CSCI 1300

Intro to Computing

Gabe Johnson

Lecture 24 Mar 11, 2013

Cellular Automata

Upcoming Homework Assignment

HW # **Due: Friday, Jan 1**

Homework Title

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam iaculis vehicula mollis. Praesent vulputate pellentesque iaculis. Phasellus dui orci, euismod in dignissim in, sodales eget magna. Proin in gravida felis. Curabitur vel nunc nisl. Aliquam vel purus eu velit adipiscing ornare eu a ligula. Cras nec porttitor purus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia.

Lecture Goals

1. Cellular Automata
2. More Graphics

Cellular Automata

- Code that *lives*
- Cellular: dealing with *cells*: discrete values. Can think of like cells in an organism
- Automata: seems to run itself

Look for YouTube videos on *Langton's Ant* and *Conway's Game of Life* for examples of 2D automata.

One Dimensional C.A.

We are going to work with one dimensional cellular automata. This means that within any time frame we can index any cell with *one* value.

In the assignment we will use an integer that references into a character array.

Simple Rule

If previous row had *, I am a space.
If previous row had a space, I am *.

0	1	2	3	4	5	6
*		*	*			*
	*			*	*	
*		*	*			*

Next Row will be:

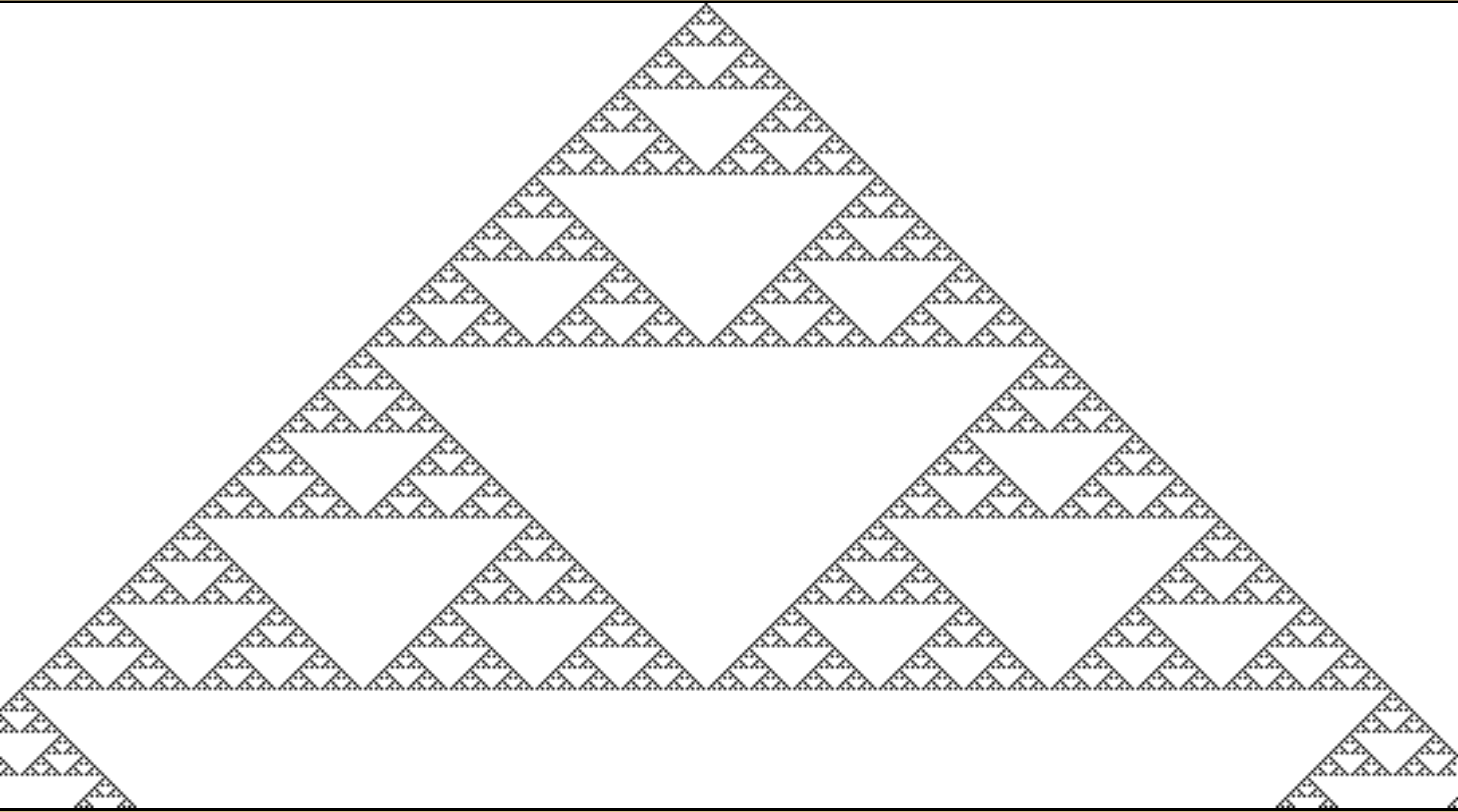
```
      *                    *       *
```

Neighborhood

In the previous example the *neighborhood* was only one. In more interesting examples the neighborhood may have more items. In the homework our neighborhood involves three elements:

* previous state, column i-1
* previous state, column i
* previous state, column i+1

“Rule 90”



HW Instructions

There are several methods that are already implemented. Study them and learn! But you shouldn't edit them in the file you turn in.

Look for the methods with the comment:

    // IMPLEMENT THIS METHOD.

Implement the indicated methods (four? of them)

setInitialPattern

```
public void setInitialPattern(String input)
```

Here the idea is to turn the input string into a character array called ‘prev’.

run

public void run(int iterations)

This runs your automata through the given number of iterations. You'll call cycle() to evolve your organism one stage, and then printPrev() to display its resulting state.

cycle

```
public void cycle()
```

This will run the cellular automaton through one more stage of its life cycle. At the end, it updates the ‘prev’ character array with the resulting state.

getPattern

```
public String getPattern(int idx)
```

This method looks in the character array called ‘prev’ and produces a String that describes the neighborhood around *idx*.

To do this you’ll need to create a String based on three primitive char items.

Hint on concatenating

If you want to combine anything and get a String, one trick in Java is to use the *empty string* in the concatenation statement. The empty string is just two double quotes in a row: “”

```
int a = 4;
char b = ‘x’;
float c = ‘4.3’;
String abc = “” + a + b + c;
```

More Graphics (Code)