Developing a JavaScript Game Engine

Who's this geek?

HTML

gradient
studios, inc

Michael "Z" Goddard

@ZofGames

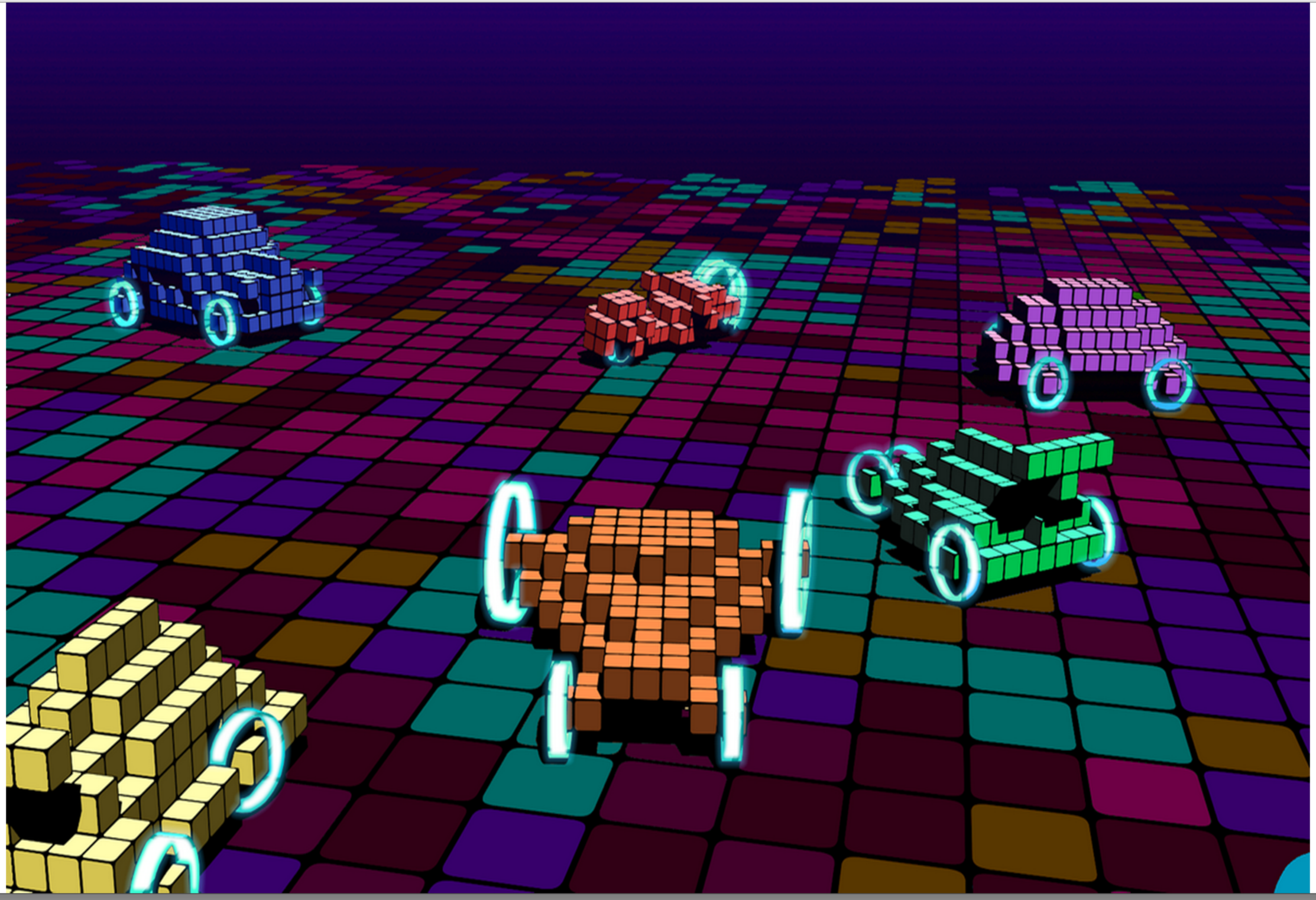
#JSGameEngine

What have I been working on?

Fieldrunners

© Subatomic Studios

Dawn Glider



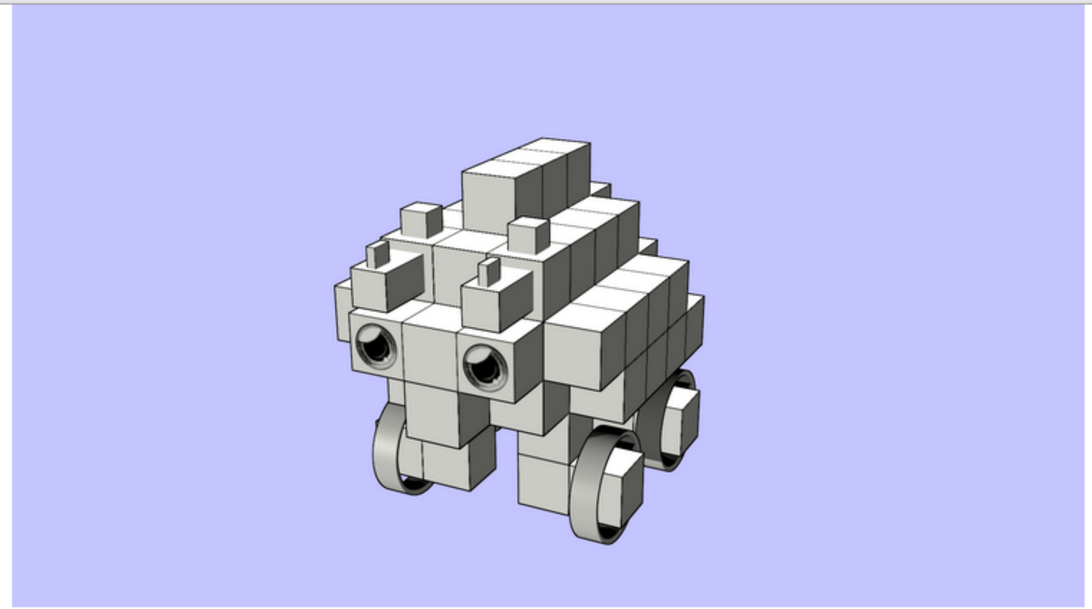
Talking about

- Component Based Architecture fits and works with declarative design using JSON documents
- Promises, a design pattern for one time events
- Dealing with scaling HTTP is important
- The Browser comes with many builtin tools for User Interface

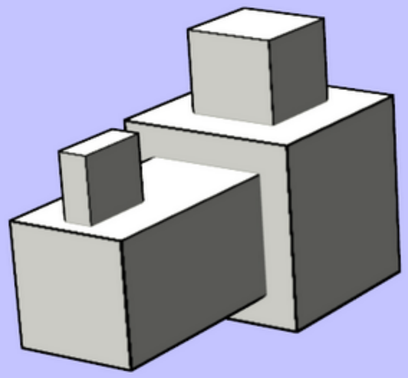
Component Based Architecture

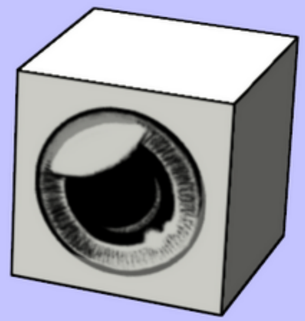
- Game
- Service
- Entity
- Component

It's like an Automobile.



It's like an Automobile.





Game

```
1  Game.prototype.add( entity /* Entity */ )
2  Game.prototype.emit( name /* String */, ... )
3  Game.prototype.main()
```

Entity

```
1  Entity.prototype.add( component /* Component */ )
2  Entity.prototype.destroy()
3  Entity.prototype.emit( name /* String */, ... )
4
5  Entity.prototype.get(
6    componentType /* Object */
7  ) /* Component */
8
9  Entity.prototype.get(
10   componentType /* Object */,
11   callback /* function( /* Component ) */
12 )
13
14 Entity.prototype.getAll(
15   componentType /* Object */
16 ) /* [ Component, ... ] */
```

Component

```
Component.prototype.destroy()

// Events
add
destroy
update
lateUpdate
fixedUpdate
lateFixedUpdate
collisionEnter
collisionExit
collisionStay
preDraw
draw
postDraw
```

Service

```
Service.prototype.destroy()

// execute actions related to this service,
// generally emitting events for entities
// and their components
Service.prototype.task()
```

Build A Driving Game

```
carGame = lib.game({
  services: [
    lib.input.service(),
    lib.network.service(),
    lib.physicsService( { gravity: [ 0, -5, 0 ]
    lib.graphicsService( { canvas: '#draw-here'
  ]
});
```

My Car

```
myCar = lib.entity({
  components: [
    lib.transform(),
    lib.input({
      accelerate: [ 'w', 'up' ],
      brake: [ 's', 'down' ],
      left: [ 'a', 'left' ],
      right: [ 'd', 'right' ]
    }),
    lib.network.view({
      watch: ['tranform', 'input']
    }),
    lib.body( { mass: 5 } ),
    lib.box( { size: [1, 1, 1] } ),
    lib.renderer({
      texture: 'crate.png',
      shader: 'glossy',
      color: 'white'
    }),
    lib.drive()
  ]
});
```

Add and Run

```
1  carGame.add(lib.myWorldObject());
2  carGame.add(myCar);
3  carGame.main();
```

Transforms

Position, Rotation, Scaling

Central point of communication

Dependent for other components

Inputs

Poll from Service

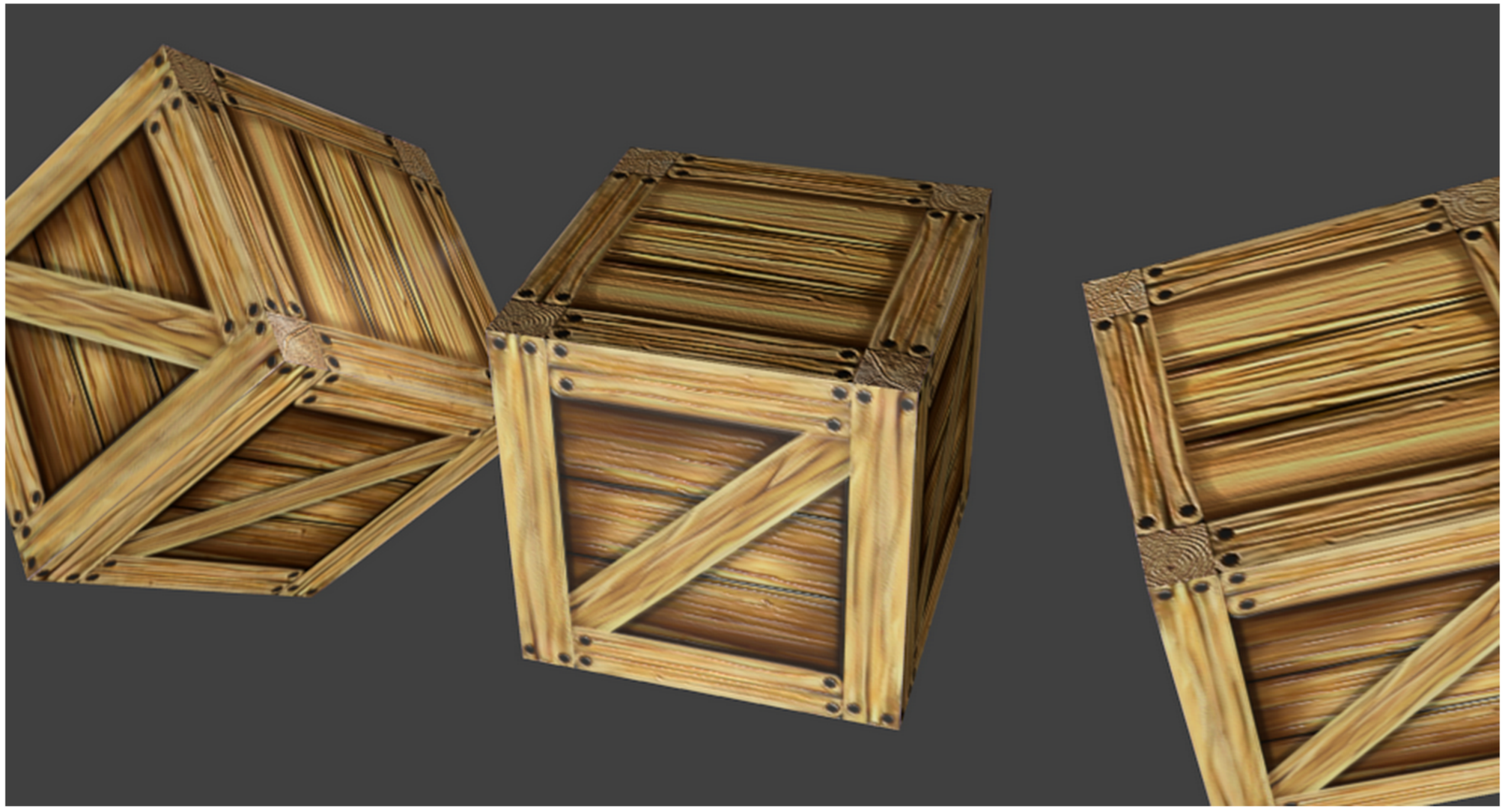
Network Playback

File Playerback

AI Generated

Renderer

How should it render? What color should it be?
Textured like a wooden or metal box?



Components in the Browser

Declaring an Entity

```
{
  "components": [
    {
      "type": "transform",
    },
    {
      "type": "body",
      "mass": 5
    },
    {
      "type": "box",
      "size": [ 1, 1, 1 ]
    },
    {
      "type": "renderer",
      "shader": "textured",
      "texture": "watcar.png"
    },
    {
      "type": "drive"
    }
  ]
}
```

Events

DOM

The browser Document Object Model (DOM) relys heavily on events.

- Mouse and Keyboard Events
  - Clicks
  - Presses
- Window Events
  - Resizing
  - Focus and Blur
- Loading Events
  - Progress
  - Loaded
  - Errors

Load an Image

```
function loadImage(options) {
  var image = new Image();

  image.addEventListener('load', function(event) {
    options.onload(image); // draw the image somewhere
  });

  image.addEventListener('error', function(event) {
    options.onerror(image); // oops, we can try reloading a few times
  });

  // image.addEventListener('progress', function(event) {
  //   options.onprogess(event.loaded / event.total);
  // });

  image.src = options.path;

  return {
    target: image
  };
}
```

Load a Bunch of Images (for an Animation)

```
1  function loadBunch(bunch, options) {
2    var loadCount = bunch.length;
3
4    bunch.forEach(function(item) {
5      item.target.addEventListener('load', handler);
6    });
7
8    function handler() {
9      if (--loadCount <= 0) {
10       options.onload(bunch);
11     }
12   }
13
14   return bunch;
15 }
```

Load a Bunch of Animations

```
1  function loadBunch(bunch, options) {
2    var loadCount = bunch.length;
3
4    bunch.forEach(function(item) {
5      if (item.target.addEventListener) {
6        item.target.addEventListener('load', handler);
7      } else {
8        loadBunch(item.target, {
9          onload: handler
10       });
11     }
12   });
13
14   function handler() {
15     if (--loadCount <= 0) {
16       options.onload(bunch);
17     }
18   }
19
20   return bunch;
21 });
```

It's Complicated.

That handles loading well ...

But what about tracking progress?

Promises

- http://en.wikipedia.org/wiki/Promise_(programming)

The term "promise" was proposed in 1976 by Daniel P. Friedman and David Wise,[1] and Peter Hibbard called it "eventual".

Where it can help.

Loading

WebWorkers (and NaCl)

WebGL Context Creation

Audio and Video playback completion

Something UMD-y

Resolver

```
Resolver.prototype.resolve( ... ) /* Resolver */
Resolver.prototype.reject( ... ) /* Resolver */
```

Promise

```
Promise.prototype.then(
  resolve /* function */,
  reject /* function */
) /* Promise */
```

If it quacks like a Duck

Deferred Implements Promise and Resolver

```
1  function Deferred() {
2    this.resolver /* Resolver */
3    this.promise /* Promise */
4  }
5
6  Deferred.prototype.resolve( ... )/* Deferred */
7  Deferred.prototype.reject( ... ) /* Deferred */
8  Deferred.prototype.then(
9    resolve /* function */,
10   reject /* function */
11 ) /* Deferred */
```

Possible Additions

Progress

```
1  Resolver.prototype.progress( {
2    target /* Object */, // optional
3    percent /* Number */, // required
4    amount /* Number */, // optional
5    total /* Number */, // optional
6  } ) /* Resolver */
7
8  Promise.prototype.then(
9    resolve /* function */,
10   reject /* function */,
11   progress /* function */
12 ) /* Promise */
```

Loading an Image

```
1   function loadImage( options ) {
2     var defer = new Deferred(),
3         image = new Image();
4
5     image.addEventListener( 'load', function() {
6       defer.resolve( image );
7     });
8
9     return defer.promise;
10  }
```

Loading an Animation (or a bunch of Animations)

```
function loadAll( bunchOfPromises ) {
  var defer = defer(),
      objects = [],
      count = bunchOfPromises.length;

  bunchOfPromises.forEach( function( promise ) {
    promise.then( handler );
  });

  function handler( object ) {
    objects.push( object );
    if ( count == objects.length ) {
      defer.resolve.apply( defer, objects );
    }
  }

  return defer.promise;
}
```

```
function loadBunch(bunch, options) {
  var loadCount = bunch.length;

  bunch.forEach(function(item) {
    if (item.target.addEventListener) {
      item.target.addEventListener('load', handler);
    } else {
      loadBunch(item.target, {
        onload: handler
      });
    }
  });

  function handler() {
    if (--loadCount <= 0) {
      options.onload(bunch);
    }
  }

  return bunch;
}
```

```
function loadAll( bunchOfPromises ) {
  var defer = defer(),
      objects = [],
      count = bunchOfPromises.length;

  bunchOfPromises.forEach( function( promise ) {
    promise.then( handler );
  });

  function handler( object ) {
    objects.push( object );
    if ( count == objects.length ) {
      defer.resolve.apply( defer, objects );
    }
  }

  return defer.promise;
}
```

Promises in the Wild

- when [https://github.com/cujojs/when]
  Library covering purely Promises pulled from another project.
- jQuery [jquery.com]
  Widely popular library for DOM operations
  including promises for responding to asynchronous operations

Loading Content

Grouping Files

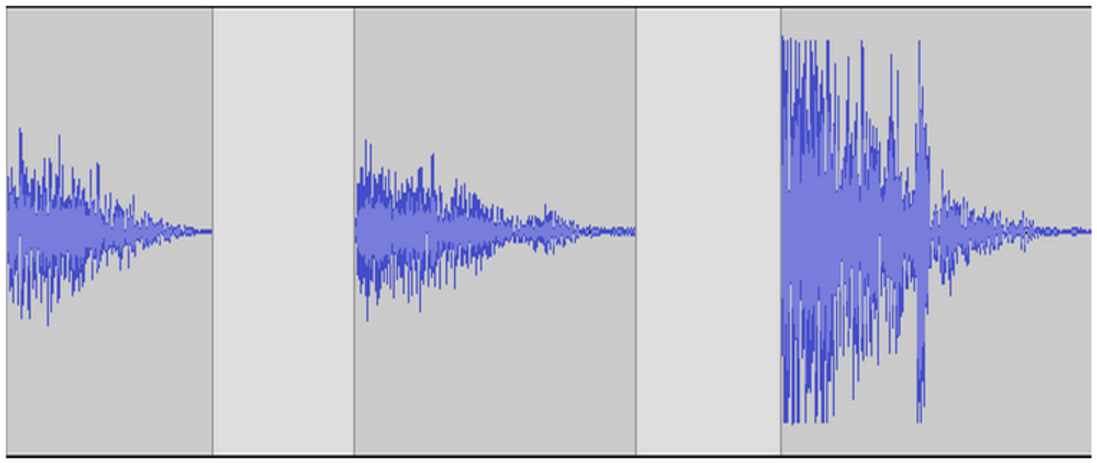
HTTP is pretty tolerant ...
But servers can only serve so many users simultaneously.

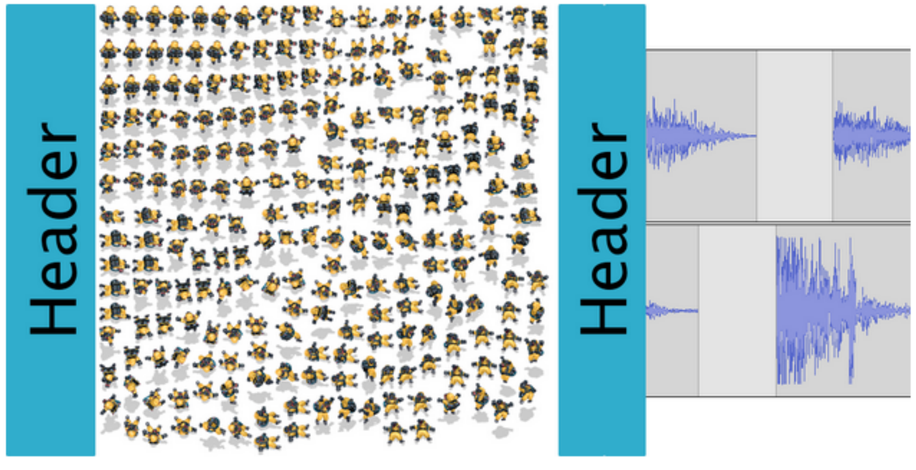
> Download individual files?

No

> Load

- Sprite Sheets
- Sound Sprites
- Tar Files





Tar Files?

Field Offset	Field Size	Field
0	100	File name
100	8	File mode
108	8	Owner's numeric user ID
116	8	Group's numeric user ID
124	12	File size in bytes
136	12	Last modification time in numeric Unix time format
148	8	Checksum for header block
156	1	Link indicator (file type)
157	100	Name of linked file

Tar Alternative

Concatenated files with JSON headers

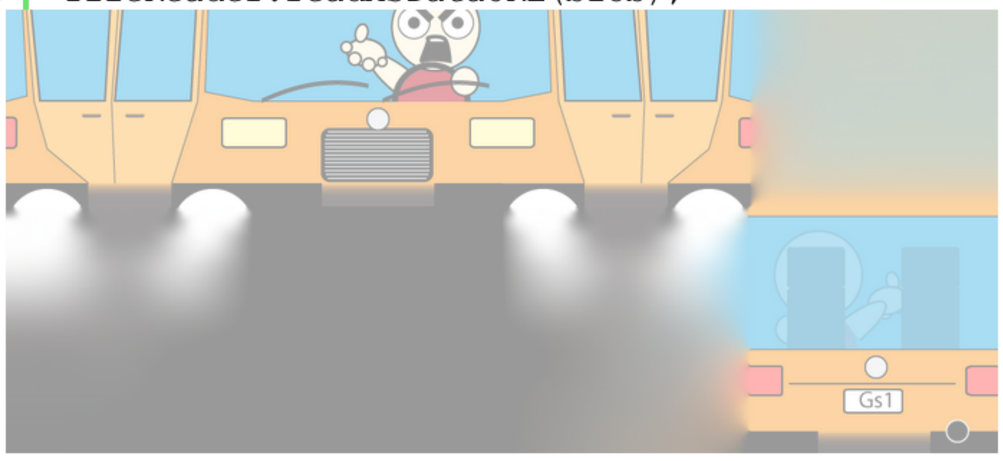
```
{
  "filename": "name",
  "size": 1024
}
```

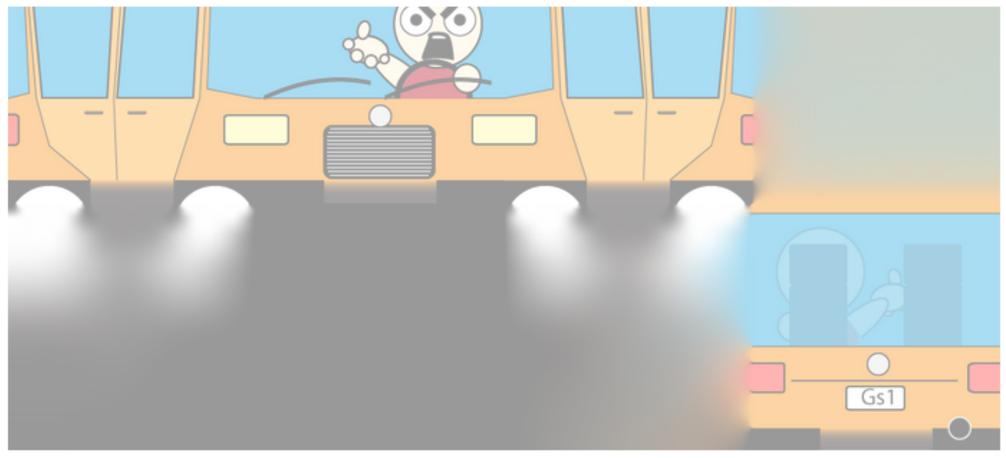
Cool Tech

- XmlHTTPRequest2
- Typed Arrays
- File and Blob APIs
- Data URI

```
<img src="images/car/cartexture.png">

<img src="" id="cartexture-datauri-target">

<img src="...+GAAAAAElFTkSuQmCC">
```

```
var blob = ...,
    fileReader = new FileReader();

fileReader.addEventListener('load', function(e) {
  $('#cartexture-datauri-target')[0].src = e.target.result;
});

fileReader.readAsDataURL(blob);
```





Simple Loading Scheme

Preload Everything

Fine for small titles

The Next Step

Load Levels

Great for level based games

Stream All of It

Group related files.

- World Scenery
- Vehicles
- Monsters
- Player Objects
    - Weapons
    - Clothing

Caching

- Browser Caching
- AppCache
- LocalStorage
- IndexedDB
- FileSystem

Browser Caching

```
HTTP/1.0 GET images/superhero.png
HOST: mygame.com
```

```
HTTP/1.0 200 OK

[Image Content]
```

```
HTTP/1.0 GET images/superhero.png
HOST: mygame.com
MODIFIED-SINCE: Yesterday
```

```
HTTP/1.0 304 NOT MODIFIED
```

Storing here

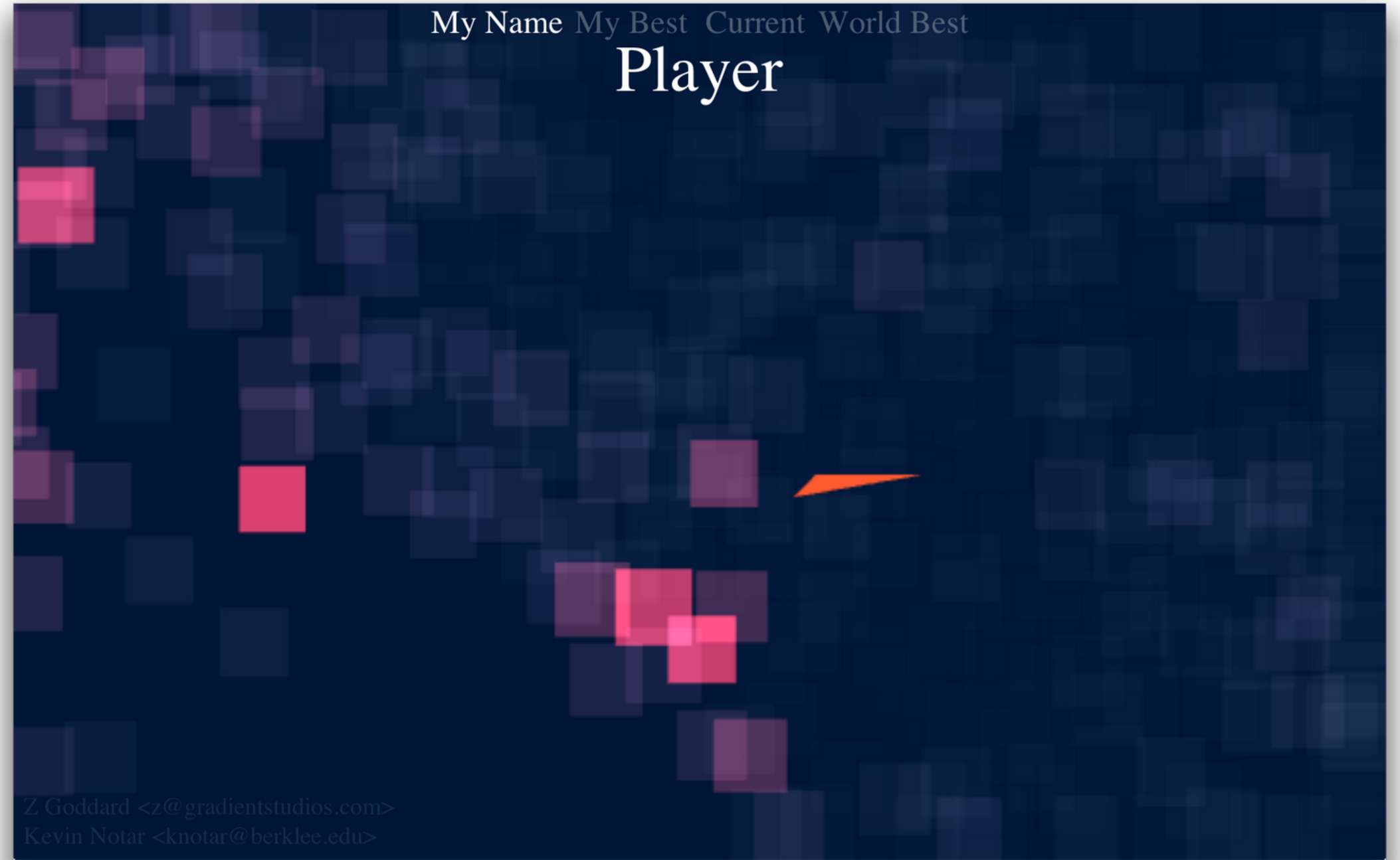
```
1  localStorage['gamesave'] = JSON.stringify({
2    playerName: 'George',
3    class: 'Paladin',
4    level: 9001
5  });
```

FileSystem API

- Sandboxed
- Directories
- Files
  - Binary Data
  - URLs with direct access

User Interface

"DOM is UI middleware you get for free."
- Ben Alman


My Name My Best Current World Best
Player
Z Goddard <z@gradientstudios.com>
Kevin Notar <knotar@berklee.edu>

Talked about

- Component Based Architecture fits and works with declarative design using JSON documents
- Promises, a design pattern for one time events
- Dealing with scaling HTTP is important
- The Browser comes with many builtin tools for User Interface

gradient
studios, inc

Michael "Z" Goddard

@ZofGames

#JSGameEngine

Help Wanted: SFX and Music
contact@gradientstudios.com

Questions?

@ZofGames
#JSGameEngine
github.com/mzgoddard