Javascript, The Swiss Army Knife of Programming Languages

David Morcillo

23-11-2013

Game design

We are going to create a simple game in the following stages using Javascript. The game will be a tiny MOBA (Multiplayer Online Battle Arena) and it will have the following features:

Game design

We are going to create a simple game in the following stages using Javascript. The game will be a tiny MOBA (Multiplayer Online Battle Arena) and it will have the following features:

Game design

We are going to create a simple game in the following stages using Javascript. The game will be a tiny MOBA (Multiplayer Online Battle Arena) and it will have the following features:

- Player choose a team: red or blue

Game design

We are going to create a simple game in the following stages using Javascript. The game will be a tiny MOBA (Multiplayer Online Battle Arena) and it will have the following features:

- Player choose a team: red or blue
- Player choose between 3 classes:
  - Soldier: low hp, ranged weapon, medium damage
  - Knight: medium hp, melee weapon, high damage
  - Protector: high hp, no weapon, can block enemy's attacks

Game design

We are going to create a simple game in the following stages using Javascript. The game will be a tiny MOBA (Multiplayer Online Battle Arena) and it will have the following features:

- Player choose a team: red or blue
- Player choose between 3 classes:
  - Soldier: low hp, ranged weapon, medium damage
  - Knight: medium hp, melee weapon, high damage
  - Protector: high hp, no weapon, can block enemy's attacks
- **Objective:** Destroy other's team base

HTML5

First of all, HTML5 is not a programming language, neither an API. It's the 5th revision of HTML but also an umbrella term about 100 specifications for the next generation web applications. Visit `http://platform.html5.org/` to have a global view about it.

HTML5

First of all, HTML5 is not a programming language, neither an API. It's the 5th revision of HTML but also an umbrella term about 100 specifications for the next generation web applications. Visit `http://platform.html5.org/` to have a global view about it.
We are going to use a small subset of these new APIs:

HTML5

First of all, HTML5 is not a programming language, neither an API. It's the 5th revision of HTML but also an umbrella term about 100 specifications for the next generation web applications. Visit `http://platform.html5.org/` to have a global view about it.
We are going to use a small subset of these new APIs:

- **Canvas:** Drawing graphics in 2D.

HTML5

First of all, HTML5 is not a programming language, neither an API. It's the 5th revision of HTML but also an umbrella term about 100 specifications for the next generation web applications. Visit `http://platform.html5.org/` to have a global view about it.
We are going to use a small subset of these new APIs:

- **Canvas:** Drawing graphics in 2D.
- **requestAnimationFrame:** Handling animations timings.

HTML5

First of all, HTML5 is not a programming language, neither an API. It's the 5th revision of HTML but also an umbrella term about 100 specifications for the next generation web applications. Visit `http://platform.html5.org/` to have a global view about it.
We are going to use a small subset of these new APIs:

- **Canvas:** Drawing graphics in 2D.
- **requestAnimationFrame:** Handling animations timings.
- **Websockets:** Two-way communication between browser and server.

Introduction to the Canvas API

HTML5 introduces a new tag called canvas. Using Javascript we can interact with this element in order to draw 2D graphics in real time.

Canvas element and 2d context

```
<canvas id=‘‘gameArea’’ width=‘‘200’’ height=‘‘200’’></canvas>
<script>
var canvas = document.getElementById(’gameArea’),
    ctx = canvas.getContext(‘‘2d’’);
</script>
```

Introduction to the Canvas API

Drawing lines

```
ctx.fillStyle = ‘‘black’’;

ctx.beginPath();
ctx.moveTo(10, 10);
ctx.lineTo(100, 10);
ctx.stroke();

ctx.beginPath();
ctx.moveTo(10, 20);
ctx.lineTo(100, 20);
ctx.stroke();

ctx.beginPath();
ctx.moveTo(10, 30);
ctx.lineTo(100, 30);
ctx.stroke();
```

Introduction to the Canvas API

Drawing rects

```
ctx.fillStyle = ‘‘blue’’;
ctx.strokeStyle = ‘‘red’’;
ctx.fillRect(100, 100, 50, 50);
ctx.strokeRect(165, 165, 25, 25);
```

Introduction to the Canvas API

Drawing rects

```
ctx.fillStyle = ‘‘blue’’;
ctx.strokeStyle = ‘‘red’’;
ctx.fillRect(100, 100, 50, 50);
ctx.strokeRect(165, 165, 25, 25);
```

Drawing arcs

```
ctx.beginPath();
ctx.fillStyle = ‘‘green’’;
ctx.strokeStyle = ‘‘orange’’;
ctx.arc(150, 50, 5, 0, 2 * Math.PI);
```

Introduction to the Canvas API

Drawing images

```
var crate = new Image();
crate.src = 'images/crate.png';
/*
* onload callback function. Called when the
* image is ready to be drawn.
*/
crate.onload = function () {
    ctx.drawImage(crate, 100, 100);
};
```

Introduction to the Canvas API

scale

```
// Draw crate 2x bigger
ctx.scale(2, 2);
ctx.drawImage(crate, 100, 100);
```

Introduction to the Canvas API

scale

```
// Draw crate 2x bigger
ctx.scale(2, 2);
ctx.drawImage(crate, 100, 100);
```

translate

```
// Same as ctx.drawImage(create, 100, 100);
ctx.translate(100, 100);
ctx.drawImage(crate, 0, 0);
```

Introduction to the Canvas API

scale

```
// Draw crate 2x bigger
ctx.scale(2, 2);
ctx.drawImage(crate, 100, 100);
```

translate

```
// Same as ctx.drawImage(create, 100, 100);
ctx.translate(100, 100);
ctx.drawImage(crate, 0, 0);
```

rotate

```
ctx.rotate(45 * (Math.PI * 180));
ctx.drawImage(crate, 0, 0);
```

Introduction to the Canvas API

save and restore

```
ctx.save();
ctx.translate(100, 100);
ctx.translate(crate.width / 2, crate.height / 2);
ctx.rotate(45 * (Math.PI / 180));
ctx.translate(-crate.width / 2, -crate.height / 2);
ctx.drawImage(crate, 0, 0);
ctx.restore();

ctx.drawImage(create, 0, 0);
```