Project 4 Rubric

Total Points: 315

Standards (100 pts)

- Follows coding standards (70 pts)
- Follows submission guidelines given in assignment pdf (30 pts)

Cellular Automata (CA) (215 pts)

- All visualizations must be done in JavaFX 14
- Your jar should give the option to choose which CA a user views

1D CA (40 pts)

- Grid is circular, left loops to right and right loops to left (5 pts)
- Successfully construct 1D CA from config file with the format given in the spec that behaves accordingly (10 pts)
- Successfully construct 1D CA from user input of an int in the range [0,255] and an initial grid that behaves accordingly (15 pts)
- Each cell must be a square (5 pts)
- Calculate a size for cells that ensures the grid will fit in the JavaFX window (5 pts)

Conway's Game of Life (75 pts)

- Grid is circular, left loops to the right, right loops to the left, up loops to down and down loops to up (5 pts)
- Uses a Moore's neighborhood (10 pts)
- Successfully implements the rules of the game (30 pts)
- Successfully construct initial grid from file (10 pts)
- Successfully read in user input for an initial grid (10 pts)
- Successfully construct initial grid from user input (10 pts)

Langton's Loop (100 pts)

- Grid is circular, see above (5 pts)
- Each state is a distinct color (10 pts)
- Uses a Von Neumann neighborhood with rotational symmetry (20 pts)
- Dynamically constructs a rule table from the "rule_table" file, this means no hard coding (35 pts)
- Generated rule table successfully implements the desired behavior (20 pts)
- Successfully construct initial grid from file (10 pts)

Extra Credit (100 pts)

- Implement a viewer for the 1D CA that grows as the visualization runs. Namely it only shows generations that have been generated. (25 pts)
- Implement a generic CA generator. Namely it must ask the user for the number of dimensions (1 or 2), how many states each cell can have, what the initial configuration is, and what the rule table is. Each state must still be associated with a unique color. (75 pts)