

ImpactJS Presentation

David Leonard

City College of New York

August 29, 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);  
    }  
}
```

main.js

- ▶ Require all modules

main.js

- ▶ Require all modules
- ▶ Define game classes

main.js

- ▶ Require all modules
- ▶ Define game classes
- ▶ Bind keys within Init()

main.js

- ▶ Require all modules
- ▶ Define game classes
- ▶ Bind keys within Init()
- ▶ Camera code, game logic within Update()

main.js

- ▶ Require all modules
- ▶ Define game classes
- ▶ Bind keys within Init()
- ▶ Camera code, game logic within Update()
- ▶ Draw images within Draw()

Loading modules

```
ig.module(  
    'game.main'  
)  
    .requires(  
        'impact.game',  
        'impact.debug.debug',  
        'game.levels.test',  
        'game.entities.player',  
        'game.entities.goomba'  
    )
```

Define your game class

```
.defines(function(){  
    MyGame = ig.Game.extend({  
  
    });  
  
    ig.main( '#canvas', MyGame, 60, 320, 240, 2 );  
});
```

Define properties

```
.defines(function(){  
    MyGame = ig.Game.extend({  
        font: new ig.Font('media/04b03.font.png'),  
        gravity: 300,  
    });  
  
    [ ... ]  
});
```

Init function

```
MyGame = ig.Game.extend({  
  init: function(){  
    ig.input.bind( ig.KEY.LEFT_ARROW, 'left');  
    ig.input.bind( ig.KEY.RIGHT_ARROW, 'right');  
    ig.input.bind( ig.KEY.X, 'jump');  
    this.loadLevel( LevelTest );  
  }  
});
```

Update function

```
MyGame = ig.Game.extend({
  init: [ ... ],

  update: function(){
    var player = this.getEntitiesByType(
      EntityPlayer )[0];
    if( player ) {
      this.screen.x = player.pos.x - ig.system.width
        /2;
      this.screen.y = player.pos.y - ig.system.height
        /2;
    }
    this.parent();
  },
});
```

Draw function

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


Player module

```
ig.module(  
    'game.entities.player'  
)  
    .requires(  
        'impact.entity'  
)  
    .defines(function(){  
        EntityPlayer = ig.Entity.extend({  
  
        });  
    });
```

Player collision

```
EntityPlayer = ig.Entity.extend({  
    type: ig.Entity.TYPE.A,  
    checkAgainst: ig.Entity.TYPE.NONE,  
    collides: ig.Entity.COLLIDES.ACTIVE,  
});
```

Player physics

```
EntityPlayer = ig.Entity.extend({  
  [ ... ],  
  animSheet: new ig.AnimationSheet('media/player.png',  
    16, 28),  
  size: {x: 16, y: 28},  
  offset: {x: 1, y: 0},  
  flip: true,  
  maxVel: {x: 100, y: 160},  
  friction: {x: 500, y: 0},  
  accelGround: 200,  
  accelAir: 310,  
  jump: 360,  
  gravity: 300,  
});
```

Player init function

```
EntityPlayer = ig.Entity.extend({  
    [ ... properties ...],  
  
    init: function(x, y, settings){  
        this.parent(x, y, settings);  
        this.addAnim('idle', 1, [1]);  
        this.addAnim('run', 0.10, [1, 0]);  
        this.addAnim('jump', 1, [2]);  
        this.addAnim('fall', 0.4, [2]);  
    },  
});
```

Player update function

```
[ ... properties ... ]
init: [ ... ]
update: function(){
    // Add left/right/jump movement logic
    var accel = this.standing ? this.accelGround : this.
        accelAir;
    if(ig.input.state('left')) {
        this.accel.x = -accel;
        this.flip = false;
    } else if(ig.input.state('right')) {
        this.accel.x = accel;
        this.flip = true;
    } else
        this.accel.x = 0;
},
```

Player update function

```
update: function(){
    [ ... ]
    if(this.standing && ig.input.state('jump')) {
        if(this.vel.y == 0) {
            this.vel.y = -this.jump;
            this.falling = false;
        }
    }

    // CASE 2: Player not standing, jump has been
    // released and we're not falling
    // we reduce the y velocity by 66% and mark us as
    // falling
    else if(!this.standing && !ig.input.state('jump') &&
        !this.falling) {
        this.vel.y = Math.floor(this.vel.y/3);
        this.falling = true;
    }
},
```

Player update function

```
update: function(){  
    [ ... ]  
  
    // Add moving logic  
    this.currentAnim.flip.x = this.flip;  
    this.parent();  
  
    if(this.vel.y < 0 && !this.standing)  
        this.currentAnim = this.anims.jump;  
    else if(this.vel.y > 0 && !this.standing)  
        this.currentAnim = this.anims.fall;  
    else if(this.vel.x !== 0)  
        this.currentAnim = this.anims.run;  
    else  
        this.currentAnim = this.anims.idle;  
},
```