

# HTML5 PARA VIDEO JUEGOS

# HTML5 PARA VIDEO JUEGOS

## Objetivo

**Introducir HTML5 para su uso en la implementación de video juegos Web.**

# HTML5 PARA VIDEO JUEGOS

## Introducción a JQuery

jQuery es una biblioteca de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con Ajax a páginas web.

Se puede descargar desde <http://jquery.com/> descargar jquery-1.4.4.min.js, y guardarlo en una carpeta js, dentro del proyecto.

# HTML5 PARA VIDEO JUEGOS

## Introducción a JQuery

Un objeto JQuery es un “*wrapper*” para un grupo determinado de nodos del DOM.

La función `$()` es un factory method que crea objetos JQuery.

Por ejemplo `$("dt")` es un objeto JQuery que contiene todos los elementos “dt” del documento.

# HTML5 PARA VIDEO JUEGOS

## Introducción a JQuery

### Formas de Selecccion en JQuery

#### CSS

p	element name
#id	identifier
.class	classname
p.class	element with class
p a	anchor as any descendant of p
p > a	anchor direct child of p

# HTML5 PARA VIDEO JUEGOS

## Introducción a JQuery

### Formas de Seleccion en JQuery

#### XPath

/html/body//div paths

a[@href] anchor with an href attr

div[ol] div with an ol inside

//a[@ref='nofollow'] any anchor with a specific value for the  
ref attribute

# HTML5 PARA VIDEO JUEGOS

## Introducción a JQuery

Selector pattern	Meaning	Examples
<code>\$("Element")</code>	Selects all elements with the given tag name	<code>\$ ("p")</code> selects all the p tags. <code>\$ ("body")</code> selects the body tag.
<code>\$("#id")</code>	Selects the element with the given ID of the attribute	Provides the following code: <code>&lt;div id="box1"&gt;&lt;/div&gt;</code> <code>&lt;div id="box2"&gt;&lt;/div&gt;</code> <code>\$ (" #box1 ")</code> selects the highlighted element.
<code>\$(".className")</code>	Selects all elements with the given class attribute	Provides the following code: <code>&lt;div class="apple"&gt;&lt;/div&gt;</code> <code>&lt;div class="apple"&gt;&lt;/div&gt;</code> <code>&lt;div class="orange"&gt;&lt;/div&gt;</code> <code>&lt;div class="banana"&gt;&lt;/div&gt;</code> <code>\$ (" .apple ")</code> selects the highlighted elements with class set to apple.
<code>\$("selector1, selector2, selectorN")</code>	Selects all elements that match the given selector	Provides the following code: <code>&lt;div class="apple"&gt;&lt;/div&gt;</code> <code>&lt;div class="apple"&gt;&lt;/div&gt;</code> <code>&lt;div class="orange"&gt;&lt;/div&gt;</code> <code>&lt;div class="banana"&gt;&lt;/div&gt;</code> <code>\$ (" .apple, .orange ")</code> selects the highlighted elements that class is set to, either apple or orange.

# HTML5 PARA VIDEO JUEGOS

## Introducción a JQuery

Algunos eventos..

Event Method	Description
<code>\$(document).ready(function)</code>	Binds a function to the ready event of a document (when the document is finished loading)
<code>\$(selector).click(function)</code>	Triggers, or binds a function to the click event of selected elements
<code>\$(selector).dblclick(function)</code>	Triggers, or binds a function to the double click event of selected elements
<code>\$(selector).focus(function)</code>	Triggers, or binds a function to the focus event of selected elements
<code>\$(selector).mouseover(function)</code>	Triggers, or binds a function to the mouseover event of selected elements



# HTML5 PARA VIDEO JUEGOS

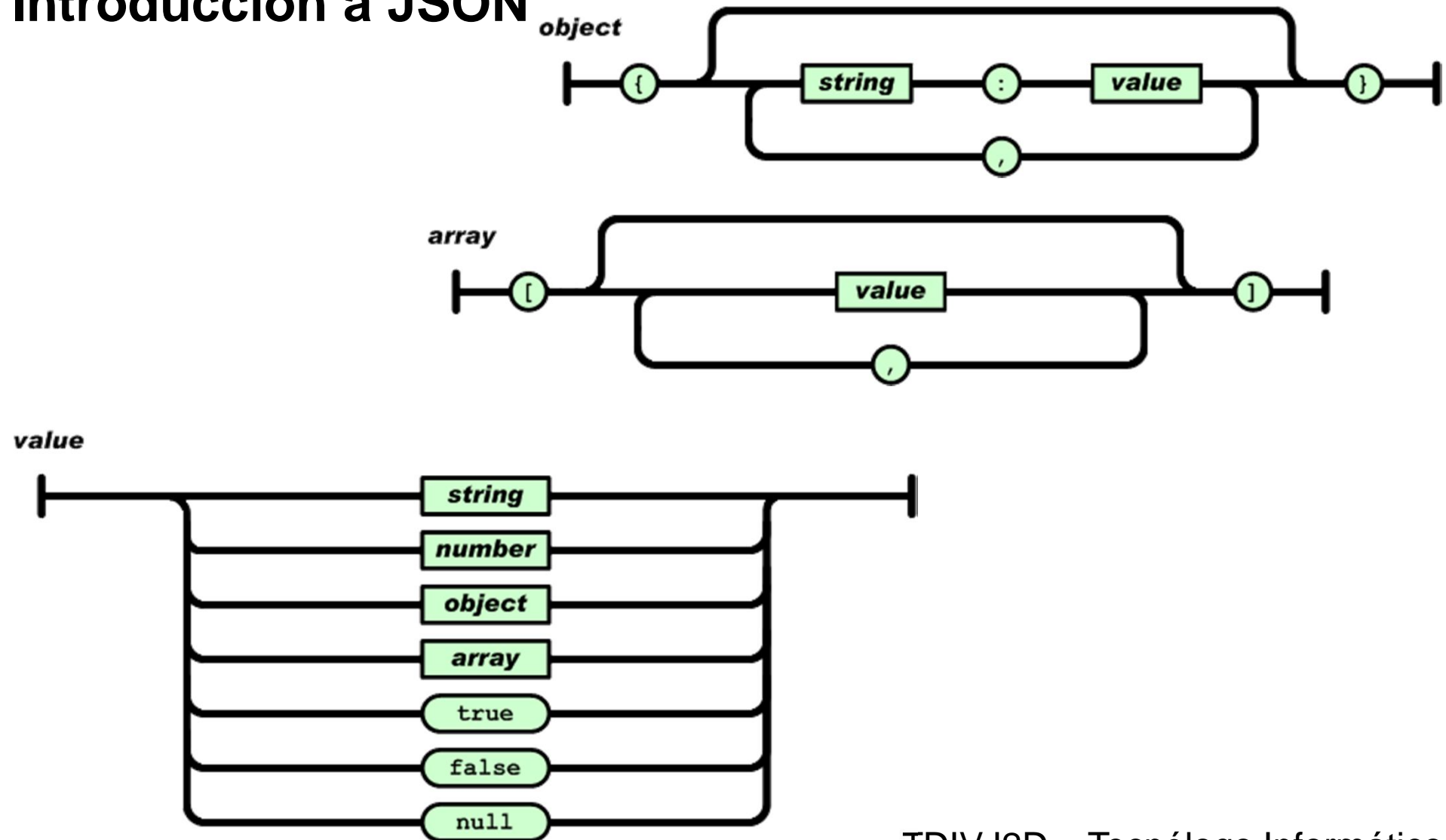
## Introducción a JSON

JSON(JavaScript Object Notation), es un formato ligero para el intercambio de datos, es una alternativa simple a XML.

En JavaScript, un texto JSON se puede analizar fácilmente usando el procedimiento `eval()`.

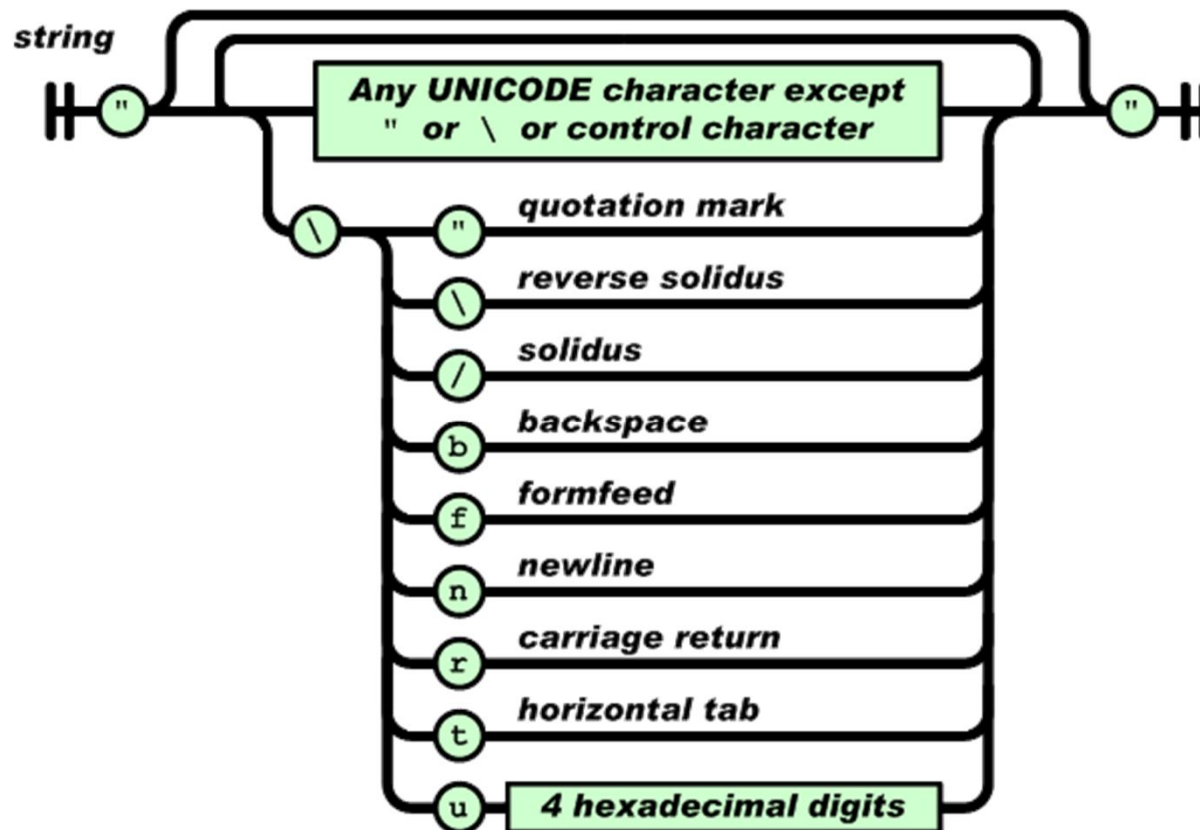
# HTML5 PARA VIDEO JUEGOS

## Introducción a JSON



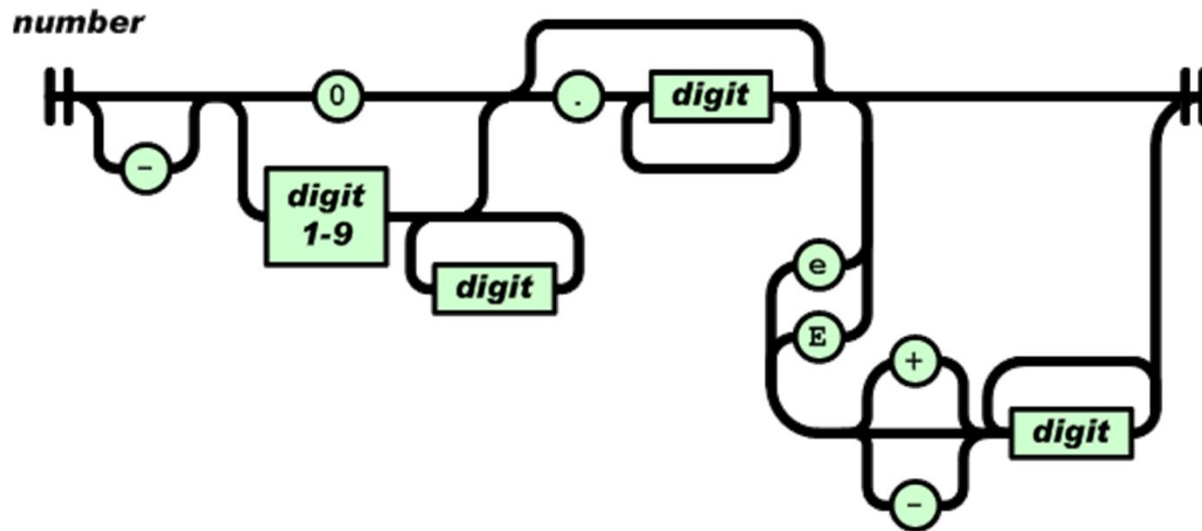
# HTML5 PARA VIDEO JUEGOS

## Introducción a JSON



# HTML5 PARA VIDEO JUEGOS

## Introducción a JSON



# HTML5 PARA VIDEO JUEGOS

## Introducción a JSON

```
<script language="javascript" type="text/javascript">
  var json = eval (
    {"poblacion": [ { "id": "0", "nombre": "Mal donado" }, { "id": "1", "nombre": "Rocha" } ]}
  );
  if (json.poblacion.length > 0 )
  {
    for (i=0; i < json.poblacion.length; i++)
    {
      alert(json.poblacion[i].nombre);
    }
  }
</script>
```

# HTML5 PARA VIDEO JUEGOS

## Introducción a HTML5

Es una colección de estándares para el diseño y desarrollo de páginas web. Esta colección representa la manera en que se presenta la información en el explorador de internet y la manera de interactuar con ella.

Permite una mayor interacción entre páginas web y contenido media (video, audio, entre otros) así como una mayor facilidad a la hora de codificar el diseño.

# HTML5 PARA VIDEO JUEGOS

## Porque hacer juegos con html5

No se necesitan instalar plugins extra, ya que los browser modernos soportan toda la nueva funcionalidad ofrecida por HTML5 como son el soporte de audio, y canvas. Funcionan en dispositivos IOS

Además de las nuevas características de CSS3 para programar jugos con html5 es necesario utilizar java script.

Existen también distintos frameworks como jquery, y distintos engine de video juegos y de física como cocos2d, box2d, instant, etc. , que se pueden utilizar.

# HTML5 PARA VIDEO JUEGOS

## Nuevas Características de HTML5

- CANVAS
- Soporte de Audio
- Geolocalizacion
- Webgl
- Web socket
- Local storage
- Offline application

## Nuevas Características de CSS3

- Transiciones
- Transformaciones



# HTML5 PARA VIDEO JUEGOS

## Canvas

El elemento canvas es un componente donde se pueden crear y manipular imágenes dinámicamente.

Todo lo que se escriba entre la apertura y cierre de la etiqueta canvas solamente será interpretado por navegadores que **no soportan** aún la nueva etiqueta.

```
<canvas id="entorno_canvas" width="360" height="240">
<p>Tu navegador no soporta canvas: Aquí deberías de ver una imagen : </p>

</canvas>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Ping Pong</title>
</head>
<body>
<header>
<h1>Ping Pong</h1>
</header>
<section id="layers">
<canvas id="bg" width="768" height="440">
    Sorry, your web browser does not support canvas content.
</canvas>
<canvas id="guide" width="768" height="440"></canvas>
<canvas id="game" width="768" height="440"></canvas>
<canvas id="ui" width="768" height="440"></canvas>
</section><footer>
</footer>
<script src="js/jquery-1.4.4.min.js"></script>
<script>
    $(function(){
        alert("jQuery cargado correctamente.");
    });
</script>
</body>
</html>
```

# HTML5 PARA VIDEO JUEGOS

## Canvas

Para utilizar el canvas se debe referenciar primero el elemento canvas y adquirir su contexto.

```
var canvas = document.getElementById('entorno_canvas');  
var context = canvas.getContext('2d');
```

Una vez adquirido, se puede empezar a dibujar en la superficie del canvas usando la API.

# HTML5 PARA VIDEO JUEGOS

## Canvas

### Introducing the line drawing API

There are some drawing APIs for us to draw and style the line stroke

Line drawing functions	Discussion
<code>MoveTo</code>	The <code>MoveTo</code> function is like holding the pen in our hand and moving it on top of the paper without touching it with the pen.
<code>LineTo</code>	This function is like putting the pen down on the paper and drawing a straight line to the destination point.
<code>lineWidth</code>	<code>LineWidth</code> sets the thickness of the strokes we draw afterwards.
<code>Stroke</code>	<code>stroke</code> is the function to execute the drawing. We set up a collection of <code>moveTo</code> , <code>lineTo</code> , or styling functions and finally call the <code>stroke</code> function to execute it on canvas.

---

[m]

---

# HTML5 PARA VIDEO JUEGOS

## Canvas

The drawImage function accepts several arguments:

```
drawImage(image, x, y);
```

Argument	Definition	Discussion
image	The image reference that we are going to draw.	We either get the image reference by getting an existing <code>img</code> element or creating a JavaScript Image object.
x	The x position to place the image in canvas coordinate.	The x and y coordinate is where we place the image with respect to its top-left corner.
y	The y position to place the image in canvas coordinate.	

# HTML5 PARA VIDEO JUEGOS

## Canvas

```
drawImage(image, x, y, width, height);
```

Argument	Definition	Discussion
image	The image reference that we are going to draw.	We either get the image reference by getting an existing <code>img</code> element or creating a JavaScript Image object.
x	The x position to place the image in canvas coordinate.	The x and y coordinate is where we place the image with respect to its top-left corner.
y	The y position to place the image in canvas coordinate.	
width	The width of the final drawn image.	We are applying scale to the image if the width and height is not the same as the original image.
height	The height of the final drawn image.	

# HTML5 PARA VIDEO JUEGOS

## Canvas

```
drawImage(image, sx, sy, sWidth, sHeight, dx, dy, width, height);
```

Argument	Definition	Discussion
image	The image reference that we are going to draw.	We either get the image reference by getting an existing <code>img</code> element or creating a JavaScript Image object.
sx	The x coordinate of the top-left corner of the clipping region.	The clipping x, y, width, height together defines a rectangular clipping area. The given image is clipped by this rectangle.
sy	The y coordinate of the top-left corner of the clipping region.	
sWidth	The width of the clipping region.	
sHeight	The height of the clipping region.	

# HTML5 PARA VIDEO JUEGOS

## Canvas

Argument	Definition	Discussion
dx	The x position to place the image in canvas coordinate.	The x and y coordinate is where we place the image with respect to its top-left corner.
dy	The y position to place the image in canvas coordinate.	
width	The width of the final drawn image.	We are applying scale to the clipped image if the width and height is not the same as the clipping dimension.
height	The height of the final drawn image.	



# HTML5 PARA VIDEO JUEGOS

## Audio

### Handling audio events

There are many other events in the audio element. The following table lists a few commonly used audio events:

Event	Discussion
ended	Sent when the audio element finishes a playback
play	Sent when the audio element plays or resumes
pause	Sent when the audio element pauses
progress	Sent periodically when the audio element is downloading
timeupdate	Sent when the <code>currentTime</code> property changes

Here we just listed a few commonly used events; you can reference the complete audio event list in the Mozilla Developer Center at the following URL:

[https://developer.mozilla.org/En/Using\\_audio\\_and\\_video\\_in\\_Firefox#Media\\_events](https://developer.mozilla.org/En/Using_audio_and_video_in_Firefox#Media_events)

# HTML5 PARA VIDEO JUEGOS

## Audio

```
<audio id="buttonactive">
<source src="media/button_active.mp3" />
<source src="media/button_active.ogg" />
</audio>
<script>
```

```
    document.getElementById("buttonactive").play();
```

```
</script>
```

The play function plays the audio from the elapsed time, which is stored in the currentTime property. The default value of currentTime is zero. The following code plays the audio from 3.5 seconds:

```
<script>
```

```
    document.getElementById("buttonactive").currentTime = 3.5;
```

```
    document.getElementById("buttonactive").play();
```

```
</script>
```

```
<script>
```

```
    document.getElementById("buttonactive").pause();
```

```
</script>
```

```
<script>
```

```
    document.getElementById("buttonactive").volume = 0.3;
```

```
</script>
```

# HTML5 PARA VIDEO JUEGOS

## Almacenamiento Local

*Using Local Storage to Store Game Data*

Argument	Definition	Description
key	The key is the name of the record that we used to identify an entry.	The key is a string and each record has a unique key. Writing a new value to an existing key overwrites the old value.
value	The value is any data which will be stored.	It can be any data, but the final storage is in a string. We will discuss this shortly.

In our example, we save the game elapsed time as the score with the following code by using the key `last-elapsed-item`:

```
localStorage.setItem("last-elapsed-time", matchingGame.elapsedTime);
```

Complementary to `setItem`, we get the stored data by using the `getItem` function in the following way:

```
localStorage.getItem(key);
```

The function returns the stored value of the given key. It returns `null` when trying to get a non-existent key. This can be used to check whether we have stored any data for a specific key.

# HTML5 PARA VIDEO JUEGOS

## **WebSocket**

WebSockets de HTML5, provee una api cliente para conectar el navegador a un servidor backend.

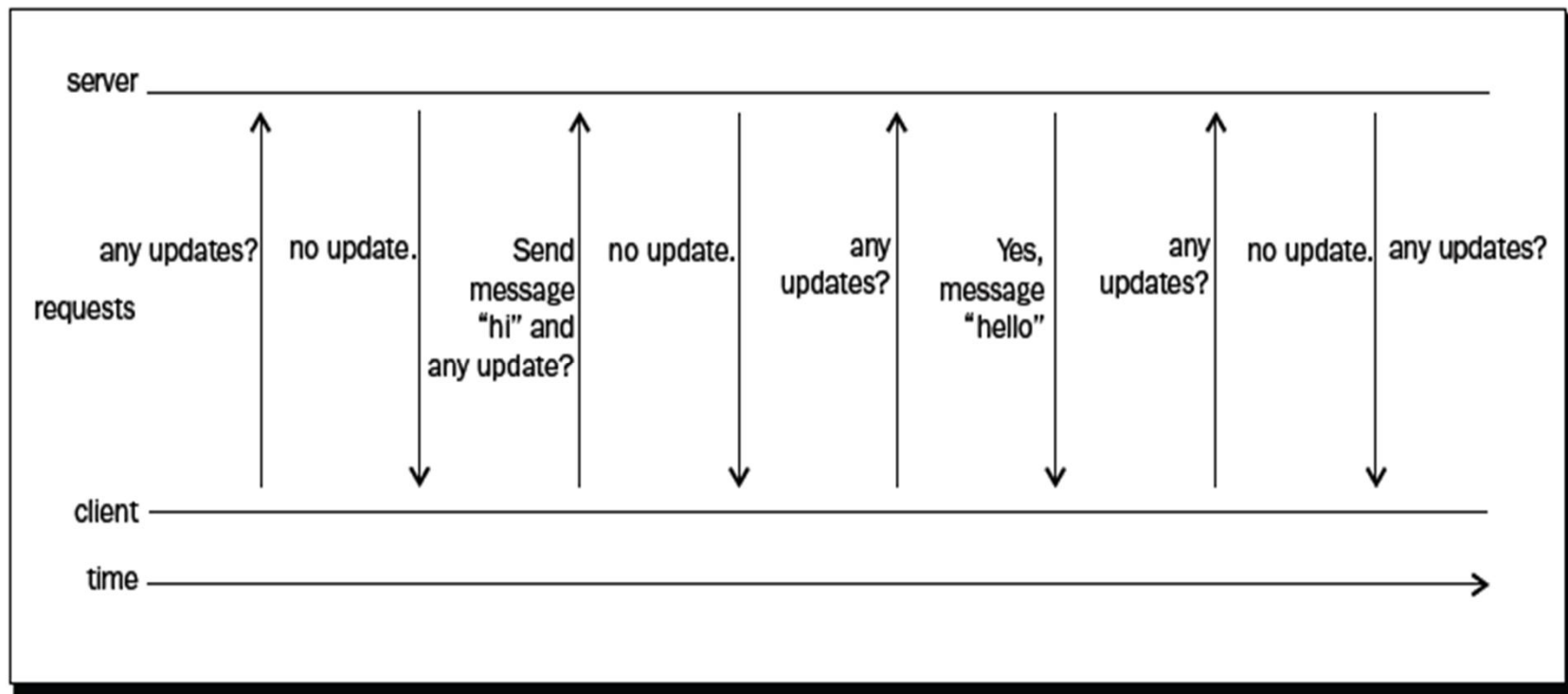
El servidor debe soportar el protocolo WebSockets para poder mantener una conexión persistente.

Ej: Node.JS, phpwebsocket, websocket-server-node.js, jWebSocket, apache-websocket

# HTML5 PARA VIDEO JUEGOS

## WebSocket

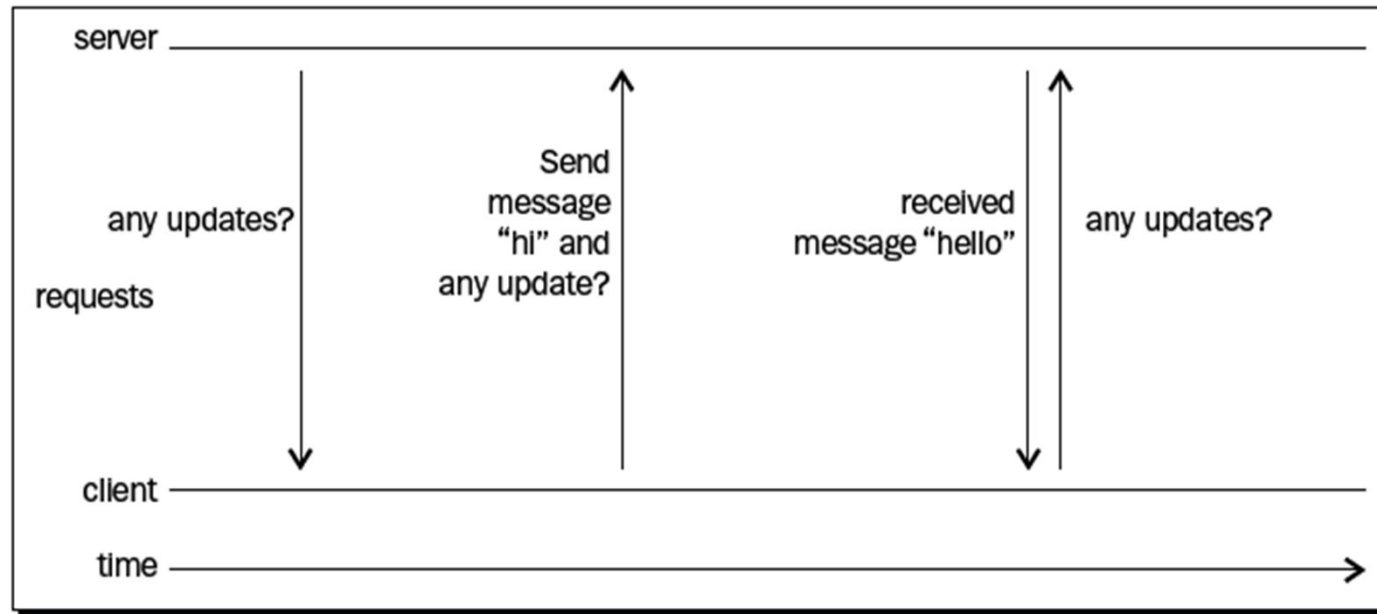
Enfoques de comunicación:  
Polling



# HTML5 PARA VIDEO JUEGOS

## WebSocket

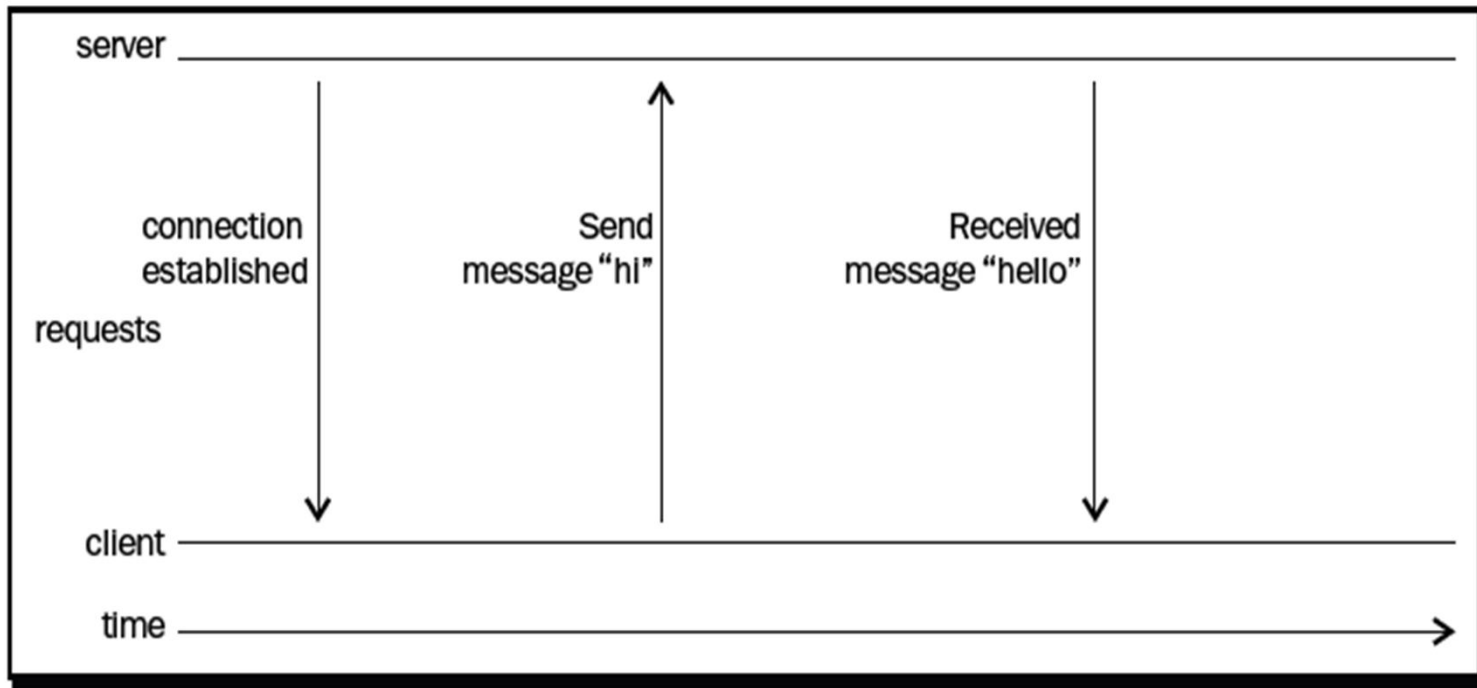
Enfoques de comunicación:  
Long Polling



# HTML5 PARA VIDEO JUEGOS

## WebSocket

Enfoques de comunicación:  
WebSocket



# HTML5 PARA VIDEO JUEGOS

## WebSocket

### Creacion y comunicación

```
var socket;
function init() {
var host = "ws://184.28.67.247:8080/dev/websocket/server.php";
    try {
socket = new WebSocket(host);
socket.onopen = function(msg){ };
socket.onmessage = function(msg){ eval('var data = ' + msg.data + ';' );...}
socket.onclose = function(msg){ };
    } catch(ex){
console.log(ex); }
$('body').bind('mousemove', function(evt){
    send(evt.clientX, evt.clientY); });
}
function send(x,y) { var msg = x + ',' + y; socket.send(msg); }
```



# HTML5 PARA VIDEO JUEGOS

## WebSocket

jWebSocket es una implementación opensource Java + Javascript del protocolo HTML5 WebSockets.

Es una solución de comunicación bidireccional entre Java y Javascript.

Ofrece:

- **WebSocket Server**: un servidor Java para comunicaciones S2C (Server To Client) y C2C
- **jWebSocket Clients**: un cliente Javascript con varios subprotocolos

# HTML5 PARA VIDEO JUEGOS

## **WebSocket**

jWebSocket ofrece la posibilidad de utilizar jWebSocket FlashBridge, para permitir el uso de websockets en navegadores que no soporten el protocolo.

Es un wrapper Flash que permite usar WebSockets en cualquier navegador que soporte y tenga instalado Flash. Esto permite usar WebSockets hasta en IE 6.

# HTML5 PARA VIDEO JUEGOS

## WebSocket

Browser	Version	SSL wss://	Comment
Chrome	8.0.x 7.0.x 6.0.x 5.0.x 4.0.249+	8.0.x 7.0.x	<b>Chrome includes native WebSockets since 4.0.249. SSL (wss) approved with Jetty 8.0.</b> Tested and verified by jWebSocket team 2010-02-28. Download latest Google Chrome <a href="#">here</a> @.
Safari	5.x 4.x	-	<b>Safari 5.x includes native WebSockets</b> - tested and verified by jWebSocket team 2010-06-08. Safari 4.x does not yet provide the WebSocket class. Installing <a href="#">Adobe Flash Player</a> @ is a reliable, quick and simple workaround - absolutely transparent to your application.
Firefox	4.0b1+ 3.7a6 3.0+	-	<b>Firefox 3.7a6 and 4.0b1+ now include native WebSockets</b> - tested and verified by jWebSocket team 2010-07-14, including new Security features (Sec-WebSocket-Key header processing), Firefox update available since Nightly Build 0.9.0.0620. Firefox versions prior to 3.7a6 tested with FlashBridge and verified by jWebSocket team 2010-03-28. Installing <a href="#">Adobe Flash Player</a> @ is a reliable, quick and simple workaround - absolutely transparent to your application. Also successfully tested on Ubuntu 10.04 and gentoo Linux with Firefox 3.6+.
Opera	11.0b 10.7b 9+ 10+	-	<b>Opera includes native WebSockets since 10.7.9067 (dev internal), 11.0 beta working as well!</b> Tested and verified by jWebSocket team 2010-10-18. Older versions successfully tested with FlashBridge. Download latest official Opera Browser <a href="#">here</a> @. latest dev internal Version <a href="#">here</a> @.
Internet Explorer with FlashBridge	6/7/8	-	Tested and verified by jWebSocket team 2010-03-31. Download latest Adobe Flash Player <a href="#">here</a> @
Internet Explorer with Chrome Frame 4/5/6/7	6/7/8	-	CF4 tested and verified by jWebSocket team 2010-02-04. Download Google Chrome Frame for Internet Explorer <a href="#">here</a> @ (for IE 6/7/8 only).  CF5 tested and verified by jWebSocket team 2010-06-09, including new Security features (Sec-WebSocket-Key header processing), available since Nightly Build 0.9.0.0609. Download Google Chrome Frame 5 (beta from Developer Channel) for Internet Explorer <a href="#">here</a> @ (for IE 6/7/8 only).

# HTML5 PARA VIDEO JUEGOS

## WebSocket

### jWebSocket

- **Despliegue:** ofrece un conjunto de JARs y de JSs que permiten incluirlo en cualquier aplicación HTML+JS+JEE.
- **Listeners:** Son el mecanismo para procesar mensajes de los clientes WebSockets además de eventos como la conexión o desconexión. Los listeners son propios de cada aplicación.

# HTML5 PARA VIDEO JUEGOS

## **WebSocket**

## **Plugins**

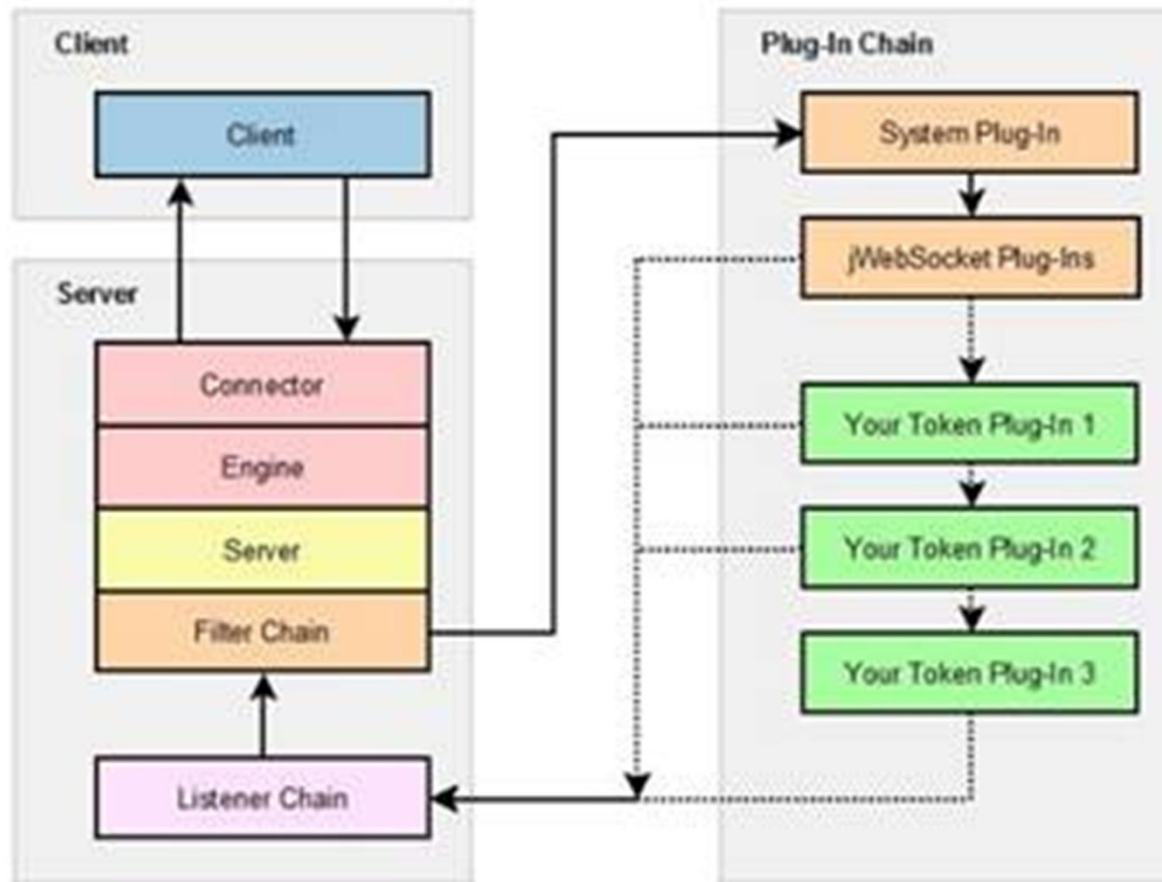
Son utilizados para crear extensiones.

JWebSocket ya ofrece una serie de plugins pero permite crear otros y añadirlos a la cadena de plugins.

Los plugins se almacenan como JARs y permiten encapsular funcionalidad.

# HTML5 PARA VIDEO JUEGOS

## WebSocket



# HTML5 PARA VIDEO JUEGOS

## WebSocket

## Plugins

RPC: permite desde un cliente web Javascript llamar funciones Java en el Server y recibir el resultado.

También permite que el Servidor llame funciones del cliente.

- Permite intercambiar objetos JSON, XML y CSV Soporta Single-Threaded y Multi-Threaded RPCs(en el modelo Single las llamadas hechas desde un cliente se responden en el mismo orden en el que se invocaron).
- Invocar funciones remotas

# HTML5 PARA VIDEO JUEGOS

## WebSocket

## Plugins

Channels: permiten que las aplicaciones abran diferentes canales lógicos independientes.

Los canales pueden ser:

- **Públicos:** todos los clientes pueden suscribirse a este tipo de canal
- **Privados:** para comunicaciones entre 2 o más clientes. Se protegen con un par Access key/secret key. Para suscribirse es necesario saben el id del canal y el Access key.
- **De Sistema:** usados por el servidor.  
Existen una serie de métodos para suscribirse, borrarse una suscripción, autorizarse.

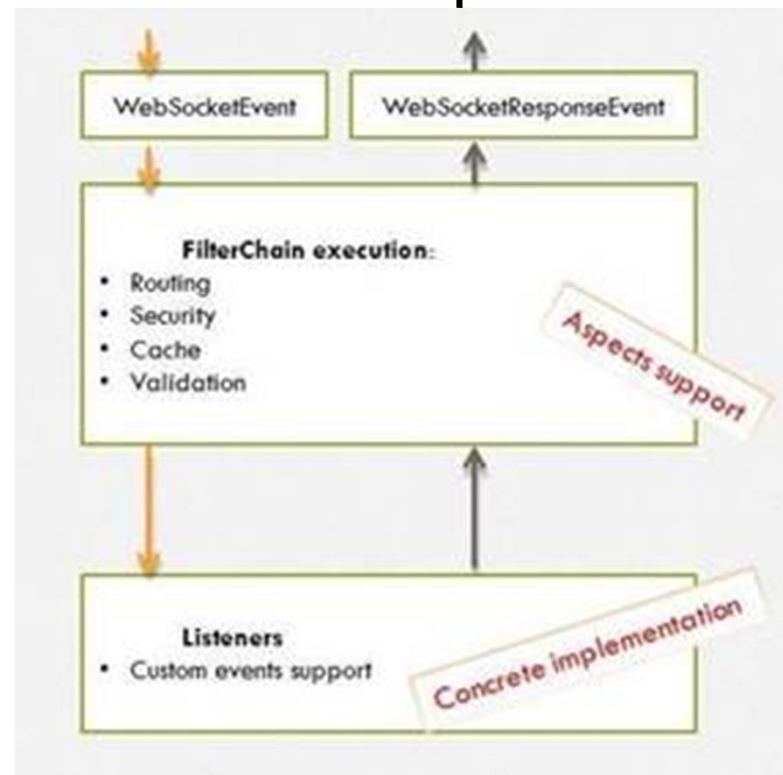


# HTML5 PARA VIDEO JUEGOS

## WebSocket

### Plugins

Events: es un plugin que permite ejecutar código en servidor en base a un evento producido



# HTML5 PARA VIDEO JUEGOS

## WebSocket

```
public class JWebSocketTokenListenerSample implements WebSocketTokenListener {

    private static Logger log = Logging.getLogger(JWebSocketTokenListenerSample.class);

    public void processOpened(WebSocketEvent aEvent) {
        log.info("Client " + aEvent.getSessionId() + " connected.");
    }

    public void processPacket(WebSocketEvent aEvent, WebSocketPacket aPacket) {
        // here you can process any non-token low level message, if desired
    }

    public void processToken(WebSocketTokenEvent aEvent, Token aToken) {
        log.info("Client " + aEvent.getSessionId() + " sent Token: " + aToken.toString() + ".");
        // here you can interpret the token type sent from the client according to your needs.
        String lNS = aToken.getNS();
        String lType = aToken.getType();

        // check if token has a type and a matching namespace
        if (lType != null && "my.namespace".equals(lNS)) {
            // create a response token
            Token lResponse = aEvent.createResponse(aToken);
            if ("getInfo".equals(lType)) {
                // if type is "getInfo" return some server information
                lResponse.put("vendor", JWebSocketConstants.VENDOR);
                lResponse.put("version", JWebSocketConstants.VERSION_STR);
                lResponse.put("copyright", JWebSocketConstants.COPYRIGHT);
                lResponse.put("license", JWebSocketConstants.LICENSE);
            } else {
                // if unknown type in this namespace, return corresponding error message
                lResponse.put("code", -1);
                lResponse.put("msg", "Token type " + lType + " not supported in namespace " + lNS + ".");
            }
            aEvent.sendToken(lResponse);
        }
    }

    public void processClosed(WebSocketEvent aEvent) {
        log.info("Client " + aEvent.getSessionId() + " disconnected.");
    }
}
```

# HTML5 PARA VIDEO JUEGOS

## WebSocket

### Registro en el servidor

```
// get the token server
TokenServer lServer = (TokenServer)JWebSocketFactory.getServer("ts0");
if( lServer != null ) {
    // and add the sample listener to the server's listener chain
    lServer.addListener(new JWebSocketTokenListenerSample());
}
```

### Registro en el cliente

```
// example how to exchange data with a server side listener
var lToken = {
    ns: "my.namespace",
    type: "getInfo"
};
jWebSocketClient.sendToken( lToken, {
    OnResponse: function( aToken ) {
        log("Server responded: "
            + "vendor: " + aToken.vendor
            + ", version: " + aToken.version
        );
    }
});
```

# HTML5 PARA VIDEO JUEGOS

## WebSocket

### Invocación JScript

```
// example how to exchange data with a server side listener
var lToken = {
  ns: "my.namespace",
  type: "getInfo"
};
jWebSocketClient.sendToken( lToken, {
  OnResponse: function( aToken ) {
    log("Server responded: "
      + "vendor: " + aToken.vendor
      + ", version: " + aToken.version
    );
  }
});
```

# HTML5 PARA VIDEO JUEGOS

## Cocos 2D

Conceptos Básicos

**Control de flujo:** Administra el flujo entre las distintas escenas.

**Sprites:** Proporciona facil manejo sobre las operaciones de sprites

**Actions:** Comando que los sprites ejecutan, pueden ser compuestos(movimiento, rotación escala)

**Effects:** Proporciona efectos predefinidos como olas, lensblour, twirl.

# HTML5 PARA VIDEO JUEGOS

## Cocos 2D

### Conceptos Básicos

- Los juegos tienen escenas
- Cada escena tiene un conjunto de capas
- Cada capa contiene sprites y captura la interacción con el usuario
- El director maneja las escenas

# HTML5 PARA VIDEO JUEGOS

## **Cocos 2D**

### Conceptos Básicos

Las escenas son las pantallas del juego, se puede pensar cada escena como una aplicación diferente.

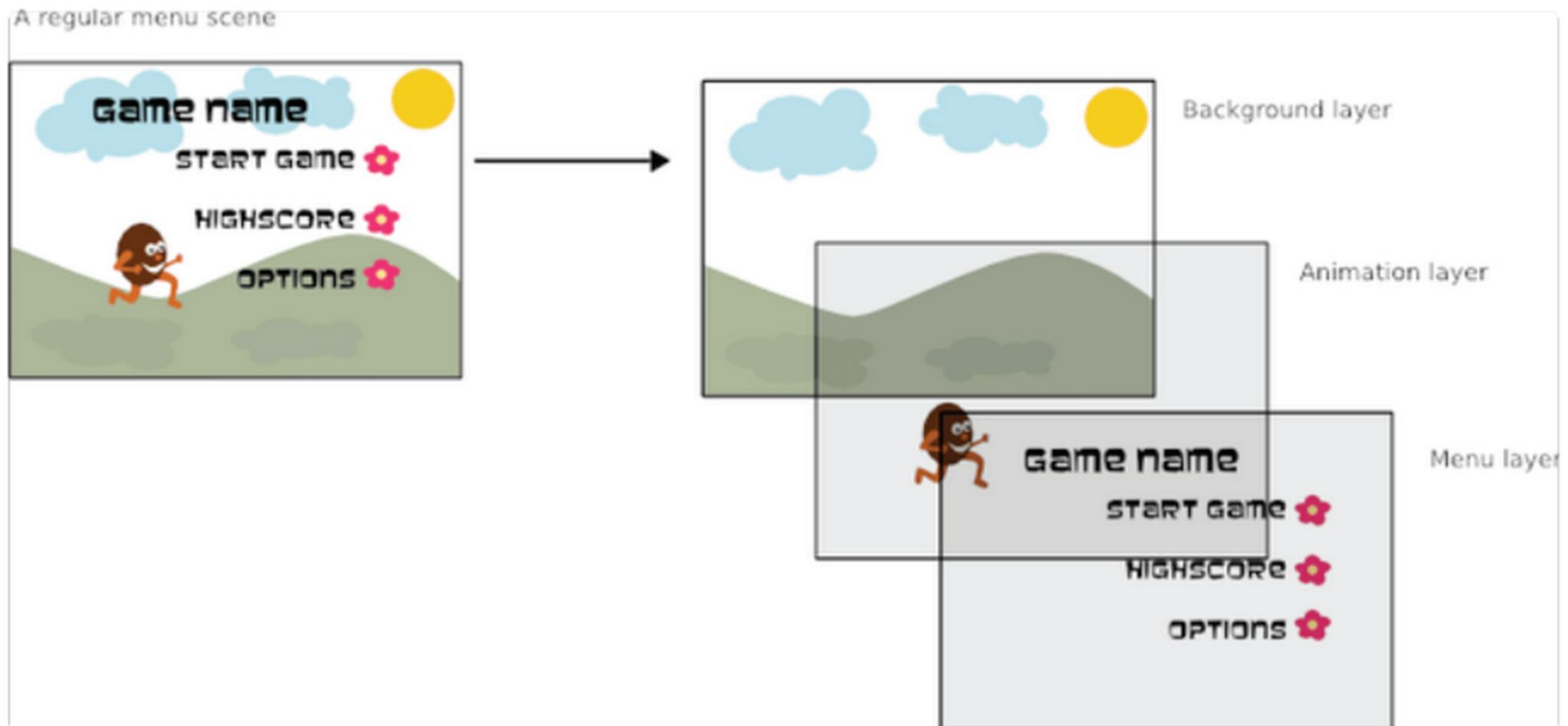
Las capas ocupan la pantalla completa manejan los eventos ingresados por el usuario, pueden a su vez contener otras capas anidadas y sprites.

El director controla la transición entre las escenas, Pausa y ejecuta escenas, las capas consultan al director para moverse.

# HTML5 PARA VIDEO JUEGOS

## Cocos 2D

### Conceptos Básicos





# HTML5 PARA VIDEO JUEGOS

## Cocos 2D

### Configuración

```
<html >
<body style="padding: 0; margin: 0; background: #fff;">
  <div style="text-align: center; font-size: 0">
    <canvas id="gameCanvas" width="600" height="600"> Your browser does not support the
    canvas tag </canvas>
  </div>
</body>
</html >
<script src="cocos2d.js"></script>
```

# HTML5 PARA VIDEO JUEGOS

## Cocos 2D

### Configuración

**COCOS2D\_Debug:** 0 debug, 1 producción

**box2d:** indica si se quiere inicializar box2d

**showFPS:** Enable/disable el contador de frames por segundo.

**tag:** indica en que tag html cocos2D debe dibujar

**EngineDir:** Directorio donde se encuentra instalado

**cocos2d**

**appFiles:** Arreglo donde se indican todos los archivos script utilizados para ser cargados por el loader.

# HTML5 PARA VIDEO JUEGOS

## Cocos 2D

### Configuración

```
(function () {  
    var d = document;  
    var c = {  
        COCOS2D_DEBUG: 2, //0 to turn debug off, 1 for basic debug, and 2 for full debug  
        box2d: false,  
        showFPS: true,  
        frameRate: 60,  
        tag: 'gameCanvas', //the dom element to run cocos2d on engineDir: '../cocos2d/',  
        appFiles: ['MyFirstApp.js'] };  
  
    window.addEventListener('DOMContentLoaded', function () { //first load engine file  
        if specified  
            var s = d.createElement('script');  
            s.src = c.engineDir + 'platform/jsloader.js';  
            d.body.appendChild(s);  
            s.c = c; s.id = 'cocos2d-html5'; });  
    })();
```

# HTML5 PARA VIDEO JUEGOS

## Cocos 2D

### Ejemplo Programa

```
var MyFirstApp = cc.Layer.extend({
  init: function()
  {
    this._super();
    var s = cc.Director.getInstance().getWinSize();
    var layer1 = cc.LayerColor.create(new cc.Color4B(255, 255, 0, 255), 600, 600);
    layer1.setAnchorPoint(new cc.Point(0.5, 0.5));
    var helloLabel = cc.LabelTTF.create("Hello world", "Arial", 30);
    helloLabel.setPosition(new cc.Point(s.width/2, s.height/2));
    helloLabel.setColor(new cc.Color3B(255, 0, 0));
    var rotationAmount = 0;
    var scale = 1;
    helloLabel.schedule(function() {
      this.setRotation(rotationAmount++);
      if(rotationAmount > 360) rotationAmount = 0;
      this.setScale(scale);
      scale+= 0.05;
      if(scale > 10) scale =1;
    });
    layer1.addChild(helloLabel);
    this.addChild(layer1); return true;
  }); //fin extend
var MyFirstAppScene = cc.Scene.extend({
  onEnter: function(){ this._super();
    var layer = new MyFirstApp();
    layer.init(); this.addChild(layer); }
});
```

# HTML5 PARA VIDEO JUEGOS

## Cocos 2D

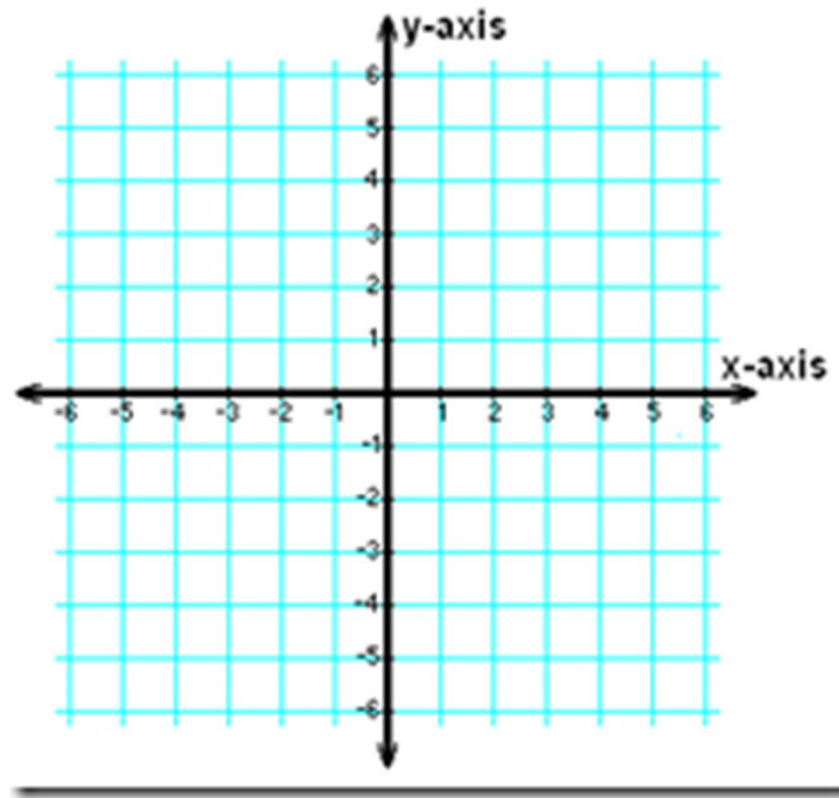
### Ejemplo Programa

```
var cocos2dApp = cc.Application.extend(  
{ config: document.querySelector('#cocos2d-html5')['c'],  
  ctor: function (scene)  
  {  
    this._super();  
    this.startScene = scene;  
    cc.COCOS2D_DEBUG = this.config['COCOS2D_DEBUG'];  
    cc.setup(this.config['tag']);  
    cc.Loader.shareLoader().onloading = function () {  
      cc.LoaderScene.shareLoaderScene().draw();  
    };  
    cc.Loader.shareLoader().onload = function () {  
      cc.AppController.shareAppController().didFinishLaunchingWithOptions();  
    };  
    cc.Loader.shareLoader().preload([]);  
  },  
  applicationDidFinishLaunching: function ()  
  {  
    var director = cc.Director.getInstance();  
    director.setDisplayStats(this.config['showFPS']);  
    director.setAnimationInterval(1.0 / this.config['frameRate']);  
    director.runWithScene(new this.startScene()); return true; }  
});  
//fin extend  
var myApp = new cocos2dApp(MyFirstAppScene);
```

# HTML5 PARA VIDEO JUEGOS

## Cocos 2D

### Sistema de Coordenadas



# HTML5 PARA VIDEO JUEGOS

## Box2D

Es una biblioteca libre que implementa un motor físico en dos dimensiones. Se encuentra disponible para distintos lenguajes de programación.

b2World es la clase que se utiliza para crear un nuevo mundo, todos los cuerpos creados en el entorno deben ser agregados al mundo.

```
var gravity = new b2Vec2(0, 300);  
var world = new b2World(worldAABB, gravity, doSleep);
```

The `b2World` class takes three arguments to initialize, which are listed in the following table with their description:

Arguments	Type	Discussion
<code>worldAABB</code>	<code>b2AABB</code>	Represents the bounding area of the world
<code>gravity</code>	<code>b2Vec2</code>	Represents the gravity of the world
<code>doSleep</code>	<code>Bool</code>	Defines whether the world ignores slept objects or not

# HTML5 PARA VIDEO JUEGOS

## Box2D

b2AABB es una clase que se utiliza para definir un área(“bounding area”) en el mundo físico. El mundo en si mismo es una “bounding area”, y es necesario crearla antes que todos los demás elementos, los elementos que se encuentren fuera de los limites de ella son destruidos.

Se puede pensar como un rectángulo del cual se da su vértice superior izquierdo y el inferior derecho. Las medidas son en metros no en pixeles.

```
var worldAABB = new b2AABB();  
worldAABB.minVertex.Set(-4000, -4000);  
worldAABB.maxVertex.Set(4000, 4000);
```



# HTML5 PARA VIDEO JUEGOS

## Box2D

Las formas definen los datos geométricos de los elementos, y al definir las formas también se definen las propiedades materiales de las cuales estarán formados los elementos, como son la densidad, la fricción, restitución, y las dimensiones.

La forma puede ser un círculo, rectángulo o polígono.

```
// definición de una forma de caja  
var groundSd = new b2BoxDef();  
groundSd.extents.Set(250, 25); //dimensiones half width and half height.  
groundSd.restitution = 0.4;
```

# HTML5 PARA VIDEO JUEGOS

## **Box2D**

La restitución indica la elasticidad que tendrá el objeto al participar en una colisión, los valores posibles son entre 0 y 1, si el objeto tiene restitución uno en caso de ocurrir una colisión con el, será perfectamente elástica, en caso de ser más cercana a cero será casi inelástica.

Si los cuerpos que colisionan tienen distinta restitución, se tomara el valor de restitución del cuerpo que tenga más.

# HTML5 PARA VIDEO JUEGOS

## Box2D

Después de crear una forma se puede crear el cuerpo que tendrá esa forma, en este paso se agrega el cuerpo al mundo, y se establece la posición inicial del cuerpo.

```
var groundBd = new b2BodyDef(); // b2PolygonDef(), b2CircleDef  
groundBd.AddShape(groundSd);  
groundBd.position.Set(250, 370);  
var body = carGame.world.CreateBody(groundBd);
```

# HTML5 PARA VIDEO JUEGOS

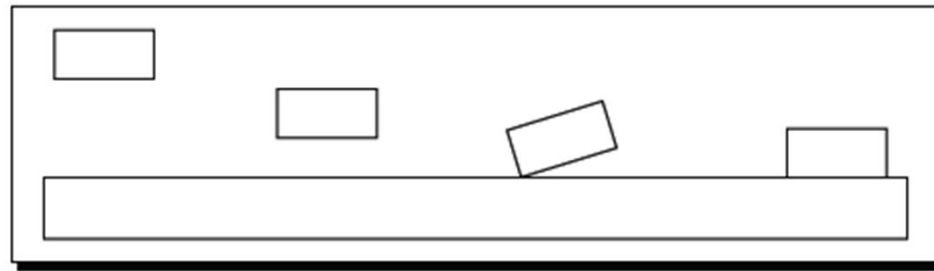
## Box2D

Si al cuerpo no se le asigna una masa, el cuerpo es considerado como un cuerpo estático o fijo. Estos cuerpos no se podrán mover en el futuro y por lo tanto no tendrán colisiones con otros cuerpos estáticos. Estos cuerpos, son usados como piso o paredes. Los cuerpos dinámicos si se podrán mover y serán afectados por la gravedad y podrán sufrir colisiones con otros cuerpos.

```
function createGround() {  
    // box shape definition  
    var groundSd = new b2BoxDef();  
    groundSd.extents.Set(250, 25);  
    groundSd.restitution = 0.4;  
    // body definition with the given shape we just created.  
    var groundBd = new b2BodyDef();  
    groundBd.AddShape(groundSd);  
    groundBd.position.Set(250, 370);  
    var body = game.world.CreateBody(groundBd);  
    return body;  
}
```

# HTML5 PARA VIDEO JUEGOS

## Box2D



# HTML5 PARA VIDEO JUEGOS

## **Box2D**

Es necesario establecer el avance del tiempo en el mundo físico que se ha creado(World step loop). En el mundo físico creado en Box2D, todos los cálculos son hechos sistemáticamente en cada iteración. El mundo calcula las transformaciones físicas de todos los elementos que contiene para el periodo de tiempo correspondiente.

# HTML5 PARA VIDEO JUEGOS

## Box2D

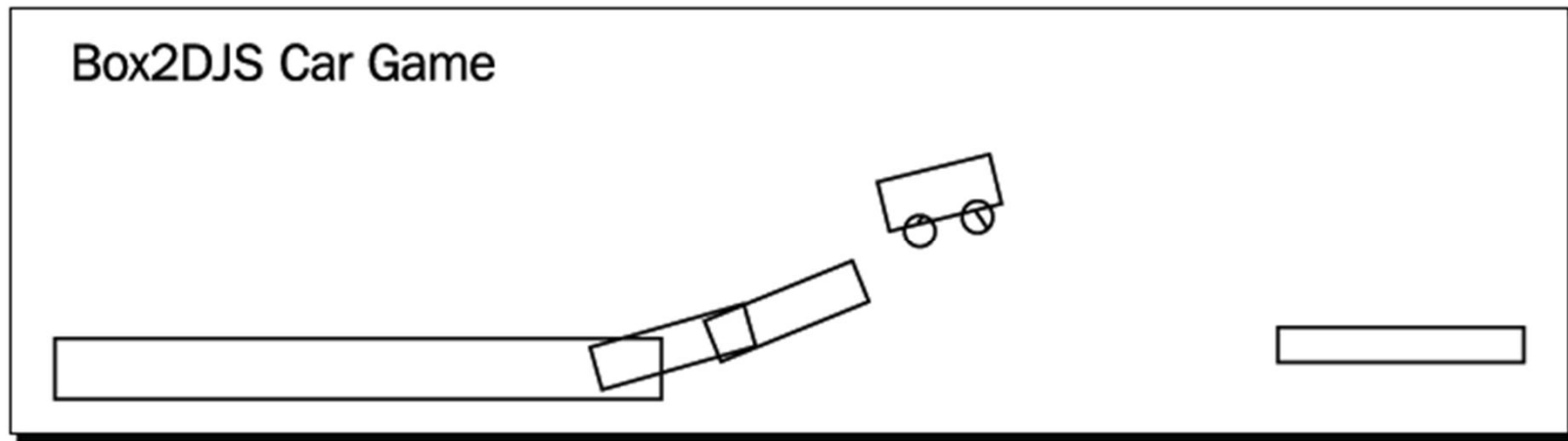
Para establecer el avance del tiempo se crea una función que se ejecuta periódicamente, cada cierta frecuencia predefinida

```
function step() {  
    world.Step(1.0/60, 1);  
    ctx.clearRect(0, 0, canvasWidth, canvasHeight);  
    drawWorld(carGame.world, ctx);  
    setTimeout(step, 10);  
}
```

Para empezar la ejecución del universo se debe invocar la función por primera vez, en alguna parte del código.

# HTML5 PARA VIDEO JUEGOS

## Box2D





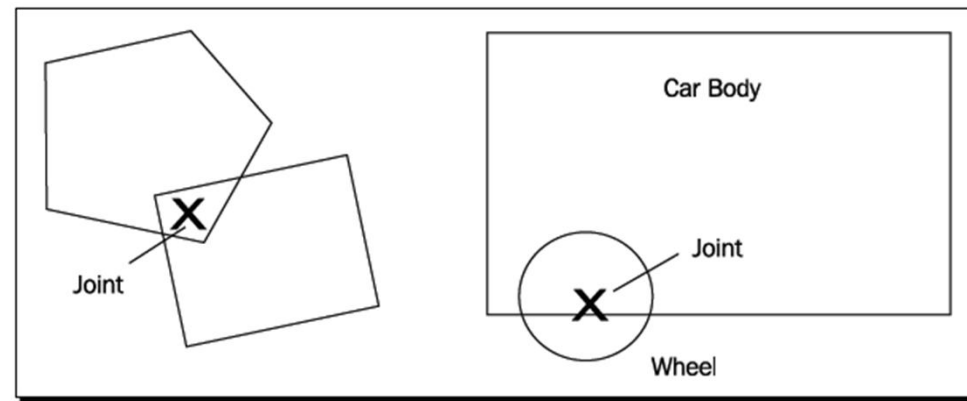
# HTML5 PARA VIDEO JUEGOS

## Box2D

```
function createWheel(world, x, y) {  
    // wheel circle definition  
    var ballSd = new b2CircleDef();  
    ballSd.density = 1.0;  
    ballSd.radius = 10;  
    ballSd.restitution = 0.1;  
    ballSd.friction = 4.3;  
    Building a Physics Car Game with Box2D and Canvas  
    [ 28888]  
    // body definition  
    var ballBd = new b2BodyDef();  
    ballBd.AddShape(ballSd);  
    ballBd.position.Set(x, y);  
    return world.CreateBody(ballBd);  
}
```

# HTML5 PARA VIDEO JUEGOS

## Box2D



# HTML5 PARA VIDEO JUEGOS

## Box2D

```
function createCarAt(x, y) {  
    // the car box definition  
    var boxSd = new b2BoxDef();  
    boxSd.density = 1.0;  
    boxSd.friction = 1.5;  
    boxSd.restitution = .4;  
    boxSd.extents.Set(40, 20);  
    // the car body definition  
    var boxBd = new b2BodyDef();  
    boxBd.AddShape(boxSd);  
    boxBd.position.Set(x, y);  
    var carBody = carGame.world.CreateBody(boxBd);  
    // creating the wheels  
    var wheelBody1 = createWheel(carGame.world, x-25, y+20);  
    var wheelBody2 = createWheel(carGame.world, x+25, y+20);  
    // create a joint to connect left wheel with the car body  
    var jointDef = new b2RevoluteJointDef();  
    jointDef.anchorPoint.Set(x-25, y+20);  
    jointDef.body1 = carBody;  
    jointDef.body2 = wheelBody1;  
    carGame.world.CreateJoint(jointDef);  
    // create a joint to connect right wheel with the car body  
    var jointDef = new b2RevoluteJointDef();  
    jointDef.anchorPoint.Set(x+25, y+20);  
    jointDef.body1 = carBody;  
    jointDef.body2 = wheelBody2;  
    carGame.world.CreateJoint(jointDef);  
    return carBody;  
}
```

# HTML5 PARA VIDEO JUEGOS

## **Box2D**

Las Articulaciones(Joints) son utilizadas para agregar restricciones entre dos cuerpos, o entre un cuerpo y el mundo.

Hay distintos tipos de Articulaciones:

- Mouse Joint
- Distance Joint
- Rope Joint
- Revolute Joint
- Prismatic Joint
- Pulley Joint
- Gear Joint
- Line Joint
- Weld Joint

# HTML5 PARA VIDEO JUEGOS

## Box2D

### Applying force to a body

We can apply force to any body by calling the `ApplyForce` function in that body. The following code shows the usage of the function:

```
body.ApplyForce(force, point);
```

This function takes two arguments, which are listed in the following table:

Arguments	Type	Discussion
<code>force</code>	<code>b2Vec2</code>	The force vector to apply to the body
<code>point</code>	<code>b2Vec2</code>	The point where the force applies

# HTML5 PARA VIDEO JUEGOS

## Box2D

La detección de colisiones es realizada de forma automática por Box2D.

Box2D calcula todas las colisiones que ocurrieron en el mundo a cada paso que es actualizado.

Esta información se obtiene del mundo.

```
Game.world.GetContactList()  
for (var cn = carGame.world.GetContactList(); cn != null; cn =  
cn.GetNext()) {  
    // We have shape 1 and shape 2 of each contact node.  
    // cn.GetShape1();  
    // cn.GetShape2();  
}
```

# HTML5 PARA VIDEO JUEGOS

## Soporte HTML5

	MAC				WIN										
															
	OPERA	FIREFOX	SAFARI	CHROME	OPERA	FIREFOX	SAFARI	IE				CHROME			
	10.63	3.6	5	7	10.63	3.6	4.03	5	6	7	8	9	7	8	
Local Storage	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	92%
Session Storage	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	93%
Post Message	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	94%
Offline Applications	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	77%
Workers	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	78%
Query Selector	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	82%
WebSQL Database	✓	✗	✓	✓	✓	✗	✗	✓	✗	✗	✗	✗	✓	✓	40%
IndexedDB Database	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	
Drag and Drop	✗	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	93%
Hash Change (Event)	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	82%
History Management	✗	✗	✓	✓	✗	✗	✓	✓	✗	✗	✗	✗	✓	✓	37%
WebSockets	✗	✗	✓	✓	✗	✗	✓	✓	✗	✗	✗	✗	✓	✓	36%
GeoLocation	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	80%
Touch	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	

## Bibliografía

HTML5 Games Development by example, Makzan,  
Packt



## Referencias

<http://www.q-rls.com/index.php/browser-compatibility-chart-html5-css3>

<http://radar.oreilly.com/2009/05/google-bets-big-on-html-5.html>

<http://blog.allanbishop.com/box2d-2-1a-tutorial-part-2-joints/>

<http://www.gamefromscratch.com/post/2012/06/04/Cocos2D-HTML5-tutorial-1-Getting-set-up-and-running.aspx>

<http://unpocodejava.wordpress.com/2012/04/30/unpoco-de-jwebsocket/>



HIGHSCORE: 127730  
SCORE: 35290

