

## Reading from files

As you know by now, programming involves taking something that you want to do and can't do in one step, and breaking it down into smaller steps that you know how to do.

You have to know the smaller steps of loops, variables, methods, strings — and you also have to know how you can put them together in a logic way to perform the larger task that you want to do. We did this with Hangman.

Another thing that programmers need to be able to do, is to look up information and find out how to do things. In this case, you're going to learn how to use Scanner to read in a text file.

*Please read all of the steps before you start this project.*

### **Today we're going to read from a text file and display it.**

1. Start with a copy of your TicTacToe program and set the grid size to 15 (or thereabouts)
2. You can remove all the checkWin and checkTie stuff. We won't be using that.
3. Recall that we have a board[][] array of ints, and that we had specific numbers for X and O, well, we're going to make some changes and use ints for colours. 0 is empty, and then 1-4 are colours that you choose.  
(Using only positive numbers makes sense because the text file is lined up much nicer, so you could actually use the digits 0-9 and have 8 colours + background.)
4. In drawGraphics() do the following:
  1. Instead of the drawOval() for drawing the O, change it to fillRect(). This will fill in a square on the board.
  2. Use some if statements, or a switch statement, to set the colour that you're drawing with:  
`if (board[row][col] == RED ) gc.setColor(Color.RED .darker());`
5. Now we need to read in a text file that will have our drawing. So use an editor and type up a text file. The text file must be the same size as the grid. So mine is 15 lines long and each has 15 digits, separated by commas.

```
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,4,4,4,4,4,4,4,4,4,4,4,4,4,0
0,4,2,2,0,0,0,0,0,0,0,0,0,4,0
0,4,2,2,0,0,0,0,3,0,0,0,0,4,0
...
```

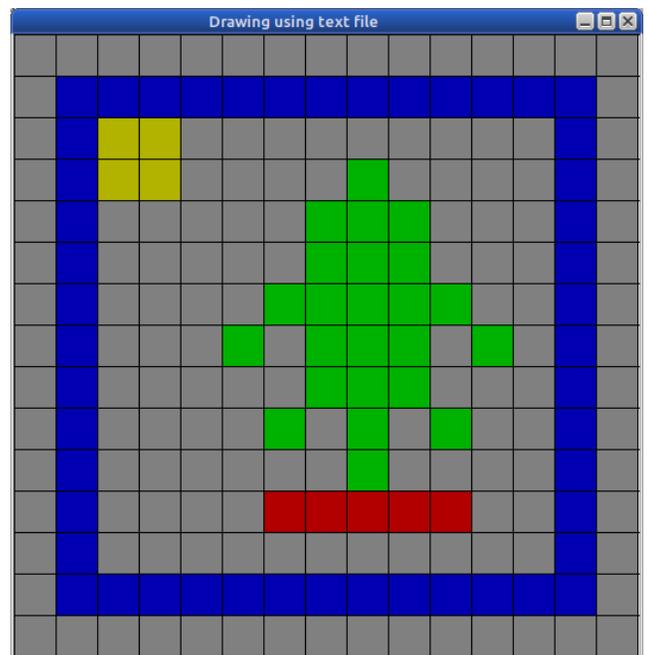
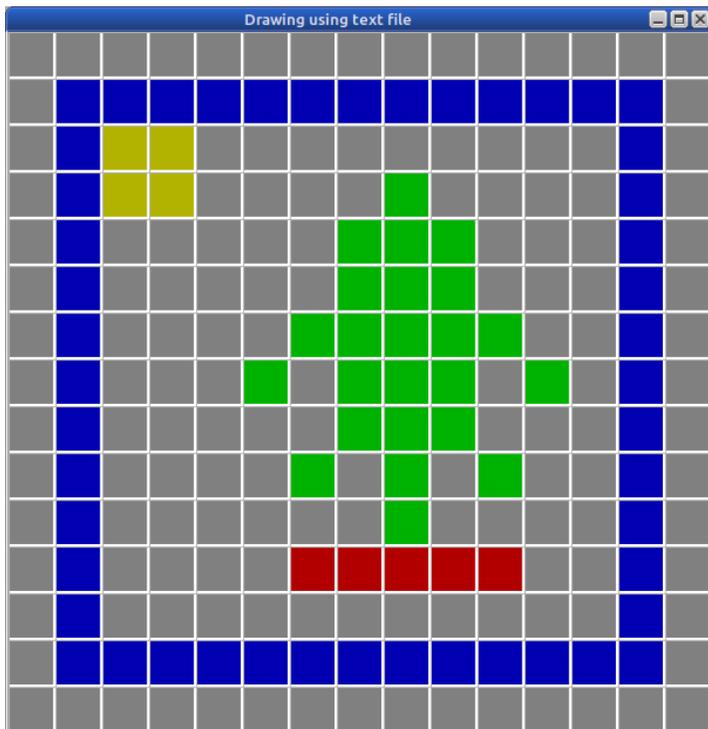
6. SCANNER can be used to read in a file. Read up on how to do this and implement it. If you can understand a little about try-catch that would be good as well. If the file does not exist, print an error message to that effect and stop the program ( System.exit(0) is fine for errors like this.)
7. Read in a line of text. Use String.split(",") to split it on commas and make a string array
8. Now go through that string array, and change each number (which is actually a string) to an integer and put it in the correct place in the board array.  
Basically, the first line of your text file is the first line of numbers that goes into your board[[]].
9. Once you have this working, you can proceed to your main game loop and you'll see the picture appear (See mine below)
10. **One final thing: WHERE do you put your text file?**
  - NOT in the package
  - NOT in the "src" folder that all of your packages are in
  - YES, in the Java project folder, whatever you java project is called.

Finally, read over your program and see where you can tidy things up. Remove repeated code if possible.

*Screenshots follow:*

### My drawing

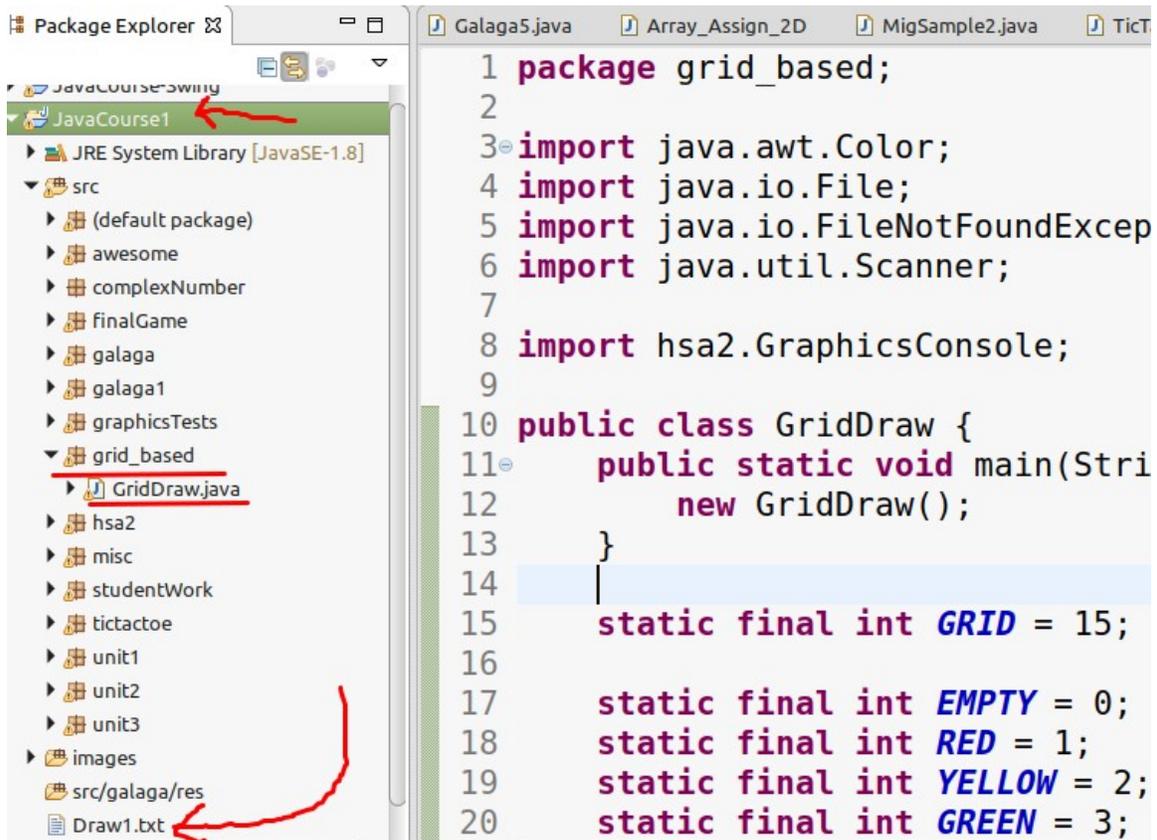
You can see that 2 is yellow and 4 is blue – from looking at the text file above. I changed the grid lines to make it look better in the second version.



Screen shot to show location of text file:

(This is an Eclipse thing. If you're not using Eclipse, don't bother with this.)

- My project is "Java Course 1"
- my package is "grid\_based"
- my java program is "GridDraw.java"
- and my text file is "Draw1.txt"



## Extra:

- If you wish, you can add more colours (I just have a total of 5)
- You could make it so that you can click on the picture and change colours
- You can increase the grid size to a larger number, then the image will not be so pixelated.
- We had a method `void printBoard() {` that prints out the array. Perhaps this could be used to print out the numbers that you're putting into your text file.