

Universidad de Guanajuato
Geometría Computacional
Proyecto Final
Profesora: Lizárraga Morales Rocío Alfonsina
17/12/2020

Emmanuel Alejandro Gallardo Domínguez
Guadalupe Carolina Aguirre Zúñiga
145884

Nenúfares - Monet

1. Resumen:

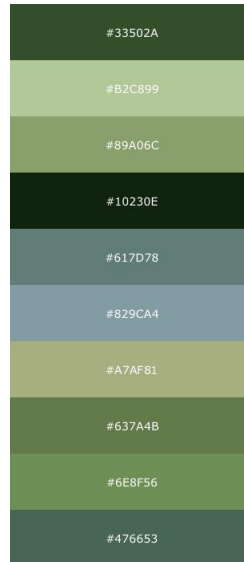
En este proyecto se analizó una obra del reconocido pintor Claude Monet, nenúfares (o mejor conocida como Japanese Bridge) para ser reinterpretada y convertida a una pieza digital con toques de aleatoriedad y geometría computacional.

2. Introducción

Como propuesta de proyecto se eligió la opción de reinterpretar una obra y convertirla en una obra en digital.

Claude Monet fue uno de los pioneros en el impresionismo, estilo artístico que ejecutaría toda su vida. Esta obra nos llamó mucho la atención por varios motivos:

2.1. Los colores a usar:



Son colores que pueden ser interpretados como "Verde, naturaleza", siempre presentes en los colores, de nuestra opinión, más neutrales y llamativos.

2.2. Los elementos que posee:

Al ser una representación de la naturaleza podíamos jugar con la aleatoriedad de esta, generando partes del terreno con funciones nativas de processing como veremos más adelante

2.3. El reto que implicaba:

Se asumió que la reinterpretación de una obra de impresionismo sería un buen reto técnico, por los colores, por la correcta colocación de los objetos o hasta por la programación de los mismos.

3. Metodología

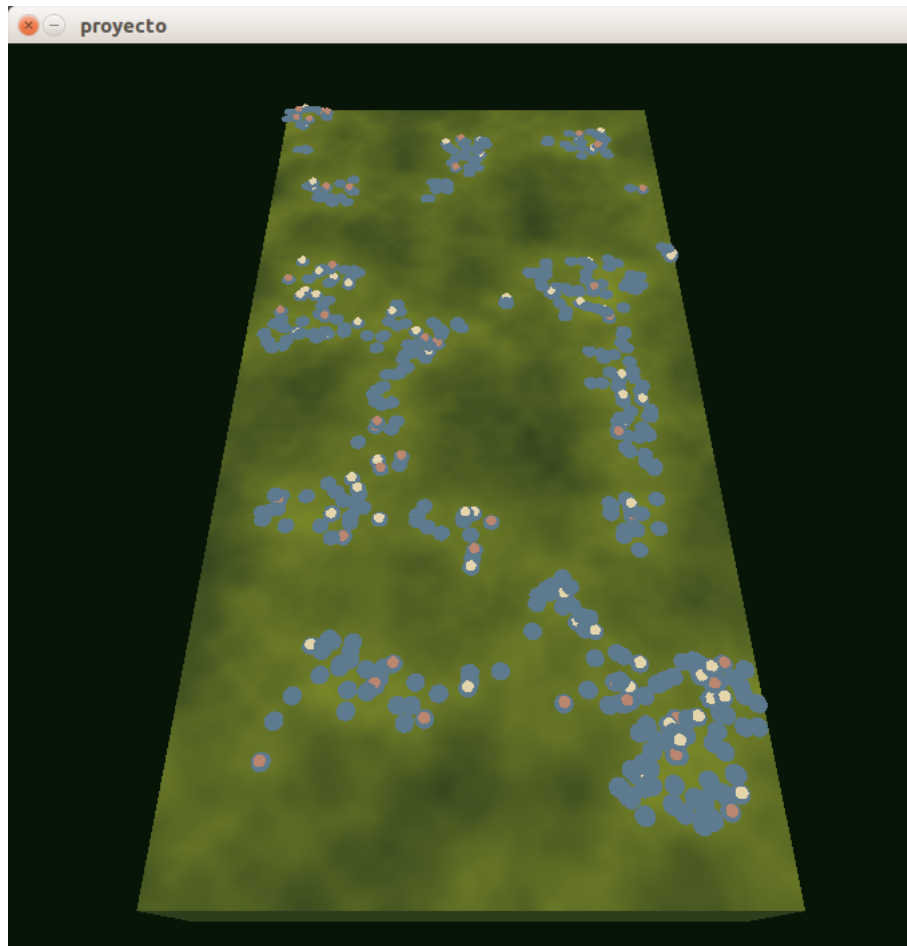
Para poder describir mejor el funcionamiento diseñado para este proyecto es útil dividirlo en dos diferentes tipos de módulos/objetos creados:

- **Objetos:** Podemos llamarles de esta forma a los módulos que son visibles en el programa, y que si bien tienen su parte de back end al ser generados, en su mayor parte se usan como decorativos.
- **Auxiliares:** Se trata de módulos que fueron implementados para incrementar la funcionalidad y a pesar de no ser visibles, hacen del programa cómodo de usar

3.1. Objetos:

3.1.1. Estanque

Se trata de un objeto tridimensional que genera un mapa de ruido aleatorio usando la función `noise()` de Processing.



4. Resultados

5. Conclusiones

6. Bibliografía