Technical Explorations

by Keith R. Bennett

Conway's Game of Life Viewer

Sep 5, 2012 • keithrbennett

Later this month I’ll be joining dozens of other coders at Ruby DCamp, where we’ll spend three days talking, coding, and camping. The first day is usually code katas (exercises), and often one of them is the implementation of Conway’s Game of Life.

By design, there’s not enough time to do a complete implementation with viewer, so I thought it would be cool to write a viewer into which you could plug your own model implementation and “play” that model visually.

In addition to using the viewer to run different implementations of the Game of Life, it could also be useful in coming up with illustrative and interesting game data, using the provided model implementation.

We often have poor or nonexistent Internet connectivity, and client/server seemed to be overkill, so I brushed off my old Java Swing skills and wrote a minimal viewer in JRuby. The code is at https://github.com/keithrbennett/life_game_viewer.

Here’s a screen shot:

Game of Life Viewer

Game of Life Viewer

In case you don’t recognize the face, it’s Alfred E. Neuman, made famous by Mad magazine, but, as I just learned this minute from Wikipedia (I read it there so it *must* be true), “[his] face had drifted through American pictography for decades before being claimed and named by *Mad*…“…but I digress…

You can install the gem in the usual way (make sure you’re in JRuby when you do):

```
>gem install life_game_viewer
```

A sample model implementation with sample initial values are provided so that you can play with the viewer before beginning the exercise. This sample implementation is available by running `LifeGameViewer::Main.view_sample` in irb, or `life_view_sample` on the command line.

You can’t see it in this image, but if you hover over a cell, a tool tip containing the coordinates and the value (alive or not) will be displayed.

One of my favorite features is the simplicity of data initialization. One of the required model methods is a static method *create*, which takes a row count, column count, and optionally, a block with which to initialize each cell. This makes it simpler and more concise to experiment with forumulas and patterns. For example, the code below would result in an X shaped board, and is all the code you’d need to run the viewer.

```
require 'life_game_viewer'

model = SampleLifeModel.create(12, 12) do |row, col|
  (row + col == 11) || (row == col)
end
LifeGameViewer::Main.view(model)
```

Game of Life Viewer

In the image above, the board is initialized with `(row + col == 11) || (row == col).`

There’s a lot more information, including instructions and troubleshooting, on the Life Game Viewer project page, and in comments in the source code.

Published with GitHub Pages