

Introducción

Recordamos que en este momento nuestro videojuego era capaz de cargar un conjunto de imágenes de forma dinámica en el documento HTML y poco más.

Desde ese punto hemos aprendido a manejar los posibles eventos que puedan ocurrir en el navegador como fruto de las interacciones del usuario o bien debido a cambios de estado en peticiones de red o eventos que nos envían otros API's.

Bien, pues esto es lo que vamos a aplicar a nuestro videojuego en este Lab. El manejo de eventos producidos por el usuario.

Por el momento lo que haremos será, para también poner en práctica lo que hemos aprendido acerca de la geometría de los elementos en pantalla, detectar sobre cuál de las imágenes el usuario ha hecho click, y resaltarla con un borde.

Existen dos maneras de hacer esto, la más sencilla sería añadir manejadores de eventos a cada objeto letter, pero así no estamos poniendo en práctica lo aprendido acerca de la geometría de los elementos en pantalla, por tanto lo que haremos, que además nos servirá para más adelante, será detectar el click a nivel del documento, y obtener el objeto situado en ese punto utilizando las funciones de geometría de los elementos aprendidas.

Tomaremos como punto de partida el proyecto resultante del anterior Lab. Haced una copia de seguridad por si acaso necesitáis volver a ese estado durante el desarrollo de este lab.

Paso 1: Detectar el click

Para detectar el click a nivel del documento, debemos añadir el manejador correspondiente en el objeto más alto del DOM que contenga todas las imágenes de las letras. Éste objeto, será el tablero, por tanto:

```
LETTERS.Board = function(htmlId) {  
    ...  
    this.el.addEventListener('click', this.clicked, false);  
};  
  
LETTERS.Board.prototype = {  
    ...  
    clicked: function(evt) {  
        var element = document.elementFromPoint(evt.clientX,  
        evt.clientY);  
        console.log(element.src);  
    }  
};
```

Hemos modificado tanto el constructor como el prototype de nuestra clase Letter y hemos añadido el manejador del evento click sobre el elemento HTML del tablero a la función clicked del propio tablero.

En esta función obtenemos el elemento que había bajo el ratón cuando se hizo click y, para comprobar que funciona, mostramos por consola su atributo src. Recordad que el elemento obtenido por `document.elementFromPoint` será un elemento HTML, no nuestro objeto Letter.

Paso 2: Resaltando el elemento clicado

En este punto vamos a adelantar un pequeño concepto que aún no hemos visto durante la explicación, pero que en breve lo haremos y es la modificación de propiedades CSS con javascript. Por el momento no es necesario que entendáis este código, aunque sin duda, lo entenderéis por que es muy sencillo.

```
clicked: function(evt) {
    var element = document.elementFromPoint(evt.clientX, evt.clientY);
    element.setAttribute('style', 'border:5px solid red');
}
```

Hemos sustituido la línea `console` una línea que añade un marco rojo al elemento sobre el que se hizo click.

Si probamos este código, veremos como efectivamente, haciendo click sobre las imágenes se les coloca el marco rojo, pero además, si hacemos click fuera de las imágenes, en el área blanca, el marco rojo se coloca en el borde del navegador. Esto es normal, ya que el área del tablero se extiende por toda la pantalla, y no hemos limitado que solamente se pueda añadir marco rojo a las imágenes de las letras.

Vamos a arreglar este código añadiendo una comprobación, para que solamente se pueda añadir el marco rojo a las imágenes de las letras, y además, el marco rojo solamente esté en una de ellas.

```
LETTERS.Board = function(htmlId) {
    ...
    this.el.addEventListener('click', clickClosure, false);

    var self = this;
    function clickClosure(evt) {
        self.clicked(evt);
    }
};

LETTERS.Board.prototype = {
    ...
    clicked: function(evt) {
        var element = document.elementFromPoint(evt.clientX,
evt.clientY);
        var clickedLetter = this.letters.filter(function(letter) {
            return letter.el == element;
        })[0];
        if (clickedLetter) {
            this.letters.forEach(function(letter) {
                letter.el.setAttribute('style', 'border: none');
            });
            element.setAttribute('style', 'border:5px solid red');
        }
    }
};
```

En este cambio hemos hecho uso de muchos conceptos estudiados en este curso.

Lo primero que hemos hecho ha sido utilizar una closure para poder definir el manejador del evento click sobre el tablero. Esto es necesario porque dentro del manejador queremos tener acceso a las propiedades del tablero y, si no lo hiciéramos así, sería imposible, ya que el contexto de ejecución del manejador es el propio elemento HTML sobre el que se hizo click.

Una vez entendido esto, y si no está entendido volved atrás y repasarlo, y sino, preguntad. Una vez entendido esto pasamos a analizar el código del método clicked.

En este método, el contexto de ejecución sí es el objeto tablero, porque se ha invocado como método en la closure.

Analizando el código del método clicked, hacemos una búsqueda en el array letters, que recordemos, contiene referencias a todos los objetos letter, para encontrar el objeto letter cuyo elemento HTML sea el que ha recibido el click. Así nos aseguramos que solamente las letras serán consideradas en este comportamiento.

Si encontramos el objeto quiere decir que el elemento clicado fue una letra, y por tanto, modificamos el atributo border de todas las letras, y a continuación se lo asignamos a la letra clicada.

Este código hace uso de dos métodos funcionales de JavaScript filter y forEach. Podréis encontrar documentación de estos métodos y otros en la documentación de array:

https://developer.mozilla.org/en-US/docs/JavaScript/Reference/Global_Objects/Array

Nuevamente, antes de terminar aseguraos de que, el proyecto funciona perfectamente y además, entendéis todo el código detalladamente.

Fin.