

# ImpactJS Presentation

David Leonard

City College of New York

August 28, 2014

# Why ImpactJS?

- ▶ Collision handling
- ▶ Camera pluginss
- ▶ Map Editor
- ▶ Player Physics
- ▶ Strong OOP Design

# JavaScript Types

► strings : "Hello World"

# JavaScript Types

- ▶ strings : "Hello World"
- ▶ number : `var x = 45`

# JavaScript Types

- ▶ strings : "Hello World"
- ▶ number : `var x = 45`
- ▶ boolean : `var flip = true`

# JavaScript Types

- ▶ strings : "Hello World"
- ▶ number : `var x = 45`
- ▶ boolean : `var flip = true`
- ▶ function
- ▶ array : `var arr = [ 1, 2, 3, 4 ]`

# JavaScript Types

- ▶ strings : "Hello World"
- ▶ number : `var x = 45`
- ▶ boolean : `var flip = true`
- ▶ function
- ▶ array : `var arr = [ 1, 2, 3, 4 ]`
- ▶ object : `person = { name: David, Age: 23 }`

# JavaScript Types

- ▶ strings : "Hello World"
- ▶ number : `var x = 45`
- ▶ boolean : `var flip = true`
- ▶ function
- ▶ array : `var arr = [ 1, 2, 3, 4 ]`
- ▶ object : `person = { name: David, Age: 23 }`
- ▶ undefined : `typeof person === undefined`



# JavaScript Types

- ▶ strings : "Hello World"
- ▶ number : var x = 45
- ▶ boolean : var flip = true
- ▶ function
- ▶ array : var arr = [ 1, 2, 3, 4 ]
- ▶ object : person = { name: David, Age: 23 }
- ▶ undefined : typeof person === undefined
- ▶ null : var x = null

# JavaScript Objects

```
var person = {  
  name: David,  
  age: 23,  
  major: Computer Science  
};
```

```
person.name; // David  
person.age; // 23  
person.major; // Computer Science
```

# Init Function

```
init: function(){  
    // Init function is only run once  
    ig.input.bind( ig.KEY.LEFT_ARROW, 'left');  
    ig.input.bind( ig.KEY.RIGHT_ARROW, 'right');  
    ig.input.bind( ig.KEY.X, 'jump');  
}
```

# Update Function

```
update: function(){  
    // update is run once per frame  
    if( typeof player === undefined ){  
        var player = this.getEntitiesByType(EntityPlayer)[0];  
        this.screen.x = player.pos.x - ig.system.width/2;  
        this.screen.y = player.pos.y - ig.system.height/2;  
    }  
}
```

## Draw Function

```
draw: function(){  
    this.parent();  
    if(this.font){  
        var player = this.getEntitiesByType(EntityPlayer)[0];  
        this.font.draw('Health: ' + player.health, 50, 10, ig.  
            Font.ALIGN.CENTER);  
    }  
}
```

## Simple slide with three points shown in succession

- Point 1 (Click “Next Page” to see Point 2)

# Simple slide with three points shown in succession

- ▶ Point 1 (Click “Next Page” to see Point 2)
- ▶ Point 2

## Simple slide with three points shown in succession

- ▶ Point 1 (Click “Next Page” to see Point 2)
- ▶ Point 2
- ▶ Point 3



# Slide with two columns: items and a graphic

- ▶ First item

Insert graphic here

## Slide with two columns: items and a graphic

- ▶ First item
- ▶ Second item

Insert graphic here

## Slide with two columns: items and a graphic

- ▶ First item
- ▶ Second item
- ▶ ...

Insert graphic here