

Programación Avanzada 2021-2

Taller de Programación 4

Inyección SVG y eventos.

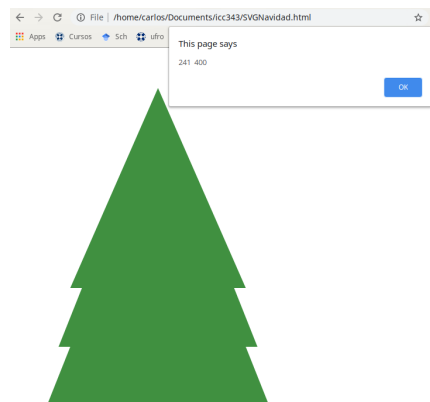
En cada problema formulado usted debe incluir claramente el fondo de pantalla de su escritorio y, en esta, la hora de su PC y parte del fondo de escritorio. Las fotos resultados deben subirse en el chat personal IDENTIFICANDO CLARAMENTE la respuesta de acuerdo a lo pedido, es decir, la etiqueta señalada y el contenido esperado.

Problema 1. (25%). Estructura básica

En esta actividad usted confeccionará un programa HTML-Javascript con gestión de eventos SVG. Replique el siguiente programa.

```
1 <!doctype html>
2 <html>
3 <head>
4   <script>
5     function clicEnArbol() {
6       alert(event.clientX+" "+event.clientY);
7     }
8   </script>
9 </head>
10 <body>
11   <svg height="650" width="500">
12     <polygon points=    "250,60 100,400 120,400
13                        80,500 100,500 60,600
14                        440,600 400,500 420,500
15                        380,400 400,400" fill="#409040"
16     onclick="clicEnArbol()" />
17   </svg>
18 </body>
19 </html>
```

Esto provoca la siguiente salida. Nota que al hacer clic sobre la figura aparece la coordenada sobre la cual se hizo el clic.



En este problema se pide lo siguiente. Usted debe “empaquetar” este código en una clase Javascript llamada `ArbolNavidad`. La clase debe tener la propiedad de inyectar el código completo (el árbol completo y el evento onclick debe invocar un método de la misma clase. Recuerde que en el Taller pasado se sugirió que use una variable global para manejar los eventos y sea posible que la misma clase los gestione. Recuerdo entonces que:

- a) Debe definir la variables `activeObjectArbolNavidad`
- b) Que en el constructor debe tener la instrucción
`activeObjectArbolNavidad = this;`
- c) Que para asignar las funciones de eventos debe usar el nombre del objeto, por ejemplo
`setAttribute("onclick","activeObjectArboNavidad.clicEnArbol()");`

El código que debe funcionar debe ser el siguiente:

```
<div id="navidad"></div>
<script>
var miarbol = new ArbolNavidad();
miarbol.muestra("navidad");
</script>
```

Adicionalmente en cada clic debe agregar un pequeño círculo amarillo, emulando una luz de navidad. No sólo debe agregar la luz de navidad, usted debe además agregar la coordenada en una estructura interna de la clase llamada `luzamarilla`, Adicionalmente en cada clic debe agregar un pequeño círculo amarillo, emulando una luz de navidad. No sólo debe agregar la luz de navidad, usted debe además agregar la coordenada en una estructura interna de la clase llamada `luzamarilla`, debe agregar las siguiente líneas en el código del evento `clicEnArbol`:

```
this.luzamarilla.push({x:event.clientX,y:event.clientY});
console.log(this.luzamarilla);
```

Para demostrar que lo ha hecho debe incluir las siguientes entregas:

TP4-1-A. Foto del código de la clase `ArbolNavidad.js`

TP4-1-B. Foto del código HTML especificado

TP4-1-C. Foto de la salida en el navegador con 10 luces.

TP4-1-D. Dos fotos consecutivas de la consola entre mostrando las posiciones de “las luces”

Problema 2. (25%). Interruptor de luces

En este problema usted debe agregar una base en el árbol partida en dos pequeños rectángulos. En este problema se pide agregar la siguiente funcionalidad a la clase ArbolNavidad.

El primer rectángulo debe cambiar de color entre gris y amarillo al recibir un clic. Este rectángulo actuará como un switch, cuando está en gris el switch está apagado que será su color inicial, cuando está en amarillo provocará que las luces amarillas parpadeen. La forma de parpadear será bajo la siguiente condición, el conjunto de luces debe estar dividido, al menos en tres grupos. Cada grupo tendrá una velocidad diferente de encendido y apagado, en 400, 600 y 1000 milisegundos.

Debe entregar:

- TP4-2-A. Foto del código de la clase ArbolNavidad.js
- TP4-2-B. Foto del código HTML especificado
- TP4-2-C. Foto de la salida en el navegador
- TP4-2-D. Código fuente HTML y JS como archivos adjuntos

Problema 3. (25%). Familia de luces

En este problema programará el segundo rectángulo. Este rectángulo tendrá el color amarillo y cambiará, con cada clic en él, a un siguiente color. La lista de colores es amarillo, luego rojo, luego azul. Ahora, cuando haga clic sobre el árbol se debe, igualmente agregar una luz en el árbol, pero del color seleccionado.

Se pide un cambio en la estructura del programa, ahora, cada objeto luz, no sólo será un par $\{x,y\}$, sino una cuádrupla $\{x,y,color,circuito\}$. El valor de circuito irá entre 1 y 3, y corresponderá al grupo de luces que se apagan y prenden juntas.

Después de estos cambios el encendido y apagado de luces debe seguir funcionando.

Debe entregar:

TP4-3-A. Foto del código de la clase ArbolNavidad.js

TP4-3-B. Foto del código HTML especificado (el mismo anterior se supone)

TP4-3-C. Foto de la salida en el navegador

TP4-3-D. Código fuente HTML y JS como archivos adjuntos

Problema 4. (25%). Efectos subrealistas

En este problema se debe provocar que las luces sean cliqueables, y, si se hace clic sobre una luz, cuando las luces están funcionando, entonces, se debe agregar una animación a cada luz que implique que la esfera se agranda y se achica en un ciclo de 2000 milisegundos de agrandado y otros 2000 milisegundos de achicado (hasta al tamaño normal que haya decidido). Cuando las luces se apagan todos estos efectos deben cesar y perderse, es decir, al volver a encender las luces estas deben estar todas de tamaño fijo. Un segundo clic sobre una luz que ya está en movimiento con tamaño variable no debe tener ningún efecto.

Debe entregar:

TP4-4-A. Foto del código de la clase ArbolNavidad.js

TP4-4-B. Foto del código HTML especificado (el mismo anterior se supone)

TP4-4-C. Foto de la salida en el navegador

TP4-4-D. Código fuente HTML y JS como archivos adjuntos