# Game of Life Challenges

http://en.wikipedia.org/wiki/Conway's\_Game\_of\_Life

Download the files from <http://community.dur.ac.uk/s.p.bradley/Life.zip>

Complete one or more of these challenges.

## Artistic

1) Write functions for a variety of basic shapes e.g. block, beehive, blinker, toad, beacon, pulsar, lightweight spaceship and use these (particularly the stills and oscillators) to build more complicated patterns. For instance, you could use them to build into letters that you then build into words by writing functions that call other functions that call other functions.

2) Drum machine. You can get javascript to play sounds using the Gibberish library http://www.charlie-roberts.com/gibberish/

Combining these with oscillators of different periods would make for a very interesting audio-visual drum machine.

3) Structures as pixels. Take an image and turn it into binary or ternary pixels: you might want to use python to do this. Then render the pixels using still-lifes or oscillators in the game of life. Or even render two images with the pixels as flyers, and make them collide with each other.

4) 3D Life. There are examples of using different shaped boards: triangles and hexagons are obvious examples but you could take it further. You could take point 1) below even further by actually showing the game as being played on a 3D surface e.g. a cube, or a toroid (doughnut) which you render with something like webGL (<http://en.wikipedia.org/wiki/WebGL>), possibly using the three.js library to make things a little easier.

## Theoretical

1) Try to adapt the code (you would need to learn a bit more about javascript to do this) so that the cells at the top are neighbours of the cells at the bottom, and cells at the left are neighbours of the cells on the right. With this approach a glider would then disappear from one edge only to reappear at the other edge. On wikipedia it calls this a toroid because you can imagine mapping the square grid onto a doughnut shape.

2) Write a javascript function to place a Gosper’s glider gun in an arbitrary position in given direction.

3) Find new or old still life/oscillators. There are a finite number of possible layouts for any given rectangle. Can you write a program which

1. Works out whether a given pattern is a still life or an oscillator (period 2)
2. Then generates all of the possible shapes of a given size (e.g. 2x2 or 4x3) to find all of the still lifes and oscillators of that size.

4) Do binary logic on streams of bits by colliding flyers. A [paper by Rennard](http://cogprints.org/4115/1/CollisionBasedRennard.pdf) explains how to build an adder out of logic gates constructed in the game of life. Doing everything in the paper would take too long, but you could try to write a javascript program which

* took two integers (in decimal or hexadecimal) as parameters
* translated them to binary
* represented them in a flyer stream
* collided them with each other and with Gosper glider guns to perform logic operations

Look here for more details:

http://cogprints.org/4115/1/CollisionBasedRennard.pdf

5) State machines. It says on the wikipedia page "For example, if two gliders are shot at a block in just the right way, the block will move closer to the source of the gliders. If three gliders are shot in just the right way, the block will move farther away. This 'sliding block memory' can be used to simulate a counter." I've never tried this but with a bit of research you could do it, but this would be pretty challenging. Look here for more details:

<http://www.radicaleye.com/lifepage/patterns/sbm/sbm.html>

## Other stuff

Just look at <http://www.conwaylife.com/wiki/Main_Page> and see if anything grabs your fancy.