

Generación Procedural de Mapas Basado en *The Binding of Isaac*

José Fernando Hernández Vera

29/06

Resumen

Este trabajo presenta una implementación de generación procedural de mapas inspirada en el videojuego *The Binding of Isaac*, utilizando el algoritmo clásico de búsqueda en profundidad (DFS). Se describe la lógica de generación, los parámetros configurables y el proceso de postprocesado para colocar salas especiales. A través de múltiples ejecuciones, se valida que el sistema genera mapas acíclicos, conectados y variados, adecuados para juegos con alta rejugabilidad y exploración estratégica.

1. Introducción

Juego de referencia

The Binding of Isaac es un videojuego de acción tipo rogue-like en el que el jugador progresa a través de niveles generados proceduralmente. Cada nivel consiste en una serie de habitaciones interconectadas, donde el jugador debe explorar, enfrentarse a enemigos, recolectar objetos y, finalmente, derrotar a un jefe. La estructura del mapa varía en cada partida, lo que incrementa la rejugabilidad y el interés por la exploración.

Características de interés a recrear

Se han identificado las siguientes características del juego original que se busca replicar mediante generación procedural:

- **Plano del piso:** El mapa se compone de una cuadrícula de habitaciones cuadradas distribuidas de manera no determinista.
- **Habitaciones especiales:** Algunas habitaciones son asignadas como especiales (tiendas, salas de recompensas, jefes) y se colocan en posiciones clave dentro del mapa.
- **Conectividad:** Las habitaciones se conectan para formar un grafo que permite un recorrido completo desde la entrada hasta el jefe. No todas las habitaciones están conectadas entre sí, y pueden existir ramas secundarias opcionales.

- **Tamaño del mapa:** El número de habitaciones y la complejidad del diseño aumentan progresivamente a medida que el jugador avanza de nivel.
- **Ausencia de bucles:** El mapa es acíclico, lo que significa que no existen ciclos; el jugador no puede regresar a habitaciones previamente visitadas en el mismo nivel.

Justificación y selección de algoritmos

Para modelar estas características, se empleó el algoritmo de generación de laberintos basado en árboles, específicamente:

- **DFS Maze Generation (Depth-First Search):** útil para crear un camino principal con bifurcaciones ocasionales.

Este algoritmo garantiza estructuras acíclicas, permite mantener conectividad global y ofrece flexibilidad para insertar reglas adicionales como habitaciones especiales, lo que lo hace adecuado para el diseño procedural de este tipo de juegos.

2. Estado del Arte

Diversos videojuegos han implementado generación procedural de mapas con características similares a las aquí descritas, algunos son:

- **Enter the Gungeon:** Utiliza una estructura de grafo acíclico, generada mediante algoritmos como Prim o DFS, para asegurar conectividad entre habitaciones clave como entrada, jefe, tienda, etc sin permitir bucles. Esto proporciona una estructura ramificada con caminos opcionales.
- **Spelunky:** Emplea una mezcla de Random Walk y plantillas predefinidas para construir niveles con múltiples rutas y espacios semiabierto. Aunque no genera árboles estrictos, introduce elementos aleatorios dentro de bloques estructurados, logrando variedad y coherencia.
- **Dead Cells:** Combina algoritmos tipo BSP (Binary Space Partitioning) y lógica de selección procedural para generar mapas acíclicos con caminos secundarios. Se enfoca en ofrecer rutas con decisiones estratégicas, manteniendo la progresión controlada.
- **Rogue:** Uno de los pioneros en usar DFS (búsqueda en profundidad) y BFS (búsqueda en anchura) modificados para crear laberintos acíclicos, con un camino principal y ramas opcionales. Esta aproximación ha influido en muchos títulos actuales del género *rogue-like*.

3. Metodología

Algoritmo a implementar

- **DFS Maze Generation (búsqueda en profundidad):** se utiliza para crear un camino principal desde la entrada hasta el jefe, con la posibilidad de incluir ramificaciones.

Este método permite equilibrar aleatoriedad con estructura, favoreciendo la diversidad de mapas sin comprometer la jugabilidad.

Variaciones propuestas

- Cambiar algunas habitaciones especiales y quitar a los enemigos, haciendo que las habitaciones especiales den tesoros o bien pueden ser una trampa.
- Reglas para controlar el tamaño y la ramificación del mapa según el nivel de complejidad que queremos en el juego.
- Validaciones post-generación para asegurar conectividad mínima y cumplimiento de restricciones sin ciclos, tamaño adecuado y habitaciones clave presentes.

Hiperparámetros a modificar

- Tamaño del mapa: número total de habitaciones por nivel.
- Número y tipo de habitaciones especiales.
- Profundidad máxima que sería la distancia desde la entrada hasta la sala del jefe.

Ejecución de experimentos

Se realizaron múltiples ejecuciones por cada configuración de hiperparámetros, generando mapas bajo diferentes condiciones. Las métricas a evaluar incluyen:

- Longitud del camino principal.
- Número de habitaciones totales.
- Cantidad y ubicación de salas especiales.
- Variedad estructural entre ejecuciones.

4. Resultados

Se realizaron múltiples ejecuciones del sistema utilizando el algoritmo DFS, variando parámetros como tamaño del mapa y profundidad máxima. A continuación, se presentan los datos recopilados y los parámetros que usamos.

Parámetros utilizados

- Tamaño del mapa: 10×10
- Profundidad máxima: 10
- Salas especiales colocadas: 1 tesoro, 1 tienda, 2 trampas

Datos generados

A continuación, se presentan los datos recopilados de 10 pruebas con distintas profundidades de generación (5 y 10). Se muestran métricas clave como el total de habitaciones, cantidad de salas especiales y trampas.

Pruebas con profundidad 10

Prueba	Total habitaciones	Tesoro	Trampas	Profundidad
1	65	1	2	10
2	73	1	3	10
3	66	3	4	10
4	49	2	3	10
5	78	4	4	10
Promedio	66.2	2.2	3.2	10

Pruebas con profundidad 5

Prueba	Total habitaciones	Tesoro	Trampas	Profundidad
6	28	3	4	5
7	46	2	1	5
8	31	2	4	5
9	33	4	2	5
10	31	2	4	5
Promedio	33.8	2.6	3.0	5

Visualización del mapa

A continuación se mostraran los mapas generados correspondientes a cada una de las pruebas realizadas donde el gato corresponde a un sala normal, el simbolo del dinero corresponde a la tienda, la S corresponde en donde empieza el jugador, los O corresponden a que no hay nada, la T corresponde a las trampas y la B corresponde a la sala del jefe.

```

--- Generando mapa con DFS ---
0 0 0 0 # #---X---# B 0
    |   |   |   |   |
0 0 0 #---#---# # #---#---#
    |   |   |   |   |
0 0 0 0 #---# # # # 0
    |   |   |   |   |
0 0 0 # #---#---# #---# #
    |   |   |   |   |
0 0 # # # T # #---#---#
    |   |   |   |   |
0 #---#---# # S $---# #---#
    |   |   |   |   |
0 0 # #---#---# # # # 0
    |   |   |   |   |
0 #---#---#---# #---#---#---#
    |   |   |   |   |
0 0 # # # 0 # X # 0
    |   |   |   |   |
0 0 0 # 0 0 0 # 0 0

```

Mapa 1

```

--- Generando mapa con DFS ---
0 0 0 0 # 0 0 0 0 0
    |   |   |   |   |
0 0 0 # # # 0 # 0 0
    |   |   |   |   |
0 0 #---#---#---#---X #---# 0
    |   |   |   |   |
0 # #---#---# # #---#---#---#
    |   |   |   |   |
#---#---# #---#---#---# X # 0
    |   |   |   |   |
#---#---#---# # S # # 0 0
    |   |   |   |   |
#---#---$---#---X---# #---#---# 0
    |   |   |   |   |
#---#---#---# # #---# # 0 0
    |   |   |   |   |
B---#---# # # # # # 0
    |   |   |   |   |
0 # #---#---#---T---#---#---#

```

Mapa 2

```

--- Generando mapa con DFS ---
0 0 0 0 # 0 # 0 0 0
    |   |   |   |   |
0 0 0 T---#---# #---# 0 0
    |   |   |   |   |
B 0 T---#---X---#---#---# 0
    |   |   |   |   |
#---# #---X---#---# #---# 0 0
    |   |   |   |   |
#---#---# $---# # # 0 0 0
    |   |   |   |   |
#---#---#---# #---S # # 0 0
    |   |   |   |   |
T # # #---#---X---#---# 0
    |   |   |   |   |
#---#---#---# # # X # 0 0
    |   |   |   |   |
#---#---# #---#---#---# 0 0 0
    |   |   |   |   |
0 # 0 #---#---# 0 0 0 0

```

Mapa 3

```

--- Generando mapa con DFS ---
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 # 0 0 0 0 0
    |   |   |   |   |
0 0 0 $ #---# 0 0 0 0
    |   |   |   |   |
0 0 #---X---#---#---# 0 # 0
    |   |   |   |   |
0 0 0 # # S---# # X #
    |   |   |   |   |
0 0 T #---# X---# #---#---#
    |   |   |   |   |
0 #---#---# #---#---#---# #
    |   |   |   |   |
0 0 #---#---# # #---#---#---#
    |   |   |   |   |
0 0 0 # #---T---# #---# B

```

Mapa 4

```

--- Generando mapa con DFS ---
0 0 B 0 0 0 0 0 0 0
    |
0 # X # 0 # 0 0 0 0
    | | | |
#---#---#---T---# # # 0 # 0
    | | | |
0 # # # #---#---#---$ #---#
    | | | |
#---# #---#---# # # #---#---#
    | | | |
#---#---# #---T S---#---# # #
    | | | |
# #---T # #---# # #---#---#
    | | | |
0 # #---#---#---#---# #---#
    | | | |
#---X---#---#---# # # X---#---X
    | | | |
0 # # # 0 # 0 #---T---#

```

Mapa 5

```

--- Generando mapa con DFS ---
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 # 0 # 0 0 0
    | |
0 0 0 # #---# # # 0 0
    | | |
0 0 #---X---T S T---X---# 0
    | | |
0 0 0 # X---#---# T 0 0
    | |
0 0 B---#---#---X 0 0 0
    | |
0 0 0 # $ # 0 0 0 0
    |
0 0 0 0 # 0 0 0 0 0

```

Mapa 6

```

--- Generando mapa con DFS ---
0 0 0 0 0 0 0 0 0 0
0 0 0 0 B 0 0 0 0 0
    |
0 0 0 #---# # 0 # 0 0
    | |
0 0 #---#---#---X---# # # 0
    | | |
0 # # # # # #---#---#---T
    | | |
#---#---#---#---#---S---# # # 0
    | | |
0 #---# # # $ # # 0 0
    | |
0 0 #---#---#---#---# 0 0 0
    | |
0 0 0 # # # 0 0 0 0
    |
0 0 0 0 T 0 0 0 0 0

```

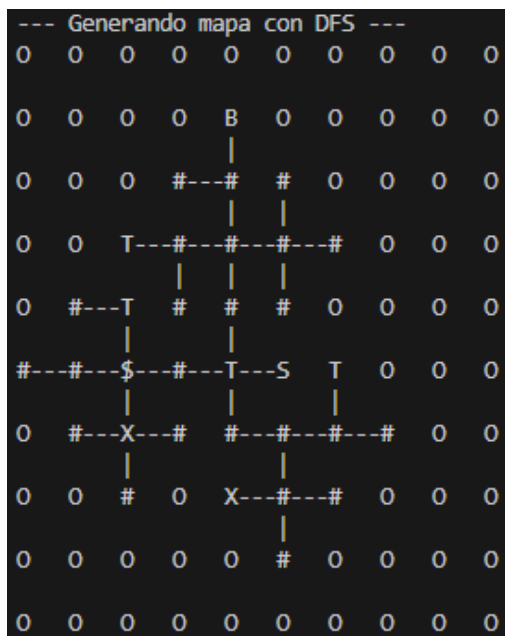
Mapa 7

```

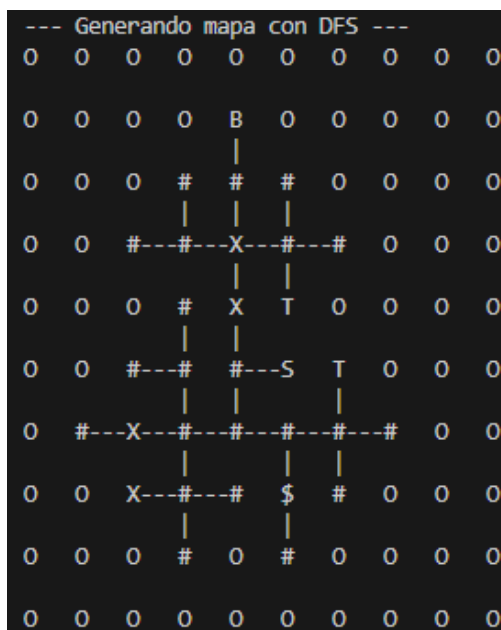
--- Generando mapa con DFS ---
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 # 0 0 0
    |
0 0 0 # 0 X---X # 0 0
    | |
0 0 T X---# S T---#---# 0
    | | |
0 B---#---#---#---# # 0 0
    | |
0 0 #---# #---#---#---# 0
    | |
0 0 0 $ 0 # X # 0 0
    |
0 0 0 0 0 0 # 0 0 0

```

Mapa 8



Mapa 9



Mapa 10

5. Discusión

El algoritmo DFS permitió generar mapas con un camino principal bien definido y ramificaciones laterales, lo cual se alinea con el estilo de exploración del juego *The Binding of Isaac*. La profundidad máxima actúa como límite de complejidad de la mazmorra, controlando la distancia desde la entrada hasta la sala del jefe.

Se pudo observar que, al limitar la profundidad, se generan mapas más compactos, y al incrementarla, se obtienen estructuras más laberínticas. Las salas especiales se distribuyen correctamente gracias a una etapa de postprocesado que selecciona habitaciones válidas de forma aleatoria, manteniendo la coherencia estructural y la rejugabilidad, produciendo que al momento de querer configurar la dificultad esta se podría hacer más sencillamente al solo tener que cambiar la profundidad.

En todas las ejecuciones, se cumplió con las restricciones deseadas: mapa acíclico, conectividad garantizada, y salas clave (inicio, jefe, tienda, trampa y tesoro) posicionadas de forma estratégica.

6. Conclusiones

La implementación del algoritmo DFS para generación procedural de mapas demostró ser efectiva para crear estructuras explorables, conectadas y sin ciclos, adecuadas para juegos tipo rogue-like. Gracias a su sencillez y flexibilidad, el sistema permite controlar la dificultad y diversidad del mapa mediante parámetros como la profundidad máxima y la cantidad de

salas especiales.

El enfoque basado en generación recursiva y validación posterior garantiza la creación de mapas coherentes que incentivan la exploración y la toma de decisiones estratégicas por parte del jugador.

Este trabajo serviría como base para futuras mejoras como generación e integración de enemigos dinámicos, o adaptación progresiva de la dificultad según el nivel alcanzado.

7. Bibliografía

- Link del juego The Binding of Isaac.
https://store.steampowered.com/app/113200/The_Binding_of_Isaac/
- Link del juego Dead Cells.
https://store.steampowered.com/app/588650/Dead_Cells/
- Link del juego Spelunky.
<https://store.steampowered.com/app/239350/Spelunky/>
- Link del juego Enter the Gungeon.
https://store.steampowered.com/app/311690/Enter_the_Gungeon/
- Link del juego Rogue.
<https://store.steampowered.com/app/1443430/Rogue/>
- Link de investigación del The Binding of Isaac.
<https://www.boristhebrave.com/2020/09/12/dungeon-generation-in-binding-of-isaac/>
- Link de investigación del Depth First Search (DFS).
<https://interviewkickstart.com/blogs/learn/depth-first-search-algorithm>
- Link de investigación en general de algoritmos de generación procedural.
<https://riull.ull.es/xmlui/bitstream/handle/915/39978/Generacion%20Procedural%20de%20Contenido%20D%20en%20Videojuegos.pdf?sequence=1&isAllowed=y>