

# Modelación del Juego *Dungeons & Dragons* cómo Sistema Dinámico.

Almeida Rodríguez, Jerónimo<sup>1,2 \*</sup> and Ruiz Melo, Jean Paul<sup>1,2 \*</sup>

<sup>1</sup>Universidad Nacional Autónoma de México, UNAM

<sup>2</sup>Facultad de Ciencias UNAM

\*These authors contributed equally.

Las redes son una herramienta muy útil para modelar sistemas complejos, poder estudiarlos, extraer información y hacer predicciones sobre el comportamiento del mismo. En general los sistemas que se estudian están basados en estructuras que existen en la naturaleza, son parte del comportamiento de ciertos fenómenos biológicos o resultado del desarrollo social de la humanidad. Este artículo toma otro enfoque. En lugar de basarnos en un sistema existente, creamos el nuestro basado en las reglas del juego de rol *Dungeons & Dragons* y simulándolo con autómatas celulares con distintos niveles de interacción. El objetivo de este trabajo es encontrar los principales tipos de estructuras que se forman después de cierto número de iteraciones según las propiedades de los actores del sistema.

**Introducción.** El sistema que diseñamos está formado de tres partes: casillas, personajes y monstruos. De esto, queremos ver que suele ser mas importante a través del tiempo.

**Descripción del Sistema.** Los tres elementos principales de nuestro sistema son:

- **Casillas:** son las localizaciones dentro del mundo. Pueden ser ciudades, contener tesoros o estar vacías. Los personajes y monstruos interactúan con ellas.
- **Personajes:** Son agentes con un cierto grado de inteligencia artificial que les permite buscar y obtener tesoros, luchar o huir de monstruos y/o comprar objetos que le permitan mejorar sus propiedades (ataque, defensa, salud) e incrementar su poder.
- **Monstruos:** Al igual que los personajes, los monstruos son agentes autónomos. Estos atacan ciudades o personajes dependiendo de que tan poderosos sean.

Un juego de rol consiste de muchas reglas y elementos, así que nosotros nos vamos a enfocar en los siguientes aspectos:

- Un número arbitrario de personajes pertenecen a cada ciudad.
- Los personajes pueden ser de una de 3 clases. Cada una de ellas tienen un rol importante al inicio, mitad y final del juego.
- Los monstruos que se mueven por el mundo.
- Los personajes salen de las ciudades para recoger tesoros.

- Los monstruos pueden atacar a los personajes.
- Si un personaje gana contra un monstruo, gana experiencia.
- Si un personaje gana suficiente experiencia, sube un nivel e incrementa su poder.
- A lo mas 6 personajes pueden unirse para formar un equipo.
- Los personajes buscan tesoros para traerlos a las ciudades a las que pertenecen. Las ciudades crecen entre más tesoros les traigan los personajes.
- Si los personajes recogen suficientes tesoros, pueden comprar armas incrementar su poder. Esto es independiente de su nivel.
- Los monstruos ganan poder a través del tiempo.
- Los monstruos pueden destruir una ciudad si el poder que tienen sobrepasa el tamaño de una ciudad.

## Marco Teórico.

**Metodología.** Esta Investigación consiste de dos programas: uno que simula el sistema y genera los datos y otro que analiza los datos y nos del cual extraeremos la información.

**Simulador.** El programa del simulador que se encuentra en la carpeta `proyecto` está escrito en JAVA y consiste de las siguientes clases:

- **Proyecto:** Es la “columna vertebral” del sistema. Se encarga de la representación gráfica del mundo, de crear a los elementos que le componen y de proporcionar los métodos para que los agentes se muevan dentro del mundo y exploren su espacio de búsqueda.
- **World:** Es la estructura del mundo en sí. Su estructura principal es una matriz bidimensional de celdas (Cell) que representa un plano del mundo. Tiene una lista de ciudades y lleva la cuenta del número de tesoros y de jugadores que hay en un momento dado dentro del mundo. También el mundo tiene un calendario de años, meses y días para representar el paso del tiempo.

- **Cell:** La celda es el elemento básico del plano del mundo. Una celda almacena la información sobre el tipo de entidades que se encuentran en su posición. Marca si hay una ciudad, algún equipo de jugadores, monstruo o tesoro. En caso de que haya un tesoro, marca el valor del tesoro.
- **City:** Una ciudad consiste de un par de coordenadas que dan su posición dentro del mundo. Tiene un nivel, una cantidad de riqueza, un rango de frontera y un número de habitantes. Estos cuatro atributos denotan su nivel de desarrollo. Además, cada ciudad tiene una lista de los equipos que añaden a su desarrollo cuándo salen a buscar tesoros.
- **Party:** Un equipo es un conjunto de personajes que pasean por el mundo en búsqueda de tesoros para llevar a su ciudad y de monstruos para subir de nivel. Cada equipo consta de distintos tipos de personajes almacenados en una lista. El equipo tiene un par de coordenadas para posicionarse en el mundo y una dirección que, al ser un autómatas celular, le sirve para conocer el siguiente “paso” que va a dar en el mundo.
- **Character:** El personaje es el actor atómico del sistema. Cada personaje tiene un nivel de ataque base, daño, salud máxima y actual y ritmo de crecimiento de ataque y salud al subir de nivel. Hay tres tipos de personajes:
  - **Fighters:** Son fuertes y tienen mucha salud al inicio del juego pero al avanzar de nivel su poder no incrementa mucho.
  - **Mage:** Inician como los personajes más débiles y con poca salud pero su incremento de poder es grande conforme van subiendo de nivel.
  - **Ranger:** Este personaje es el punto medio entre los dos anteriores. Inicia con salud y fuerza media y su ritmo de incremento es menor que el de un mago pero mayor que el de un luchador.

Los personajes pueden decidir si atacan a un monstruo o no dependiendo de su nivel y fuerza.

- **Monster:** Un monstruo deambula por el mundo buscando jugadores y ciudades para atacar y conforme pasa el tiempo va subiendo de nivel automáticamente. El monstruo lleva la cuenta de los personajes que ha matado y de los que necesita matar para subir de nivel. Además, un monstruo se mueve en el mundo más rápido que los personajes y puede acumular tesoros, el cual se reparte entre los personajes que lo derrotaron.

Este programa guarda en un archivo los datos de cada elemento de la simulación para después analizarlos. Los datos se guardan dentro de la carpeta `archivos`.

**Analizador de Datos.** Aprovechando las bibliotecas que Python tiene para análisis matemático (`numpy`) y de gráficas (`networkx`), creamos un programa que extrayera los datos generados por el simulador para recrear la gráfica y estudiarla.

**Resultados.**

**Conclusiones.**

**Bibliography**