

~ 어느 방향으로 움직일 것인가?

- 5> 사용자 1 의 말이 움직일 목적지에 따라 처리

~ 목적지 도달 불가

~ 말 종류에 맞지 않는 이동을 하려고 하면 4> 부터 재입력 받음

~ 목적지 도달 불가 시, 4> 부터 재입력 받음.

~ 목적지 경로가 비어있거나, 다른 말이 존재

~ 목적지에 있는 대상이 상대편 말일 시, 삭제하며 점수 적립

~ 목적지가 비어있을 때, 말 이동

- 6> 사용자 2 에 대하여 2 ~ 5 반복

< ------- loop ------- >

- 7> 게임 종료 메시지 출력

* 체스판

- 2차원 Piece 배열 : private Piece[][] board = new Piece[8][8];

for(int row = 0; row < 8; row++) {

for(int col = 0; col < 8; col++)

board[row][col] = null;

}

이후는 말들을 적절히 위치에 대입..

* 각 말에 대한 인터페이스 (폰, 룩, 비숍, 나이트, 킹, 퀸) : Piece

1. 말 이동 : void move() :

if(movable)

print(이동 완료한 말판 출력);

else

print(이동 불가);

2. 이동 가능 여부 체크 : boolean movable(int[] board, int x, y) { ... } ~ 상세한 상태 파악은 여기다가 구현. 매개변수로 좌표를 입력받아, 이동 가능 여부 출력

3. 킹을 잡을 수 있는 지 확인 : boolean catchKing() : 보드로부터 King의 좌표를 받아, 그 좌표로 movable 하면 체크. 이는 movable() 메소드에다가 King의 좌표를 넣는 식으로 이용.

\* 킹의 경우(특수) : 체크 상태에 따른 이동 범위 제약 -> 다음 상태의 임시 piece 변수를 선언, 그 상태에서 모든 말들에 대하여 catchKing을 판단하여 check일 시 거부.

interface Piece {

public void move();

public boolean movable(int[] board, int x, int y);

public boolean catchKing();

}

* 메시지 처리 클래스(or 메소드) : AppView ~ 사용자로부터 입력을 받거나, 출력을 담당하는 부분
* 게임 시스템 : System

1. 체크상태 확인

2. 턴( for or while ~ for 는 턴 수 제한)

3. 수행할 명령 입력

4. 특수 룰(이후 추가 ~ 말 업그레이드 등..)

5. 점수(이후 추가)

* 추가 : 체스부터 구현을 하고 나서. 나중에

역할 : 김기조 님 : 게임 시스템, 메시지 처리 등 기반 시스템

김한기 님 : 폰, 룩

최재범 님 : 킹, 나이트

안우재 님 : 비숍, 퀸