

FERNETJS

fernet + javascript

[JavaScript](#) [NodeJS](#) [Historia](#) [FAQ](#)

Animando sprites con HTML5: Canvas

2013-06-26 • [Pablo Novas](#)

Siguiendo con la serie de posts de juegos en canvas, voy a continuar los posts anteriores (si no los viste: [Dibujando en CANVAS](#), [Animando en CANVAS](#), [Construyendo un Game Loop](#)).

Vamos a armar una animación utilizando Sprites. Que es un Sprite?, simplemente una imagen con distintos estados de un objeto en el cual mostrar de a uno y en orden se ve el objeto animado. Es decir, vamos “cortando” la imagen con cada objeto y al mostrar y ocultar en un muy corto tiempo se genera una animación.

Para el caso de este ejemplo me descargué un Sprite del ActionDoom desde [acá](#), que luego organicé un poco para que me sea mas simple manejarlo desde el javascript, quedando en lo siguiente:



Seccionando el sprite

Para poder usarlo desde javascript, tenemos que pensar en una grilla virtual, lo que podemos traducir a una matriz de posiciones, mientras que cada celda tenga el mismo ancho y alto:



Ahora que tenemos nuestra grilla imaginaria, vamos a definir que celdas corresponden a cada estado, en este caso la separe que colores para que sea visible, siendo:

Libre

Acción

Recarga



Todo listo, ahora armemos la matriz de referencia y animaciones en el código:

```
// defino los tipos de armas que voy a tener
var armas = ['cuchillo', 'pistola', 'escopeta', 'grosa'];

// defino para cada arma donde están sus estados en la matriz
var sprites = {
  dimension: {
    ancho: 200,
    alto: 200
  },
  cuchillo: {
```

```

    libre: [0,0],
    accion: [[1,0],[2,0],[1,0]]
  },
  pistola: {
    libre: [0,2],
    accion: [[0,2],[1,2],[2,2],[3,2],[4,2],[5,2]],
    recarga: [[0,1],[1,1],[2,1],[3,1],[4,1],[0,1]]
  },
  escopeta: {
    libre: [0,3],
    accion: [[3,0],[4,0],[5,0]],
    recarga: [[1,3],[2,3],[3,3],[4,3],[5,3]]
  },
  grosa: {
    libre: [0,4],
    accion: [[1,4],[2,4],[3,4],[4,4],[5,4]]
  }
};

```

Como se puede ver, hay celdas que se repiten y es la idea, es común que usemos una “celda” varias veces en la misma animación.

Animando en el Game Loop

La animación se produce sola, simplemente por cada paso del game loop (actualizar y dibujar), actualizamos las coordenadas en la imagen, borramos lo que hay y dibujamos uno nuevo.

Lo que hacemos primero es dejar el game loop corriendo, cuando el usuario presiona una tecla, guardamos que arma seleccionó y su estado:

```

// Asignamos el callback al evento keypress
document.addEventListener('keypress', asignarArmaYEstado);

// estadoAnterior, estado y arma son variables globales
// del módulo en el game loop.

function asignarArmaYEstado(evento){
  var codigo = evento.keyCode;
  // dependiendo del codigo asignamos el arma y el estado

  switch(codigo){
    case 97: // Letra A
      estadoAnterior = estado;
      estado = 'accion';

```

```

        break;
    case 114: // Letra R
        estado = 'recarga';
        break;

    // Números del 1 al 4
    case 49:
    case 50:
    case 51:
    case 52:
        arma = armas[codigo-49];
        break;
    }
}

```

En la próxima corrida del actualizar en el game loop, nos fijamos cual es el arma y su estado actual, para actualizar la posición de la matriz:

```

// La variable pos mantiene la posición a dibujar
// y es otra variable global del modulo en el game loop

function actualizar(){
    var x, y,
        sprite = sprites[arma][estado]; //la celda en la matriz

    // si recién cambio de estado, volvemos el índice a 0
    if (estado !== estadoAnterior){
        animContador = 0;
    }

    // si el índice es es el último no hay mas para animar,
    // así que volvemos el estado a "libre"
    if(animContador === sprite.length-1){
        estado = 'libre';
        sprite = sprites[arma][estado];
    }

    // si no es "libre", hay que seguir con la animación,
    // así que incrementamos el índice de la animación
    else if(estado !== 'libre') {
        sprite = sprite[animContador];
        animContador++;
    }
}

```

```
// calculamos cual es la x e y real en la imagen
// partiendo de la posición de la "celda" y la dimensión de cada sprite
x = sprite[0] * sprites.dimension.ancho;
y = sprite[1] * sprites.dimension.alto;

// asignamos las coordenadas para dibujar
pos = {
  x: x,
  y: y,
  w: sprites.dimension.ancho,
  h: sprites.dimension.alto
};
}
```

Por último al dispararse el dibujar, utilizamos el `drawImage` de canvas, pero con unas cuantas opciones mas para poder dibujar **sólo** la “celda” del estado y arma que queremos:

Para información sobre el `drawImage`, está muy bien explicado [acá](#)

```
function dibujar(){
  //context.drawImage(<Objeto Image>, <X Sprite>, <Y Sprite>, <Ancho Sprite>, <Al
  contexto.drawImage(imgSprites, pos.x, pos.y, pos.w, pos.h, 0, 0, pos.w, pos.h);
}
```

Les dejo un Fiddle con el resultado (click adentro del fiddle para usar):

<http://jsfiddle.net/pjnovas/4PvSm>

Compartí este post en [Twitter](#) o en [Facebook](#). [Seguinos en twitter](#) para mantenerte al tanto, o seguí al [autor](#) de este artículo. ¿Algún problema? [Creá un issue!](#).

6 Comentarios fernetjs

1 Iniciar sesión ▾

Recomendar 1  Compartir

Mejores primero ▾



Unite a la discusión...



Mauricio • hace 4 años

Mauricio

```
// Hacele caso a tu sed
var sprites = ['normal', 'light', 'zero'];
2 ^ | v • Responder • Compartir ›
```



José Alejandro Urieles Montaña • hace 2 años

en ejemplo no funciona, así me es difícil confiar en que la info está bien, gracias de todas maneras

^ | v • Responder • Compartir ›



pjnovas Moderador ➔ José Alejandro Urieles Montaña • hace 2 años

José, como estás?. en qué navegador estás abriéndolo?, a mí me funciona con Chrome.
Saludos!

^ | v • Responder • Compartir ›



Andkristur • hace 4 años

El jsfiddle no funciona.

^ | v • Responder • Compartir ›



pjnovas Moderador ➔ Andkristur • hace 4 años

fíjate la consola, tira algún error?, con qué navegador lo estás abriendo?
Saludos!

^ | v • Responder • Compartir ›



Ezequiel Alvarez • hace 4 años

var armas = ['cuchillo', 'pistola', 'escopeta', 'grosa'];
Muy bueno locos.

^ | v • Responder • Compartir ›

TAMBIÉN EN FERNETJS

Aprendiendo Bootstrap en 5 días

13 comentarios • hace 4 años •



pjnovas — Agrego un link para los que están en la misma o parecida situación, por ahí les ayuda <http://www.layoutit.com/>

jQuery .animate y .stop

1 comentario • hace 2 años •



Elkin — Waaoo, que gran información..
Gracias por compartir.

NodeJS a la nube con NGINX en un VPS

18 comentarios • hace 4 años •



Brian Eduardo Craig — Ya lo subí a GitHub, <https://github.com/BrianCraig>, igual todavía le falta muuucho, cuando se pueda probar ...

NodeJS

22 comentarios • hace 2 años •



Kevin Villalobos — ¡Hola! La verdad es que este blog me parece un muy buen recurso para perfeccionar mi conocimiento en NodeJS. Me

El código de este sitio está disponible en [github](#) y quien quiera puede colaborar. Si encontrás algún problema reportalo [en github issues](#) . Cualquier ayuda será más que bien recibida!