

# GENERADOR PROCEDURAL DE DUNGEONS PARA VIDEO JUEGOS 3D

*Juan David Gomez Quiñones*

*juandavidgomez@javerianacali.edu.co*

Pontificia Universidad Javeriana Cali, Colombia

## RESUMEN

Las Dungeons (o mazmorras por su traducción al español) son escenarios de juego muy comúnmente usados en varios géneros de vídeo juegos, sobre todo en el genero RPG (Role-playing game). Las Dungeons son escenarios donde los jugadores pueden encontrar diferentes tipos de misiones, acertijos, enemigos y recompensas. Tradicionalmente en el proceso de desarrollo de vídeo juegos, la arquitectura y diseño de estos escenarios se realizan de forma manual y debido a la alta demanda de este tipo de escenarios dentro de un juego, el desarrollo termina siendo demasiado costoso. Debido a lo anterior, es muy común encontrar muchas Dungeons lineales, monótonas, poco entretenidas y con una baja probabilidad de rejugabilidad. En este trabajo se desarrolló un algoritmo capaz de generar la representación 3D de una Dungeon y algunos de sus elementos principales, tomando como entrada un conjunto de parámetros y mallas 3D que permiten la personalización de la Dungeon a las necesidades del usuario.

**Palabras Clave**— Dungeon, Vídeo Juegos, RPG, Representación 3D, Generación Aleatoria, Parámetros, Rejugabilidad.

## 1. INTRODUCCIÓN

Los escenarios en los vídeo juegos son uno de los elementos más importantes dentro de la mayoría de géneros, dado a que son los espacios donde los jugadores experimentan todo lo que un vídeo juego tiene para ofrecer. Así mismo, Las Dungeons se

han convertido en un tipo de escenario muy común y recurrente en múltiples géneros de vídeo juegos, sobre todo, en el género RPG.

Es por eso que muchos desarrolladores tratan de agregar una gran cantidad de escenarios a sus juegos con diseños increíbles e inmersivos que ayuden a explotar todos los aspectos positivos que el juego puede ofrecerle al jugador, sin embargo, ofrecer la cantidad ideal de escenarios con un buen diseño no es tan trivial como suena [1]. Normalmente el proceso para desarrollar un conjunto de Dungeons es un proceso costoso donde es necesario personal especializado como diseñadores, ingenieros multimedia o de sistemas, tiempo, y por supuesto dinero.

Este proyecto plantea un algoritmo que busca ayudar a los desarrolladores a generar Dungeons complejas interesantes de forma aleatoria y automática en base a una serie de parámetros que se adaptan a los requisitos del escenario. Donde no solo se busca establecer su estructura topográfica, sino que también posicionar elementos claves como recompensas y enemigos.

## 2. FUNDAMENTACIÓN TEÓRICA

Para el desarrollo del presente proyecto, se deben de tener conocimientos tanto de ciencias de la computación como de mecánicas de Vídeo Juegos. A continuación se presentan las bases teóricas sobre las que se trabajó el proyecto.

**Grafo No Dirigido.** Un grafo no dirigido es un tipo de grafo  $G = (V, E)$ , donde  $V$  es un conjunto finito de nodos o vértices y  $E$  es un conjunto finito de

aristas, que relacionan los nodos de forma simétrica y no tienen un sentido definido. En el proyecto se usa un grafo no dirigido como representación de una Dungeon, donde  $V$  representa el conjunto de habitaciones o sub zonas y  $E$  el conjunto de puertas que conecta dos habitaciones.

**Modelo de Erdős-Rényi.** Roney [2] muestra que el modelo de *Erdos - Rényi* es uno de los modelos clásicos que permiten la construcción de un grafo aleatorio  $G(V,p)$  a través de una distribución de Poisson, partiendo de un conjunto  $V$  de vértices. Para todo  $V \in G$ , se asigna una probabilidad  $p \in (0,1)$  tal que para cada  $V_i, V_j \in G$ ,  $p$  determinará si existe una conexión entre  $V_i, V_j$ , entre mas alto sea el valor de  $p$  existirá una mayor probabilidad de conexión entre los dos vértices y aumentara la conectividad del grafo  $G$ .

En el presente proyecto fue necesario hacer una modificación al algoritmo para garantizar que todos los vértices  $V \in G$  tengan al menos una conexión y no tengan más conexiones que el número de puertas máximas definidas por la malla 3D de la habitación. La figura 1 muestra el algoritmo de *Erdős - Rényi*.

---

**Algorithm 1:** Original ER

---

**Input:**  $v$  : number of vertices, indexed 0 to  $v - 1$ ;  
 $p$  : inclusion probability  
**Output:**  $G$  : Erdős-Rényi graph

```

1  $G = \emptyset$ ;
2 for  $i = 0$  to  $v - 1$  do
3   for  $j = 0$  to  $v - 1$  do
4     Generate a uniform random number  $\theta \in [0, 1)$ ;
5     if  $\theta < p$  then
6        $G \leftarrow (i, j)$ ;
```

---

**Figura 1:** Algoritmo de Dijkstra [3]

**Algoritmo de Dijkstra.** Para poder posicionar los jefes y tesoros dentro de la Dungeon de una forma balanceada, es necesario conocer que tan cerca está un nodo con los elementos mencionados anteriormente de otros nodos importantes como el inicio o el final de la Dungeon. Para esto se planea usar el algoritmo de Dijkstra para grafos no dirigidos. Erickson [4] muestra que dado un grafo  $G(V,E)$  el algoritmo de Dijkstra permite encontrar el camino mas corto desde un nodo a todos los demás nodos.

El algoritmo de Dijkstra funciona de forma similar al algoritmo BFS, solo que en lugar de usar una cola de tipo FIFO (First In First Out), se usa un cola de prioridad donde la clave de un vértice  $V$  es su distancia tentativa  $dist(V)$ , ver figura 2.

```

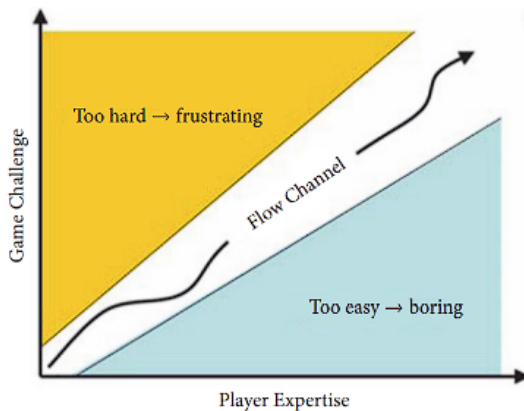
DIJKSTRA( $s$ ):
  INITSSSP( $s$ )
  INSERT( $s, 0$ )
  while the priority queue is not empty
     $u \leftarrow$  EXTRACTMIN()
    for all edges  $u \rightarrow v$ 
      if  $u \rightarrow v$  is tense
        RELAX( $u \rightarrow v$ )
        if  $v$  is in the priority queue
          DECREASEKEY( $v, dist(v)$ )
        else
          INSERT( $v, dist(v)$ )
```

**Figura 2:** Algoritmo de Dijkstra

**Fundamentos de Dungeons & Dragons.** Es un juego de mesa de fantasía de tablero con lápiz y papel donde la mayoría de los jugadores toman el rol de un personaje de fantasía y un jugador toma el papel de Maestro de Juego y es el encargado de guiar las dinámicas de cada sección de juego. La importancia de este juego en este proyecto es que Dungeons & Dragons es un juego bien conocido por ser el principio de los juegos RPG modernos y ha venido evolucionando en múltiples ediciones desde la fecha de su lanzamiento en 1974. Debido a la larga historia de este juego, se han creado varios libros guía oficiales para los Maestros de Juego, los cuales explican aspectos importantes para el desarrollo de este proyecto, como lo son los modelos a seguir para el posicionamiento de enemigos y recompensas para cada tipo de Dungeon [5].

**Dificultad en los Video Juegos.** Hace referencia a el reto que supone un vídeo juego, o un componente de este a los jugadores y es directamente dependiente de las intenciones de los desarrolladores. Sin embargo Csikszentmihalyi [6] señala que es importante saber medir el nivel de dificultad y propone un canal de flujo en el que la dificultad del juego debería de ser establecida en base a el nivel del desafío y la experiencia del jugador. De esta for-

ma se evitan niveles demasiado difíciles y por ende frustrante al igual que niveles demasiado fáciles.



**Figura 3:** Curva de Dificultad propuesta por Csikszentmihalyi

Por lo tanto es común encontrar dos tipos modelos de dificultad en los video juegos:

- Un modelo adaptativo o dinámico que según Zohaib [7] tiene como objetivo es generar siempre escenarios correspondientes al nivel del jugador adaptando las características de los enemigos y el entorno para poder proporcionar siempre un desafío placentero que no sea ni muy fácil ni muy difícil. Es posible encontrar este tipo de dificultad en juegos como "The elder scrolls Online" donde el nivel de los enemigos en las Dungeons avanza según el nivel del jugador.
- Un modelo constante donde al contrario de la dificultad adaptativa, cada escenario tiene una dificultad predefinida y es fija según los criterios del desarrollador. Si el jugador se encuentra en el momento del juego adecuado podrá encontrar una gran experiencia las primeras veces que juegue la Dungeon ya que un desarrollador definió de forma manual los parámetros de dificultad en base a su experiencia. Sin embargo, los parámetros de la Dungeon no progresan con el jugador, por lo que si este vuelve en una etapa posterior del juego, encontrará un nivel mucho

más sencillo. Por otro lado, si el jugador trata de saltar a un nivel muy por encima de su nivel actual encontrará un nivel demasiado difícil y letal [8]. Este modelo se puede ver en juegos "World of Warcraft" donde cada Dungeon está pensado para un momento en específico del juego.

### 3. RESULTADOS

La implementación de este proyecto se realizó en el motor de video juegos Unreal Engine 4 y consta de dos módulos fundamentales. En primer lugar se tiene un módulo de lógica en el lenguaje de C++ el cual es el encargado de crear la representación en un grafo de una Dungeon y la definición de sus elementos claves como enemigos y recompensas. En segundo lugar se tiene un módulo de visualización construido con la tecnología de Blueprints de Unreal Engine, el cual construye la Dungeon Jugable en el mundo virtual a partir de las salidas del módulo lógico.

Los módulos mencionados anteriormente necesitaran de parámetros de entrada proporcionados por el usuario como lo son la longitud de la Dungeon, la probabilidad de conexión entre habitaciones, la cantidad de salidas máximas que tiene una habitación, la recompensa total de la Dungeon, el modelo de dificultad (Constante o adaptativo), el nivel promedio de los jugadores objetivo, la dificultad de los enemigos, entre otros. Con estos parámetros el módulo lógico genera un grafo aleatorio mediante el modelo de *Erdős – Rényi* para luego convertirlo en un grafo conexo verificando que existe al menos un camino que recorra todos los nodos del grafo y de esta forma garantizando que todas las habitaciones de la Dungeon podrán ser alcanzadas por los jugadores.

Posteriormente el módulo lógico define de que manera se deben posicionar las recompensas y los enemigos dentro de la Dungeons. Una iteración del algoritmo de *Dijkstra* calculara cual es el vértice más lejano al inicio de la Dungeon y posicionara al

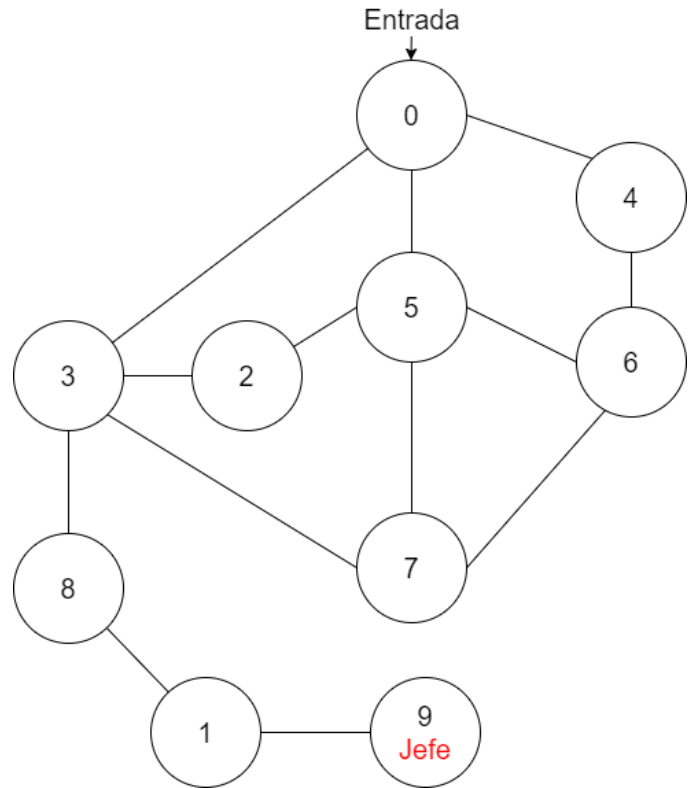
Jefe final en dicho vértice. Seguidamente se posicionaran los enemigos y las recompensas según el modelo de dificultad establecido en los parámetros de entrada. Un modelo constante posicionara de modo uniforme la cantidad de enemigos definida por el desarrollador y establecerá una recompensa por cada enemigo según el monto recompensa total de la Dungeon y la cantidad correspondiente al Jefe. En contraparte, un modelo adaptativo se basara en los modelos de *Dungeons&Dragons* y los atributos de los jugadores para definir cuantos enemigos posicionar en la Dungeon y cual seria la recompensa por cada enemigo abatido según el monto de recompensa total y la bonificación por dificultad de juego.

Finalmente el modulo de visualización creara la representación de la Dungeon a través de las herramientas de Unreal Engine 4. Permitiéndole al usuario importar mallas 3D personalizadas para que algoritmo escoja de forma aleatoria una de las mallas 3D para la construcción de una habitación, siempre y cuando la malla 3D cumplan con los atributos de la habitación. Por ultimo se crean las conexiones entre habitaciones a través de teletransportadores instantáneos posicionados según la topología del grafo generado en el modulo lógico.

La figura 4 muestra un grafo generado bajo parámetros pensados para una Dungeon de dificultad media con una distribución uniforme de enemigos y recompensas, La tabla 1 muestra el posicionamiento de enemigos y recompensas por habitación para la misma Dungeon.

#### 4. CONCLUSION

El algoritmo desarrollado demostró ser capaz de crear diversos tipos de Dungeons de forma aleatoria y flexible a partir de los parámetros proporcionados, permitiendo al usuario definir que tipo de Dungeon esta buscando y personalizarla a sus necesidades. Adicionalmente la matriz de mallas 3D permite no solo crear Dungeons diferentes a partir de un mismo conjunto de entradas si no que



**Figura 4:** Grafo de la Dungeon

| Habitación | Enemigos | Recompensa |
|------------|----------|------------|
| 0          | 2        | 9332       |
| 1          | 2        | 9332       |
| 2          | 2        | 9332       |
| 3          | 1        | 4666       |
| 4          | 1        | 4666       |
| 5          | 2        | 9332       |
| 6          | 1        | 4666       |
| 7          | 2        | 9332       |
| 8          | 2        | 9332       |
| 9          | Jefe     | 30000      |

**Tabla 1:** Atributos por Habitación

también, permite no solo crear Dungeons diferentes a partir de una misma topología de un grafo, si no que también permite que el usuario agregue su toque personal a la Dungeon creada por el generador.

Debido a que este trabajo ayuda a automatizar la generación de Dungeons de diferentes tamaños y

características, el costo de desarrollo por cada Dungeon se vería disminuido y permitiría a los desarrolladores implementar Dungeons complejas y entretenidas con una mayor frecuencia dentro de su vídeo juegos. Adicionalmente, las Dungeons creadas mediante este algoritmo le otorgaría a los jugadores experiencias mucho más completas y dinámicas. Así mismo, al poder crear varias topologías y definir varias formas de distribuir los enemigos y recompensas para una misma Dungeon, se le ofrece a los jugadores la oportunidad volver a jugar una Dungeon como si fuera la primera vez, aumentando el nivel de diversión y por ende la cantidad de horas invertidas en el juego.

## 5. REFERENCIAS

- [1] D. Cox, “Interior design and environment art: Mastering space, mastering place.”
- [2] L. G. Rooney, “Random graph models and matchings,” 2019.
- [3] S. Nobari, X. Lu, P. Karras, and S. Bressan, “Fast random graph generation,” 2011.
- [4] J. Erickson, *Algorithms*. 1st ed., 2019.
- [5] M. Mearls, J. Crawford, C. Perkins, J. Wyatt, R. J. Schwalb, R. Thompson, P. Lee, S. F. Gray, M. Carter, C. Sims, J. C. Wilkes, G. Bilsland, and I. W. of the Coast, *Dungeon master’s guide*. 2014.
- [6] M. Csikszentmihalyi, “The psychology of optimal experience,” *Harper Row*, 2009.
- [7] M. Zohaib, “Dynamic difficulty adjustment (dda) in computer games: A review,” 2018.
- [8] D. Soham, “What is video game difficulty and how does it work?,” 2021.