

MEMENTO

Javier Juaristi

PROYECTO PERSONAL

Diseño Generativo

Enero 2023

Índice

3	Briefing
4	Memento [Propuesta]
5	Memento [Proceso]
6	Memento [Proceso ii]
7	Memento [Proceso y iii]
8	Memento [Reglas]
9	Reflexión
10	Recursos y enlaces



▲ Los copos de nieve como elemento generativo natural

Briefing

Copos de nieve

Los copos de nieve se generan debido a infinitas variaciones de temperatura y degradación en un proceso recursivo y aleatorio donde existe un elemento fijo, el agua, y otros variables a lo largo del tiempo de generación. Los cambios de temperatura y presión o los flujos de aire, degradan las dendritas del copo de nieve durante su formación produciendo formas únicas.

Podría decirse que los copos de nieve son un diseño generativo de la naturaleza.

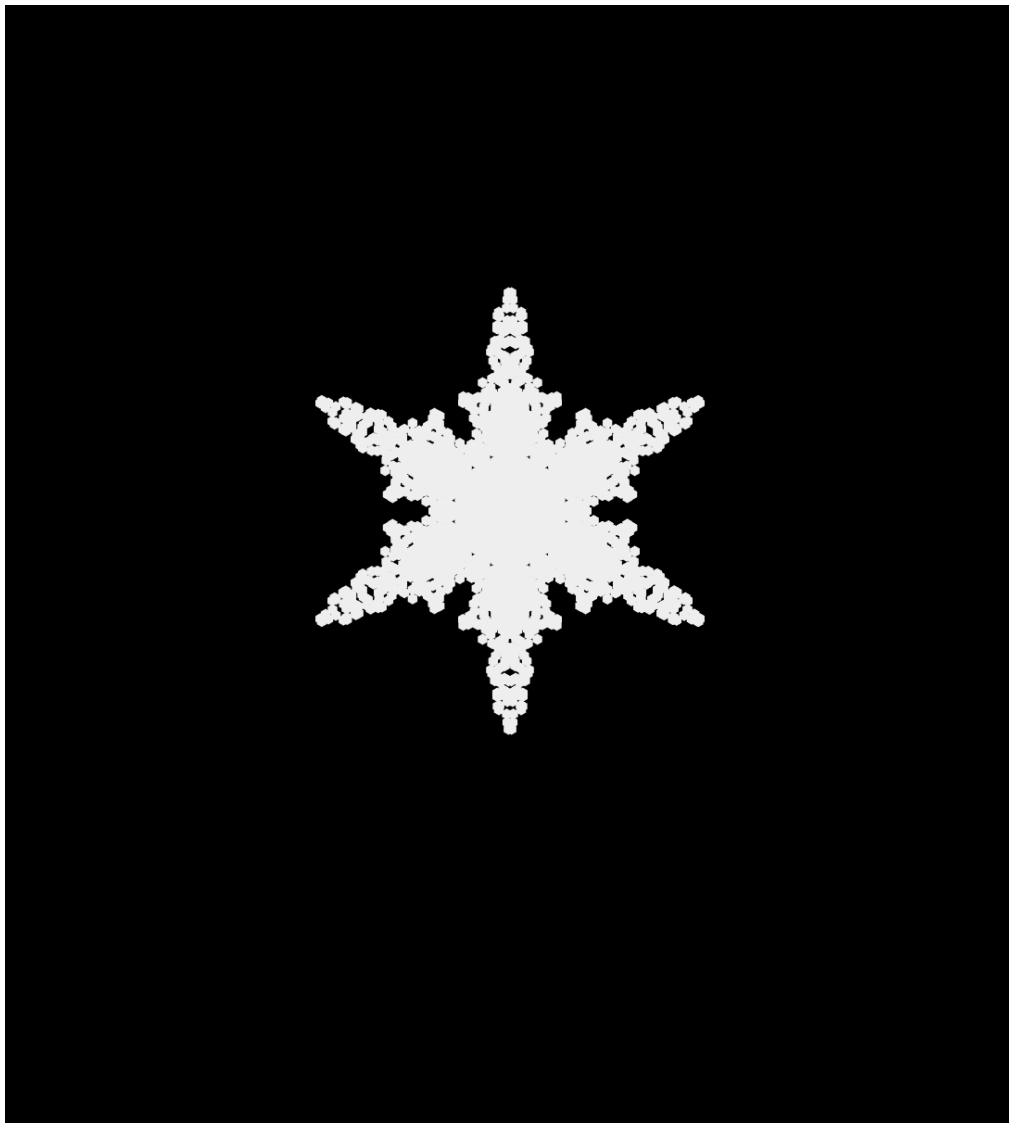
Recuerdos

El ser humano juega a controlar su destino y la vida lo sacude con eventos al azar. Cada recuerdo almacenado en las sinapsis de un ser humano es único. Es tan único que dos seres humanos íntimamente relacionados con un suceso guardarán un recuerdo ligera o ampliamente diferente del mismo, pero nunca idéntico.

Al igual que la naturaleza con los copos de nieve, la memoria, al generar recuerdos, realiza un proceso generativo con un evento como elemento fijo y nosotros como elemento con infinitas variables físicas, emocionales, perceptivas o conductuales.

El proyecto

Con estos dos conceptos de procesos generativos en la cabeza, mi subconsciente me despertó una noche con la idea una aplicación web que uniera de forma permanente ambos procesos.



▲ Copo generativo en una fase avanzada de las pruebas

memento [Propuesta]

La propuesta, tal y como me vino la idea, consiste en unir estos dos eventos generativos, los copos de nieve y los recuerdos, en un espacio en la red.

Allí, una persona podría entrar e introducir la fecha de un momento a recordar, el nombre de la persona o ser querido asociado a ese momento, y un breve texto descriptivo del recuerdo.

Con estos datos se produciría una semilla que daría origen a la generación de un copo de nieve único que serviría de almacén de ese recuerdo.

Una vez generado, este copo de nieve se uniría a una nevada perpetua que se visualizaría en la pantalla de aquel que visite el sitio.

Un visitante podría tocar un copo de nieve y se leería algo así como: “Aitor, tu padre te lleva en su corazón”, un snippet que la persona que creó el copo habría escrito.

El recuerdo en si, sería solo visible para la persona que lo creó o a quien permita verlos. Podría ser un texto, una imagen, un vídeo o un audio que viviría en la nube junto a los demás recuerdos.

Como “extra”, el copo podría descargarse en formato vectorial y utilizarse para fabricar un colgante, un adorno para el árbol de Navidad, un dibujo para colorear, grabar la tapa de un libro... Mil cosas “únicas” e irrepetibles.

Mirando más allá, esta propuesta podría trasladarse a un memorial sobre una proyección envolvente, o de forma más experimental, transportando del mundo virtual a la realidad los copos de nieve de la nevada mediante hologramas táctiles como las Fairy Lights.

Las Luces de Hada, es una tecnología de hologramas hápticos desarrollada por un equipo de investigadores de la universidad de Tsukuba. Emitiendo ráfagas concentradas y ultra cortas de luz láser, en el orden de un cuatrillón de segundo (femtosegundo), consiguen que las moléculas de aire alcanzadas liberen un electrón produciendo luz y plasma. Al entrar en contacto la piel humana con las moléculas de plasma, estas se perciben como si tocásemos papel de lija, consiguiendo así un holograma táctil.

Acotando

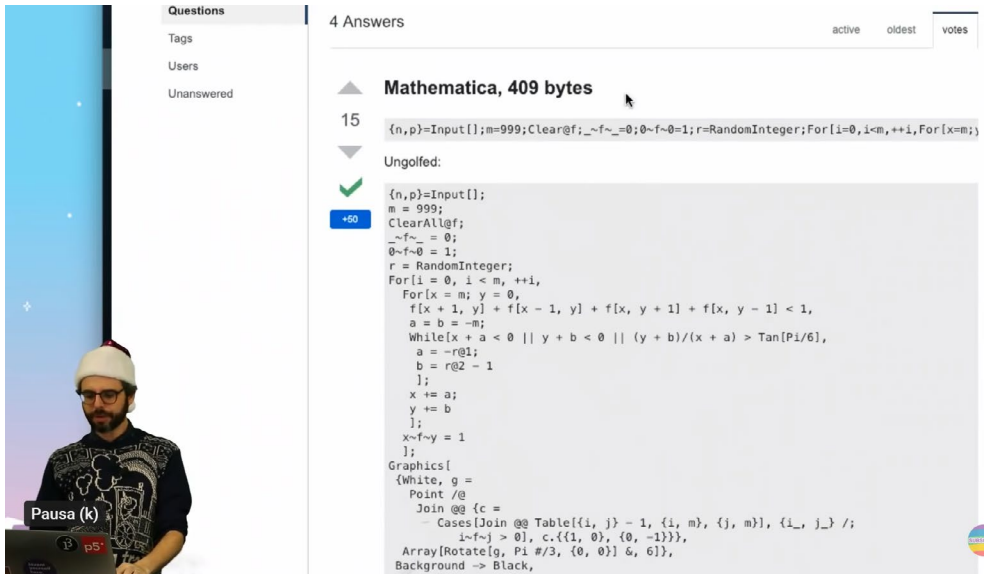
Como la propuesta base, aunque en apariencia simple, es bastante compleja y extensa, la propuesta final para este proyecto será la realización de una aplicación desarrollada en P5.js capaz de generar y mostrar los copos de nieve y la nevada en base a la entrada de un archivo JSON con semillas, fechas, nombres y recuerdos.

Ontología [elementos fijos]

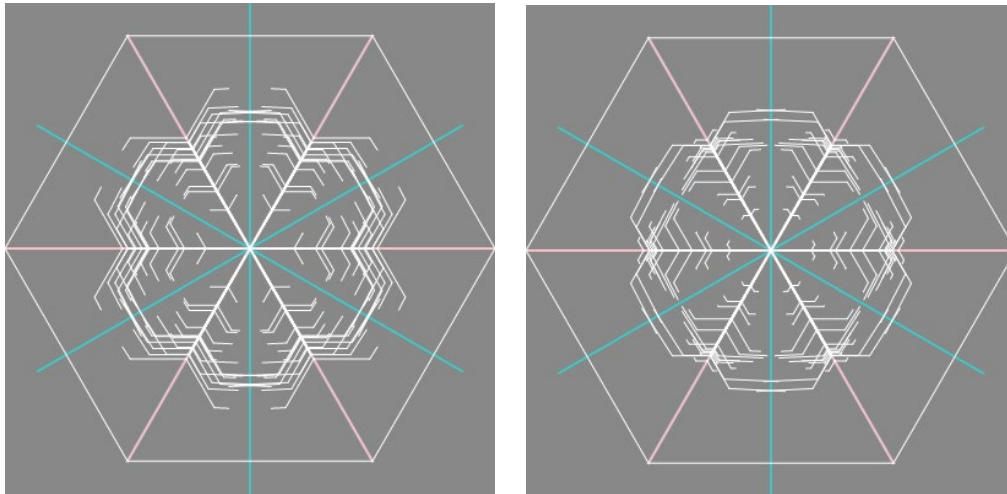
Número de ejes
Número de secciones

Ontología [elementos variables]

Fecha
Nombre
Memento



▲ Daniel Siffman y el algoritmo de code.golf escrito en Mathematica.



▲ Pruebas con un algoritmo recursivo empleando líneas y vectores en lugar de partículas.

memento [Proceso]

El proceso comenzó realmente con un vídeo de Daniel Siffman, **Coding Challenge #127: Brownian Tree Snowflake**, donde trataba de reproducir los resultados que vio en la página code.golf, un sitio donde se compete por generar el código más corto posible que de respuesta aun reto propuesto.

El desafío consistía en generar un copo de nieve. La solución que analizaba Siffman empleaba un árbol browniano de matrices de enteros, y estaba desarrollado en **Mathematica** (Wolfram).

Sin embargo, para el desarrollo del proyecto pensé en buscar una solución basada en la recursividad con vectores.

Los primeros resultados eran prometedores, pero mis ganas de avanzar rápidamente en el proyecto me obligaron a descartar esa vía y regresar a las partículas viajeras del árbol browniano. Algo más sencillo con lo que lidiar.

Datos

Antes de empezar, y para poder testar el sitio, debía generar un conjunto de datos de prueba. Para ello recurrí al **INE** (Instituto Nacional de Estadística) y descargué dos archivos CSV: uno con todos los nombres de personas empadronadas en España hasta 2021 y con más de 20 registros de entrada y otro de las mismas características con los apellidos. Esto incluye nombres y apellidos de todas las

nacionalidades, así que algunos resultados son bastante llamativos.

Curiosamente, tanto los nombres como los apellidos se encontraban en mayúsculas, cosa que pude solucionar fácilmente en una hoja de cálculo, y sin tildes, asunto que no he corregido dado el volumen de registros y su empleo tan solo como modelo de pruebas.

Después de unir los nombres de hombres y mujeres en un archivo, exporté este en formato CSV he hice lo mismo con el de los apellidos.

Para simular el relato de los recuerdos recurrí a **lipsum.com**, un generador de textos para maquetación. El clásico “Lorem ipsum...” basado en “de Finibus Bonorum et Malorum” de Cicero. El sitio genera un máximo de 150 párrafos, lo que tampoco supone un problema.

El siguiente paso fue escribir un programa que generase un archivo **JSON** con los datos.

Semillas

En el sitio completamente desarrollado, los visitantes introducirían una fecha, un nombre y un recuerdo, y con cada una de esas entradas se generaría una semilla única que se emplearía para generar cada una de las tres secciones que componen el copo. Y aquí ya estoy adelantando algunas reglas del proceso.

```

1  [
2  {
3    "seedA": 991745357140,
4    "seedB": 9797105117,
5    "seedC": 46105115105,
6    "date": "2001-06-05T12:49:17.139Z",
7    "name": "Lucia Vanessa Gual Cortina",
8    "memento": "\nUt pellentesque elit ex, sit amet vestibulum hendrerit felis sapien, quis molestie diam luctus accumsan. Nulla ultricies dolor a lorem sedales, nec pretium.

```

▲ Fragmento del archivo JSON con los datos de prueba.

Al igual que dos copos de nieve que caen en la palma de tu mano no pueden ser iguales por más que iniciaran su viaje juntos, el ser humano juega a controlar su destino y la vida lo sacude con eventos al azar.

Cada recuerdo almacenado en las sinapsis de un ser humano es único. Es tan único que dos seres humanos íntimamente relacionados con un suceso guardarán un recuerdo ligero o ampliamente diferente del mismo, pero nunca idéntico.

En la nevada, pulsa y arrastra el ratón para generar viento.
Haz clic sobre un copo para ver el recuerdo que guarda.

[HAZ CLIC EN LA PANTALLA Y PULSA UNA TECLA PARA COMENZAR]

Mementos v0.1 | Javier Juaristi

▲ Las instrucciones aparecen tras la carga de los recursos

memento [Proceso ii]

El programa que he escrito para producir el archivo JSON con los datos de prueba genera una fecha al azar tomando como base del cálculo el tiempo actual en milisegundos: `Date.now()`. Esto apunta a un día aleatorio entre el 1 de enero de 1970 y el momento de generar el archivo de datos. El propio valor en milisegundos compondrá la **primera semilla**.

Para los nombres, el programa elige al azar uno de la lista con 54916 nombres preparada en el archivo CSV. El nombre seleccionado se retira de la lista para evitar posibles repeticiones.

Se hace lo mismo con dos apellidos y se une todo en un único string.

Para generar la **segunda semilla** usamos cuatro caracteres del nombre completo: los valores ASCII para caracteres alfabéticos van del 65 al 122 así que generaremos una cadena que contendrá entre 8 y 12 dígitos. Esta cadena la convertiremos en semilla.

Durante el proceso de selección de estos 4 caracteres se van eliminando los seleccionados y, a fin de que el proceso sea reproducible, antes de comenzar empleamos la primera semilla.

Encadenar las semillas también nos permitirá verificar la integridad de los datos si fuese necesario.

En la selección de párrafos, de 1 a 4, para evitar seleccionar dos párrafos

iguales seguidos, trabajamos sobre una copia de la lista a fin de poder eliminar los párrafos seleccionados durante una iteración y recuperar la lista completa para la siguiente. Esto se hace dado que tan solo contamos con 150 párrafos.

Nuestra única intervención consistirá en decidir el número de elementos a generar e introducirlo en la variable asignada al efecto. Hay que tener en cuenta que el límite de elementos sería la mitad de la cantidad total de apellidos.

Ejecutaremos el programa y se descargará un archivo JSON con las semillas y recuerdos de prueba.

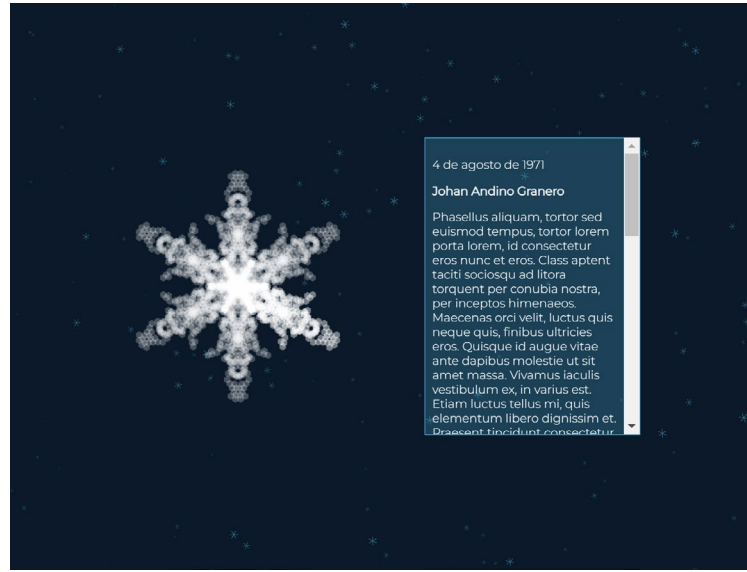
La nevada

Una vez el JSON de prueba en su carpeta, podemos ejecutar el programa.

Tras unos segundos de inicialización (carga del JSON, fuentes y clonación del JSON en forma de array), comienza una nevada.

Los copos, representados por asteriscos de diferentes tamaños, velocidades y transparencia, caen hacia el suelo portando cada uno de ellos un identificador o índice del array de recuerdos.

Como se indica en las instrucciones durante el proceso de inicialización, podemos pulsar el ratón y arrastrarlo para simular una corriente de aire o hacer clic en la proximidad de un copo para observar su contenido.



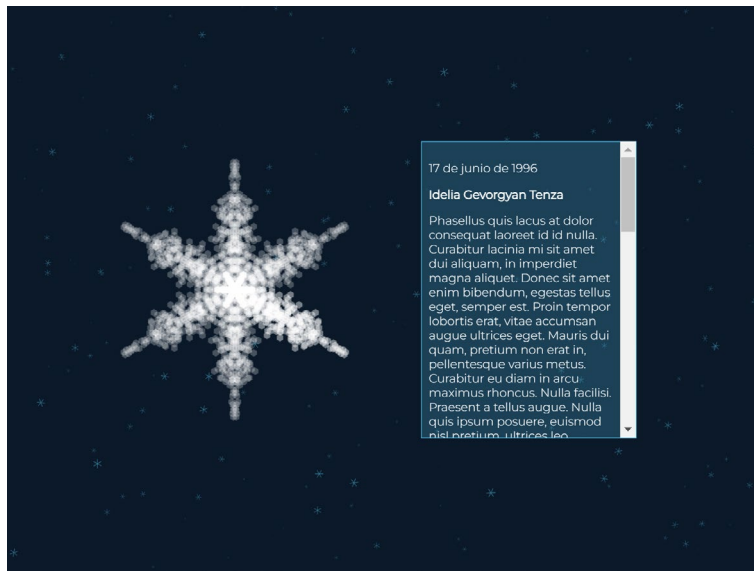
memento [Proceso y iii]

En la nevada nunca habrá dos copos iguales. Los copos se seleccionan aleatoriamente del listado de recuerdos (mementos) y se retiran de la lista antes de incorporarlos a la nevada.

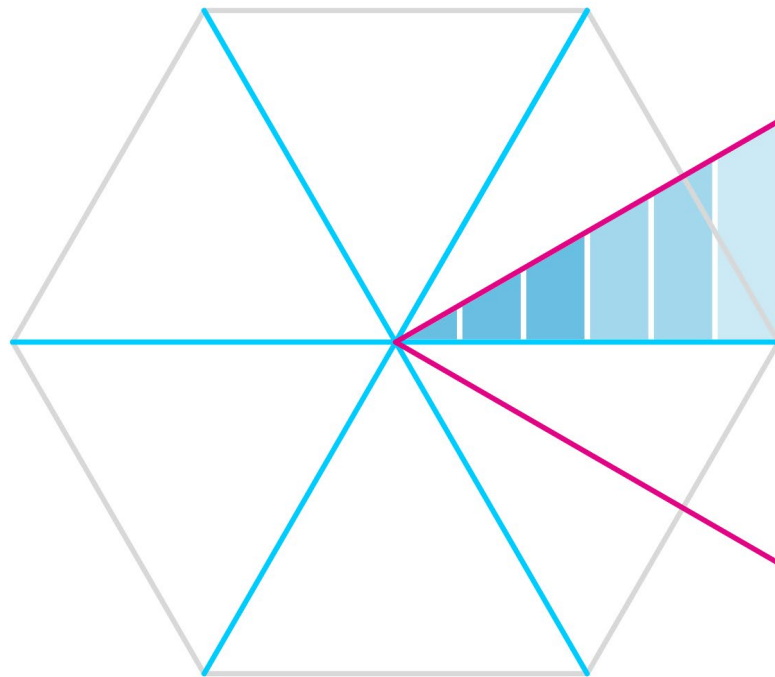
Cuando un copo desaparece por la base de la pantalla, un nuevo copo lo reemplaza, repitiéndose el proceso hasta terminar con todos los recuerdos. En ese momento se regenera la lista y la nevada continúa permanentemente.

Caza de recuerdos

Al hacer clic en la proximidad de un copo accedemos a los datos mediante su índice y generamos un gráfico del copo que mostramos junto a una ventana con los datos de fecha, nombre y recuerdo.



▲ Capturas de copos con la nevada de fondo.



▲ En el dibujo se puede ver la forma hexagonal del copo, los radios, la bisectriz y las secciones que conforman un copo.

memento [Reglas]

Los pocos de nieve se generan por la acumulación de “partículas viajeras” que rigen su comportamiento por las siguientes reglas:

Aunque el algoritmo permite generar copos de 3 o más radios, los copos tendrán siempre 6 radios.

Cada radio divide su longitud en 6 partes para siguiendo la secuencia de Fibonacci, dividir el copo en tres secciones: Comenzando desde el exterior tendremos una primera sección de un módulo de longitud, una segunda de dos módulos y una tercera de tres.

Cada sección lanzará un número de partículas inversamente proporcional a su tamaño.

El número de partículas por sección, además, estará determinado por los valores que representa: fecha, nombre y recuerdo.

La sección del núcleo del copo está determinada por la fecha y constreñida a un rango de parámetros predeterminado.

La sección central está definida por el doble de la longitud del nombre.

La sección exterior, por el número de renglones de 35 caracteres que contiene el mismo.

Cada sección utilizará la semilla correspondiente recibida en el JSON.

Las partículas parten del extremo del radio y avanzan a lo largo del eje X hacia el centro del copo variando su posición Y aleatoriamente.

Las partículas no pueden superar la bisectriz entre dos ejes. Para ello se restringe el ángulo del vector que almacena la posición de la partícula a un sexto de π .

Si una partícula llega al centro del copo, termina su viaje y se almacena su posición.

Si una partícula colisiona con otra partícula, termina su viaje y se almacena su posición.

Para representar el copo de nieve se dibujan las partículas en 12 ocasiones: en cada uno de los 6 radios en sus valores positivos y negativos.

Dado que el ángulo 0 estaría sobre el eje horizontal de visualización, a la hora de mostrar el copo se rotará su posición un sexto de π a fin de mostrarlo en posición “vertical”.



Reflexión

Antes de iniciar mi camino en el diseño generativo, el concepto lo asociaba a potentes ordenadores, sistemas expertos, redes neuronales e inteligencia artificial.

Ahora entiendo que el diseño generativo es natural e innato en el ser humano y en la naturaleza en general, y que su potenciación con la tecnología es lo que quizás nos ha llevado a visualizarlo casi con una mirada cyberpunk o de ciencia ficción.

Reflexionando sobre este proyecto personal aun en desarrollo, creo que la idea inicial era bastante ambiciosa. Aunque en apariencia sea simple, el número de tecnologías involucradas para poder llevarlo a cabo hace que su desarrollo sea más complejo de lo que esperaba inicialmente.

P5.js no es quizás el mejor marco para trabajarlo. Puede que Three.js tenga más potencia. He visto demos realizadas con millones de partículas sin hacer sudar a mi viejo portátil así que quizás la próxima versión sea sobre Three.js

La parte aun no desarrollada en este proyecto para la captación de datos en el front-end quizás sería mejor realizarla con React en lugar de emplear el JavaScript nativo dada su capacidad de usar un DOM virtual para acelerar los procesos de renderización de las páginas. Aunque quizás esto no influya demasiado en el resultado global.

En cualquier caso, el mayor reto puede que tenga que superar, será la fusión de las formas geométricas para lograr un único objeto sin intersecciones en sus vectores.

Recursos y enlaces de interés

Coding Challenge #127: Brownian Tree Snowflake

<https://www.youtube.com/watch?v=XUA8UREROYE>

Code Golf

<https://code.golf/>

Lorem Ipsum

<https://lipsum.com/>

Javier Juaristi

PROYECTO PERSONAL
Diseño Generativo

Enero 2023