VISUALIZACIÓN DE INFORMACIÓN EN EL ÁMBITO DEL ARTE GENERATIVO A PARTIR DE PARSEO DE HTML

David Bedoian
Universidad Nacional de las Artes
Universidad Nacional de La Plata
bedoiandavid@gmail.com

Resumen

La representación de información de datos se ha abierto paso en el ámbito artístico, dando lugar a una vasta producción de obras de múltiples ramas del arte. Ya hace tiempo que la medición de datos o fenómenos cuantificables y su representación ha dejado de ser una actividad exclusiva de la ciencia y de fines únicamente informativos o estadísticos. En este trabajo se describe el potencial inherente en tomar el origen de datos para ser trasladados a un sistema de representación que cuadre en el ámbito del arte generativo. Este origen de datos será específicamente el que pueda recopilarse de contenidos de páginas web, describiendo técnicas de *parseo* de código fuente. Para ejemplificar la temática se describe la obra *Rain Running*, una aplicación de arte generativo de mi autoría que representa datos de una tabla de texto publicada en Internet.

Arte generativo, Visualización de información, Parseo html, Data art

INTRODUCCIÓN

El caos y el orden son dos componentes de gran protagonismo en el arte generativo. Las obras o producciones de este ámbito suponen

I CONGRESO INTERNACIONAL DE ARTES
ISBN 978-987-3946-13-4

un conjunto de reglas y lineamientos que determinan el proceso de producción y por ende su resultado. Al mismo tiempo estas reglas se ven afectadas o permiten considerar parámetros aleatorios, idealmente impredecibles. Estas reglas determinan el predominio que tendrá el azar en el proceso constructivo.

De todos modos, no deberíamos considerar el caos y el orden como dos extremos de la tensión entre lo simple y lo complejo; y pretender alcanzar un equilibrio entre ambos. La búsqueda debería llevarnos más bien a experimentar en qué medida incluir estos parámetros de manera que pueda evidenciarse el contraste entre ambos.

Para exponer la temática he desarrollado una pequeña aplicación de llamada *Rain Running*. Esta aplicación representa gráficamente los tiempos de llegada de los corredores de una media maratón. Cabe aclarar que el desarrollo de los conceptos se plantea desde un contexto del ámbito informático, incluyendo lenguajes de programación y otros aspectos técnicos relacionados; pero el arte generativo no se limita exclusivamente a la inclusión de esta tecnología. En el mejor de los casos, estos conceptos explicados podrían trasladarse a otros ámbitos y vincularse a otras herramientas de producción.

DESCRIPCIÓN DE LA APLICACIÓN DE EJEMPLO

La aplicación *Rain Running* (conceptualmente podría traducirse como lluvia de corredores o carrera de lluvia), es una aplicación de arte electrónico desarrollada en Processing, vinculada al arte generativo. Esta aplicación representa los tiempos de llegada de los corredores de una media maratón. Esta representación se hace en el transcurso del tiempo, colapsando el tiempo en una relación de 30 a 1. En la medida que avanza el tiempo, la llegada de cada corredor se representa en una gota de lluvia que cae en el escenario de la pantalla. Opcionalmente, estas gotas pueden dejar ver el nombre del corredor y su numeración; o mostrarse con un color correspondiente al sexo o la categoría del

I CONGRESO INTERNACIONAL DE ARTES
ISBN 978-987-3946-13-4

corredor. Como es habitual en este tipo de competencias, los primeros corredores llegan de manera esporádica. En la medida que transcurre el tiempo, la llegada es cruzada por una muchedumbre. En el transcurso de la aplicación, esto resulta en que en los primeros segundos se ven solo algunas gotas aisladas y avanzando el tiempo una gran precipitación; similar a como suelen sucederse estos fenómenos climáticos en la realidad. Lo particular de la aplicación en este caso es que los datos son reales y se obtienen de los publicados en los micrositios de este tipo de competencias.

ORÍGENES DE DATOS

Por mi parte, lo que veo interesante en esta aplicación es el origen de los datos a representar como fuente de información aleatoria, pero con evidencia de ciertos patrones. Por ejemplo: nadie podría predecir el tiempo en el que llegarán alrededor de quince mil corredores a la meta. Sin embargo, es muy probable que sólo unos pocos tarden aproximadamente una hora. También, que en el transcurso del tiempo aumente la cantidad de corredores que cruzan la meta con tiempos muy cercanos entre sí. Que la mayoría tardará un promedio de dos horas. Y finalmente disminuirá progresivamente la cantidad de corredores que alcancen la menta simultáneamente, llegando los últimos nuevamente de manera esporádica cerca de las cuatro horas.

De la misma manera, existe publicada en Internet una vasta cantidad de información susceptible de ser relevada para levantarse como fuente de información para aplicaciones de arte generativo. En lo personal, las que encuentro muy interesantes son las que evidencian de manera cuantificable el comportamiento humano. En el caso de *Rain Running*, evidencia dos aspectos: por un lado el interés a la participación masiva en este tipo de competencias y por otro lado la capacidad promedio del hombre y de la mujer para correr ciertas distancias.

Otros orígenes de información podrían ser las redes sociales, los

| CONGRESO INTERNACIONAL DE ARTES
ISBN 978-987-3946-13-4

portales de noticias, los censos poblacionales, información estadística de problemáticas sociales o del medio ambiente, etc. Cabe aclarar que es muy importante tener presente las restricciones de difusión de la información y repasar principalmente las posibles implicaciones éticas de la información que se está manipulando.

VINCULANDO LA INFORMACIÓN CON LA APLICACIÓN

Lo primero es conseguir que la información relevada pueda cargarse en nuestra aplicación. Para esto se lleva adelante un proceso circular que ayudará a alcanzar la depuración de los datos con el mejor resultado final.

ANÁLISIS DE LA ESTRUCTURA DE DATOS

En este punto podemos hacernos preguntas tales como: "¿Los datos que quiero mostrar los puedo medir?"; "Si no es así ¿hay algún parámetro que resulte en un valor cuantificable?"; "Los registros que encuentro, ¿qué características tienen en común?", "¿Pueden compararse entre sí?". Estas y preguntas similares al responderse generarán nuevos cuestionamientos y nos ayudarán a recorrer y definir conceptos y fenómenos cuantificables presentes en la temática que hemos decidido abordar.

EL MEDIO Y DISPONIBILIDAD DE LA INFORMACIÓN

Algo importante a determinar es si la información se encuentra publicada en Internet o puede estar almacenada de manera local (esto es guardar un archivo en el ambiente donde se ejecuta la aplicación). Aún si la fuente de información fuera una publicación online, si esta

I CONGRESO INTERNACIONAL DE ARTES
ISBN 978-987-3946-13-4

información no va a cambiar regularmente (tal como es el caso de los resultados de la carrera), conviene realizar una copia de la información para asegurarse el acceso a los datos. Ahora bien, esto no sería posible si justamente nuestro origen de datos tiene que ver con monitorear en tiempo real publicaciones web que cambian de forma dinámica.

PROCESANDO LA INFORMACIÓN

El *parseo* (en inglés 'parsing') es el proceso de transformar una entrada de texto o un origen de datos una estructura de datos normalizada (usualmente un árbol, o una tabla). Esta estructura obtenida debe resultar apropiada para ser procesada. Generalmente los parseadores primero identifican los símbolos de la entrada y luego construyen el árbol de parseo para esos símbolos.

Existen numerosas herramientas para realizar este proceso. En la aplicación del ejemplo, el proceso se realiza con Processing, mediante la función *loadStrings()*. Para simplificar el procesamiento en tiempo de ejecución de la aplicación, el origen de datos ha sido preprocesado de formato de código fuente de la publicación en HTML a CSV (*Comma Separated Values*, o Valores Separados por Coma). En este formato, se almacena un registro por línea.

Por ejemplo, cada registro de la tabla se vería convertido de la siguiente manera:

De:

<tr><td>755</td><td>41</td><td>M</td><td>JAVIER</td>LOPEZ<td>01:59:03</td></tr>

<tr><td>756</td><td>213</td><td>M</td><td>MARTIN</td>PADILLA<td>01:59:05</td></tr>

A:

755,41,M,JAVIER,LOPEZ,01:59:03

756,213,M,MARTIN,PADILLA,01:59:05

I CONGRESO INTERNACIONAL DE ARTES
ISBN 978-987-3946-13-4

Una de las herramientas más simples para esta conversión está presente en la mayoría aplicaciones de ofimática estándar (como *Excel*), en la que se puede pegar una tabla copiada de una página web, y al guardar seleccionar el formato .csv.

Al leer el archivo en formato CSV con la función *loadStrings()*, el archivo de texto de origen se carga en un arreglo de cadenas de texto. Cada salto de línea del archivo de origen se corresponde con un índice de este arreglo. De esta manera, el arreglo tiene tantas casillas como saltos de líneas (por ende, equivalente a la cantidad de registros).

En código:

```
String[] datos = loadStrings( “resultados_media_maraton.csv” );
println( (datos.length-1) + " registros." );
```

La primera línea de código del ejemplo carga el contenido del archivo en el arreglo denominado *datos*. La siguiente línea es una instrucción para escribir en consola la cantidad de casillas que tiene el arreglo, que sirve por ejemplo para testear y monitorear la cantidad de saltos de línea o de registros del archivo origen.

El siguiente paso es recorrer uno a uno los registro obtenidos e identificar los campos en cada uno. Por ejemplo, como vimos en la estructura del archivo CSV, en cada línea la coma separa los valores de cada registro. En este caso en particular, se corresponden de la siguiente manera:

```
Posición,Dorsal,Sexo,Nombre,Apellido,Tiempo Oficial
755,41,M,JAVIER,LOPEZ,01:59:03
756,213,M,MARTIN,PADILLA,01:59:05
```

Resulta oportuno mencionar que por norma los archivos CSV pueden incluir opcionalmente una primera línea con los nombres de los campos para cada registro, como es el caso citado.

Continuando el ejemplo en el lenguaje de Processing, el paso mencionado anteriormente se realiza dentro de una estructura repetitiva (por ejemplo, ciclo *for*):

I CONGRESO INTERNACIONAL DE ARTES
ISBN 978-987-3946-13-4

```
for (int i=0 ; i<datos.length ; i++ ) {
String[] campos = split( datos[i], “,” );
corredores.add( new Corredor( campos[0], campos[1], campos[2],
campos[5] ) );
}
```

Esta estructura de código recorre todas las casillas del arreglo *datos*. Dentro de la estructura que se repite para cada ciclo, la primera línea de código contiene la función *split()*, que sirve para dividir una cadena de texto en porciones y almacenar cada una de ellas en las casillas de un arreglo resultante. Este resultado se almacena en un nuevo arreglo denominado justamente *campos*, en el que cada casilla corresponderá a cada uno los campos que toman su valor correspondiente al texto entre cada coma.

La siguiente línea invoca una acción llamada *new Corredor(...)*, procedimiento que se corresponde con la POO (Programación Orientada a Objetos). Explicado de manera abreviada, permite definir de manera personalizada una estructura de código que pueda almacenar las propiedades de cada instancia de un corredor y comportamiento dentro de la aplicación. Esta porción se denomina constructor y justamente realiza la acción de *construir* una nueva instancia de la definición de la clase *Corredor*. En este caso, se invoca dentro de la función *corredores.add()*. Esto permite acumular cada una de estas instancias en un tipo de dato denominado *arrayList*. Este tipo de estructura nos dará la posibilidad de recorrer todo el conjunto de elementos almacenados y realizar las acciones correspondientes a cada instancia.

Por ejemplo, a continuación, la parte del código que se utiliza para evaluar si una instancia de *corredores* debe mostrarse o no, comparando el puntero del tiempo global con la de cada corredor:

I CONGRESO INTERNACIONAL DE ARTES
ISBN 978-987-3946-13-4

```
for (int i=0 ; i<corredores.size() ; i++ ) {
 Corredor  c = (Corredor) corredores.get(i);
 if (puntero.getPuntero() == c.getSegundos() ) {
 elementos.add( new Elemento( c ) );
 }
}
```

Lo que resulta del código citado es que se agrega al elemento *arrayList elementos* una instancia del objeto que debe activarse en ese momento, según se corresponda el tiempo *puntero*, con el de cada elemento de *corredores*.

De ahí en adelante, lo que resta a la aplicación es mostrar cada objeto del *arrayList elementos* que se encuentren activos. Esto es:

```
for (int i=0 ; i<elementos.size() ; i++ ) {
 Elemento  e = (Elemento) elementos.get(i);
 e.actualizar();
}
```

REFINAR LOS RESULTADOS Y EXPERIMENTAR

Una vez que se han definido las reglas que conjugan la información procesada se sigue a una etapa de pruebas y experiencias, incorporando control o aleatoriedad en la medida que los resultados sean los esperados. Por ejemplo, el momento que un *elemento* se activa para comenzar a mostrarse con su correspondiente animación, he agregado en su constructor las siguientes líneas de código:

```
x = floor(random(width));
y = floor(random(height));
```

Estas dos instrucciones asignan un valor aleatorio para cada una de las dimensiones de las coordenadas de ubicación del elemento que se está mostrando.

I CONGRESO INTERNACIONAL DE ARTES
ISBN 978-987-3946-13-4

CONCLUSIÓN

Durante la elaboración de los contenidos del presente material y el desarrollo de la aplicación de ejemplo he procurado indagar el potencial inherente en diversas fuentes de información como origen de datos para la producción de obras que cuadren en el ámbito generativo. Las experiencias llevadas a cabo en este recorrido han dejado en manifiesto que diferentes fuentes, particularmente en este caso las disponibles en Internet, pueden resultar en un componente de mucho provecho y de gran potencial. He encontrado que administrar estos orígenes de datos mediante las reglas de una obra de arte generativo amplía su potencial tanto estético como conceptual. Ambos aspectos se ven enriquecidos y en el resultado final se aprecia una obra interesante en su apariencia y también en su relato. Al apreciar la obra que aplica estos elementos, el concepto de la misma además de estar presente en la técnica de representación aplicada, se refuerza al despertar cierto interés del espectador o participante con el origen de datos seleccionado y su temática.

I CONGRESO INTERNACIONAL DE ARTES
ISBN 978-987-3946-13-4

BIBLIOGRAFÍA

- Causa, Emiliano (compilador). (2014). *Invasión Generativa*. La Plata, Buenos Aires, Argentina: Invasores de la Generatividad. http://www.invasiongenerativa.com.ar/
- Fry, Ben. (2008). *Visualizing Data*. Gravenstein Highway North, Sebastopol, USA: O'Reilly Media Inc.
- Pearson, Matt. (2011). *Generative Art*. Shelter Island, USA: Manning Publications Co.

ISBN 978-987-3946-13-4