

Guía de uso de programación: Etiquetas de vino para Sottosopra

UOC

Índice

1. Etiqueta_vinozeta
2. Etiqueta_vinozeta_simplificado
3. Referencias

Autoría: Guillermo Sevilla

PID_00267137



CC BY-NC-ND
Primera edició: setembre 2019
Autoria: Guillermo Sevilla
Licència CC BY-NC-ND de esta edició, FUOC, 2019
Avda. Tibidabo, 39-43, 08035 Barcelona
Realització editorial: FUOC

Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-NoComercial-SinObraDerivada (BY-NC-ND) v.3.0 España de Creative Commons. Podéis copiarlos, distribuirlos y transmitirlos públicamente siempre que citéis el autor y la fuente (FUOC. Fundación para la Universitat Oberta de Catalunya), no hagáis de ellos un uso comercial y ni obra derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>

Guía de uso de programación: Etiquetas de vino para Sottosopra

En este manual veremos qué código os proporcionamos, cómo funciona a alto nivel y explicaremos algunos conceptos necesarios para entender el código.

Os proporcionamos dos esbozos de *Processing*:

- etiqueta_vinozeta: código completo para generar las etiquetas y guardarlas en archivos.
- etiqueta_vinozeta_simplificado: código para generar la textura de fondo y visualizarla en pantalla.

Para poder ver y ejecutar estos esbozos, el proceso es el mismo que con el código del cartel generativo de La Volta de la actividad anterior.

1. Etiqueta_vinozeta

Se trata del esbozo completo, que permite generar las etiquetas para todas las botellas y guardar el resultado en archivos para poder enviarlas a Impresión.

Dentro del directorio data, se encuentra, aparte del código, dos archivos: un CSV y un SVG. El archivo CSV contiene la información de las botellas para las cuales hay que generar las etiquetas: número de serie, cliente, año y medida.

Nota

El formato .csv es un formato sencillo y textual para representar datos en forma de mesa. Se puede abrir el fichero con algún editor de hojas de cálculo para ver los datos organizados, pero si se abre con un editor de textos se podrá ver cómo está estructurado. Se trata sencillamente de tener cada fila de datos en una línea de texto y, dentro de esta línea, las columnas separadas por comas (u otros símbolos). Un ejemplo posible es:

```
a1,b1,c1,d1  
a2,b2,c2,d2
```

Cuando se revise el código, se verá que solo usa el número de serie, puesto que se trata de una versión simplificada. Por otro lado, el archivo SVG contiene el diseño textual de la etiqueta.

En el código, primero se lee el listado de botellas con la información de cada una del archivo CSV y después, para cada botella, se genera una etiqueta y se guarda en un archivo JPG. El proceso para generar cada etiqueta consta de dos pasos: primero se genera la textura de fondo y después se sobrepone el diseño del SVG. Para generar la textura, se emplea una técnica parecida a la que se puede encontrar en el apartado 2.2.3

«Shapes from agentes» del libro *Generative Design* (pág. 224). Con esta técnica, se dibuja un círculo que se va desplazando, deformando y cambiando de color, y esto hace que a medida que se desplaza se sobreponga a él mismo y cree esta textura de apariencia orgánica.

Estas transformaciones que sufre el círculo se hacen de manera aleatoria para que cada etiqueta sea diferente. Pero a la vez nos interesa que el resultado sea el mismo si el esbozo se ejecuta dos veces. ¿Cómo lo hacemos? Las funciones para generar números aleatorios se pueden inicializar con un valor, denominado *semilla* (o *seed*, en inglés), de forma que los valores que generan una vez han sido inicializados con una misma semilla sean los mismos. Por lo tanto, si se inicializa con una semilla se pueden obtener siempre los mismos resultados. ¿Con qué semilla inicializamos? Pues si se quiere que cada etiqueta sea única, lo podemos hacer con un dato de la etiqueta que sea único, y en este caso se puede usar el número de serie (número de botella).

En el esbozo se usa una función llamada *noise*, que puede ser que no sea conocida. Esta función nos permite generar una secuencia de números aleatorios más natural, de forma que dos números consecutivos no sean «muy diferentes». En cambio, dos llamamientos consecutivos a *random* pueden devolver valores completamente diferentes. La función *noise* se define en un espacio, de forma que nos devolverá valores «parecidos» por coordenadas cercanas. Estas coordenadas son los parámetros que le pasamos, y pueden ser de una, dos o tres dimensiones. En el código siguiente se pueden ver los resultados de generar tres números aleatorios con *random* y tres con *noise* de coordenadas cercanas (en este caso en un espacio unidimensional).

```
println(random(1));  
println(random(1));  
println(random(1));
```

```
resultado:  
0.053666353  
0.8991752  
0.6556985
```

```
println(noise(0.0));  
println(noise(0.1));  
println(noise(0.2));
```

```
resultado:  
0.3648655  
0.4140035  
0.47199562
```

Para ampliar esta información, puede ser útil el capítulo «Randomness and Noise» (pág. 320-345) del libro *Generative Design*.

En el esbozo se usa con dos coordenadas la función *noise* para calcular la deformación del círculo. La primera se incrementa con cada uno de los puntos que definirán un círculo, de forma que de un arco al siguiente no hay grandes variaciones. La segunda se

incrementa cada vez que dibujamos un círculo, de forma que dos círculos consecutivos no sean excesivamente diferentes.

2. etiqueta_vinozeta_simplificado

En este esbozo solo se dibuja la textura de fondo, que se muestra en pantalla. Esto permite hacer pruebas modificando parámetros y ver el resultado sin tener que generar todas las etiquetas.

3. Referencias

Bohnacker, H.; Groß, B.; Laub, J.; y otros (2012). *Generative Design: Visualize, Program and Create with Processing*. New York: Princeton Architectural Press.