Práctica 8

Juego de *"Piedra, papel o tijera"* - II

Integración de Servicios Telemáticos
Máster Universitario en Ingeniería de Telecomunicación
ETSIT-UPV

Ismael de Fez Lava
Marisol García Valls
F. J. Martínez Zaldívar

Índice

1. Introducción y objetivos

En la presente práctica vamos a seguir utilizando sobre todo JavaScript y el entorno Node.js y quizá puntualmente HTML y CSS, para acabar de diseñar un sencillo juego como es el de *"Piedra, papel o tijera"*, finalizando algunos detalles que quedaban pendientes de realizar, por lo que el marco de trabajo y el contexto siguen siendo los mismos que los de la práctica anterior.

Las peculiaridades que se añadirán al juego tendrán implicación en la modificación del código tanto en el servidor como en el cliente, tanto en la lógica del juego como en las comunicaciones.

En esta serie de prácticas vamos a seguir utilizando lo estudiado en clases de teoría. Aunque las especificaciones pueden ser incompletas, deberían ser suficientes para conseguir un aspecto bastante similar a las referencias gráficas o sugerencias que aquí puedan aparecer.

Durante el transcurso de la práctica, el alumno podrá apreciar que una misma idea puede implementarse de múltiples formas; por ello se pretende que se reflexione sobre estas alternativas y elija la opción que le parezca más conveniente, tratando de justificarla adecuadamente en forma de comentarios dentro del código HTML, CSS o JavaScript.

Asimismo, es probable que deba consultar algunos detalles de implementación todavía no impartidos en la asignatura; la intención de ello es ejercitar y valorar oportunamente la competencia transversal de *aprendizaje permanente*.

1.1. Organización de la práctica

Los grupos de prácticas pueden ser de uno o de dos alumnos. Se sugiere que estos grupos sean estables a lo largo de todas las prácticas, y esta especialmente, ya que aprovecha los resultados de la práctica anterior y además tendrá continuidad y reaprovechamiento de todo el trabajo realizado para la siguiente. Se dedicará una sesión de dos horas para la realización de la presente práctica.

Se seguirá trabajando con el mismo repositorio GitHub que la práctica anterior (`Peri-digi-`), acumulando nuevos *commits* con los que se va a ir dotando al juego de características adicionales. Antes de finalizar la sesión de prácticas en el laboratorio, debe realizarse un *commit/push* con etiqueta `pr8SesionFin` con el contenido de la práctica, esté en el estado que esté. Si quedan detalles pendientes de finalizar, recuérdese que se dispone de una semana adicional para finalizarlos, en cuyo caso, el *commit* que se considere que representa el estado final de esta práctica deberá tener la etiqueta `pr8DefFin`.

1.2. Material necesario

El material necesario para llevar a cabo la práctica será un ordenador de sobremesa o un portátil, con cualquier sistema operativo convencional, un editor de texto plano y un navegador (preferiblemente alguno que posea y del que se conozcan y manejen capacidades de depuración).

Adicionalmente es necesario tener instalado Node.js, lo cual se supone de la realización de la práctica anterior.

Muy probablemente será necesario consultar información sobre HTML, CSS, JavaScript y Node.js. Recuérdese que un sitio bastante útil es `http://www.w3schools.com`, entre otros.

2. Adición de peculiaridades en el juego

Se van a añadir las siguientes características:

- Un servidor sólo atenderá una partida jugada por dos jugadores, por lo que:
  - Habrá que controlar que sólo participen en el juego, dos clientes
  - Si un tercer cliente pide jugar, entonces se le enviará un mensaje de tipo *rechazo*. La reacción del cliente frente a un mensaje de rechazo será ir a una página web denominada por ejemplo `rechazo.html` en la que se mostrará un texto en el que se indica que debe esperar y un botón con el que se reintente de nuevo la conexión al juego, es decir, la *reapertura* de la página `ppt.html`.

    Recuérdese que una forma de abrir una página web desde JavaScript puede ser:

    ppt.js (parte)
    ```
    window.open(URL_string, TARGET_string);
    ```

    donde `URL_string` sería el *string* que representa la URL que se pretende abrir (por ejemplo, en nuestro caso podría ser `"ppt.html"`) y el `TARGET_string` el *string* que indica dónde abrir dicha URL, si aquí mismo (`"_self"`), o bien en una nueva pestaña (`"_blank"`); en nuestro caso se sugiere que sea *aquí* para verlo en el mismo *frame*.

- Si un cliente es aceptado como jugador, entonces debe aparecer un *prompt* preguntando el nombre del jugador

  ppt.js (parte)
  ```
  let mi_nombre = prompt("Nombre:");
  ```

El nombre será enviado al servidor y este lo reenviará oportunamente al jugador contrincante para que lo pueda escribir en su pantalla. Debe gestionarse correctamente la comunicación de los nombres para que en cualquier caso, ambos jugadores posean el nombre propio y el del contrincante.

Puede observarse del código HTML en el fichero `ppt.html` que existen sendos `span` cuyos `id` son `"mi_nombre"` y `"nom_cont"` respectivamente y cuya intención es albergar el nombre del jugador y el del contricante. Utilizando `jQuery` resulta muy fácil actualizar dichos `span`:

ppt.js (parte)

```
$("#nom_cont").text(NOMBRE_DEL_CONTRINCANTE);
...
$("#mi_nombre").text(MI_NOMBRE);
```

- Un jugador podrá apostar (recuadrándose la imagen clicada) sólo si hay un contrincante ya conectado y solo podrá hacerlo una vez por partida, es decir, frente a sucesivos *clicados* sobres las figuras, solo el primero tendrá efecto.

  Una forma de controlar esta situación es *activar* o *reactivar* la función a ejecutar cuando ocurra un evento *click* sobre las figuras. Ya estaba incluido en la plantilla el código asociado a la función a ejecutar con el clicado:

ppt.js (parte)

```
// envío del mensaje apuesta, con el string
//      <eleccion> = 'piedra' | 'papel' | 'tijera'
// Ademas, se debería poner borde a la figura clicada
$("div.yo div.tercio:nth-child(2) div.imagenes img").click(function(e) {
    let eleccion = this.id;
    poner_borde(YO, eleccion)
    socket.emit('apuesta', eleccion);
});
```

  Para desactivar esta acción, podría llevarse a cabo empleando el método jQuery `.off()` de la siguiente manera:

ppt.js (parte)

```
$("div.yo div.tercio:nth-child(2) div.imagenes img").off('click');
```

  Reflexiónese dónde puede introducirse la anterior sentencia para conseguir el objetivo deseado.

- Por defecto, las imágenes del contrincante deberían ser traslúcidas. Si el contrincante ya ha apostado, pero nosotros no, entonces deberían aparecer todas las imágenes de los logos del contrincante con interrogantes; para ello solo habrá que ejecutar la instrucción

ppt.js (parte)

```
interrogantes(ON);
```

- Una vez que ambos jugadores hayan apostado y consecuentemente el servidor haya difundido las apuestas de ambos, en la página de cada jugador debe aparecer en modo *opaco* y recuadrada la imagen o logo correspondiente a la apuesta del contrincante (sin ningún interrogante en ninguna figura: `interrogantes(OFF);`), junto con la suya también recuadrada.

- Una vez mostrados los resultados de las apuestas, para arrancar una nueva partida, ambos jugadores deben pulsar el botón `"Jugar otra vez"`, cuyo id es `"otra_vez"`. Hasta ese momento, el boton debería estar *deshabilitado*, lo cual se consigue con:

```
$("#otra_vez").attr("disabled", true);
```

  Para habilitarlo, entonces habrá que ejecutar:

```
$("#otra_vez").removeAttr("disabled");
```

  El primero que pulse dicho botón deberá provocar el envío de un mensaje al servidor, y dicho servidor enviará un mensaje de cierto tipo al contrincante, reaccionando dicho contrincante poniendo en intermitente el botón `"Jugar otra vez"` (cuyo `id` es `"otra_vez"`), adivirtiendo así al jugador que su contrario ya pulsó su botón y lo está esperando para reanudar una nueva partida.

ppt.js (parte; intermitencia ON y OFF)

```
// intermitencia
let intermitencia;
...
intermitencia = setInterval(function(){
    $("#otra_vez").fadeTo(250,0);
    $("#otra_vez").fadeTo(250,1);
}, 550);
...
// no intermitencia
clearInterval(intermitencia);
```

- Mientras no se cambie de jugadores hay que ir acumulando la cantidad de partidas que cada uno va ganando, para lo cual de nuevo puede accederse a ciertos `span` para ello:

ppt.js (parte)

```
$("#mis_puntos").text(MI_PUNTUACION);
...
$("#sus_puntos").text(PUNTUACION_CONTRINCANTE);
```

- En cualquier estado de la partida, cualquier jugador puede abandonarla, lo cual deberá ser tenido en cuenta, con la advertencia oportuna al jugador que sigue en la partida.

2.1. Estrategia de programación

Tal y como ocurría con prácticas anteriores, se sugiere que se vayan afianzando poco a poco los pasos que se vayan dando. Se sugiere que cada nueva característica añadida al juego sea verificada antes de añadir la siguiente, si estas son independientes.

Empléese la consola del inspector de código HTML y utilícese en el código sentencias de tipo `console.log()` que ayuden a que el seguimiento del código sea sencillo. Apóyese en el depurador o *debugger* para encontrar fallos en el funcionamiento del software diseñado.

3. Resultados a entregar

Actualícese el repositorio de GitHub con los ficheros que cumplan las especificaciones requeridas a lo largo del enunciado de la presente práctica. Recuérdese las condiciones de la entrega para evitar una merma en la calificación de la práctica.