Game of Life due **tonight** at 11:00 PM

Fritter (phase 1) due next **Friday** at 11:00 PM
Overview

- Ajax recap
- Node.js & NPM
- Express & MVC
AJAX

- Asynchronous requests to server
  - Usually through HTTP requests
- Modify your current view / model
- Allows for enhanced UX
  - No need to reload page for updated content
Node.js

- It is not a web server. It can’t do anything on its own
- It is a JavaScript runtime using Google’s V8 engine
- Cross-platform
- Modular
- NPM (Node Package Manager)
  - dependency manager
  - handle multiple versions of same package
Example: reading from a file

```javascript
var fs = require('fs');

fs.readFile('input.txt','utf8', function (err, data) {
    if (err) throw err;
    console.log(data);
});

console.log('DONE!');
```
Example: simple web server

```javascript
var http = require('http');

http.createServer(function (req, res) {
    res.writeHead(200, {'Content-Type': 'text/plain'});
    res.end('Hello World
');
}).listen(3000);

console.log('Server listening on port 3000');
```

Example: npm, package.json

```json
> package.json
{
  "name": "application-name",
  "version": "0.0.1",
  "dependencies": {
    "moment": "^2.10.6"
  }
}
```

```bash
> npm install;
moment@2.10.6 node_modules/moment

> npm install moment --save

> npm install moment -g

> npm shrinkwrap
```
Model + View + Controller

- Separate presentation (view) from our data (model).
- Controller handles communication between the view and the model.
Express

- Minimalist framework with robust set of features
  - ~3700 LOC
- Most popular NPM package
- Demo
Express: Routes

- Responsible for mapping URL paths to views and controller

//Use module routes/users, a router() instance
var users = require('./routes/users');
app.use('/users', users);

//Use module routes/index, a router() instance
var index=require('./routes/index');
app.use('/', index);
Express: Controllers

- Communication bridge between views and models
- Controllers are triggered by the HTTP requests sent to your application.

```javascript
app.get('/users/:from-:to', function(req, res, next){
    var from = req.params.from;
    var to = req.params.to;
    var names = users.map(function(user){
        return user.name;
    });
    res.send('users ' + names.slice(from, to).join(', '));
});
```
Express: View

- Display content, must be invoked by the controller.

/*GET signup page*/
router.get('/signup', function(req,res){
    res.render('signup',{title:"Signup"});
});
Demo: Let’s build an app!