

Conway's Game of Life

Alex Chubaty

17 November 2016

```
library(magrittr)
library(raster)

## Loading required package: sp

##
## Attaching package: 'raster'

## The following object is masked from 'package:magrittr':
##
##      extract

library(SpaDES)

## Default paths for SpaDES directories set to:
##   cachePath: /tmp/RtmpthVcz0/SpaDES/cache
##   inputPath: /tmp/RtmpthVcz0/SpaDES/inputs
##   modulePath: /tmp/RtmpthVcz0/SpaDES/modules
##   outputPath: /tmp/RtmpthVcz0/SpaDES/outputs
## These can be changed using 'setPaths()'. See '?setPaths'.

##
## Attaching package: 'SpaDES'

## The following objects are masked from 'package:stats':
##
##      end, start, time
```

Overview

This is a simple module illustrating the implementation of Conway's Game of Life.

From Wikipedia:

The universe of the Game of Life is an infinite two-dimensional orthogonal grid of square cells, each of which is in one of two possible states, alive or dead, or “populated” or “unpopulated” (the difference may seem minor, except when viewing it as an early model of human/urban behavior simulation or how one views a blank space on a grid). Every cell interacts with its eight neighbours, which are the cells that are horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:

1. Any live cell with fewer than two live neighbours dies, as if caused by under-population.
2. Any live cell with two or three live neighbours lives on to the next generation.
3. Any live cell with more than three live neighbours dies, as if by over-population.
4. Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.

The initial pattern constitutes the seed of the system. The first generation is created by applying the above rules simultaneously to every cell in the seed—births and deaths occur simultaneously, and the discrete moment at which this happens is sometimes called a tick (in other words, each

generation is a pure function of the preceding one). The rules continue to be applied repeatedly to create further generations.

Usage

```
library(igraph)
library(raster)
library(SpaDES)

setPaths()

X = 10
Y = 10
TYPE <- "blinker" ## see below for other types

modules <- list("gameOfLife")
parameters <- list(
  gameOfLife = list(X = X, Y = Y, initialType = TYPE)
)
times <- list(start = 1, end = 10)

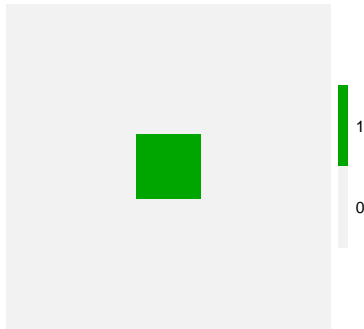
clearPlot()
mySim <- simInit(times = times, params = parameters, modules = modules)
mySim1 <- spades(copy(mySim))
```

Alternate seeds

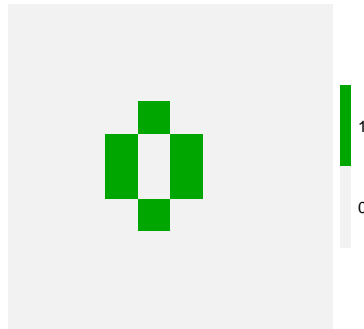
Below are some alternate seeds that demonstrate the model's behaviour. Names correspond to those in the Wikipedia article.

Stills

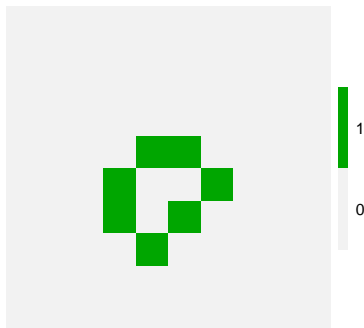
block



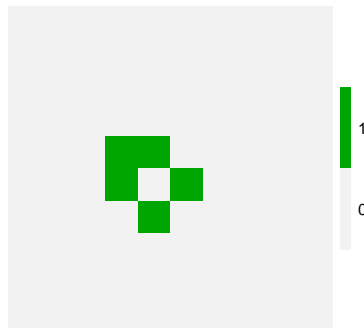
beehive



loaf



boat



Oscillators



Events

Generation

Each generation (iteration), the rules above are applied to the grid/raster.

Plotting

The grid/raster (world output object) is plotted. Plotting occurs every generation by default.

Data dependencies

Input data

NONE

Output data

A raster with dimensions $X * Y$ is output by this module.

Links to other modules

NONE