cmput 355 fall 2020 assignment 1

You can work on this assignment in groups of up to 5 but . . . do not just copy from your group members (or anyone else) and . . . if we suspect there is mindless copying, we might ask you later to explain your answers, and if you cannot, we might deduct some or all marks, and of course report any suspected plagiarism to the appropriate authorities.

The assignment is due SUNDAY SEPT 27 23:59 Edmonton time. Submit your answers in the form of a `.py` python3 file. Use life-asn1.py from the course github repo as your starting template.

For this assigment, each student's secret number is the 4th and 5th integer of their ccid, interpreted as a 2-digit number. E.g. if your ccid is `***91**`, then your secret number is 91.

1. **Answer as a comment at the indicated location in the template file.** For the game of life, for each property below, describe a small (in terms of grid needed to show it, and the number of live cells) pattern:

   a) grid needed for the whole simulation is unbounded (hint: course reading)

   b) the number of cells alive at any one time is unbounded (hint: Conway's bet)

2. **Answer as a comment at the indicated location in the template file.** Briefly compare and contrast `life.py` and `life-np.py`, from `simple/life/` in the class github repo. As part of your answer, explain

   - what happened to `num_nbrs` in `life-np.py`,
   - how an infinite grid is implemented in `life.py`,
   - how the use of a guarded board simplifies `num_nbrs`.

3. To implement the game of life on a bounded torus, start with a 2-dimensional bounded grid, and assume that the top and bottom rows are adjacent (so, for each fixed j, the jth cell in the top row is a neighbor of the jth cell in the bottom row). At this point we have a cylinder. Now turn the cylinder into a torus by further assuming that, for each fixed k, the kth cell in the leftmost row is adjacent to the kth cell in the rightmost row. See

   `https://www.youtube.com/watch?v=lxIeaotWIks` .

   On the supplied `life-asn1.py`, **make only the changes requested below: do not change anything else!**

   a) In your version of `life-asn1.py`, complete functions `next_state2` and `num_nbrs2`.

   b) Provide the code for functions `num_nbrs_torus`, and `next_state_torus`, which assume that the bounded grid is a torus, not a plane, and modify `interact` so that it calls `next_state_torus` instead of `next_state`.

   c) Run your torus program with input file `t/g-asn1`. As a final comment in your program, show the state after n iterations, where n is the sum of the secret numbers of the students in your group. Check your answer to make sure that it is the same for everyone in your group!