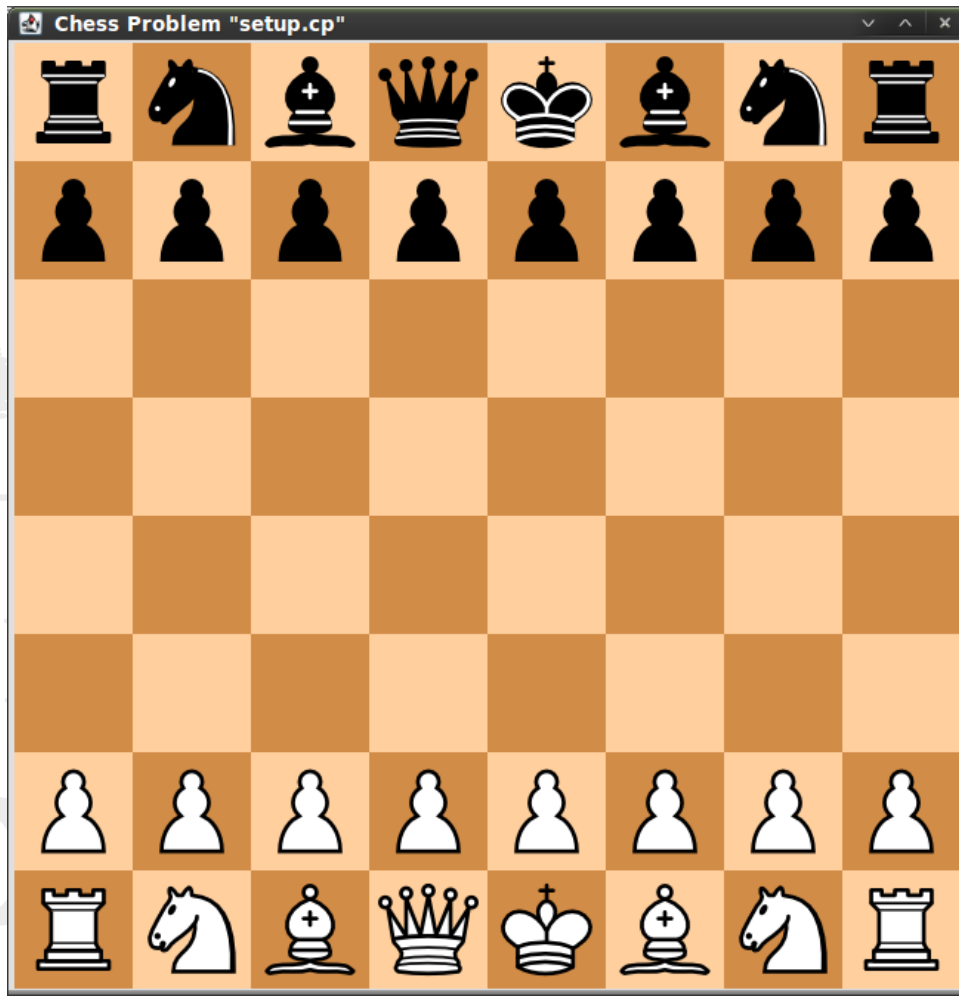


Homework 11: Drawing Chess Problems

Read in a file, and use it to display a chess problem. The output should look like this:



Start by downloading the graphics file (`chess-pieces.png`) and chess problem files (`setup.cp` and `problem1.cp`) to the active directory on your machine. The first thing your program should do is to query the user for the name of a chess problem file. It will then draw the board according to the setup described in that file. If it cannot find the file, it should issue a polite error message and terminate rather than throwing an exception.

A `.cp` file designates which pieces are in a chess problem. It has several lines of text, and will look similar to the following:

```
WQ g7  
BN f5
```

Each line describes a single piece and its location. There are 5 characters in each line:

- The first character is either B or W, and determines the piece's color ("black" or "white").

- The second character is P, N, B, R, Q, or K, and determines the type of piece. (P means “pawn”, N means “knight”, B means “bishop”, R means “rook”, Q means “queen”, and K means “king”.)
- The third character is a space (and will be ignored).
- The fourth character is the file (column) of the piece. It will be a lower-case letter between a (for the left file) and h (for the right file).
- The fifth character is the rank (row) of the piece. It will be a digit between 1 (for the bottom rank) and 8 (for the top rank).

So, the whole file represents a setup of the whole board. There will be as many pieces on the board as there are lines in the file. The file shown above has only two pieces on the board: a white queen at g7, and a black knight ready to capture her at f5.

Use the `GraphicsWindow` class from the last assignment. You can make the board using `fillRect()`, with two colors of your choice (though they should be noticeably light and dark) and dark in the bottom-left. A chess board is 8 tiles by 8 tiles. Each square should be 80 pixels on a side. Note that the size of the pieces is 68 pixels by 68 pixels, so you will need to center each piece on its square. Be sure to use Java constants, so these values can be changed quickly.

Be careful not to flip the board! The file `setup.cp` describes the proper board setup. Be sure that it looks like the image: white is on the bottom, and both queens are on the left.

If you need help understanding the different pieces, read the Wikipedia article on chess. It describes the game, including the rank/file system, in more detail.

Your program will be tested using both `.cp` files that you can download, and on a third mystery file that will remain unknown to you. This class will be called `Chess`. Be careful with your coding style, and make use of static final variables to hold constant data. You don't need to turn in the graphics files or chess problem files (unless you make your own).

Extra Credit

A small amount of bonus credit will be given for more ornate graphics. For example, you could label the ranks and files on the board with the appropriate letters and numbers. Or you could print out descriptions to the user (e.g. “White to mate in two.”) beneath the board. Use your imagination.