CodeHub

Curran Kelleher
8/18/2012

Programming is Overly Complex

- Development environment setup
- Revision control management
- Dependency management
- Deployment

= time and effort **learning tools**,
not writing code.

- The barrier to entry is higher than necessary
- This can be solved through automation

Simplification by Automation

Automation of what?

- Development environment setup
  - It's all on the server
  - Just open a Web page
- Revision control management
  - It's all on the server
  - Just click "save"
- Dependency management
  - It's all on the server
  - Just use "require(moduleName)'
- Deployment
  - Just click "run" and share that link

Prior Art in "Web-Based IDEs"

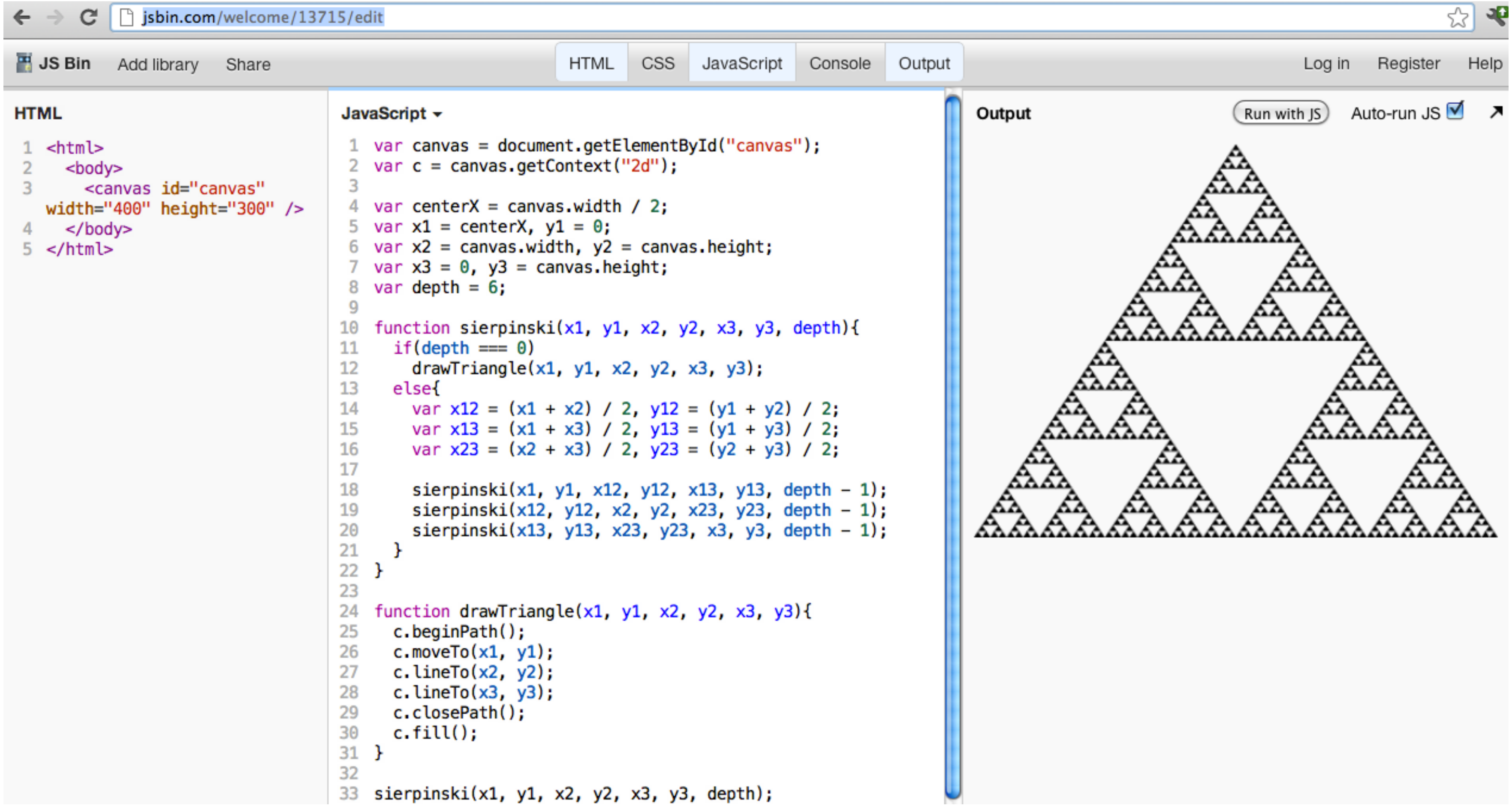
We consider a "Web-Based IDE" something that

- runs in a browser
- lets users edit, save, and run source code

The following are notable examples:

- JSBin
- JSFiddle
- CSSDesk
- Cloud9 IDE
- GitHub

Prior Art - JSBin

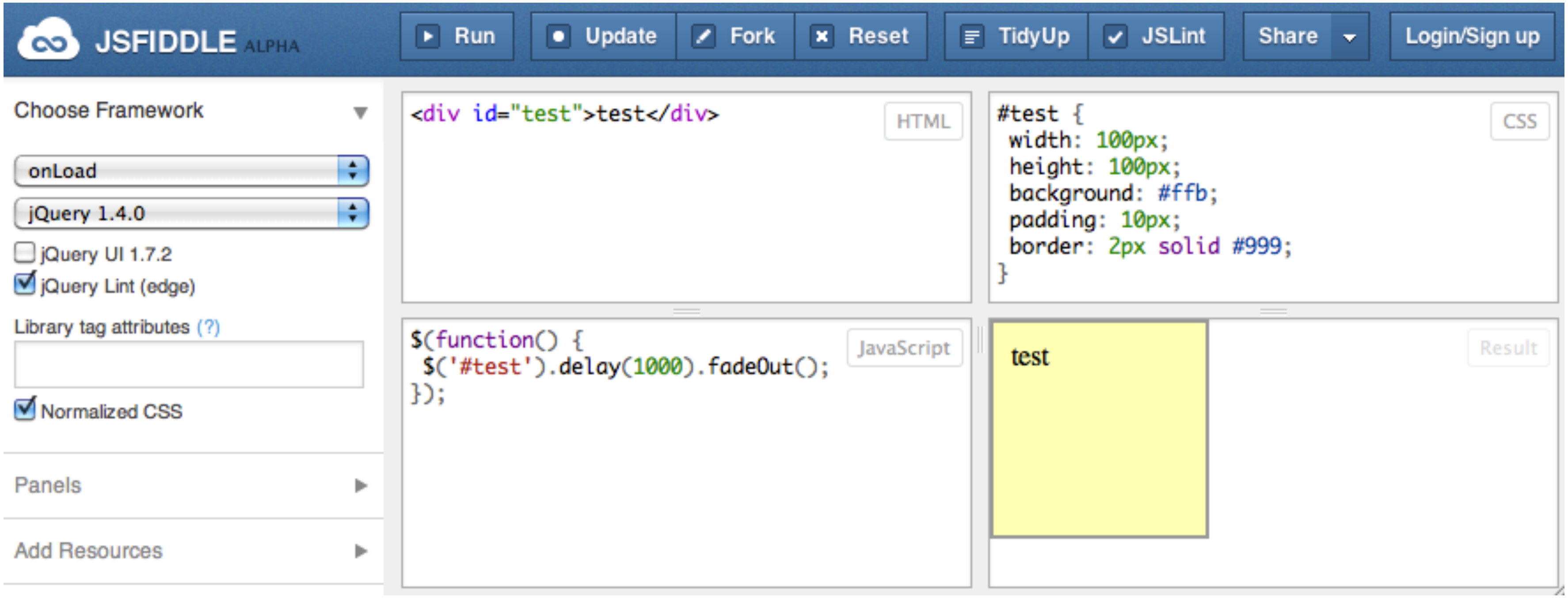


```html
<html>
  <body>
    <canvas id="canvas" width="400" height="300" />
  </body>
</html>
```

```javascript
var canvas = document.getElementById("canvas");
var c = canvas.getContext("2d");

var centerX = canvas.width / 2;
var x1 = centerX, y1 = 0;
var x2 = canvas.width, y2 = canvas.height;
var x3 = 0, y3 = canvas.height;
var depth = 6;

function sierpinski(x1, y1, x2, y2, x3, y3, depth){
  if(depth === 0)
    drawTriangle(x1, y1, x2, y2, x3, y3);
  else{
    var x12 = (x1 + x2) / 2, y12 = (y1 + y2) / 2;
    var x13 = (x1 + x3) / 2, y13 = (y1 + y3) / 2;
    var x23 = (x2 + x3) / 2, y23 = (y2 + y3) / 2;

    sierpinski(x1, y1, x12, y12, x13, y13, depth - 1);
    sierpinski(x12, y12, x2, y2, x23, y23, depth - 1);
    sierpinski(x13, y13, x23, y23, x3, y3, depth - 1);
  }
}

function drawTriangle(x1, y1, x2, y2, x3, y3){
  c.beginPath();
  c.moveTo(x1, y1);
  c.lineTo(x2, y2);
  c.lineTo(x3, y3);
  c.closePath();
  c.fill();
}

sierpinski(x1, y1, x2, y2, x3, y3, depth);
```

code, run, save, deploy - in the browser try it!

Prior Art - JSFiddle



onLoad

jQuery 1.4.0

□ jQuery UI 1.7.2

☑ jQuery Lint (edge)

Library tag attributes (?)

☑ Normalized CSS

```html
<div id="test">test</div>
```

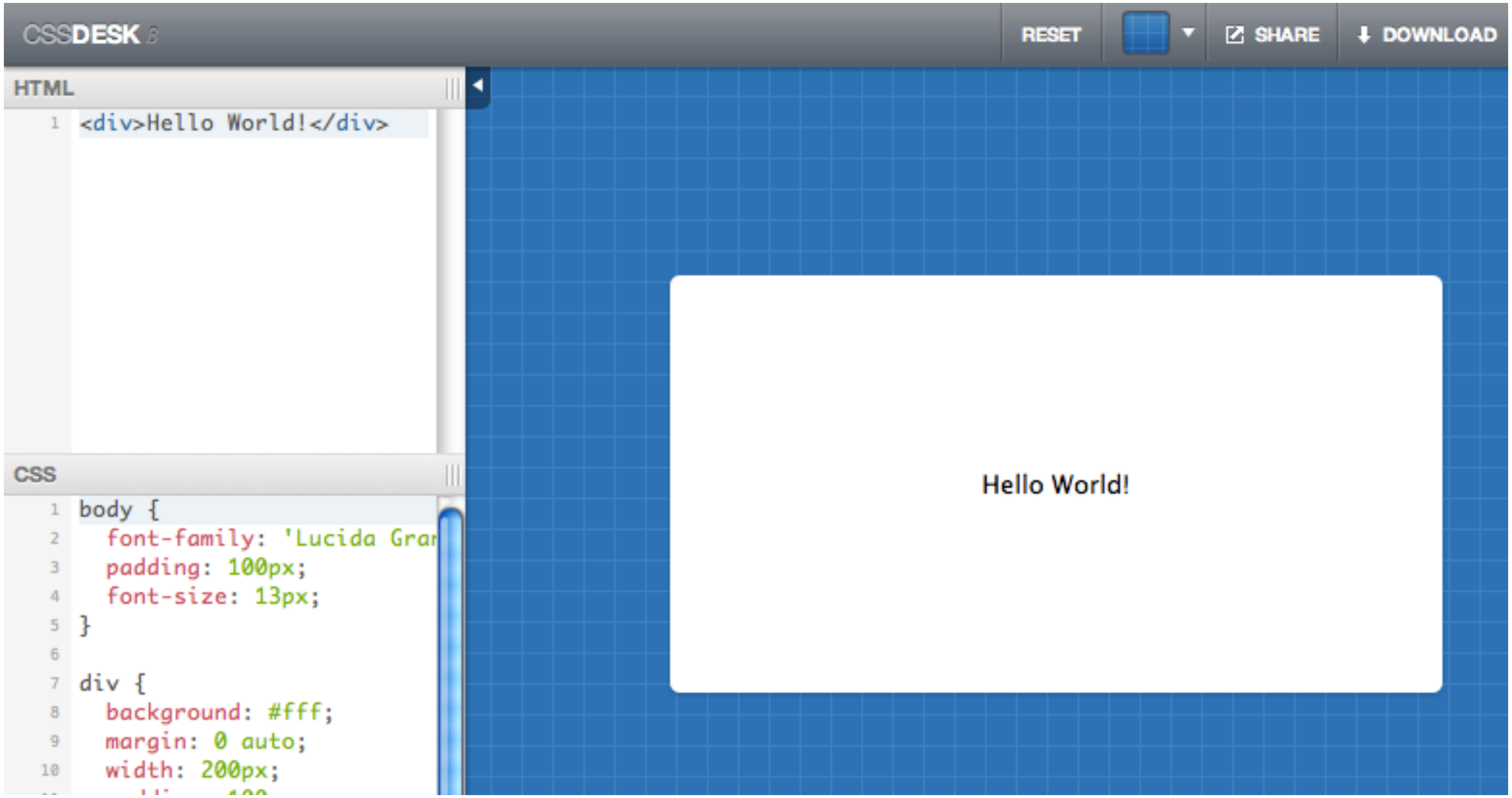
```css
#test {
 width: 100px;
 height: 100px;
 background: #ffb;
 padding: 10px;
 border: 2px solid #999;
}
```

```javascript
$(function() {
 $('#test').delay(1000).fadeOut();
});
```

test

code, run, save, deploy - in the browser

Prior Art - CSSDesk



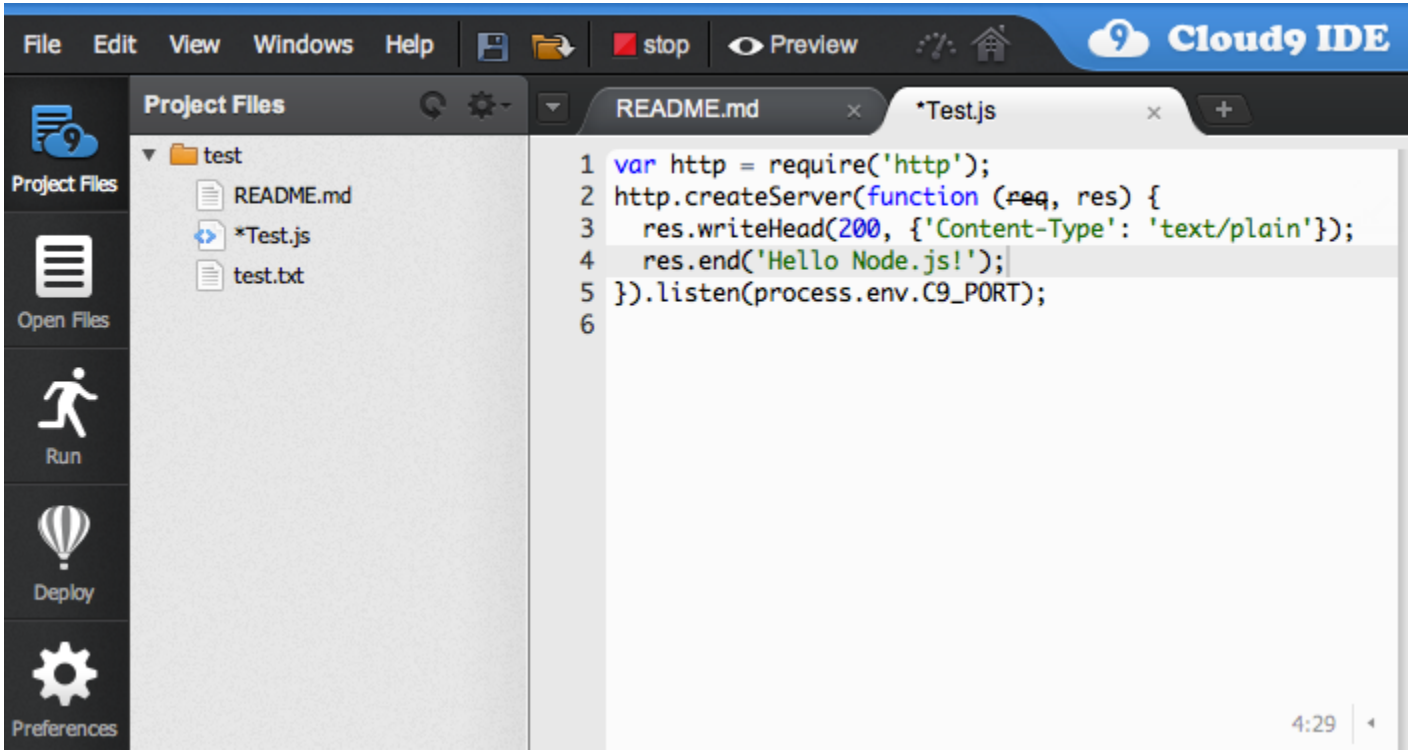
```
<div>Hello World!</div>
```

```
body {
  font-family: 'Lucida Gran
  padding: 100px;
  font-size: 13px;
}

div {
  background: #fff;
  margin: 0 auto;
  width: 200px;
```

Hello World!

code, run, save, deploy (CSS) - in the browser

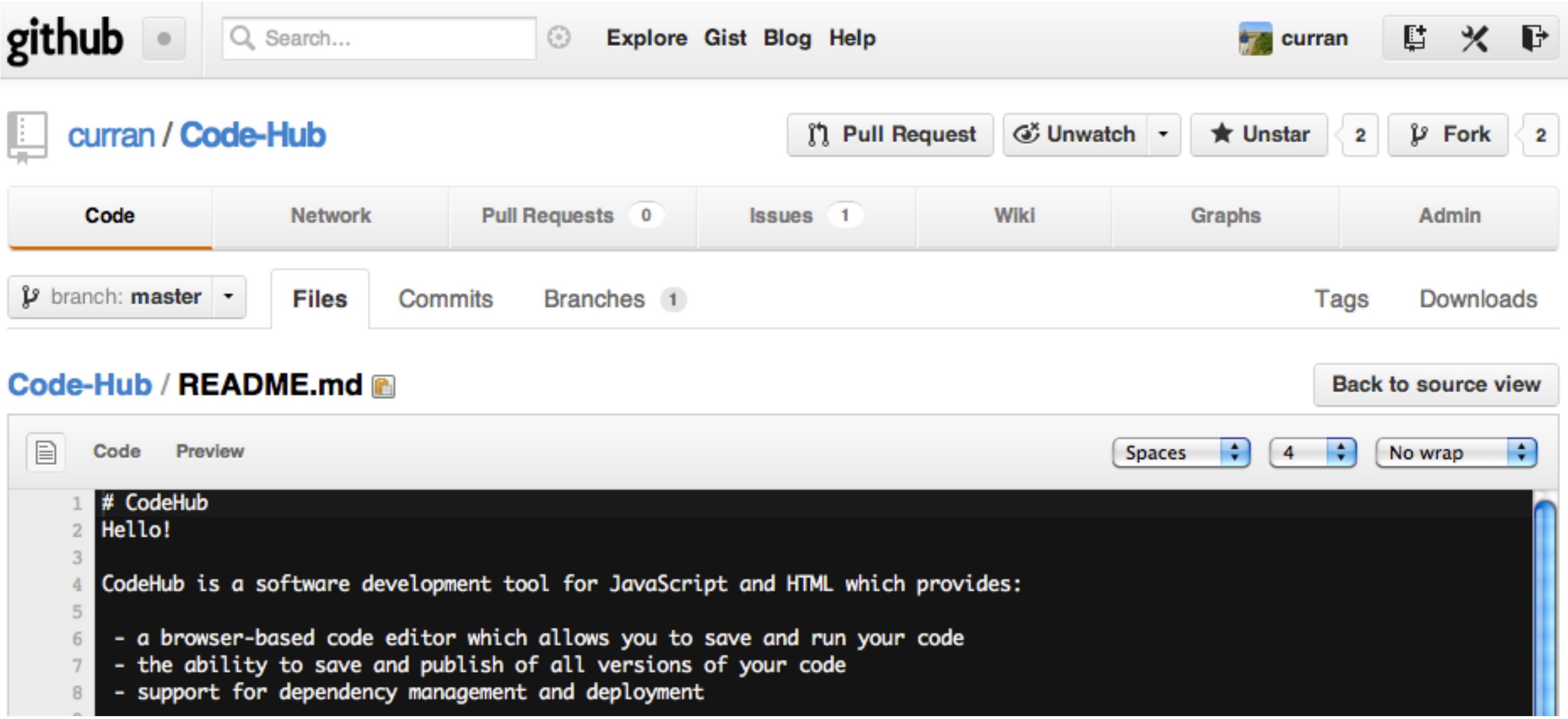
Prior Art - Cloud9 IDE



```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello Node.js!');
}).listen(process.env.C9_PORT);
```

- Full featured IDE, runs Node.js code
- Integrates with GitHub and Heroku
- Integrated terminal for Git and Unix commands

Prior Art - GitHub



```
# CodeHub
Hello!

CodeHub is a software development tool for JavaScript and HTML which provides:

 - a browser-based code editor which allows you to save and run your code
 - the ability to save and publish of all versions of your code
 - support for dependency management and deployment
```

- An in-browser text file editor for Git repositories
- Can be coupled with "GitHub Pages", a service that serves GitHub repositories as Web sites

Prior Art in Software Repositories

Developers can publish reusable packages with support for dependency management

- The Maven Repository (for Java)
- RubyGems (for Ruby)
- The Node Package Manager (for Node.js)
- The CommonJS module specification
- Asynchronous Module Definition (AMD)
- The Require.js AMD module loader
- ...many more out there
- The point: *packages and dependency management empower the platforms*

The CodeHub Vision

No existing tools have all of these features:

- Web-based source code editing
- Definition and use of reusable modules
- Web-based deployment, with
- Automated dependency management

Such a tool would support

- Computer science education
- a public Wikipedia-like software repository
- a research testbed for interactive graphics

The CodeHub Architecture

- Limited to JavaScript and HTML software
- Based on CommonJS modules
- All versions are published
    - So when an application is linked to or embedded, its behavior does not change (or break) over time
- The server tracks
    - Script content, for all versions
    - The dependency graph, for all versions
- Scripts can be run
    - At runtime, dependencies are evaluated and bundled together into a single page
    - Compilation strategy from a CommonJS Wiki Page

CodeHub Scripts

- Every piece of code stored is a "script"
- Each script has an id number
- Each script version has a revision number
- Saving a script creates a new revision
- All script revisions are published
  - For example
    http://code-hub.org/edit/7.1
    edits the script whose id is 7, revision number 1
- There are three types of scripts:
  - Modules
  - Templates
  - Applications

CodeHub Modules

- CodeHub supports CommonJS Modules
- Each module must have a unique name
- Modules can be required with the syntax `foo = require('foo')`
- Modules can be defined with the syntax `@module foo`
- Modules define their exported API by adding properties to an `exports` object

CodeHub Templates

- Templates are HTML pages with placeholders of the form `${parameterName}`
- Parameter values are passed in from applications that use the template
- `${scripts}`gets replaced by script tags including application source code
    - dependencies are bundled together and included also
- Each template has a unique name
- Templates can be defined with the syntax `@template templateName`

CodeHub Applications

- Applications are scripts that can be run
- Applications can depend on modules
- Applications are defined using the syntax

  ```
  @app template templateName
  ```

- Applications can pass arbitrary parameters to the template using the syntax

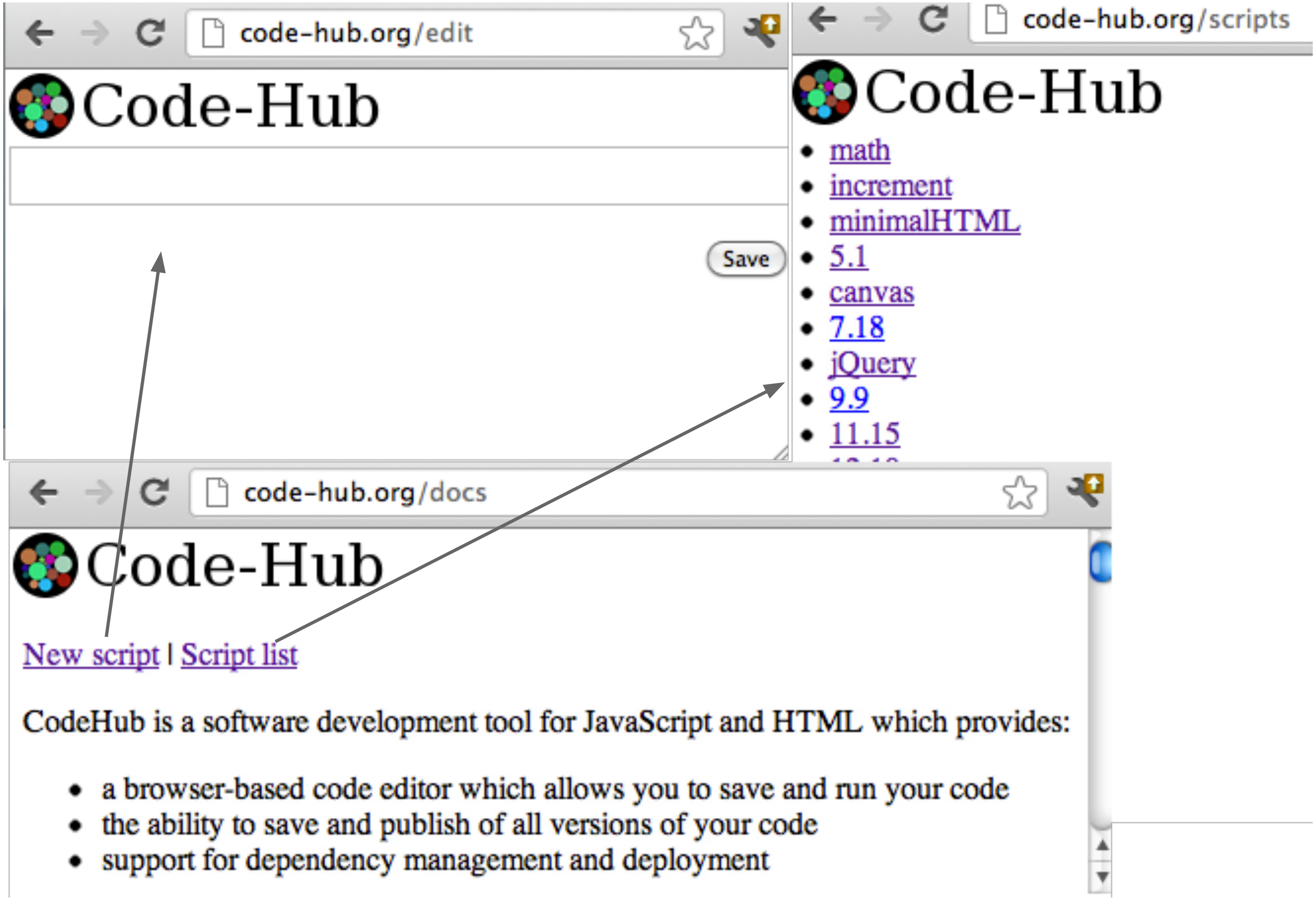
  ```
  @app parameterName value
  ```

CodeHub Implementation

CodeHub was implemented using

- Node.js as a server platform
- The Express.js Web Framework
  - With Jade and Markdown templates
- MongoDB via the Mongoose API
- Git via the Node.js child process API
- Hosted on the Rackspace cloud
  - In a single Ubuntu server VM
- Live now at code-hub.org

CodeHub Screenshots: New Script and Script List



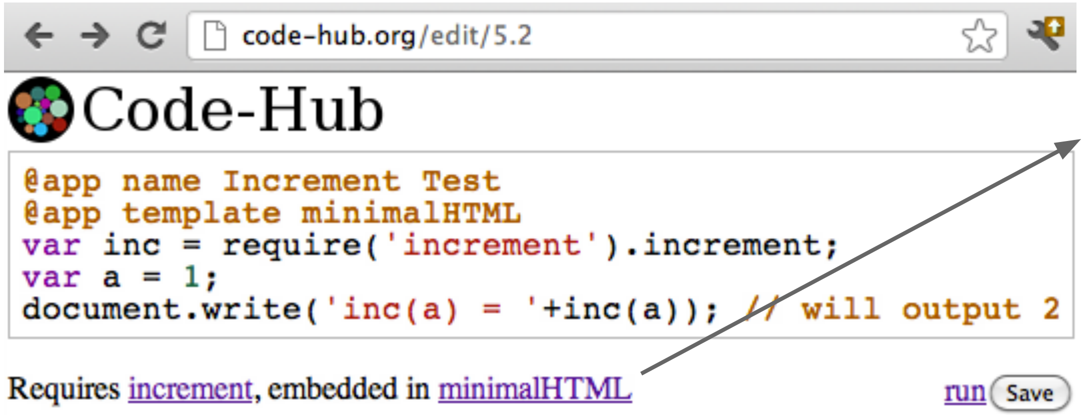
- math
- increment
- minimalHTML
- 5.1
- canvas
- 7.18
- jQuery
- 9.9
- 11.15

New script | Script list

CodeHub is a software development tool for JavaScript and HTML which provides:

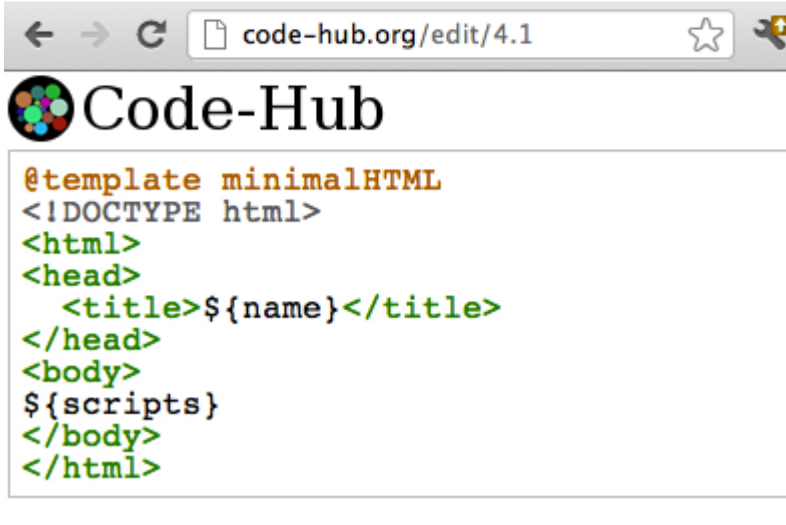
- a browser-based code editor which allows you to save and run your code
- the ability to save and publish of all versions of your code
- support for dependency management and deployment

CodeHub Screenshots: Example Scripts

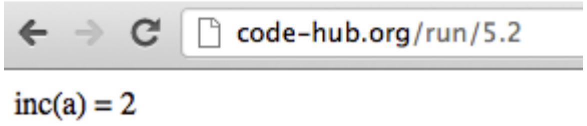


```
@app name Increment Test
@app template minimalHTML
var inc = require('increment').increment;
var a = 1;
document.write('inc(a) = '+inc(a)); // will output 2
```

Requires increment, embedded in minimalHTML

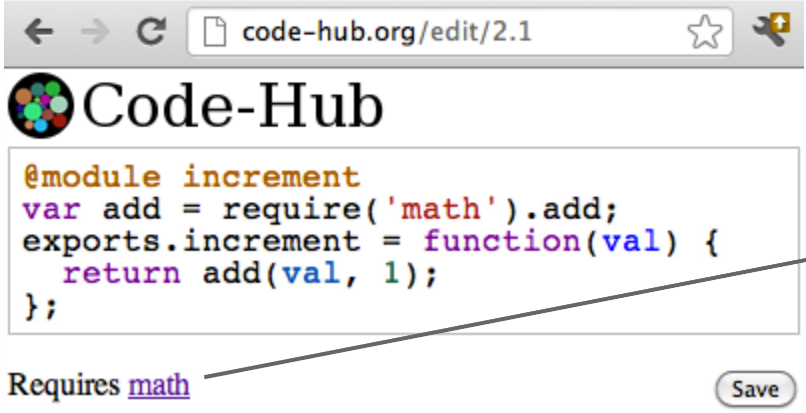


```
@template minimalHTML
<!DOCTYPE html>
<html>
<head>
  <title>${name}</title>
</head>
<body>
${scripts}
</body>
</html>
```



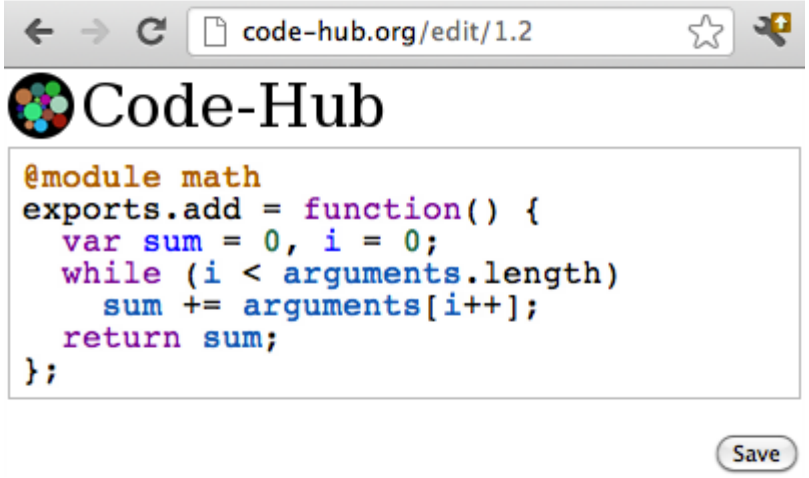
```
inc(a) = 2
```

Try it!



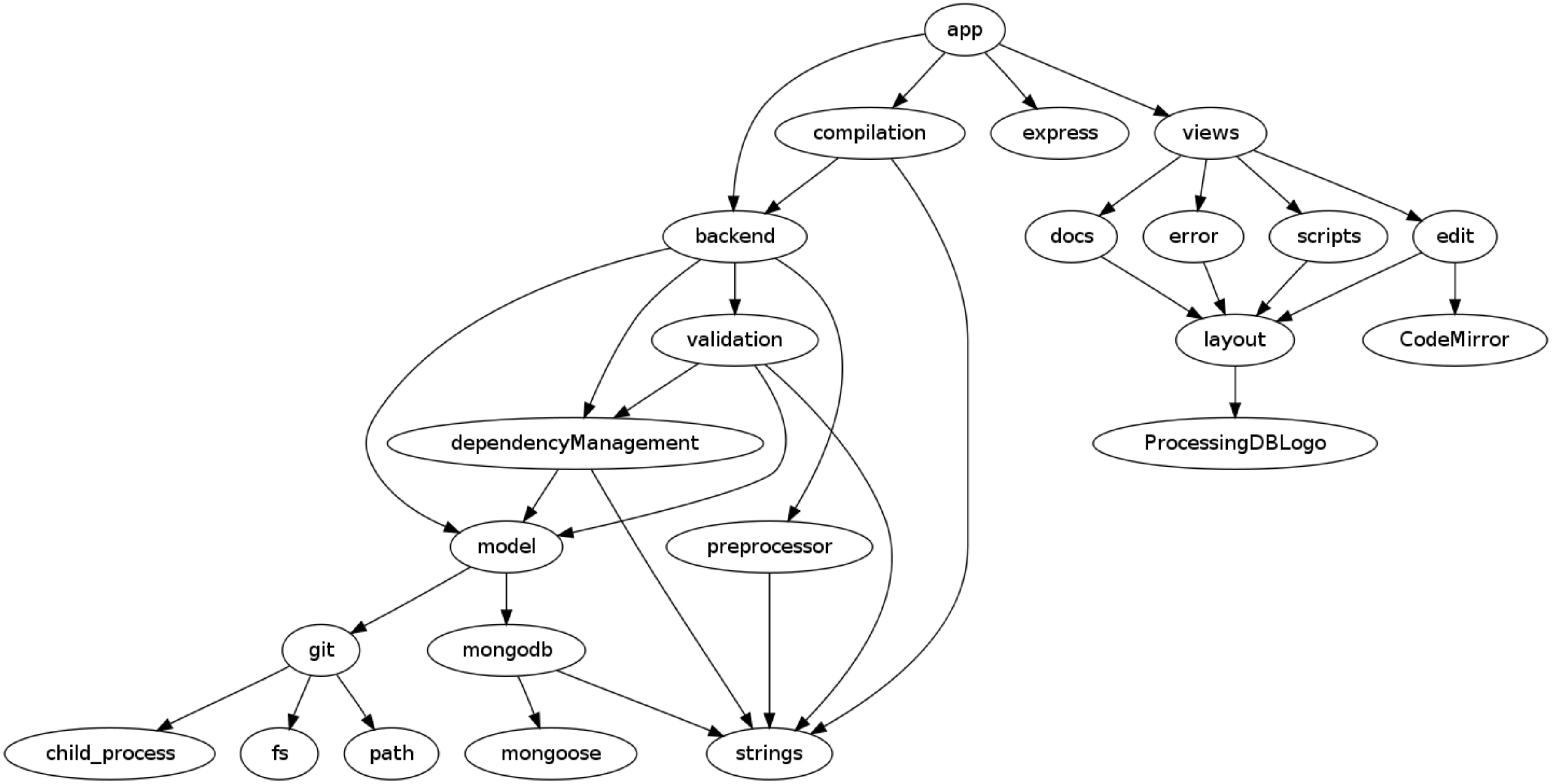
```
@module increment
var add = require('math').add;
exports.increment = function(val) {
  return add(val, 1);
};
```

Requires math



```
@module math
exports.add = function() {
  var sum = 0, i = 0;
  while (i < arguments.length)
    sum += arguments[i++];
  return sum;
};
```

CodeHub Implementation Modules



Case Study:
An Interactive Graphics Course

- Web-based IDEs were used in a course
  - "Computer Programming and Interactive Graphics"
  - a 50-hour summer course for high school students
  - of the MIT Junction program, July-August 2012
- The class blog contains
  - Links to the "edit" pages in CodeHub
  - Embedded programs from CodeHub using iFrames
  - Comment sections for students to post their work
- The class used
  - JSBin
  - CodeHub
  - Cloud9 IDE
  - GitHub and GitHub Pages

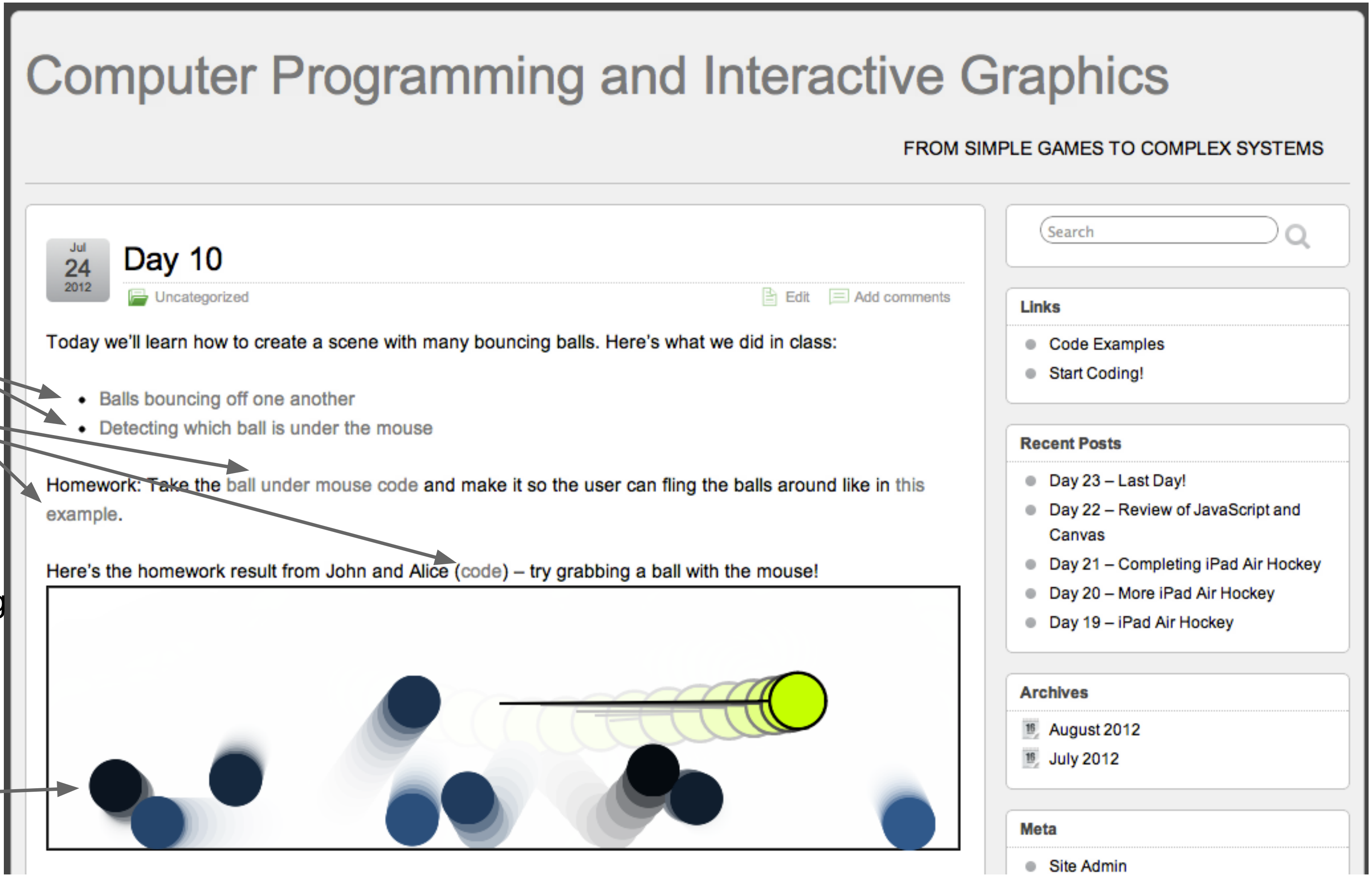
Case Study:
An Interactive Graphics Course

- Students were first exposed to JSBin
  - Basic JavaScript and HTML5 Canvas features were introduced in "code as I code" fashion using JSBin
  - JSBin's "auto-run" feature provided instant feedback
- Students were then exposed to CodeHub
  - ..but did not use the module functionality
    - as it was beyond their knowledge
  - ..but preferred JSBin because
    - The editor and output are on the same page
    - The code is re-run automatically when changed
- Some students liked
  - the simplicity of CodeHub
  - that CodeHub adds nothing extra when running
    - whereas JSBin adds an "edit in JSBin" button

Case Study:
An Interactive Graphics Course

- Students were then exposed to GitHub
    - Students created GitHub accounts and a repository
    - Students used the in-browser GitHub editor
        - just for learning basic Markdown
    - Students set up Web sites using GitHub Pages
- Students were then exposed to Cloud9 IDE
    - Students learned the basics of Git, and merging
    - Students used Cloud9 IDE to
        - construct an interactive graphical program
        - publish it to the Web using GitHub Pages
- The overall response was fear and dislike
    - Students perceived GitHub and Cloud9 IDE as overly complex and cumbersome to use
    - JSBin was their most preferred tool overall

Case Study:
An Interactive Graphics Course Blog



Computer Programming and Interactive Graphics

FROM SIMPLE GAMES TO COMPLEX SYSTEMS

Jul 24 2012

Day 10

Uncategorized Edit Add comments

Today we'll learn how to create a scene with many bouncing balls. Here's what we did in class:

- Balls bouncing off one another
- Detecting which ball is under the mouse

Homework: Take the ball under mouse code and make it so the user can fling the balls around like in this example.

Here's the homework result from John and Alice (code) – try grabbing a ball with the mouse!

Links

- Code Examples
- Start Coding!

Recent Posts

- Day 23 – Last Day!
- Day 22 – Review of JavaScript and Canvas
- Day 21 – Completing iPad Air Hockey
- Day 20 – More iPad Air Hockey
- Day 19 – iPad Air Hockey

Archives

- August 2012
- July 2012

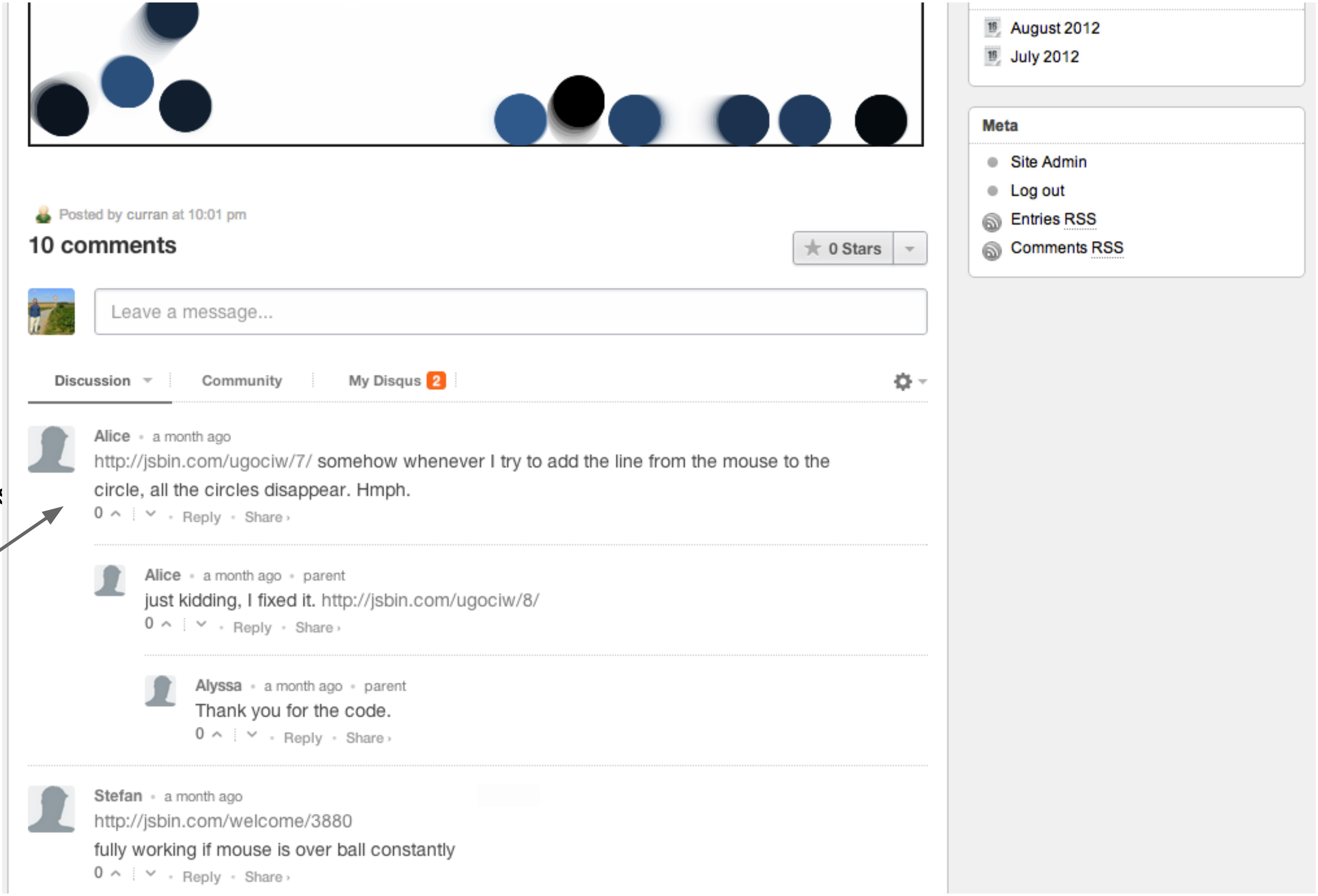
Meta

- Site Admin

Links to source code in CodeHub

The running program embedded directly from CodeHub using an iFrame

Case Study:
An Interactive Graphics Course Blog



August 2012

July 2012

Meta

- Site Admin
- Log out
- Entries RSS
- Comments RSS

Posted by curran at 10:01 pm

10 comments

★ 0 Stars

Leave a message...

Discussion | Community | My Disqus 2

Alice · a month ago

http://jsbin.com/ugociw/7/ somehow whenever I try to add the line from the mouse to the circle, all the circles disappear. Hmph.

0 · Reply · Share ›

Alice · a month ago · parent

just kidding, I fixed it. http://jsbin.com/ugociw/8/

0 · Reply · Share ›

Alyssa · a month ago · parent

Thank you for the code.

0 · Reply · Share ›

Stefan · a month ago

http://jsbin.com/welcome/3880

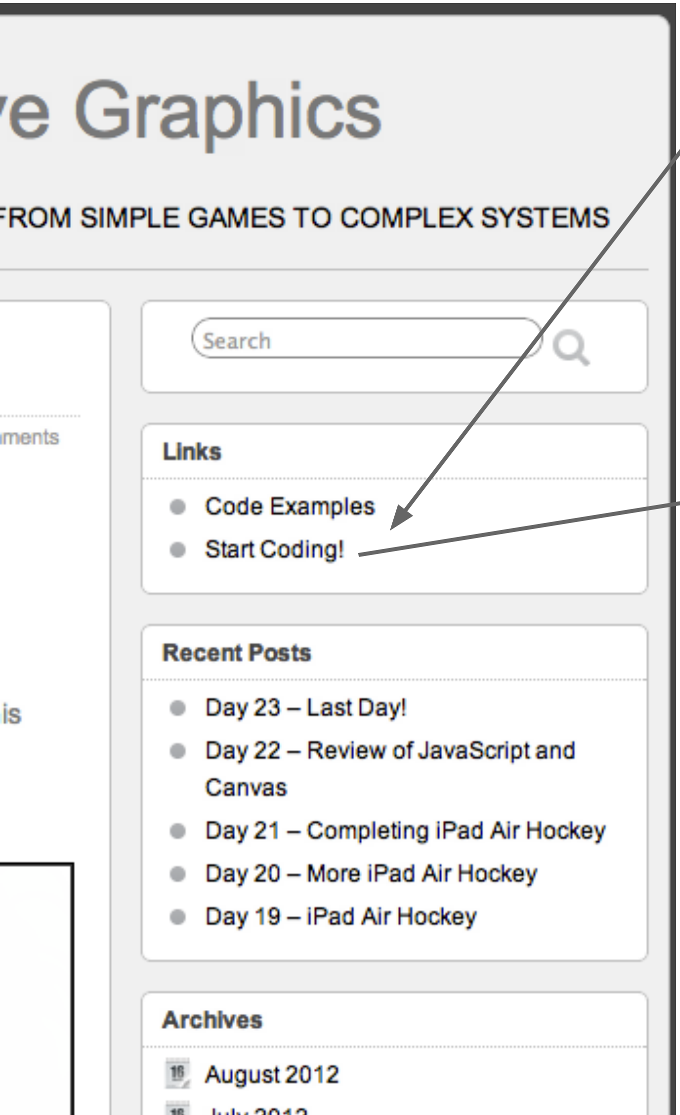
fully working if mouse is over ball constantly

0 · Reply · Share ›

Comments allowed students to showcase their work and collaborate

Case Study:
An Interactive Graphics Course Blog

A starting point for graphics programming was provided.



ve Graphics

FROM SIMPLE GAMES TO COMPLEX SYSTEMS

Links

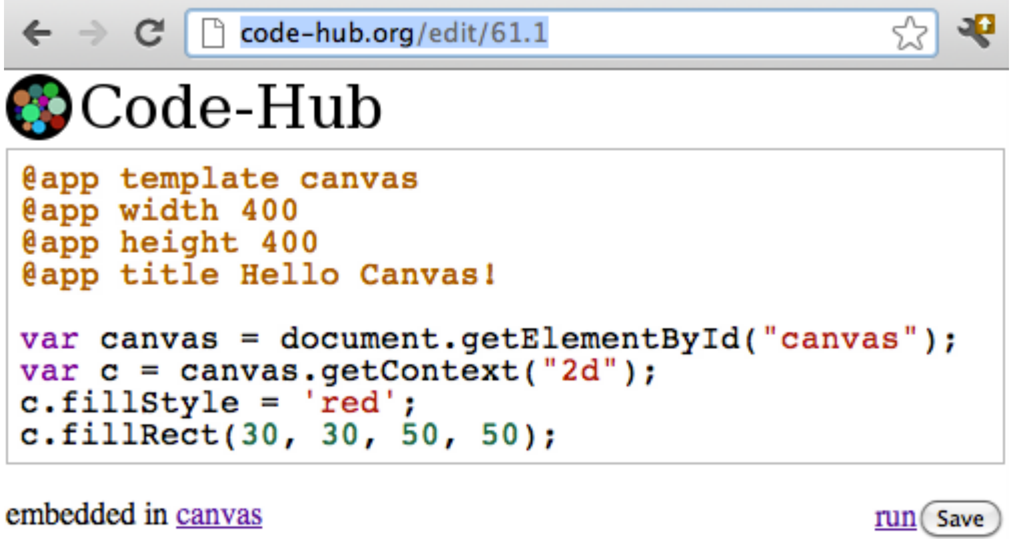
- Code Examples
- Start Coding!

Recent Posts

- Day 23 – Last Day!
- Day 22 – Review of JavaScript and Canvas
- Day 21 – Completing iPad Air Hockey
- Day 20 – More iPad Air Hockey
- Day 19 – iPad Air Hockey

Archives

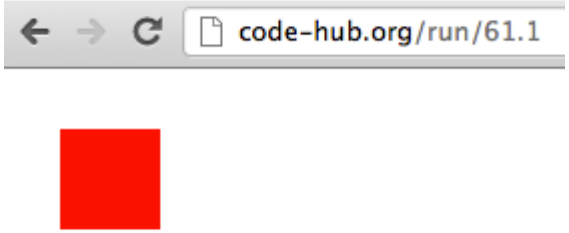
- August 2012
- July 2012



code-hub.org/edit/61.1

Code-Hub

```
@app template canvas
@app width 400
@app height 400
@app title Hello Canvas!

var canvas = document.getElementById("canvas");
var c = canvas.getContext("2d");
c.fillStyle = 'red';
c.fillRect(30, 30, 50, 50);
```



code-hub.org/run/61.1

Shortcomings

- Apps do not have names
  - So the list of scripts includes things like "12.4"
  - Perhaps apps should have names
- Modules are in a single global namespace
  - So modules for specific applications might use "appName.moduleName" for the module name
- Modules are editable by anyone
  - Therefore breakable by anyone
  - No built-in concept of ownership or authorship
- Trumped by JSBin in terms of usability
  - CodeHub needs the "Auto-run JS" feature

Future Work

- A "showcase" feature
  - Apps can be showcased on their own page
  - Including a comments section
  - This would enable discussion-based collaboration
- The "Auto-run JS" feature from JSBin
- Visualization of dependencies and apps
  - The content of CodeHub is difficult to navigate
  - A node-link visualization of the dependency graph would be useful for navigation
- A documentation editor
  - Each module could have documentation
  - This would make CodeHub more usable
- Implementation of Information Visualization software within CodeHub

The End