

# Generación de contenido por procedimientos para la creación de espacios de juego

Generation of content by procedures for the creation of game spaces

Luis Eduardo Domínguez Hurtado

Estudiante de la Licenciatura en Artes Digitales  
División de Ingenierías, Campus Irapuato-Salamanca.  
Universidad de Guanajuato

[le.dominquezhurtado@ugto.mx](mailto:le.dominquezhurtado@ugto.mx)

## Resumen

A la hora de crear videojuegos nos encontramos con varios desafíos uno de ellos es el manejo del tiempo, ya que el proceso de desarrollo suele ser extenso. Sin embargo, la generación de contenido procedural puede ahorrar grandes cantidades de tiempo y a su vez, encontrar un nuevo enfoque. En la generación de contenido procedural se utilizan algoritmos que generan contenido automáticamente, lo que permite ahorrar tiempo y crear experiencias únicas. En este trabajo se explorará el algoritmo de Colapso de la Función de Onda que se implementará en el software Processing para la creación de un videojuego único y diverso. En el videojuego desarrollado, siempre se tendrá un nuevo acertijo por resolver, específicamente, a través del algoritmo de función de colapso de onda, se crearán nuevos escenarios y niveles de manera pseudo aleatoria, para que nunca sea el mismo nivel y siempre tenga una diferencia, aunque sea mínima.

**Palabras clave:** videojuegos; programación; algoritmo; colapso; processing; procedural.

## Introducción

En el ámbito de los videojuegos la generación procedural de escenarios ha sido una técnica fascinante para crear mundos virtuales únicos y emocionantes. En este reporte se presenta el algoritmo de Colapso de la Función de Onda, un innovador algoritmo basado en los principios de mecánica cuántica que ha sido implementado en este trabajo para el desarrollo del videojuego al que llamamos "Buscando un diamante".

El algoritmo permite crear terrenos pseudoaleatorios de manera inteligente, generando un ambiente diverso y cambiante en cada partida. Mediante la utilización de tiles y reglas predefinidas, este algoritmo logra construir un espacio de juego coherente y atractivo.

En este reporte se examinará en detalle el funcionamiento del algoritmo, explicando su proceso paso a paso y destacando los resultados obtenidos en el videojuego.

## Metodología

En esta sección se presenta la descripción del funcionamiento del Algoritmo de colapso de la función de onda paso a paso en base a lo que propuso Maxim Gumin (2016) se le agregaron algunas ideas de Robert Heaton (2018) que plantea restringir posibilidades entre sus vecinos más cercanos.

El primer paso para generar un terreno pseudoaleatorio en el videojuego es contar con una variedad de bloques o elementos que les llamaremos "tiles", que son las piezas que utilizaremos para construir el entorno del juego. Estos tiles representarán diferentes elementos del entorno, como el mar, la tierra, árboles, rocas, cuevas, entre otros. Cada tile tendrá sus propias reglas y restricciones para su ubicación y conexión con otros tiles.

A continuación, describiremos el proceso paso a paso para generar el entorno pseudoaleatorio utilizando el algoritmo de colapso de la función de onda:

Preparación de los tiles: Se reúnen y diseñan los diferentes tiles que se utilizarán en el juego. Cada tile debe ser coherente en términos de diseño y encajar correctamente con los demás tiles.

En la Figura 1 se presentan los tiles utilizados en esta aplicación. Como se puede observar tenemos una variedad de elementos como esquinas en diferente orientación, bloques y orillas creadas por mar y tierra. Sin embargo, en esta figura los tiles no tienen coherencia en el orden presentado. En la Figura 2 se presenta una posible interconexión entre tiles que ya presenta sentido para un escenario.

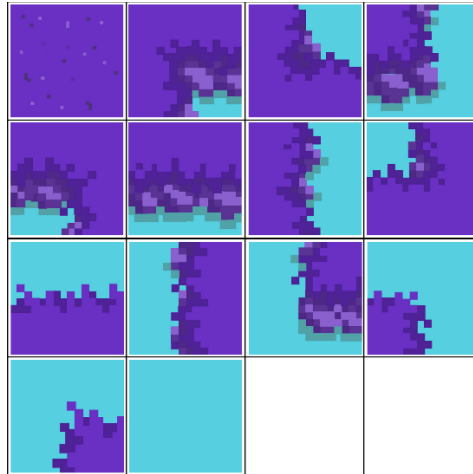


Figura 1. Tiles que se usaran

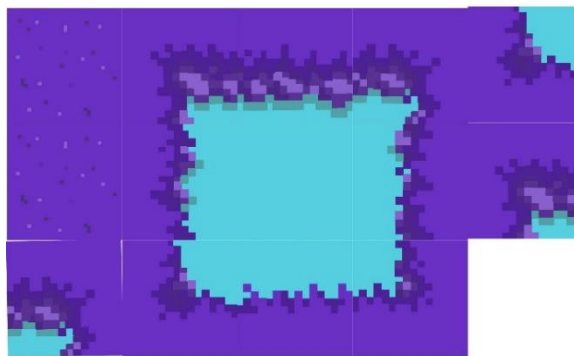


Figura 2. Tiles ordenados ya con sentido

14	14	14
14	14	14
14	14	14

Figura 3. Cuadrícula de 3x3 donde cada celda tiene 14 posibilidades

Viendo la anterior imagen podemos darnos cuenta de que cada celda tiene 14 posibilidades ya que tenemos 14 tiles y todavía ninguna celda ha sido colapsada.

Parche inicial: Se coloca una pieza pequeña en el terreno como parche inicial. Esta pieza puede ser seleccionada pseudoaleatoriamente para comenzar el proceso de generación.

14	4	14
4	1	4
14	4	14

Figura 4. Cuadrícula de 3x3 con una celda colisionada

Análisis y colapso: El algoritmo analiza las reglas y restricciones del juego para asegurarse de que cada nueva pieza se ajuste lógicamente y coherentemente a las piezas ya colocadas en el terreno. El algoritmo colapsa o expande de manera inteligente el parche inicial para determinar qué otros tiles pueden encajar de manera adecuada en su entorno justo como se ve en la figura 4.



Figura 5. Ejemplo de tiles que no conectan

Figura 6. Ejemplo de tiles que conectan

Probabilidades: A medida que se van colocando más tiles en el terreno, el algoritmo utiliza probabilidades para decidir qué pieza se colocará en cada posición. Esto implica que cada vez que se ejecute el algoritmo, se generarán terrenos ligeramente diferentes debido a las probabilidades utilizadas.

Construcción progresiva: El algoritmo continúa colocando tiles y expandiendo el terreno de manera progresiva, asegurándose de mantener la coherencia y las restricciones establecidas previamente.

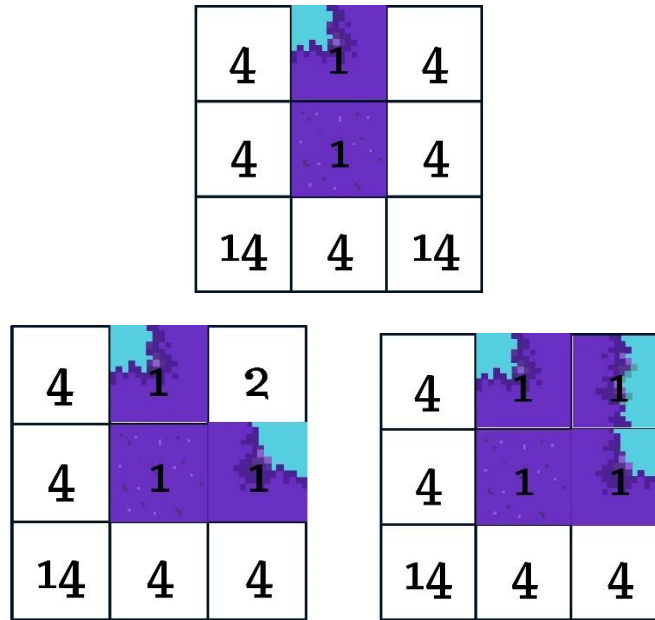


Figura 7. Terreno con dos, tres y cuatro celdas colapsadas

Finalización del terreno: El proceso continúa hasta que todo el espacio de juego esté lleno con tiles y se haya generado el terreno completo.

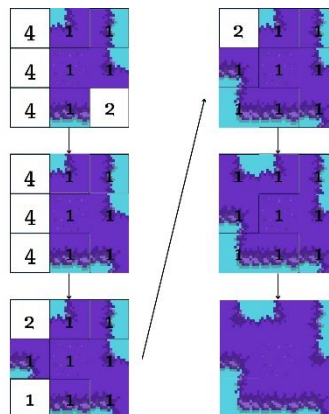


Figura 8. Terreno completándose de manera aleatoria.

Con este proceso, el algoritmo logra generar un escenario pseudoaleatorio y diverso en cada partida.

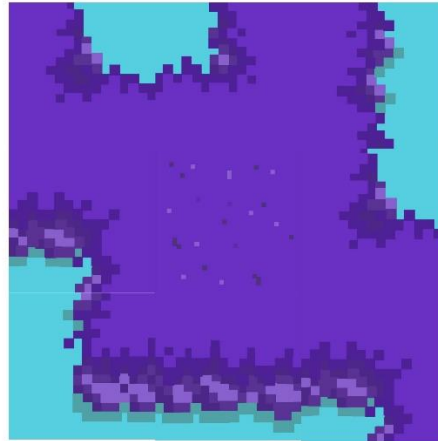


Figura 8. Terreno generado pseudoaleatorio

## Implementación

La creación de este videojuego requirió la generación de un escenario procedimental en el que se escondería un tesoro en una casilla aleatoria. Para lograr esta parte procedural, se desarrollaron dos clases fundamentales: "Celda" y "Opciones". La clase "Celda" representa cada una de las celdas que conforman la cuadrícula, que, a su vez, constituye nuestro terreno. Por otro lado, la clase "Opciones" hace referencia a las diferentes posibilidades o "tiles" dentro de cada celda.

Antes de adentrarnos en los detalles del desarrollo, es imprescindible definir las reglas del videojuego. Para ello, es necesario realizar un análisis de los "tiles" y cómo se componen, tal como se puede observar en la figura a continuación:

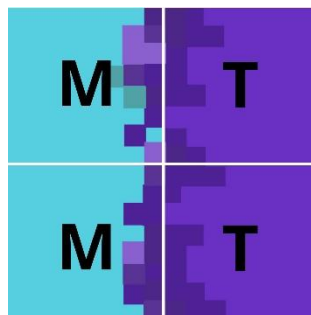


Figura 9. Muestra de un tile

Con esta información como base, podemos explicar con mayor detalle el proceso de implementación del videojuego "Buscando un diamante".

Observamos que las "tiles" que utilizaremos están compuestas por dos tipos: M y T, donde la M representa el Mar y la T la Tierra. A partir de esta observación, podemos deducir que cada borde del "tile" tiene 4 opciones posibles, las cuales serían: MT, TM, TT y MM.

La opción MT indica que el borde en cuestión tiene una transición de Mar a Tierra, mientras que TM representa una transición de Tierra a Mar. Por otro lado, TT indica que el borde está completamente compuesto por Tierra, y MM señala que es completamente Mar.

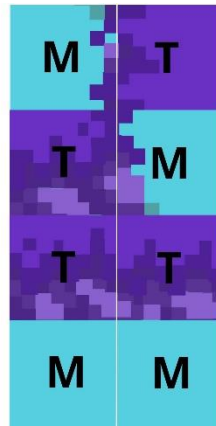


Figura 10. Las diferentes orillas que puede tener un tile

En cuanto a la cuestión del orden y la etiquetación de cada "tile", se presenta la siguiente figura para clarificar y estructurar visualmente las opciones de las orillas:



Figura 11. Reglas para nombrar tile

En esta figura, se ha establecido un etiquetado para cada "tile". Las opciones de la orilla superior se representan como "AB", mientras que las opciones de la orilla inferior se denotan como "CD". A partir de estas opciones, es posible obtener la configuración para las orillas derecha e izquierda, donde la orilla derecha sería representada por "BD" y la izquierda por "AC".

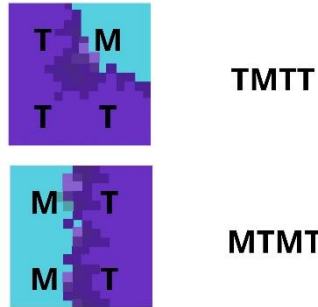


Figura 12. Tiles, TMTT, MTMT

Con esta información, comprendemos que conocemos las orillas de cada "tile" y podemos compararlas para determinar si pueden unirse entre sí o no. Cada instancia de la clase "Celda" cuenta con un arreglo dinámico de instancias de la clase "Opciones", lo que nos permite agregar y eliminar opciones según sea necesario.

Para facilitar la representación del terreno y su disposición, se crea una cuadrícula utilizando una matriz bidimensional de instancias de la clase "Celda". En este caso, se ha establecido un tamaño de terreno de 10x10, como se muestra en la siguiente figura:

(0),(0)	(1),(0)	(2),(0)	(3),(0)	(4),(0)	(5),(0)	(6),(0)	(7),(0)	(8),(0)	(9),(0)
(0),(1)	(1),(1)	(2),(1)	(3),(1)	(4),(1)	(5),(1)	(6),(1)	(7),(1)	(8),(1)	(9),(1)
(0),(2)	(1),(2)	(2),(2)	(3),(2)	(4),(2)	(5),(2)	(6),(2)	(7),(2)	(8),(2)	(9),(2)
(0),(3)	(1),(3)	(2),(3)	(3),(3)	(4),(3)	(5),(3)	(6),(3)	(7),(3)	(8),(3)	(9),(3)
(0),(4)	(1),(4)	(2),(4)	(3),(4)	(4),(4)	(5),(4)	(6),(4)	(7),(4)	(8),(4)	(9),(4)
(0),(5)	(1),(5)	(2),(5)	(3),(5)	(4),(5)	(5),(5)	(6),(5)	(7),(5)	(8),(5)	(9),(5)
(0),(6)	(1),(6)	(2),(6)	(3),(6)	(4),(6)	(5),(6)	(6),(6)	(7),(6)	(8),(6)	(9),(6)
(0),(7)	(1),(7)	(2),(7)	(3),(7)	(4),(7)	(5),(7)	(6),(7)	(7),(7)	(8),(7)	(9),(7)
(0),(8)	(1),(8)	(2),(8)	(3),(8)	(4),(8)	(5),(8)	(6),(8)	(7),(8)	(8),(8)	(9),(8)
(0),(9)	(1),(9)	(2),(9)	(3),(9)	(4),(9)	(5),(9)	(6),(9)	(7),(9)	(8),(9)	(9),(9)



Figura 13. Cuadrícula de 10x10

La figura permitirá visualizar la cuadrícula generada, donde cada celda contendrá sus respectivas opciones, representadas por los "tiles". Al tener la información estructurada de esta manera, se podrá realizar una evaluación y comparación sistemática entre las opciones de las celdas adyacentes para determinar si pueden encajar entre sí o si existen restricciones para su unión.

El algoritmo de generación procedimental para el videojuego "Buscando un diamante" se desarrolla siguiendo los siguientes pasos:

1. Se realiza un análisis de todas las celdas para encontrar aquella que tenga la entropía más baja. La entropía hace referencia a la cantidad de opciones que tiene cada celda. En el inicio del proceso, todas las celdas tienen la misma entropía, por lo que se selecciona la última celda analizada que no tenga una entropía menor que la penúltima celda analizada.
2. La celda seleccionada se colapsa, lo que significa que se elige una opción aleatoriamente de entre las disponibles y se le asigna ese valor.
3. Cada opción tiene asociado un "tile" de los 14 posibles, y con ello, sus reglas. Cada "tile" cuenta con 4 orillas: superior, inferior, izquierda y derecha. Al colapsar una celda y asignarle un valor, sus reglas se propagan a las celdas vecinas. Cada celda posee un arreglo dinámico de opciones, lo que permite agregar o quitar opciones según corresponda. Al colapsar una celda, las celdas vecinas eliminan las opciones que no son compatibles con el valor asignado en cada orilla.
4. Al quitar opciones, la entropía de las celdas disminuye, lo que hace que las celdas con menor cantidad de opciones vayan colapsando progresivamente. Este proceso evita posibles fallos en el algoritmo.
5. El proceso de análisis y colapso de celdas se repite en ciclos hasta que todo el terreno esté lleno y todas las celdas tengan valores asignados.

En resumen, el algoritmo avanza iterativamente evaluando cada celda y seleccionando la de menor entropía para colapsarla y asignarle un valor. Luego, las reglas de ese valor se propagan a las celdas vecinas, lo que conlleva a eliminar opciones incompatibles y reduce la entropía en el proceso. El ciclo se repite hasta que todas las celdas tengan sus valores y el terreno esté completamente generado.

## Resultados

El algoritmo de Colapso de función de onda se ha implementado exitosamente en el entorno de programación "Processing", lo que ha permitido crear el videojuego "Buscando el diamante". Este videojuego presenta un terreno generado proceduralmente, lo que garantiza que cada partida sea única y en cierto sentido infinita, ya que la disposición del terreno cambia en cada ocasión.

El funcionamiento del videojuego ha sido probado exhaustivamente y no se han detectado errores o comportamientos inesperados. El algoritmo ha demostrado su eficacia en la generación coherente y visualmente atractiva del terreno, lo que contribuye a una experiencia de juego fluida y entretenida.

El objetivo del videojuego es encontrar un tesoro oculto en alguna parte del terreno, el cual también es generado pseudo aleatoriamente. Los jugadores cuentan con un total de 3 intentos o palas para excavar y encontrar el tesoro. Para ayudar en la búsqueda, disponen de un mapa que proporciona una pista sobre la ubicación del terreno. Este mapa siempre se presenta rotado en 90, 180 o 270 grados, también de forma aleatoria, lo que aumenta la rejugabilidad y desafío del juego.



La elaboración del escenario procedural se ha logrado mediante la utilización de 14 tiles diferentes, los cuales han sido previamente mostrados en este informe en diversas ocasiones. Estas piezas se interconectan para generar un terreno que combina elementos de mar y tierra, con colores y ambientación que evocan una atmósfera de ciencia ficción o de un mundo extraterrestre.



Figura 14. Pantalla de jugabilidad del juego.

También se hizo una pantalla principal donde se puede pasar a jugar o leer las instrucciones.



Figura 15. Pantalla de inicio del juego.



Figura 16. Pantalla de instrucciones del juego.

Cabe aclarar que los controles del juego son mediante las teclas de W, A, S, D y la tecla de ESPACIO para cavar, pero el control de los botones de jugar, regresar, instrucciones es mediante clic del ratón en ellos.

## Conclusión

La aplicación del algoritmo de Colapso de la función de onda en el desarrollo del videojuego ha demostrado ser un acierto en la generación de un escenario proceduralmente generado y brinda una experiencia de aleatoriedad y dinamismo en el juego. La utilización de tiles y reglas establecidas ha permitido la creación de un entorno coherente y estéticamente atractivo, donde cada partida se presenta como una aventura impredecible y única, ya que asegura que ningún escenario sea igual al anterior.

El algoritmo ha superado las pruebas con éxito, demostrando un funcionamiento sin errores ni comportamientos inesperados y ha sido una herramienta invaluable y nos muestra todo el potencial que tiene en la creación de mundos infinitos y la diversidad en cada partida al menos en este videojuego ha sido su sello distintivo.

Es emocionante poder seguir explorando la generación procedural y sus otros muchos algoritmos para ver hasta donde es capaz de llegar en la creación de videojuegos y arte.

## Bibliografía/Referencias

- Gumin, M. G. (s. f.). GitHub - mxGMn/WaveFunctionCollapse: Bitmap & Tilemap generation from a single example with the help of ideas from quantum mechanics. GitHub. <https://github.com/mxgmn/WaveFunctionCollapse>
- The wavefunction collapse algorithm explained very clearly | Robert Heaton. (2018, 17 diciembre). Robert Heaton. <https://robertheaton.com/2018/12/17/wavefunction-collapse-algorithm/>
- Martin Donald. (2020, 31 julio). Superpositions, sudoku, the wave function collapse algorithm. [Video]. YouTube. <https://www.youtube.com/watch?v=2SuvO4Gi7uY>
- Cheng, D., Han, H., & Fei, G. (2020). Automatic generation of game levels based on controllable wave Function collapse algorithm. En Lecture Notes in Computer Science (pp. 37-50). [https://doi.org/10.1007/978-3-030-65736-9\\_3](https://doi.org/10.1007/978-3-030-65736-9_3)