Info I | 02.06.2025 |

LAB ASSIGNMENT 7

DUE TO NEXT LAB

1. Save a copy of the **Brain** project as game-of-life and modify the Cell class so that it implements the rules for the Game of Life. It should not be necessary to modify any other classes. In Conway's *Game of Life*, a cell has only two states, alive or dead. Its next state is determined as follows:
   - A count is taken of the number of its neighbors that are alive.
   - if the cell is dead and has exactly three live neighbors, then its next state will be alive, otherwise its next state will be dead.
   - If the cell is alive, then its next state also depends on the number of live neighbors. If there are fewer than two or more than three, then the next state will be dead, otherwise alive.
   - Save a copy of the **Brain** project as game-of-life and modify the Cell class so that it implements the rules for the Game of Life. It should not be necessary to modify any other classes.

Commented [NH1]: Exercise 7.38

```
   /**
    * Determine this cell's next state, based on the
    * state of its neighbors.
    * This is an implementation of the rules for
    * Conway's Game of Life.
    * @return The next state.
    */
   public int getNextState()
   {
      // Count the number of neighbors that are alive.
      int aliveCount = 0;
      for(Cell n : neighbors) {
         if(n.getState() == ALIVE) {
            aliveCount++;
         }
      }
      if(state == DEAD) {
         return aliveCount == 3 ? ALIVE : DEAD;
      }
      else {
         return aliveCount < 2 || aliveCount > 3 ? DEAD :
ALIVE;
      }
   }
```

In addition, the possible states should be defined as:

```
   // The possible states.
   public static final int ALIVE = 0, DEAD = 1;
   // The number of possible states.
   public static final int NUM_STATES = 2;
```

2. Modify the Environment class in your version of the Game of Life so that the environment is not **toroidal**. In other words, the next state of a cell at an outer row or column of the environment is determined by assuming that neighbors outside the bounds of the environment are always dead. Is there any difference between this behavior and the toroidal one?

3. Look up the initial patterns on the Wikipedia that oscillate and move. Try them out on your game-of-life. Do they work?

4. Program the rules for this **Corona-virus puzzle**: Cells now only have 4 neighbors, not 8. What do you have to change?

The status of the cells is either infected or not infected. A cell that is not infected becomes infected if two or more of its neighbors are infected. Can you find a state that eventually fills an entire 8x8 board? https://www.spiegel.de/karriere/epidemie-auf-dem-schachbrett-raetsel-der-woche-a-8bf324e1-20a3-44ae-b35d-0ba1bdd8d9a5

5. (Addon /For the bored) Can you add colors to your board? A cell has a color that depends on the number of rounds it has stayed alive (or the number of neighbors it has, be creative). What do you have to change to get this to work?