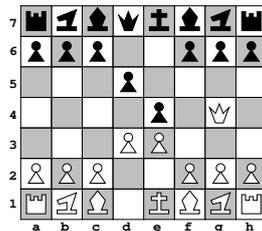


Victoria University of Wellington, School of Engineering and
Computer Science

SWEN221: Software Development

Mid-term Test (worth 10% of overall mark)

Simplified Chess



This test is about a variation on *Chess* called *Simplified Chess*. You may read about this variation here: <https://www.chessvariants.com/rules/simplified-chess>. This makes the following changes to a standard game of chess:

- *There is one less row. The board is 7×8 .*
- *There is no check/check mate — you win either by taking the opposition King or when the opposition player cannot legally move any piece other than the king.*
- *There is no double move for pawns. Pawns will always move one space. There is no en-passant and there is no castling.*
- *A player's pawn may only promote to one of the players pieces which has been previously captured (Queen, Rook, knight, Bishop). Pawns cannot move to the last row unless there is a piece to promote to.*

Long Algebraic Notation

Long algebraic chess notation is a way for writing down the moves taken during a chess game. The following illustrates an example game (which is not valid) in this notation:

White	Black
e2-e4	e7-e5
d2-d4	e5xd4
Nb1-c3	Qd8-f6
1-0	

The first move by White, e2-e4, indicates the pawn at position e2 will advance to position e4. When indicating pawn movement, no piece specifier is given. However, when indicating the movement of other pieces a specifier is always given. The specifiers are: N=kNight; B=Bishop; R=Rook; Q=Queen; K=King. So, for example, White's last move above, Nb1-c3, indicates his Knight moves from b1 to c3. Finally, a move where one piece takes another is indicated using an 'x', as in Black's move e5xd4 — the (black) pawn at position e5 takes the (white) pawn at d4. In addition, *pawn promotion* is a permitted move (e.g. b6-b7=N). And, finally, *game over* is signalled by either "1-0" (indicating white won) or "0-1" (indicating black won).

Download. You can download the code provided for the Chess program here:

http://ecs.victoria.ac.nz/~djp/files/tt_chess_2019.jar

You will find several Java source files, including a JUnit test file.

ASCII Representation. In the JUnit tests, the board is represented using an array of ASCII characters. The following illustrates such a board:

7								
6								
5								
4								
3								
2								
1								
	a	b	c	d	e	f	g	h

```
7|r|n|b|q|k|b|n|r|
6|p|p|p|_|_|p|p|p|
5|_|_|_|p|_|_|_|
4|_|_|_|p|_|Q|_|
3|_|_|_|P|P|_|_|
2|P|P|P|_|_|P|P|P|
1|R|N|B|_|K|B|N|R|
  a b c d e f g h
```

Here, uppercase characters represent the White player, and lowercase characters the black player. Furthermore, “P/p” indicates a pawn, “N/n” indicates a knight, “B/b” indicates a bishop, “R/r” indicates a rook, “Q/q” indicates a queen and, finally, “K/k” indicates a king.

1 Getting Started (worth 20%)

Begin by importing the code provided into Eclipse and running the JUnit tests. A large number of tests should be failing. There is a simple problem in each of the following classes:

- `SinglePieceMove`
- `SinglePieceTake`
- `Position`

Having fixed these bugs, you should find tests `test_01...test_45` now pass.

2 Invalid Pawn, King and Knight Moves (worth 20%)

You should find that some or all of the tests `test_46,...,test_77` currently fail. This is because the following methods are not implemented correctly:

- `Pawn.isValidMove()`
- `King.isValidMove()`
- `Knight.isValidMove()`

Having fixed these bugs, you should find tests `test_46...test_77` now pass.

3 Invalid Bishop, Rook and Queen Moves (worth 20%)

You should find that some or all of the tests `test_78,...,test_110` currently fail. This is because of problems in the way that invalid moves are detected for Bishops, Rooks and Queens. Having fixed these issues, you should find tests `test_78...test_110` now pass.

4 Pawn Promotion (worth 20%)

You should find that some or all of the tests `test_111`, ..., `test_118` currently fail. This is because pawn promotion has not been implemented at all. Having implemented this feature, you should find tests `test_111`...`test_118` now pass.

5 Game Over (worth 20%)

You should find that some or all of the tests `test_119`, ..., `test_124` currently fail. This is because code for detecting when the game is over has not been implemented at all. Having implemented this feature, you should find tests `test_119`...`test_124` now pass.

Submission.

Your test solution should be submitted electronically via the *online submission system*:

`http://ecs.victoria.ac.nz/cgi-bin/auth/submit?course=SWEN221`

Late submissions will get zero marks (unless you have arranged this with us, which will only be in exceptional circumstances). The minimum set of required files is:

```
simplechess/Game.java
simplechess/moves/Move.java
simplechess/moves/SinglePieceMove.java
simplechess/moves/SinglePieceTake.java
simplechess/pieces/Bishop.java
simplechess/pieces/King.java
simplechess/pieces/Knight.java
simplechess/pieces/Pawn.java
simplechess/pieces/Piece.java
simplechess/pieces/Queen.java
simplechess/pieces/Rook.java
simplechess/util/Position.java
simplechess/util/Round.java
```