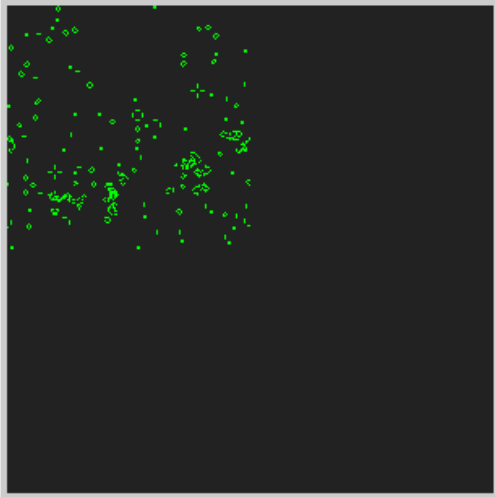


ASSIGNMENT

PROTOVIS

Conway's Game of Life



Source

```
<html>
<head>
  <title>Conway's Game of Life</title>
  <link type="text/css" rel="stylesheet" href="ex.css?3.2"/>
  <script type="text/javascript" src="../protovis-r3.2.js"></script>
  <script type="text/javascript" src="life.js"></script>
  <script type="text/javascript" src="breeder.js"></script>
  <style type="text/css">

body {
  background: #222;
}

#fig {
  width: 300px;
  height: 300px;
}
```

```

    </style>
</head>
<body><div id="center"><div id="fig">
  <script type="text/javascript+protovis">

var vis = new pv.Panel()
  .width(life.size * 2)
  .height(life.size * 2);

vis.add(pv.Image)
  .def("init", function() life.update())
  .imageWidth(life.size)
  .imageHeight(life.size)
  .image(pv.colors(null, "#0f0").by(function(i, j) life[j * life.size + i]))
  .event("click", function() life.reset());

vis.render();

setInterval(function() vis.render(), 42);

  </script>
</div></div></body>
</html>

```

Data

```

var life = [];

life.neighbors = [];

life.neighbors.add = function(x, y, v) {
  for (var i = x - 1; i <= x + 1; i++) {
    for (var j = y - 1; j <= y + 1; j++) {
      this[i * life.size + j] += v;
    }
  }
  this[x * life.size + y] -= v;
};

```

```

life.reset = function(source) {
  if (!arguments.length) {
    source = [];
    source.size = 150;
    for (var x = 0, p = 0; x < source.size; x++) {
      for (var y = 0; y < source.size; y++, p++) {
        source[p] = Math.random() > .5 ? 1 : 0;
      }
    }
  }
  life.size = source.size;
  for (var x = 0, p = 0; x < life.size; x++) {
    for (var y = 0; y < life.size; y++, p++) {
      life[p] = source[p];
      life.neighbors[p] = 0;
    }
  }
  for (var x = 0, p = 0; x < life.size; x++) {
    for (var y = 0; y < life.size; y++, p++) {
      if (life[p]) {
        life.neighbors.add(x, y, 1);
      }
    }
  }
};

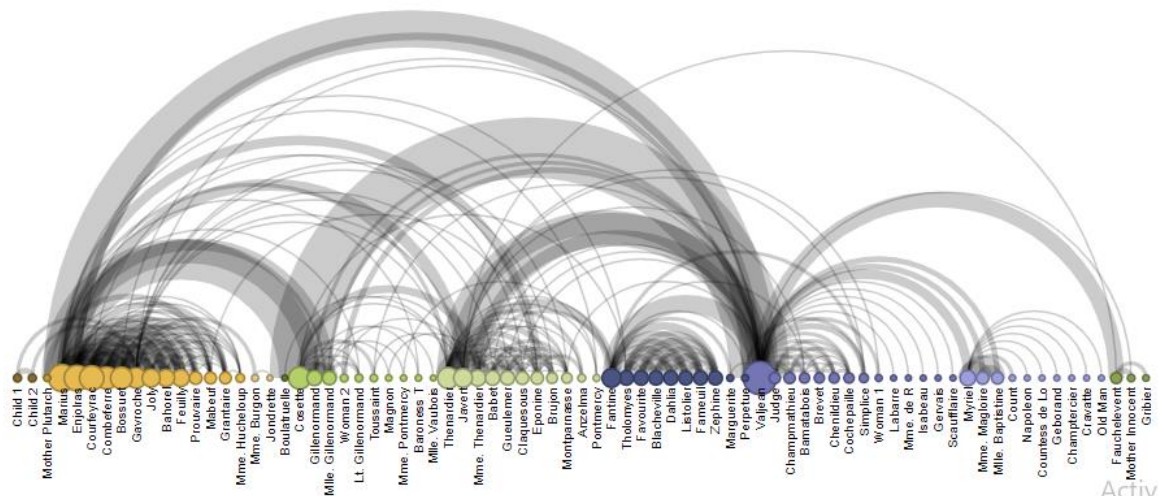
life.update = function() {
  var neighbors = this.neighbors.concat();
  for (var x = 0, p = 0; x < life.size; x++) {
    for (var y = 0; y < life.size; y++, p++) {
      if (this[p]) {
        if ((neighbors[p] < 2) || (neighbors[p] > 3)) {
          this.neighbors.add(x, y, -1);
          this[p] = 0;
        }
      } else if (neighbors[p] == 3) {
        this.neighbors.add(x, y, 1);
        this[p] = 1;
      }
    }
  }
}

```

```
}  
};
```

Arc Diagrams

Arc Diagrams



Activate \
Go to Setting

Source

```
<html>  
<head>  
  <title>Les Misérables Arc</title>  
  <link type="text/css" rel="stylesheet" href="ex.css?3.2"/>  
  <script type="text/javascript" src="../protovis-r3.2.js"></script>  
  <script type="text/javascript" src="miserables.js"></script>  
  <style type="text/css">
```

```
#fig {  
  width: 880px;  
  height: 400px;  
}
```

```
</style>  
</head>  
<body> <div id="center"> <div id="fig">
```

```
<script type="text/javascript+protovis">

var vis = new pv.Panel()
    .width(880)
    .height(310)
    .bottom(90);

var arc = vis.add(pv.Layout.Arc)
    .nodes(miserables.nodes)
    .links(miserables.links)
    .sort(function(a, b) a.group == b.group
        ? b.linkDegree - a.linkDegree
        : b.group - a.group);

arc.link.add(pv.Line);

arc.node.add(pv.Dot)
    .size(function(d) d.linkDegree + 4)
    .fillStyle(pv.Colors.category19().by(function(d) d.group))
    .strokeStyle(function() this.fillStyle().darker());

arc.label.add(pv.Label)

vis.render();

</script>
</div> </body>
</html>
```

Submitted by:

M.Subitcha

19BIT031