

Animando SVG con GreenSock

@evaferreira92

¿Por qué no CSS?

GreenSock

- GreenSock Animation Platform
- Data de 2007/2008 con Adobe Flash
- Anima elementos, no solo SVG



GreenSock

- IE7+ con elementos de HTML
- IE9+ con elementos SVG



GreenSock

— — —

- Foro fantástico:
www.greensock.com/forums
- Ejemplos:
www.greensock.com/examples-showcases



Animando SVG con GreenSock

Animando SVG con GreenSock

— — —

1. Planificar la animación
2. Preparar los assets
3. Conseguir GSAP
4. ¡Animar!

1

Planificar la animación

Vaca abducida por un OVNI

— — —

2

Preparar los assets

- Silueta de vaca
- OVNI
- Opcionales
 - Árboles
 - Estrellas
 - Nubes
 - Luz abductora (?)

Preparar los assets

— — —

- https://www.freepik.com/free-vector/variety-of-colorful-ufo_799060.htm#term=spaceship&page=1&position=2
- https://www.freepik.com/free-vector/pack-of-farm-animals-ilhouettes_1058750.htm#term=cow&page=1&position=8

Preparar los assets

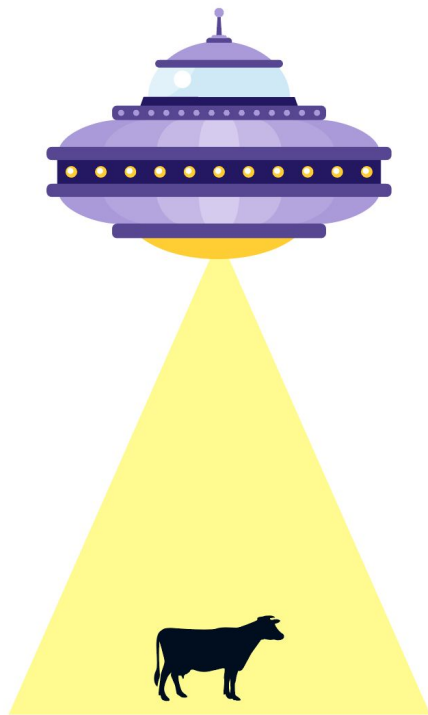


Preparar los assets



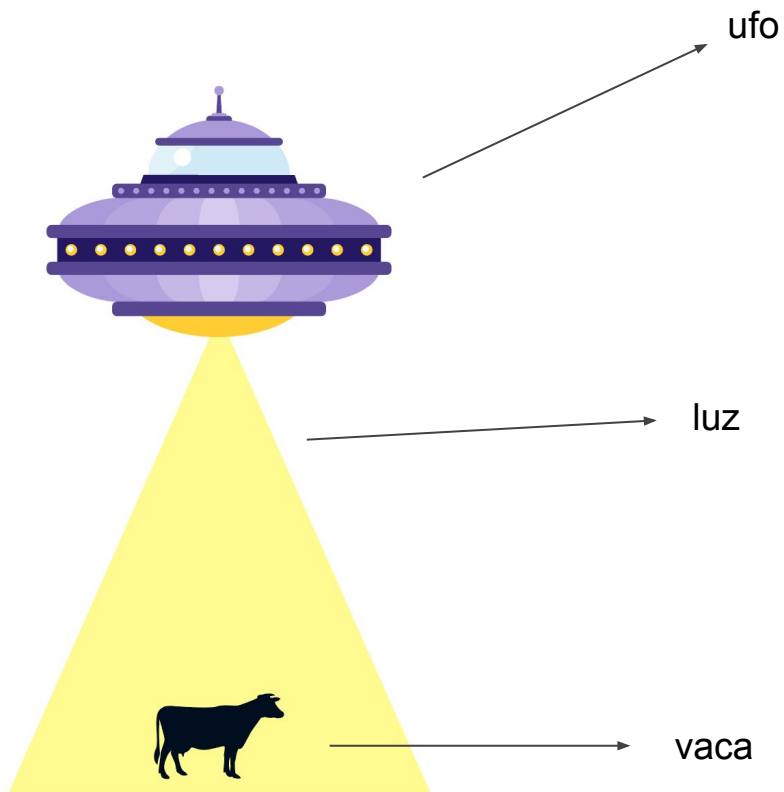
Escena

— — —



Verificar nombres

— — —



Exportar SVG

— — —

Archivo -> Exportar para pantallas -> SVG

Optimizer SVG

— — —

<https://jakearchibald.github.io/svgomg/>

- Prettify code
- No eliminat IDs

Verificar que todo siga funcionando

3

Conseguir GSAP

Creamos un HTML y enlazamos
GreenSock.

Pero... ¿Cuál de todos?

Enlazando GreenSock

— — —

- TweenLite
- TimelineLite
- TweenMax
- TimelineMax

Enlazando GreenSock

— — —

- TweenLite: Simple e independiente
- TimelineLite: Chaining
- TweenMax: Chaining y controles
- TimelineMax: Chaining, controles, repetición y más

4

¡Animar!

Yay!

Beware of localhost

Seleccionando elementos del SVG

Trabajando SVG con GreenSock

— — —

Archivo SVG inline en el HTML

```
<svg>  
  ...  
</svg>
```

Archivo SVG externo

```
<iframe src="escena.svg"></iframe>
```

Trabajando SVG con GreenSock

— — —

Archivo SVG inline en el HTML

```
let vaca =  
document.getElementById("vaca");  
  
console.log(vaca)
```



Archivo SVG externo

```
let vaca =  
document.getElementById("vaca");  
  
console.log(vaca)
```



Archivos externos

— — —

```
<iframe src="escena.svg" id="ilustracion"></iframe>
```

```
document.addEventListener("DOMContentLoaded", () => {  
    window.addEventListener("load", () => {  
        let doc = document.getElementById("ilustracion").contentDocument;  
        let vaca = doc.getElementById("vaca");  
    }, false);  
});
```

Tweens

Animaciones independientes

```
TweenLite.to(element, duration, {transformations});
```

Tweens, from 0 to

```
TweenLite.from(vaca, 6, {x: -500});
```

```
TweenLite.to(vaca, 6, {x: 500});
```

Tweens, set initial state

```
TweenLite.set(vaca, {opacity: 0});
```

Opciones

— — —

Delays:

```
TweenLite.to(element, 5, {x: 300, delay: 2});
```

Easings:

```
TweenLite.to(element, 5, {x: 300, ease: Power0.easeNone});
```

<https://greensock.com/ease-visualizer>

Opciones

— — —

Callbacks:

```
TweenLite.to(element, 5, {x: 300, onComplete: myFunc});
```

<https://greensock.com/docs/TweenLite/TweenLite>

Encadenando animaciones

Encadenar animaciones con Tweens

```
TweenLite.to(element1, 2, {x: 30, rotation: 0});
```

```
TweenLite.to(element2, 1, {x: 30, rotation: 0, delay: 2});
```

Timeline

Chaining animations

Encadenando animaciones

```
var tl = new TimelineLite();
```

```
tl.from(vaca, 2, {x: -1000})  
  .to(vaca, 5, {y: -240});
```

Controles

¡PAUSA!

Controles

```
var tl = new TimelineLite({paused: true});
```

```
button.addEventListener("click", () => {  
    tl.play()  
});
```