You can (should) use the provided starter file “gol.py” as the Python/Tkinter user interface for your project. A picture of that interface is shown here. The starter program provides functionality for drawing a cell, setting generations, and running a simulation. The starter program has a canvas for drawing and selecting cells; a scale for setting the number of generations; and a button for running a simulation. A leftMouseClick(self, event) function is also provided that uses the x y location of a mouse click in the canvas area to set a cell alive. The starter program displays all the cells alive/dead on alternating generation. I have tried to place comments in gol.py to describe, or print statements to show, some of the operations of Tkinter. The screen capture below shows some cells set alive by left mouse clicks (some of these patterns are stable).
Conway's Game of Life. Consider a grid of cells (N by N). The cells can either be alive or dead. On each generation determine for each cell its next state. All cell state determination is done "simultaneously" – that is you determine the next state for each cell without changing any cell's state. After determining the next state for all the cells you then change the state of each cell. This process of determining, and then possibly changing, a cell’s state is called a generation. A cell’s state is determined by its neighborhood – its possibly 8 adjacent cells. Do not consider wrapping around the grid. If a cell is at the edge of the grid then its neighborCell that is off the grid is considered dead. If the count of alive cells in a cell's neighborhood is 0, 1, or greater than 3 the cell dies (no resources or overcrowding). If the count is 2 the cell maintains its current state. If the count is 3 the cell stays alive, or becomes alive (is born). The simulation should start with an initial configuration of alive cells and should run for the specified number of generations. The initial configuration should be set by clicking the mouse (left button) on a cell. In fact clicking a mouse button on a cell toggles its alive/dead state. You can continue generations by clicking “run” without changing the state of the cells.

Submission. Each group should submit their solution on removable storage media (floppy disc, CD, USB drive, etc.). The media should be labeled with the names of the group members. Please submit the entire project directory – source files and executable files. In addition there should be a printed UML lite class diagram of the solution. The names and email addresses of every group member must be in printed in the comments in the source of each class, on the submitted media and printed on the UML lite class diagram. There should also be a printed statement about how to run the solution. You can included any additional documentation that you think will help me grade your submission (more documentation does not imply a higher grade). Do not print the source file of the classes for submission. Submit media and printed documentation in an 8.5 by 11” folder. I will provide folders.

Try some of these patterns: