## Introducción

Ha llegado el momento de dar algo de visibilidad a nuestro juego. Algo visualmente atractivo.

Lo que haremos será programar una animación de entrada de los elementos.

## Paso 1: Función de utilidad

Igual que anteriormente programamos una función de utilidad que manejaría el comportamiento de una ficha cuando ésta estuviera siendo arrastrada, ahora haremos lo mismo cuando queramos animar una ficha para que se mueva de un punto a otro.

Creamos un nuevo archivo utils/slide.js y rellenamos con este código:

```
function slide(element, target, duration, oncomplete) {
  var xInterval = target.x - element.getBoundingClientRect().left;
  var yInterval = target.y - element.getBoundingClientRect().top;

  var start = (new Date()).getTime();

  var css = window.getComputedStyle(element, null);
  var initialPosition = {x: parseInt(css.left), y: parseInt(css.top)};
  animate();

  function animate() {
    var elapsed = (new Date()).getTime() - start;

    if (elapsed < duration) {
      var deltaX = xInterval * (elapsed / duration);
      var deltaY = yInterval * (elapsed / duration);

      element.style.left = (initialPosition.x + deltaX) + "px";
      element.style.top = (initialPosition.y + deltaY) + "px";

      setTimeout(animate, 1000 / 24);
    } else {
      element.style.left = target.x + "px";
      element.style.top = target.y + "px";
      oncomplete();
    }
  }
}
```

Lo que esta función hace es calcular el intervalo que debe desplazar el elemento en cada eje. A continuación almacena la posición inicial y el instante inicial y arranca la animación.

En la función recurrente animate (ojo, he dicho recurrente, no recursiva. Una función es recurrente cuando se invoca repetidas veces, recursiva es cuando se invoca a sí misma) calcula el tiempo transcurrido hasta ese instante de tiempo y basándose en ese tiempo, determina la posición intermedia en que debería estar el objeto y lo coloca. En caso de que el tiempo de animación haya transcurrido, nos

aseguramos de colocar al elemento en la posición final e invocamos al callback proporcionado.

Ahora nos falta utilizar esta función. La utilizaremos desde el tablero, ya que es éste quien sabe cuándo ha colocado el elemento en el tablero y, por tanto, cuándo es visible. Modificaremos el código de la función loadLetters para que quede así:

```
loadLetters: function(images) {
  for (var i = 0 ; i < images.length ; ++i) {
    var img = images[i];
    var letter = new LETTERS.Letter(img);
    this.letters.push(letter);
    this.el.appendChild(letter.el);
    var css =  window.getComputedStyle(letter.el, null);
    slide(letter.el,
      { x: parseInt(css.width) * i, y: parseInt(css.top) },
      1000,
      function() {});
  }
}
```

Estas son las modificaciones, fijaros que hemos quitado un argumento del constructor de Letter, porque solo se utilizaba para colocarlo en pantalla, y como ahora se hace aquí, ese código sobra. Por otra parte, hemos añadido la invocación a la función slide.

Finalmente, el código del constructor de Letter quedará así:

```
LETTERS.Letter = function(source) {
  this.el = document.createElement('img');
  this.el.src = source;

  this.el.classList.add('letter');

  this.el.addEventListener('mousedown', clickClosure, false);

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

Nada de especial, simplemente hemos quitado todo el código relativo a la posición del elemento en pantalla, ya que ahora se hace desde otro sitio más adecuado porque es el tablero quien tiene la información de dónde debe colocar cada ficha.

Aquí finaliza el Lab, como siempre, no lo déis por terminado con dudas sin resolver.