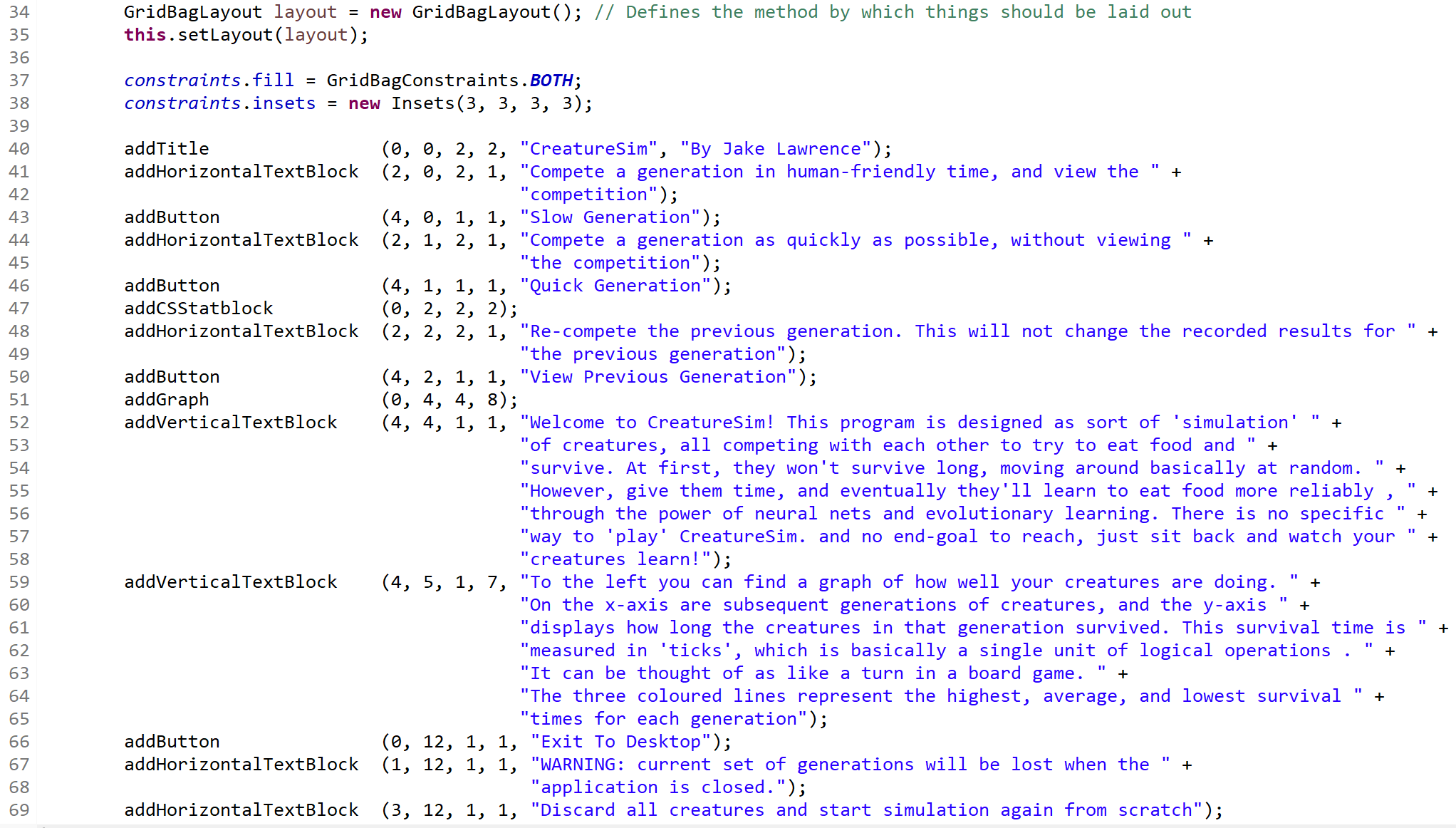
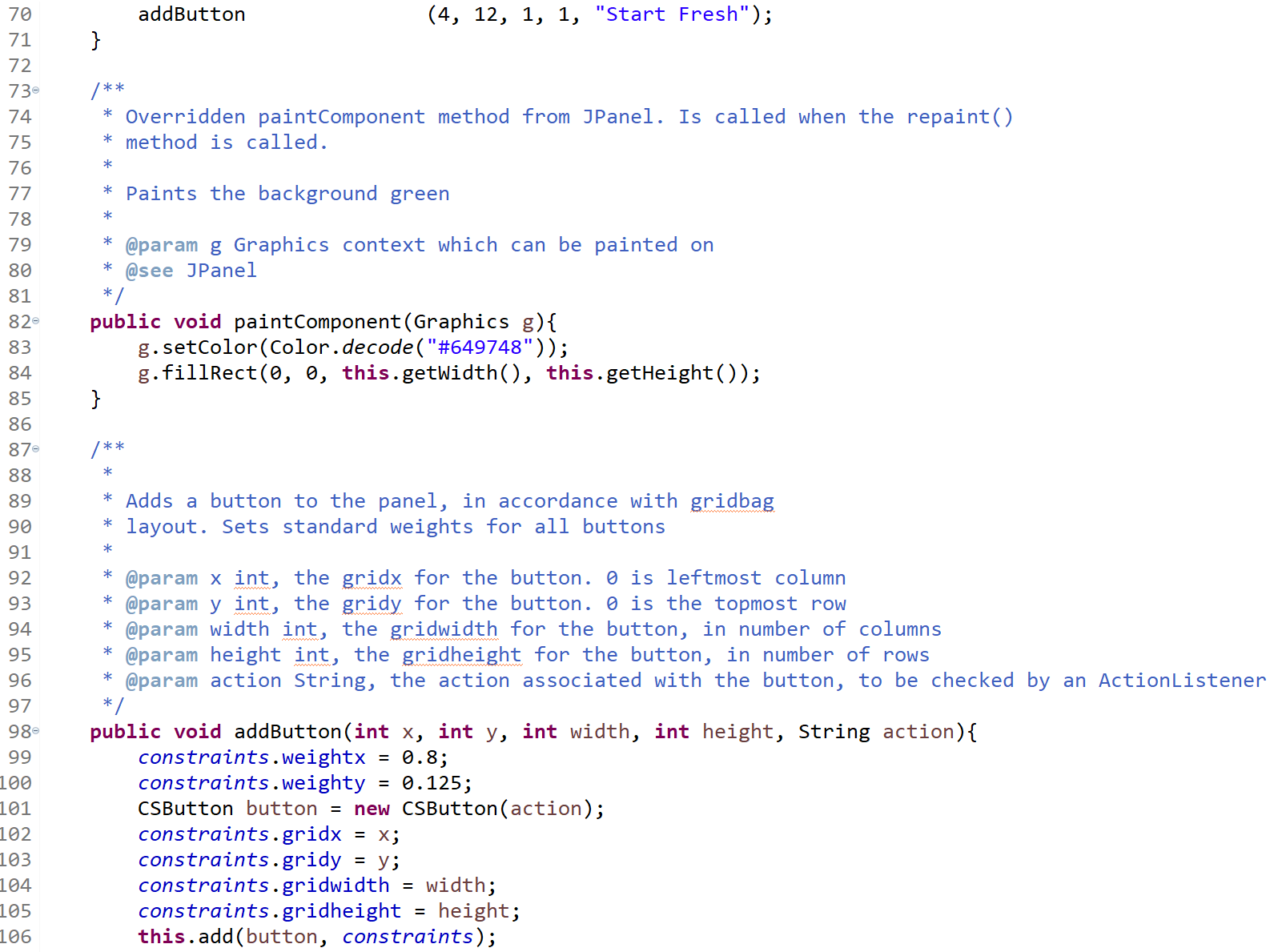
Final Implementation

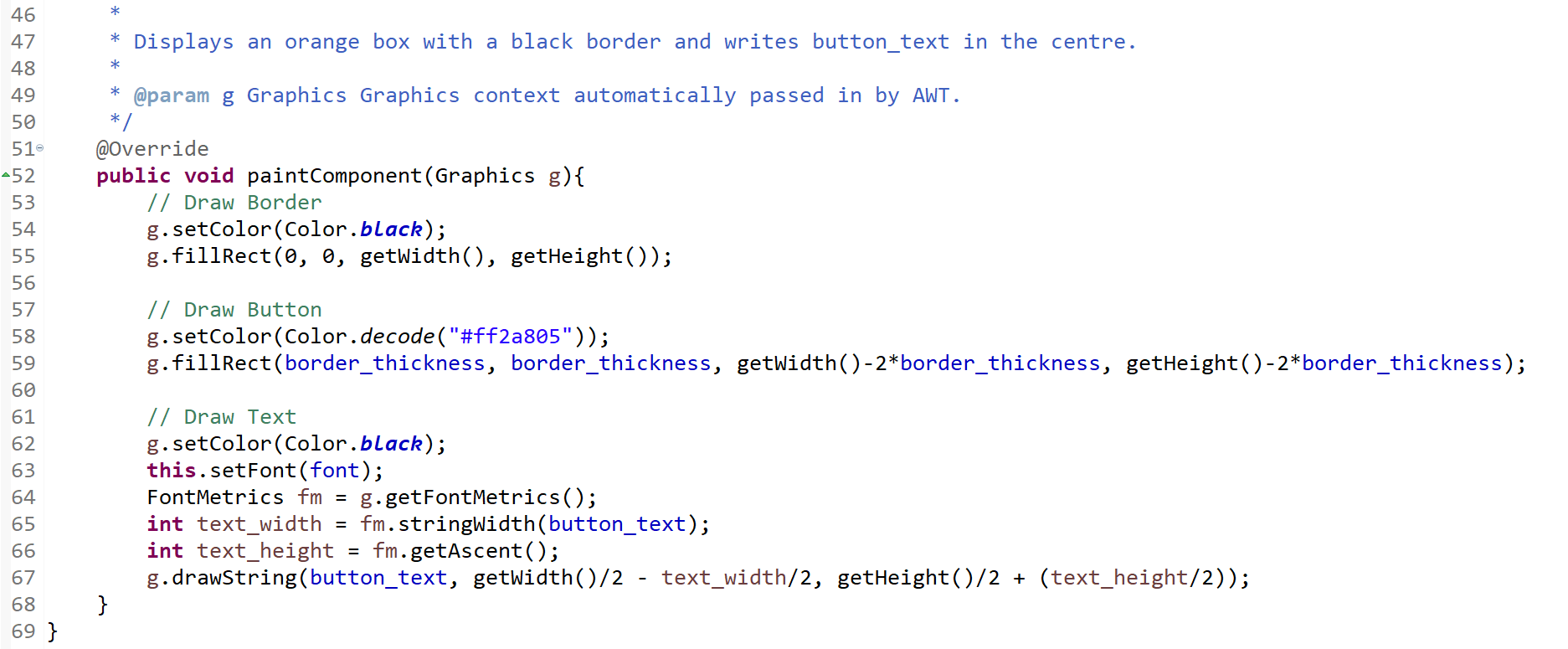
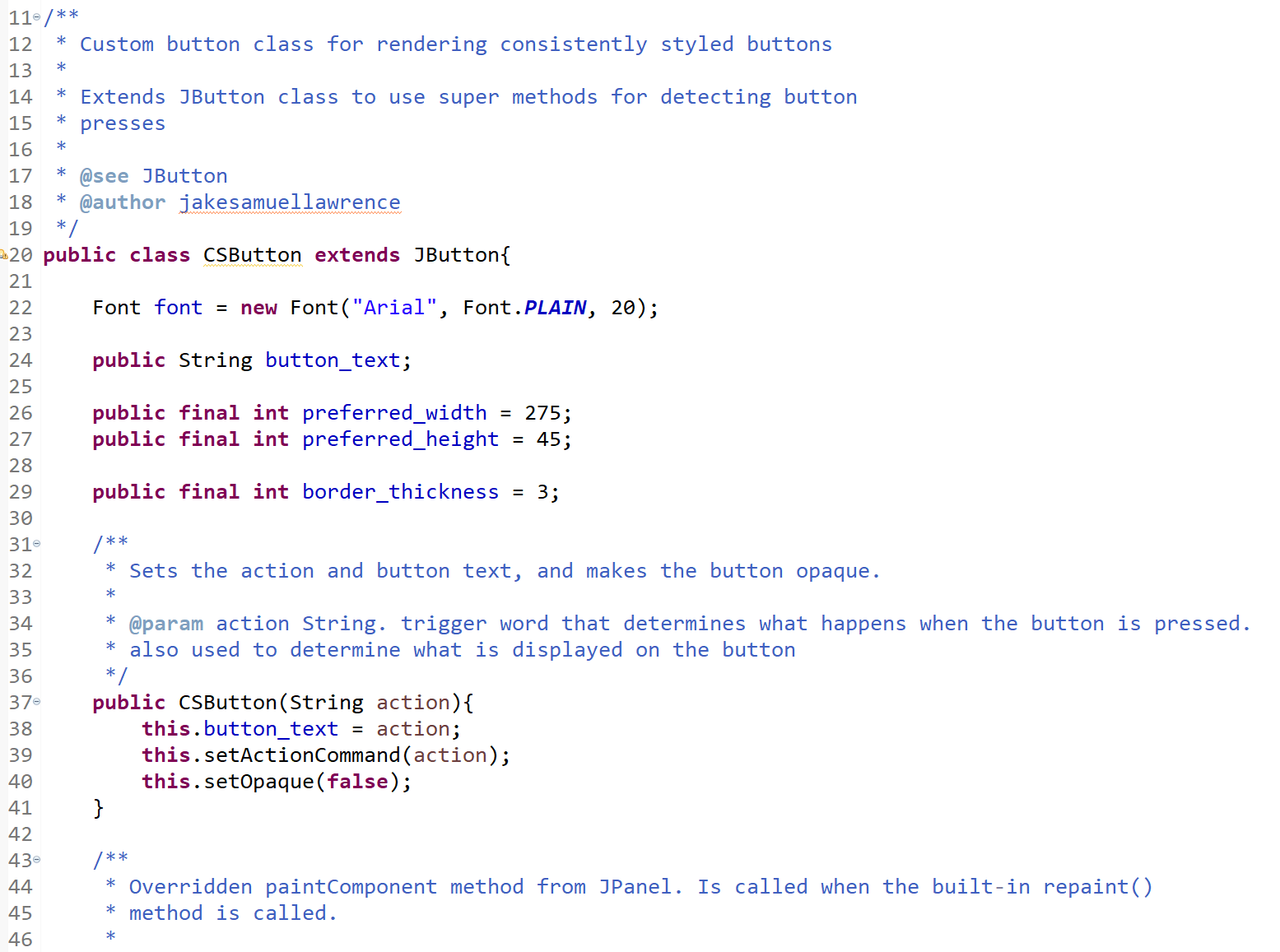
Due to the very large nature of this project, it is not feasible to provide printouts of all code involved. Instead, certain sections will be selected which I believe demonstrate the required techniques.

Evidence of User Interface implementation

The following is the declaration of all UI elements being added to the display:

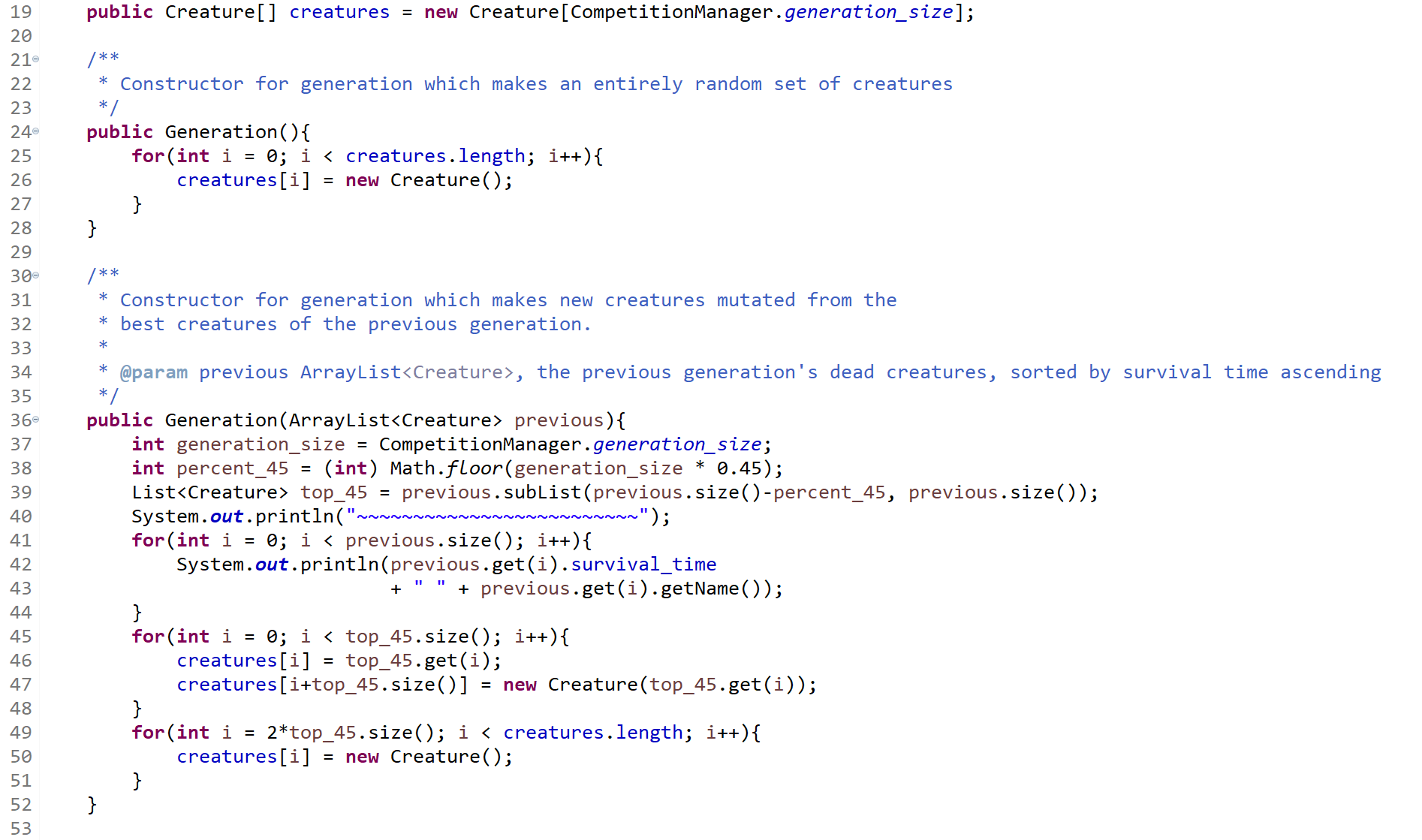


Further methods exist below, similar to AddButton() but for other UI elements. An example UI element class, CSButton, is shown below:

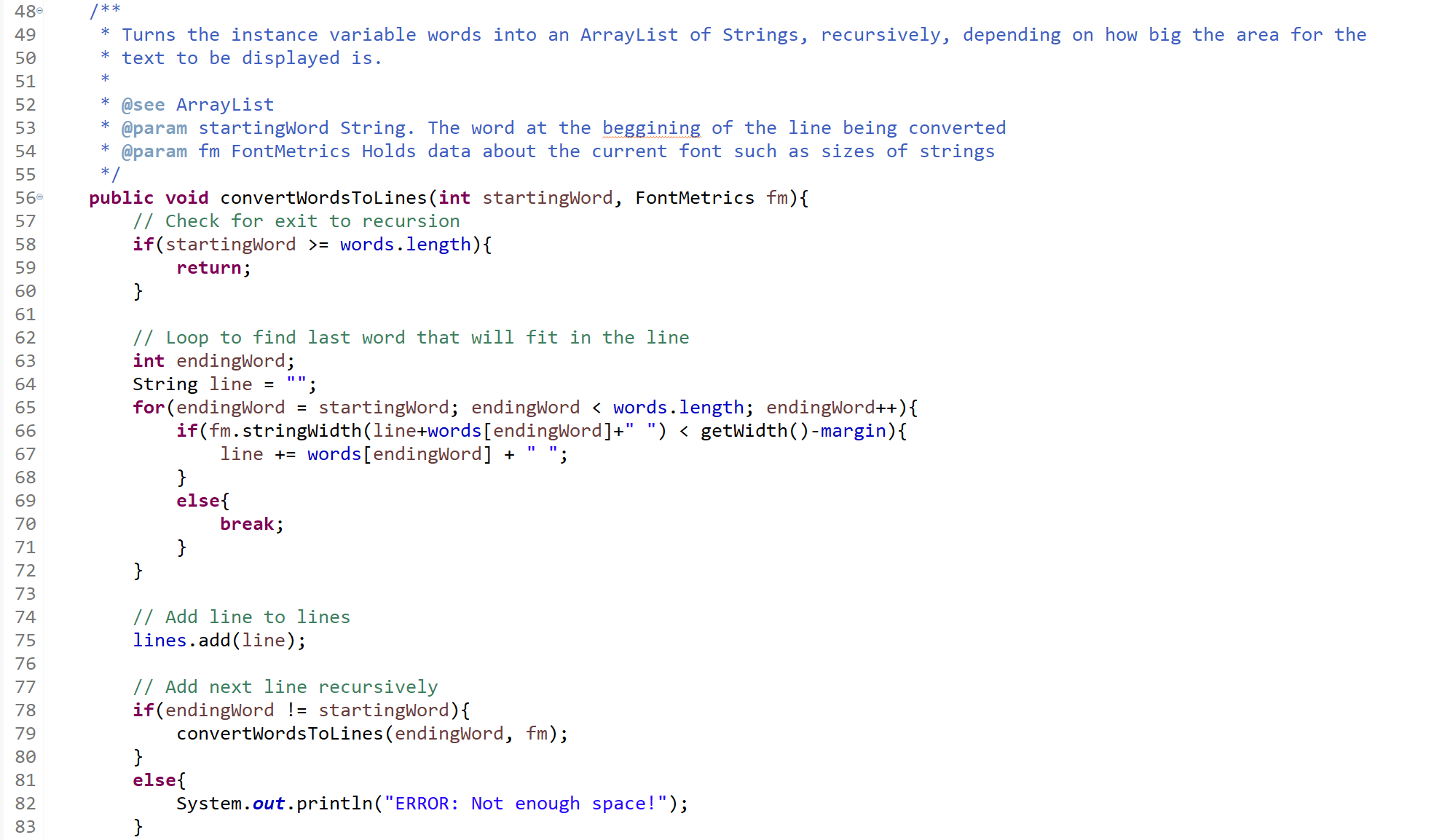


All other classes contain as much detailed commentary as this class, e.g. javadoc descriptive commentary for every method explaining how it works and what the inputs and returns are of it. A majority of methods also have commentary within them explaining the steps of the algorithms involved.

Below is evidence of storing and manipulating arrays of objects, taken from the Generation class where arrays of creatures are generated based on previous arrays of creatures



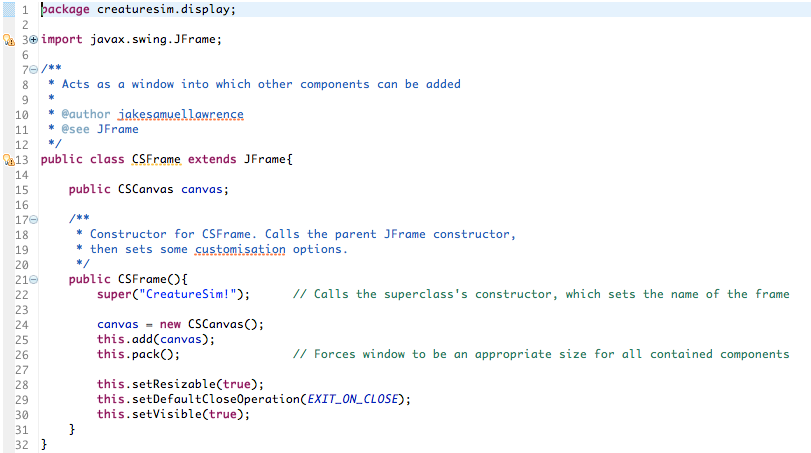
Below is evidence of the use of recursion in the word-wrap algorithm from the CSTextBlock class:

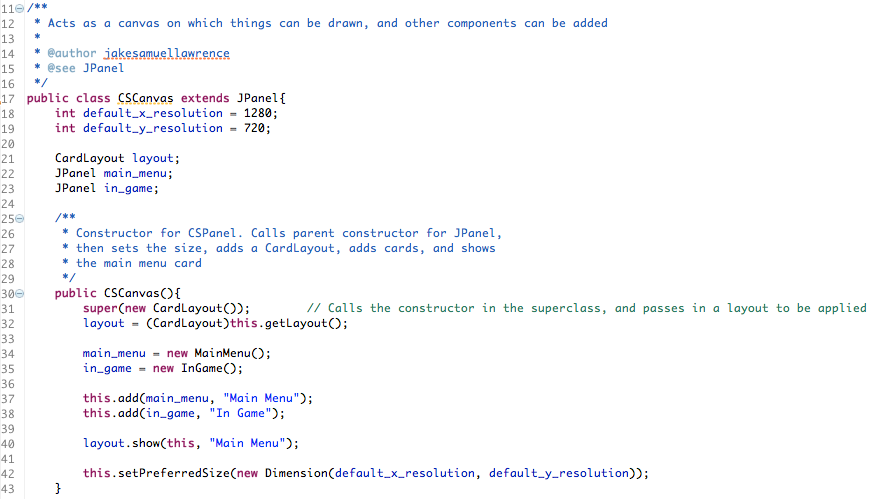


Evidence of functional requirement 1: Frame and Panel

Main.java: Macintosh HD:Local:Users:AHCS:Desktop:Screen Shot 2019-04-09 at 10.59.38.png

…Macintosh HD:Local:Users:AHCS:Desktop:Screen Shot 2019-04-09 at 11.00.07.png

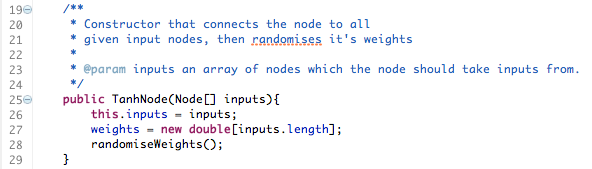
CSFrame.java:

CSCanvas.java:

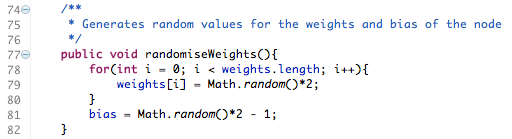
…

Functional Requirement 2: Random Neural Nets

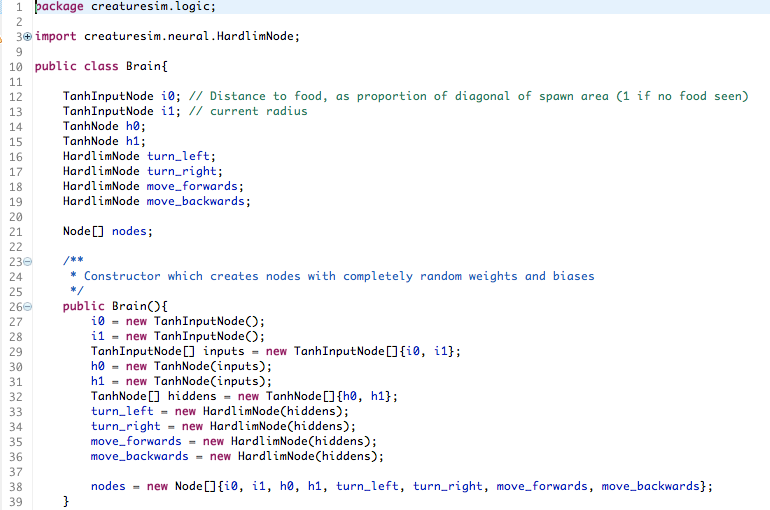
Example from TanhNode.java . Similar methods exist for the other types of neural net node:



…

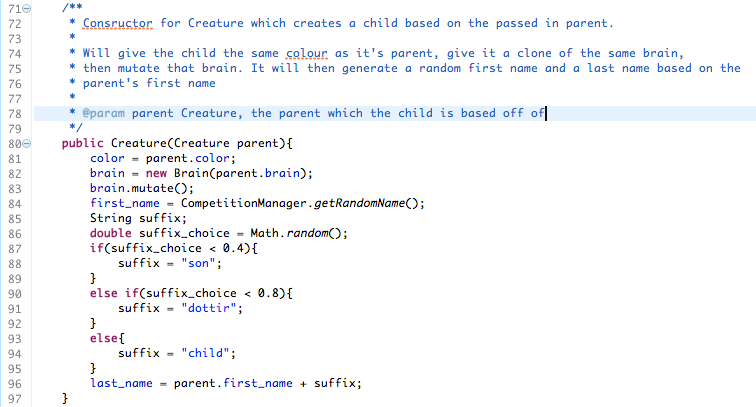


Sample from Brain.java, which creates the network of nodes:

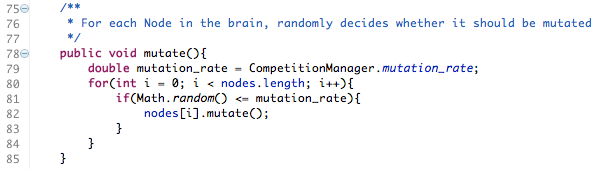


Functional Requirement 3: Children with random mutations

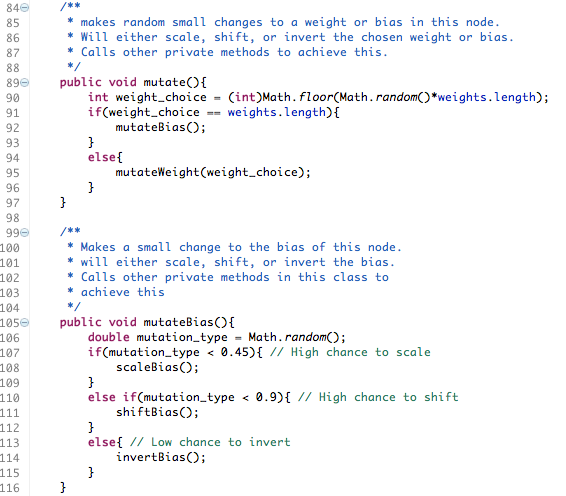
Sample constructor from Creature.java which is fed a parent creature and creates a child from it:

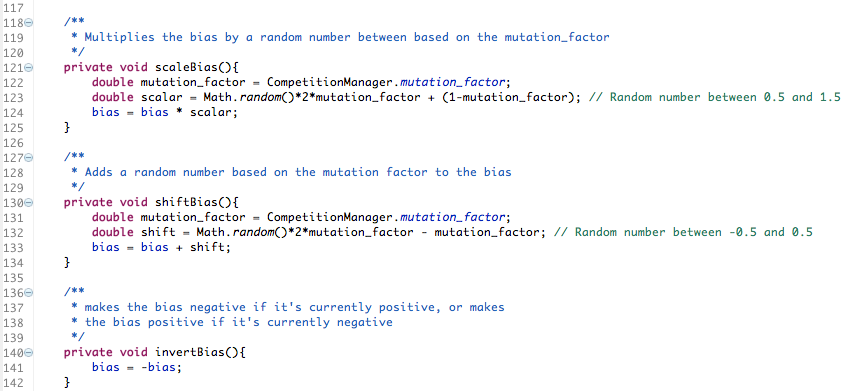


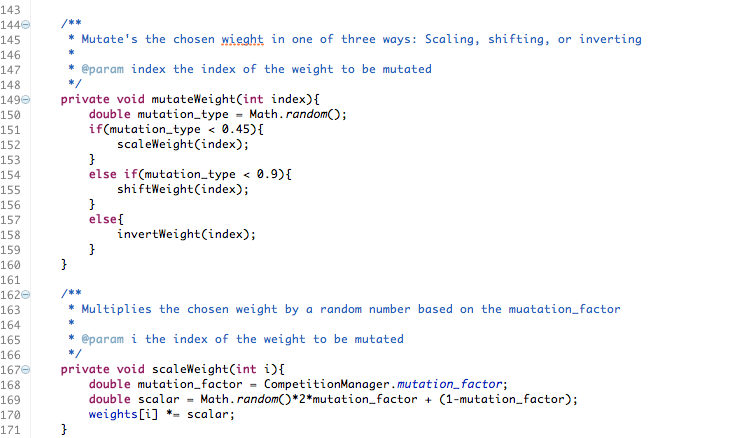
Sample from Brain.java which shows mutate method:

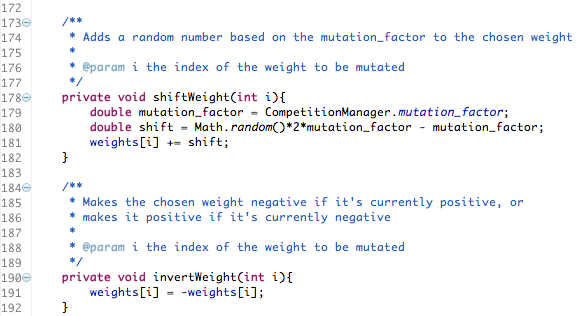


Sample from TanhNode.java which shows mutation methods. All other neural net node classes contain similar methods:







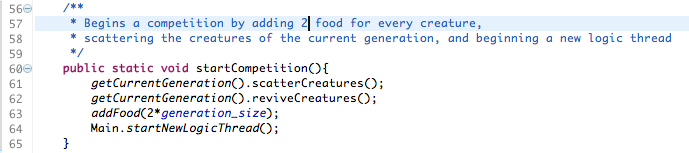


Functional Requirement 4: Quick Generation

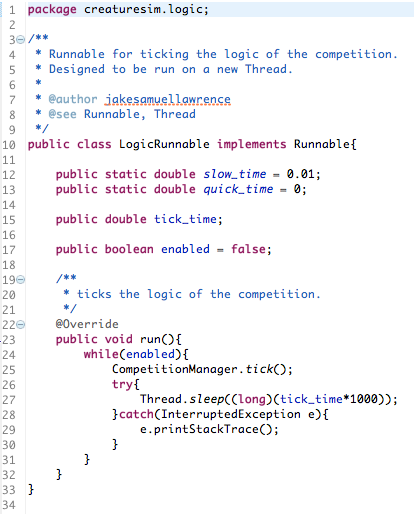
Sample from CSActionListener.java, which handles UI button presses:

Macintosh HD:Local:Users:AHCS:Desktop:Screen Shot 2019-04-09 at 11.24.52.png

startCompetition() method from CompetitionManager.java, which statically holds all the creatures and manages the way they compete:

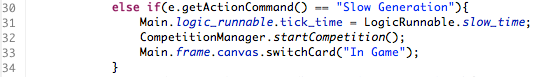


LogicRunnable.java:

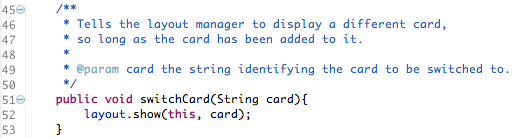


Functional Requirement 5: Slow Generation

Sample from CSActionListener.java:

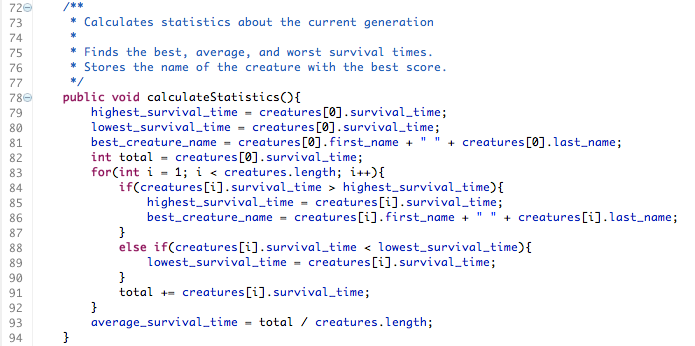


switchCard() method from CScanvas.java:

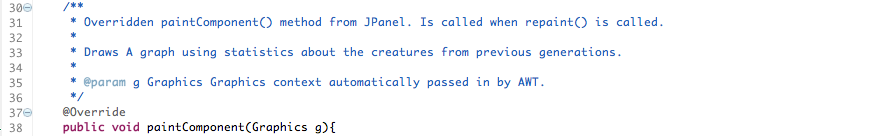
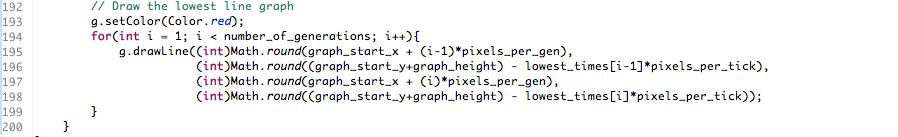
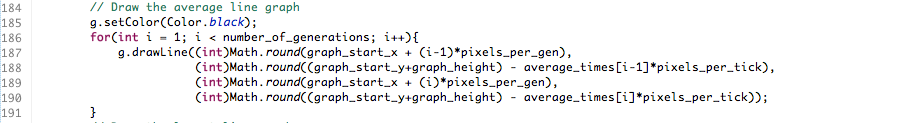
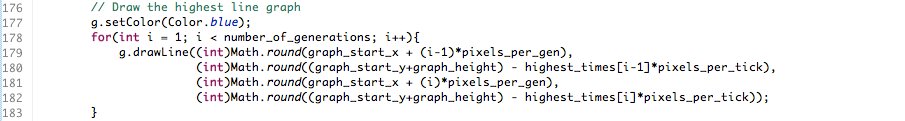
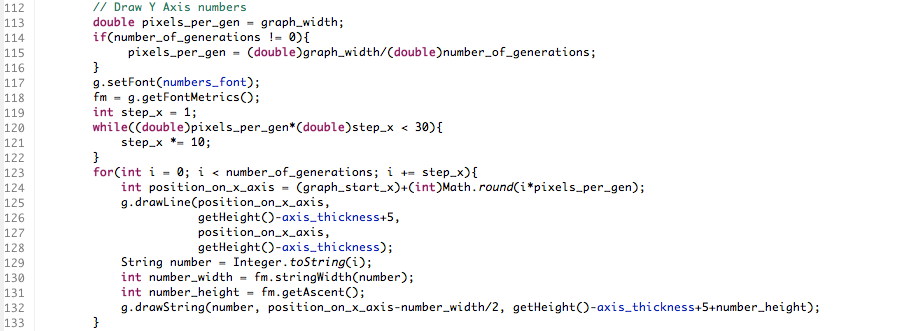
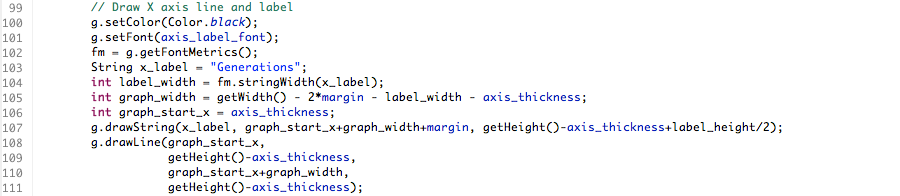
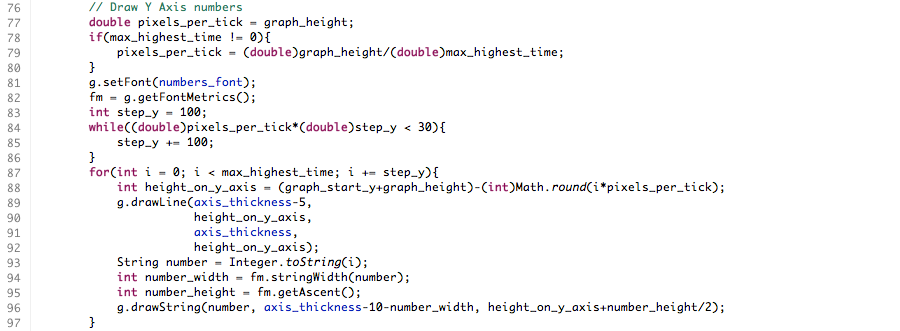
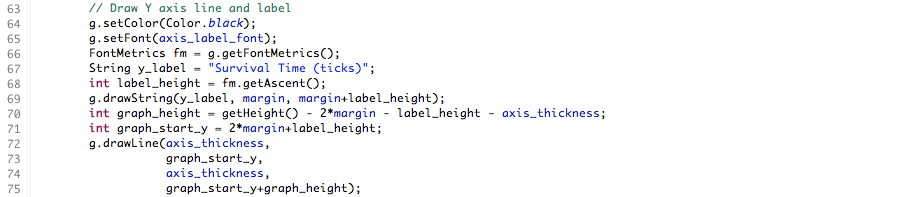
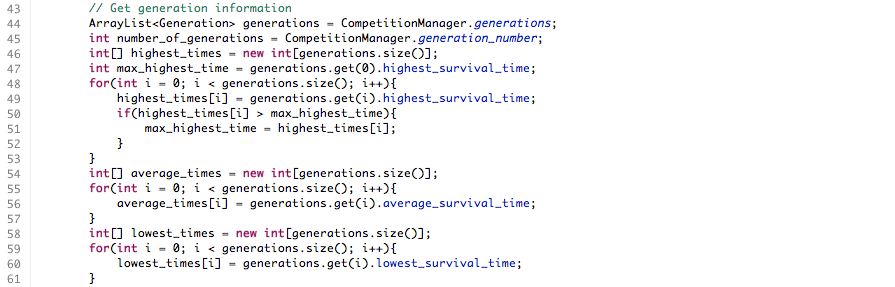


Functional Requirement 6: Record and display highest survival times

calculateStatistics() method from Generation.java which is called when a competition ends:



paint method of CSGraph.java, which displays the calculated and stored results and draws them in a graph:

Macintosh HD:Local:Users:AHCS:Desktop:Screen Shot 2019-04-09 at 11.37.57.png

Functional Requirement 7: Creature Cannibalism

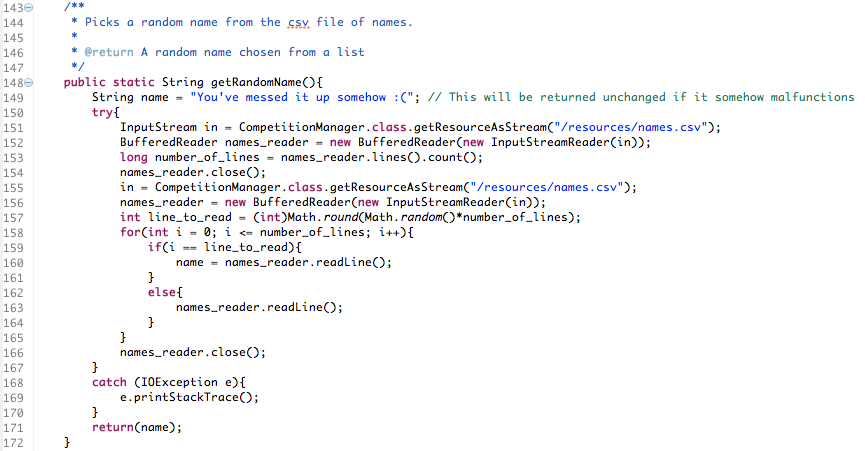
This feature was removed intentionally, as can be seen in the reflections of the progress diary

Functional Requirement 8:

This feature was removed intentionally, as can be seen in the reflections of the progress diary

Functional Requirement 9: Draw names for creatures from csv file

getRandomName() from CompetitionManager.java . This method is called when a creature is created in order to provide it with a first name:

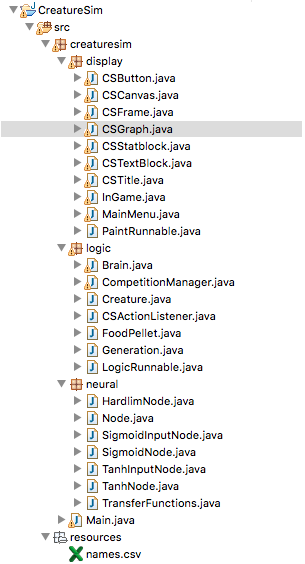


Miscellaneous Notes

No external libraries were used other than the ones which are in-built into Java. This means that any classes referenced in the above examples were created by me.

If further evidence of any of the implementation of other classes is needed, a request can be sent to either me or St Thomas of Aquin’s School.

The following is the directory structure of the whole project:



Evidence of ongoing testing can be found in the Test Plan document.

Evidence of reflective commentary can be found in the record of progress document.