

Complejidad y videojuegos, un acercamiento a los comportamientos sociales en los *Role Playing Games* (RPG) a través de la programación de los *Non Player Characters* NPC: una reflexión sobre la construcción de sus narrativas

Fernando H. Gallegos Borunda ⁽¹⁾

Leonardo A. Moreno Toledano ⁽²⁾

Erika A. Rogel Villalba ⁽³⁾

Resumen: La complejidad presente en nuestro entorno actual se ve usualmente reflejada en los productos culturales, entre ellos, los producidos por el diseño y los medios lúdicos, como lo son el cine y el videojuego, en este último es posible observar la transición de los sistemas sociales y su complejidad en juegos de mundo abierto, como el caso de *Second life* y en diversos juegos del tipo RPG (*Role Playing Game*), en los cuales se replican narrativas complejas propias de nuestro entorno social. La construcción de narrativas y formas de juego, realizadas a través de softwares generadores de historias que simulen con cierta veracidad la complejidad social, así como los modelos para lograrlo, son cada vez más complejos. Esto puede ejemplificarse a través de la figura del *Non playable character* (NPC), personaje que aparece comúnmente en un videojuego y que cumple cierta tarea de apoyo sobre la jugabilidad del juego, ya sea informando, interactuando, luchando o resolviendo alguna tarea junto al jugador. Una premisa actual, en este ejemplo, es que se busca que este personaje sea indistinguible de una persona normal, es decir, que, aun siendo una IA, cuente con la capacidad de interactuar y reaccionar en un momento determinado como un ser humano, herramientas como los árboles de comportamiento, sistemas que establecen diversas reacciones que puede llevar a cabo un NPC y los simuladores sociales, sistemas que tienen como objetivo el simular los comportamientos cuando interactúan y se relacionan las personas, aunado a las nuevas tecnologías, buscan generar una relación de apofenia entre el jugador y los NPC's, esto es, la generación de líneas de interpretación e interacción acordes a las ideas y pensamientos de las personas.

El presente ensayo, presenta una reflexión sobre la evolución de los procesos mediante los cuales los NPC's responden e interactúan con el jugador y los modelos generativos que se encuentran detrás de ello, la complejidad que supone la construcción de dichas narrativas, así como los alcances actuales y los posibles caminos a seguir al respecto.

Palabras clave: Complejidad - Videojuegos - Narrativas no lineales - RPG - NPC

[Resúmenes en inglés y portugués en las páginas 133-134]

⁽¹⁾ **Fernando Humberto Gallegos Borunda.** Licenciado en Diseño de Medios Interactivos, actualmente se encuentra realizando sus estudios como becario CONACHyT en la Maestría en Estudios y Procesos Creativos de la Universidad Autónoma de Ciudad Juárez. al239644@alumnos.uacj.mx ORCID: 0009-0009-0027-8698

⁽²⁾ **Leonardo Andrés Moreno Toledano.** Coordinador de la Maestría en Estudios y Procesos Creativos en Arte y Diseño de la Universidad Autónoma de Ciudad Juárez. Miembro fundador del Cuerpo Académico: Diseño, Usuario y Entorno. Miembro del SIN nivel 1. Co-responsable del proyecto Gender Design STEAM en la Universidad Carleton Canadá. lemoreno@uacj.mx ORCID: 0000-0002-9447-6362

⁽³⁾ **Erika Anastasia Rogel Villalba.** Profesora por la Universidad Autónoma de Ciudad Juárez (UACJ). Líder del Cuerpo Académico: Diseño, Usuario y Entorno. Miembro del Sistema Nacional de Investigadores SNI nivel 1. Responsable del proyecto Gender Design STEAM en la Universidad Carleton Canadá. Cuerpo Académico: Diseño, Usuario y Entorno. erogel@uacj.mx ORCID: 0000-0002-0219-0173

Introducción

La complejidad presente en nuestro entorno actual se ve usualmente reflejada en los productos culturales, entre ellos, los producidos por el diseño (Thakara, J. 2010; Morales A., 2017), así como los medios lúdicos, como lo son el cine y los videojuegos, en estos últimos es posible observar la transición de los sistemas sociales y su complejidad al menos desde dos perspectivas, por una parte, en juegos de mundo abierto, como el caso de *Second Life*, un espacio tridimensional creado por Philip Rosedale en 2003, en los que los jugadores (personas) participan en entornos digitales creando su entorno, sus identidades a través de avatares, y con ello, las relaciones que en él se desarrollan. Por otra parte, en diversos juegos del tipo *Role Playing Game* (RPG), en los cuales se replican narrativas complejas propias de nuestro entorno social que son simuladas en gran medida por lo que conocemos como *Non Playable Character* (NPC), personajes digitales que interactúan con los jugadores, en este último caso, es imprescindible la construcción de una narrativa que guíe el juego y, al mismo tiempo, haga sentir al jugador que se encuentra interactuando con un mundo lo más real posible.

En la primera década del siglo XXI, Steven Poole sostenía que la vida interior de los videojuegos se encontraba ligada a la vida interior del jugador mediante una experiencia estética. Examinaba cómo se construían mundos, se explicaban historias y cómo los personajes se convertían en ídolos. Poole se preguntaba: “¿Qué tan precisamente podía el videojuego aspirar a recrear algo del mundo real y cómo otros tipos de juego, como aquellos que presentan mundos fantásticos, podían persuadirnos de que son de alguna manera reales?” (2007, p. 84).

Como podemos observar, la recreación del mundo real a través de los videojuegos no es un asunto novedoso; muchos juegos, como *Shen Mue* (1999), *Grand Theft Auto* (1997) o *F.E.A.R.*, (2005), lo han hecho en mayor o menor medida al recrear la cultura material, la estructura urbana y los objetos de uso cotidiano, así como conceptos sociales como la guerra, la delincuencia y las estructuras laborales y de clase. Sin embargo, lo que aquí nos interesa es la complejidad que subyace detrás de la recreación de los mundos creados actualmente y la manera en que los desarrolladores trabajan para persuadirnos de que estos mundos y las relaciones sociales generadas entre los diversos actores digitales que lo habitan son, de alguna manera, reales.

Para lograr lo anterior, uno de los elementos que juega un papel imprescindible es la narrativa, Jakub Majewski nos explica que la narrativa es una historia que consiste en una serie de eventos y que la narración es el acto mediante el cual se lleva esta historia a la audiencia (2003, p. 3). Para ejemplificar lo anterior, Majewski compara los juegos *Tetris* (1985) y *Space Invaders* (1977), explica que mientras el primero es no narrativo porque no busca contar una historia, el segundo es algo más complejo ya que sugiere que una invasión se está llevando a cabo y, por lo tanto, existe una causa implícita entre la lucha del jugador contra los alienígenas.

Por otra parte, Majewski describe la estructura básica presente en los videojuegos narrativos, a través de tres modelos:

- **Modelo Spring of pearls:** el jugador avanza a través de varios eventos, y aunque en algunos casos parezca que tiene libertad, al final sólo avanzará al siguiente evento, *Shenmue* (2000) y *Legend of Zelda Skyward Sword* (2011).
- **Modelo Branching storyline:** en este modelo el jugador puede generar cambios en la historia, al seleccionar entre varias opciones que se le presentan en momentos estratégicos de la narración, *Fable* (2004, 2008 y 2010) y *Call of Duty-Black Ops II* (2012).
- **Modelo Amusement park:** el jugador posee cierta libertad en la historia para moverse de un lugar a otro, y lo que haga en algún momento afectará la manera en que se desarrolle el juego, *Massive Multiplayer Online Roleplaying Games* (mmorpg), *Final Fantasy XIII* (2009) y *The Elder Scrolls Skyrim* (2011).

Lo anterior nos permite establecer que existen diferentes formas de contar historias, mismas que van desde narrativas lineales a narrativas complejas no lineales, estas nuevas expresiones de videojuegos se centran en lo que se conoce como la capacidad de agencia para los jugadores.

Por agencia, nos referimos a la capacidad de influenciar el desarrollo de las propias historias que se desarrollan en un espacio determinado. Por ejemplo, la existencia de una historia de un mundo de zombis puede mostrar las ideas del propio programador al integrar elementos que se corresponden con lo que él considera como características e interacciones fundamentales del zombi (Adams, 2019, p. 4). Así entonces, visto desde una perspectiva rizomática, un agenciamiento es precisamente ese aumento de dimensiones en una multiplicidad de acciones que cambian necesariamente de naturaleza a medida que aumentan sus conexiones, en un rizoma no hay puntos o posiciones, como ocurre en una estructura, un árbol o una raíz (Deleuze et al., 2002, p. 14).

Empero, si bien los videojuegos se han vuelto cada vez más profundos y complejos, el fin de cualquier juego es que el jugador pueda disfrutar de una experiencia lúdica sin sufrir dificultades en el desarrollo de la historia y las acciones que se desarrollan durante una partida, empero, lo anterior, como hemos descrito, no implica una reducción de la complejidad; la jugabilidad y la narrativa de un juego como *Space invaders* no tiene comparación con juegos como *Elder Scrolls V: Skyrim* (2011), si bien, la complejidad existente en los videojuegos tiende a ser proporcionalmente mayor conforme avanzan las tecnologías digitales, quien tiende a lidiar con ella no es el jugador, sino los desarrolladores, lo anterior puede entenderse a través de lo que conocemos como Ley de Tesler, también conocida como ley de la conservación de la complejidad, es un principio fundamental en el desarrollo de experiencias de usuario, plantea eliminar o reducir al mínimo las barreras cognitivas y los pasos innecesarios que puedan dificultar la realización de una tarea a través de la eliminación de funciones redundantes, agrupando lógicamente elementos relacionados y adoptando un enfoque centrado en el usuario. Al simplificar la experiencia del usuario, se mejora la eficiencia, se reducen los errores y se aumenta la satisfacción del usuario. Lo anterior no implica, como mencionamos, la reducción de la complejidad en sí, pero plantea que, en el diseño de interfases, entre ellas, las que conciernen a los videojuegos, es el diseñador quien tiene que trabajar con la complejidad del sistema, simplificando las tareas que el usuario debe entender y realizar.

Así entonces, para lograr una buena jugabilidad y un fluido desarrollo de la historia, el NPC debe reflejar una serie de capacidades específicas, entre ellas y la más importante, es la comunidad con el jugador.

Community feels good. It feels like belonging, fitting in, and actively caring about something together. Community typically arises when a group of people who have a common interest start to interact with each other to further that interest. It requires positive participation from everyone in the group. (McGonigal, 2011, p. 191)

In the case of virtual characters, artificial emotions are labels for states that may not exactly replicate human feelings but are intended to initiate behavior that we would expect from someone in that state ([Picard, 1997] en Shirvani et al., 2023, p. 24).

Esto determina al agente como un elemento capaz de simular una personalidad y un ambiente social, pero no significa que sea un patrón común observado en los videojuegos. Algunos NPC, actúan únicamente ofreciendo cierta información al jugador con el fin de darle continuidad a la historia, otros NPC como es el caso de *The Sims 4* se acercan más a un agente inteligente, esto debido a su libertad y agencia para poder reaccionar a los eventos, necesidades y relaciones con otros Sims dentro del videojuego.

Para lograr una adecuada interacción con el jugador, es necesario entender que dificultades enfrenta un agente inteligente para funcionar dentro del videojuego. Una de éstas es la capacidad de reaccionar rápidamente a los cambios que se presentan en el ambiente del juego, lo que requiere del desarrollo de técnicas de aprendizaje y memoria por parte del NPC (Butt et al., 2023, p. 4). Lo anterior nos indica la necesidad de un sistema rápido

y eficiente para resolver problemas y dar la ilusión de inteligencia, “*Illusion of life requires agents to appear to have goals and be able to pursue them at the same time*” (Shirvani et al., 2023, p. 25). En este sentido, el jugador y el agente inteligente se someten a los mismos tipos de reglas del juego que se pueden establecer para crear o generar una conexión o interacción al tener metas, objetivos o reglas similares.

Del NPC al Agente Inteligente

Lo explicado hasta aquí ha dado paso al desarrollo y modelación de diversas formas de programación para lograr una mejor simulación del comportamiento de los agentes inteligentes (NPC). Para poder lograr este cometido, el desarrollador debe pensar primero en las formas de modelizar los comportamientos del agente inteligente, como describe Nigel y Klaus en *What Is Simulation?* como una serie de comportamientos simplificados para ejecutar algún tipo de teoría social (2005, pp. 2-3), esto permite generar nuevos ambientes y alterar la forma en que se percibe la historia, considerando la teoría social como la narración.

Aun así, para Johnson-Bey et al., este campo se encuentra rezagado por la distancia que existe entre los trabajos académicos y la implementación de éstos como herramienta (2022, p.2). Sobre la simulación de comportamientos sociales, Johnson-Bey propone la idea de inspirarse en los Motores Físicos, herramientas que facilitan la utilización de elementos para la simulación de físicas (2022, p. 4). En sí, lo que plantea Johnson-Bey es el tomar múltiples patrones e ideas desarrolladas por diferentes proyectos académicos y utilizarlas en la construcción de un motor social propio, el cual permita facilitar a los desarrolladores *indie* generar y explorar nuevas formas de narrar historias en los videojuegos. Para lograr lo anterior se han utilizado distintos patrones, como los Árboles de Comportamiento, Maquinas de Estados, *Goal Oriented Action Planing* (GOAP) y Generadores de Contenido Procedural.

El primero de ellos, los Árboles de Comportamiento, son un patrón popularizado a partir de su uso en el juego Halo 2 (2004) desarrollado por Microsoft. Estos fueron implementados principalmente para el desarrollo cooperativo entre el escritor y los programadores, usando como punto principal la capacidad de tener una interfaz visual que permita convertir la visualización de nodos en comportamientos fáciles de identificar, además de integrar herramientas de depuración para observar de forma más cercana el comportamiento de los agentes. Funciona creando un nodo base el cual tiene nodos hijos, estos se leen de izquierda a derecha y comprenden acciones básicas, como moverse a algún lugar o interactuar con objetos. Incluyendo nodos de decisión los cuales ayudan a escoger otras ramas de comportamiento o decoradores para tener más ramas de comportamiento, similares a una rama de árbol, y un Blackboard, herramienta que almacena información de interés para el agente inteligente de tipo local, cosas que a él le afectan, y de tipo global, cosas que suceden en su mundo. Esta herramienta se usa para modelar comportamientos, probar, depurar y hacer que actúen acorde a las ideas del escritor. Por ejemplo, en Halo 2 cada

enemigo tiene un comportamiento, dependiente de su raza y su rango militar, además de tener comportamientos específicos dentro del combate cómo reaccionar ante la muerte de sus compañeros, denostando así el origen y formas de organización de los alienígenas, enemigos en Halo 2.

Por su parte, las Máquinas de Estados son patrón mucho más sencillo de desarrollar y con mayor tiempo entre los programadores. Al ser un patrón común para diseñar programas, este suele estar centrado en facilitar el control y los posibles comportamientos de un sistema.

Visto por primera vez en el éxito comercial, Half-Life (1998) juego desarrollado por Valve Corporation, en él se integra este patrón para dotar de mayor reactividad a los agentes, desde cucarachas que reaccionan a condiciones de luz, hasta enemigos que tienen la capacidad de pegarse a planes para poder atacar al jugador con jerarquía de ataque. Este patrón, aun cuando es sencillo de aplicar, se realiza a través de un mayor nivel de programación, cada una de las instrucciones está escrita en un pequeño script, y los estados son establecidos que script se está ejecutando en cada momento, cada posible estado está conectado a otros estados como los nodos de la red de una araña.

Este patrón funcionó dentro de Half-Life para crear enemigos capaces de elaborar una serie de acciones más complejas, como buscar refugio y poder diferenciar enemigos para tener un orden o jerarquía de ataque, incluyendo la capacidad de tener comportamientos centrados en el escenario. Aunque sus beneficios suenan cercanos a los descritos para el árbol de Comportamiento, debe tenerse en cuenta lo innovador que fue presentar estos comportamientos, cuando la mayoría de los enemigos desarrollados exhibían solo uno o dos comportamientos diferentes.

Otro de los planificadores de acciones orientados a objetivos es el *Goal Oriented Action Planner* (GOAP), los cuales se utilizaron por primera vez en el juego F.E.A.R. (2005), desarrollado por Monolith Productions. Se implementó dejando a un agente inteligente que actuaba como director de la escena, los agentes inteligentes esperan por un plan y el director lo desarrolla mediante una ponderación de sus heurísticas.

Este patrón está diseñado para tener una serie de objetivos específicos, como atacar al jugador, reparar un auto o repartir algo. Para cumplir estos objetivos se diseñan acciones con precondiciones y efectos. Después un sistema apila las acciones para realizar cadenas de acciones y poder realizar el objetivo considerando el estado actual del personaje. Para escoger la ruta de acciones más apta se usa una heurística definida por el narrador, que consta de reglas para establecer cual plan, serie de acciones, es más deseable para la narración o la situación. Por ejemplo, si para atacar al jugador tenemos disponibles tres planes, cada uno de ellos recibirá una puntuación en la cantidad de tiempo que tome realizar cada acción y la cantidad de puntos de acción de costo por cada acción, después se selecciona el plan más barato y este le es asignado al agente.

Este ayuda a los narradores en la capacidad de generar acciones más inteligentes y desplegar acciones más complejas. Gracias a esto, se puede crear una falsa sensación de coordinación por la variabilidad de planes entre los agentes, lo cual puede dar como resultado agentes conscientes del entorno, así como del estado y ubicación de sus compañeros. Aunque esto permitió integrar más comportamientos fáciles para el narrador, significa que cada posible acción debe estar programada y un sistema debe poder apilarlas para buscar

rutas y después calcular costos a cada ruta. Aun así, por medio de la heurística, se pueden declarar objetivos o estados específicos para poder construir agentes más atentos al medio ambiente que los rodea y diversidad en cada acción.

Por último, tenemos un caso especial. Aun cuando nos gustaría llamar a los generadores de narrativa procedurales como un patrón de diseño, su uso se puede observar como un término paraguas, el cual puede tomar prestadas múltiples técnicas o patrones de diseño para realizar las activades deseadas o realizar las acciones deseadas. En esta línea tenemos trabajos como *The Ensemble Engine: Next Generation* (Sammuel et al., 2015) el cual busca diseñar un sistema centrado en la generación de inferencias para la construcción de respuestas válidas en las situaciones que se desarrollan, usando como motor principal la relación entre dos agentes como base para la modelización de los comportamientos. Además de incluir el trabajo del narrador Michael Cook, quien diseñó un sistema para simular una sociedad ya desaparecida por medio de la simulación de sus personas, sus relaciones a grandes rasgos y sus elementos culturales, como su arquitectura y el uso cotidianos de sus objetos. La construcción de este tipo de escenario permitía a un jugador explorar y generar sus propias ideas sobre la historia de la zona a partir de las pistas abandonadas en ese escenario (2020).

Aun cuando el desarrollo de estas herramientas se vean como únicas para cada uno de los casos del narrador, se empieza a vislumbrar el uso de patrones ya establecidos para modificarlos y permitir adaptar estos a las necesidades propias de los narradores y sus propuestas narrativas. Como se puede observar en *Moody 5* (Garavaglia et al., 2022) una aplicación que sirve para la generación, interacción y construcción de narrativas a través de la simulación de las emociones de los agentes inteligentes. Para ello, se hace uso de un Goal-Oriented Action Planner (GOAP) modificado para hacerlo consciente de las emociones y usar esto como heurística para la selección de tareas. En pocas palabras, toma en cuenta las emociones y sus personalidades para crear planes, los cuales buscan reflejar las emociones y la personalidad de los personajes. Este grupo de patrones permite mayor libertad, ya que representa la necesidad de ejecutar elementos narrativos obtenidos a partir de cambiar el estado de ánimo de los agentes inteligentes para desbloquear misiones y poder avanzar en la trama.

Aun cuando, como explicamos, existen una serie de herramientas para la generación de historias o narrativas a través de los agentes inteligentes, entendiendo estas historias como narrativas generadas por computadora (Picucci, 2014, pp.111-113), éstas corren por medio de la base del ya mencionado patrón GOAP, modificado para tomar en cuenta múltiples tipos de heurísticas, formas de selección de acciones, las cuales son asignadas a los personajes. Esto lo podemos comprobar en cuanto a los avances del investigador de Square Enix, Boeda, quien señala como es que los personajes (NPC) generan un proceso para indicar cuando no son capaces de resolver un problema, por lo que inician un proceso de comunicación para desarrollar un plan para cada uno de ellos, logrando cooperar para resolver sus problemas (2021).

Es importante mencionar que los temas técnicos o de patrones para la modelización de los simuladores sociales no vienen desarrollados como herramientas abstractas, sino como una serie de implementaciones específicas para cada videojuego. El principal problema de esto es la agencia narrativa del académico (el diseñador del patrón hacia su interés

narrativo) se encuentra establecida en relación con su trabajo. Como apunta Johnson-Bey et al. la utilización de herramientas ya disponibles imposibilita al desarrollador a crear sus propios modelos de comportamiento, estructuras y herramientas en beneficio de sus propias agencias narrativas, y la dificultad de crear sus propias herramientas crea una barrera para los desarrolladores para el desarrollo de software, en el campo indie (2022). Este limitado acceso ha provocado un estancamiento en la generación de sistemas narrativos generativos y su exploración dentro del área.

Como hemos mencionado, existen técnicas para generar y construir comportamientos más avanzados o específicos para cada agente. En el caso de los árboles de comportamiento, como describe Francis, se pueden emplear cadenas de texto sintácticas convertidas en expresiones de árboles de comportamiento para medir la capacidad de resolver diferentes problemas, además de poder ser mutable y realizar interacciones sobre estas para su selección e integración (2017, pp. 11-14). Lo anterior no significa que sea la única forma de aprovechar este mecanismo, como por ejemplo los modificadores Guimarães et al. especifican utilizar un sistema que le permite traducir el sistema de escritura de las misiones de *The Elder Scrolls V: Skyrim* (2011) para asignarles nuevas actividades a los NPC cercanos al jugador, mediante esto, no solo permite integrar comportamientos específicos, sino también agregar rasgos de personalidad y dificultades de diferentes tipos para el desarrollo de los agentes. Esto lo logra por medio de la transcripción de elementos de script a un texto entendible para Chat GPT y generar una respuesta acorde a la descripción del personaje y sus estados como sed o alegría (Guimarães et al., 2023).

Aunque podemos considerar usar estas técnicas como un punto importante, es necesario explicar cómo funciona un elemento crucial para sostener una idea de coherencia dentro del propio videojuego. Para esto, se usa un sistema que permita el acto de recordar los sucesos de alguna forma, la principal es la conservación del estado del agente por una serie de variables las cuales representan de forma abstracta su pasado al modificarse durante el desarrollo. Como ejemplos clásicos podemos mencionar el uso de medidores que recurren a un rango de 0 a 100 para indicar el nivel de salud, sed, hambre y las relaciones con otros personajes durante el desarrollo del juego. Aunque estas variables son comunes, también deben almacenar fragmentos de conocimientos específicos, como almacenar sucesos importantes dentro de la historia. Aun cuando sea algo común en el desarrollo de agentes inteligentes, no existe una forma específica para ello, sino una serie de libertades, desde variables específicas que funcionan como banderas, hasta solo pequeños fragmentos de relación descritos como frases básicas.

Con lo visto hasta aquí podemos observar las diversas propuestas para la implementación de las herramientas necesarias para el desarrollo de un Motor de Físicas Sociales. Para tener un mejor campo de visión, hay que considerar el perfil de acceso de los desarrolladores, que pueden usar estas herramientas y aplicarlas a su propio espacio de desarrollo. Facilitar y tener una comunicación cercana con los desarrolladores nos puede permitir entender las necesidades, las herramientas y como aprovecharlas dentro del propio espacio de desarrollo para los videojuegos. No por ello, podemos evitar una disección a profundidad de las ventajas y desventajas de cada una de las herramientas aquí expuestas.

Técnica	Implementación	Emergencia	Diseño de comportamiento	Facilidad de diseño de comportamientos	Facilidad de desarrollo
Máquinas de estados	Su implementación pasa por abstraer las tareas en actividades cortas y éstas en verbos sencillos, después se crea una red de nodos fácil de describir en código	La red de nodos desarrollada solo es capaz de operar a nivel agente, sin tener la capacidad de reaccionar ante los demás personajes o propuestas nuevas.	El comportamiento pasa por la abstracción de sus acciones a acciones atómicas fácilmente recurribles y capaces de representarse, después se pasan a sistemas de validación de estos comportamientos	Es sumamente fácil integrar pequeños sistemas, difícil de escalar y se requiere de una enorme habilidad de programación para agregar o quitar cosas.	Para los programadores, es relativamente sencillo de integrar, difícil de mantener más de 100 acciones con sus respectivas jerarquías. Y sumamente complejas para desarrolladores poco experimentados que no tengan nociones de programación.
Árboles de Comportamiento	Ligeramente más compleja por tener que traducir de un editor de nodos a un sistema capaz de representar a nivel de programación el comportamiento	Idéntico a una máquina d estados, pero mucho más rígido para integrar nuevos comportamientos.	El comportamiento se describe como una serie de pasos que se pueden leer de izquierda a derecha. Toda posible interacción es una desviación de la lectura.	El comportamiento se diseña fácil requiere de la ayuda de un Blackboard, espacio donde almacena elementos de interés, para poder estar al tanto de lo que sucede alrededor	El sistema se vuelve ligeramente más complejo para los desarrolladores a implementar, ahora las personas sin conocimiento pueden integrarse para el desarrollo y creación de agentes inteligentes.
GOAP	Se debe realizar un algoritmo de búsqueda de nodos, programar un sistema el cual almacene las diferentes tareas a realizar y por último una con la capacidad de seleccionar planes.	Al ser un sistema diseñado alrededor de tareas y posibles acciones, las acciones pueden alterar la forma de obtener el resultado y, por tanto, resulta más fácil obtener resultados inesperados integrando nuevas tareas o acciones.	El comportamiento de retira como serie de acciones específicas, pasa a ser una serie de acciones construidas a demanda y, por tanto, mutable sin problema. Pero sin límites, estas acciones pueden ser replicadas por todos los agentes inteligentes.	El diseño ahora pasa por los algoritmos de selección más que las acciones mismas, al ser estos quienes escogen las acciones a realizar.	Complejo de llevar a cabo sin una noción sólida de estructuras de datos además de tener una idea clara de crear las acciones que deberán realizar los agentes inteligentes.
Procedural Content	Implementación libre para el desarrollador, no tiene restricciones de algún tipo.	Al ser sistemas a discreción del desarrollador, estos solo responden a las herramientas e ideas traídas por el desarrollador.	El diseño pasa a ser libre y, por tanto, requiere la habilidad del desarrollador para transferir sus ideas de sistemas sociales a código fácil de usar para él.	El diseño debe pasar por las ideas y capacidades del desarrollador para lograr transferir los comportamientos al agente social.	Herramientas fáciles de implementar o difíciles. Requiere afinidad del desarrollador y conocimientos de patrones y formas de combinación.

Tabla 1. Cuadro comparativo entre las técnicas y su desarrollo e implementación frente al desarrollador (Elaboración propia).

Técnica	Personalidad	Cultura	Relaciones sociales	Cumplimiento de objetivos
Máquinas de Estados	La personalidad se desarrolla en el marco de la cantidad de reacciones que pueden introducirse a nivel de programación.	A nivel cultura, el agente debe recibir instrucciones para resolver conforme las reglas determinadas para simular la sociedad.	Las relaciones sociales funcionan como indicadores, sus posibles reacciones están limitadas al contexto representativo de dicha variable.	Los objetivos son estáticos y específicos, se requiere un cambio en las máquinas de estados para realizar un cambio de objetivos y el cómo cumplirlos.
Arboles de comportamiento	Existe más espacio para escribir distintos árboles, pero sus sistemas son más limitados al evitar la complejidad de programación.	La cultura se puede representar como una serie de diferentes pasos para representar las posibles interacciones, requiere de una asistencia de una IA directora para resolver posibles conflictos.	Las relaciones sociales se limitan por el nivel de escritura de árboles de comportamiento, sus posibles ramas y direcciones. EL contexto puede ser ofrecido por variables de apoyo para representar el estado de estas relaciones.	Los objetivos son intrínsecos a las lecturas del árbol, no puede cambiar, agregar o plantear un nuevo objetivo si no está soportado por el árbol de comportamiento.
GOAP	La personalidad se rige con base a los algoritmos de selección y la capacidad de acceder a acciones de diferentes tipos para cumplir sus objetivos, además se pueden indicar diferentes tipos de variables de apoyo para representar diferentes tipos de acciones.	El sistema es capaz de plantear múltiples operaciones para resolver o mantener una regla, puede realizar una ruptura si el objetivo es más importante que la cultura a característica del agente inteligente.	Las relaciones sociales están expresadas en función a la cantidad de tareas que se pueden desarrollar para resolver una determinada acción, una variable de estado de relación y posibles listas dependientes del agente a relacionar son necesarias para establecer las acciones a performar.	Los objetivos deben ser mutuados por un algoritmo exterior, pero la capacidad de resolverlos es múltiple y necesita de un procedimiento para seleccionar el plan más deseable. Esta selección puede ser determinada por las mismas variables de apoyo para definir su personalidad.

Tabla 2. Cuadro comparativo entre las técnicas y las agencias de los agentes inteligentes y sus posibles implementaciones (Elaboración propia)

En el último cuadro comparativo se omitió la técnica de Procedural Content para evitar conflictos, ya que usa una variedad y pluralidad de técnicas donde cada una de éstas pueden compararse como técnicas o patrones de diseño específicos. Es por esto, que la complejidad de estas técnicas que construyen sus propios patrones de diseño provoca dificultades para su análisis como solo un patrón de diseño capaz de englobar diferentes mecanismos.

Según lo discutido hasta ahora, se puede observar la singularidad de cada patrón para tratar los operadores de símbolos, como referencia al estado del mundo y la situación del agente inteligente. Estos operadores tienen la capacidad de representar solo los comportamientos especificados por el narrador y un espacio de libertad, para ejemplificar

los comportamientos, dejando así la agencia narrativa del narrador dentro de la propia obra. Esta capacidad permite establecer fórmulas que permiten alterar y contar historias diferentes cada que se inicia una partida, permitiendo así, una mayor libertad del jugador para cualquier situación que surja. Para lograr lo anterior, el procesador simbólico y el desarrollador se pueden valer de algoritmos diferentes como el *Wave Function Collapse* (Gumin, 2016) el cual utiliza otro método para completar las acciones dejando más espacio al narrador. Además, la intención de construir herramientas para el desarrollo a conducido al desarrollo del *framework FANTASIA* (Origlia et al., 2019) el cual se destaca por usar una base de datos orientada a grafos como forma de almacenar la relación entre los agentes inteligentes para después contextualizar las respuestas y diálogos narrados dentro de la experiencia virtual.

Conclusión

Con lo planteado, podemos encontrar que cada herramienta presentada se centra en perfiles específicos, que permiten desarrollar y modelar comportamientos específicos en cada uno de los agentes inteligentes (NPC) que actúan en un videojuego. Esto implica la necesidad de herramientas más avanzadas capaces de operarse como procesadores simbólicos en sí mismos, para alterar y modificar su comportamiento, según se enfrente a situaciones específicas. Para lograr esto, se plantea la existencia de un patrón central que permita la integración de otros patrones individuales para poder ofrecer soluciones y con ello, mayores libertades narrativas. Siendo esta herramienta capaz de operar integrando distintas fuentes de información para evaluar los conocimientos de cada agente y sus relaciones sociales. Así como de establecer rutas y comunicarlas para asignarlas en forma de cooperación, además este patrón debe soportar la capacidad de alterar sus formas de selección considerando el estado emocional del agente para potenciar ciertas acciones y, si es necesario, entregar planes, misiones al jugador, como resultado de una clase de acciones, que permiten un sentido de cooperación y relación entre el jugador y el agente inteligente. Aunque herramientas como Chat GPT permiten el uso de un lenguaje capaz de crear misiones para cada agente según su descripción, el GOAP permite generar sistemas conscientes de sus emociones y alterar las acciones acorde a estas, esta máquina imaginaria debería permitir su uso con tecnologías emergentes para desarrollar nuevos tipos de procesadores y analizadores de símbolos (acciones) acorde a descripciones del mismo tipo y descripciones del medio donde vive para crear respuestas coherentes en el universo planteado por el narrador.

Esto nos permite delimitar las capacidades del sistema, como algún tipo de acción primitiva expandible para la traslación a otros patrones y, mediante el uso de esta primitiva, poder indicar sistemas más complejos para el desarrollo de éstos, otros sistemas o compartir información entre los distintos tipos de acciones.

Aun así, aunque el sistema pueda operar a este nivel, faltaría considerar la cantidad de recursos: espacio, programación y procesamiento, necesarios para implementar en cada personaje un sistema propio. Cabe señalar, que un sistema cuyo rol sea la gestión de otros

agentes inteligentes deberá hacerlo solo en el aspecto de recursos y como sistema para indicar el nivel de carga de detalle de los comportamientos. Así se evita el problema de tener un sistema que funcione como director de orquesta de los demás agentes inteligentes y, por tanto, necesite incorporar sus propios comportamientos y termine con las implicaciones de un *Deus Ex Machina* y los problemas que suele implicar dentro de las narraciones.

Referencias

- Adams, T. X. S., Tarn (Ed.). (2019). *Procedural Storytelling in Game Design*. A K Peters/CRC Press. <https://doi.org/10.1201/9780429488337>
- Boeda, G. (2021). Multi-Agent Cooperation in Games with Goal Oriented Action Planner: Use Case in WONDER Prototype Project. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 17(1), 204–207. <https://doi.org/10.1609/aiide.v17i1.18909>
- Butt, N. A., Mahmood, Z., Rehman, G. U., Nasralla, M. M., Zubair, M., Farman, H., & Khattak, S. B. A. (2023). The Development of Intelligent Agents: A Case-Based Reasoning Approach to Achieve Human-Like Peculiarities via Playback of Human Traces. *IEEE Access*, 11, 78693–78712. <https://doi.org/10.1109/ACCESS.2023.3274740>
- Cook, M. (2020). *Generative Forensics: Procedural Generation and Information Games* (arXiv:2004.01768). arXiv. <https://doi.org/10.48550/arXiv.2004.01768>
- Deleuze, G., Guattari, F., & Deleuze, G. (2002). *Mil mesetas: Capitalismo y esquizofrenia* (5. ed). Pre-Textos.
- Garavaglia, F., Nobre, R. A., Ripamonti, L. A., Maggiorini, D., & Gadia, D. (2022). Moody5: Personality-biased agents to enhance interactive storytelling in video games. *2022 IEEE Conference on Games (CoG)*, 175–182. <https://doi.org/10.1109/CoG51982.2022.9893689>
- Guimarães, M., Santos, P. A., & Jhala, A. (2023). *Emergent social NPC interactions in the Social NPCs Skyrim mod and beyond* (arXiv:2207.13398). arXiv. <https://doi.org/10.48550/arXiv.2207.13398>
- Gumin, M. (2016). *Wave Function Collapse Algorithm* (1.0) [C#]. <https://github.com/mxgmn/WaveFunctionCollapse> (Obra original publicada en 2016)
- Jager, W., Abramczuk, K., Komendant-Brodowska, A., Baczk-Dombi, A., Fecher, B., Sokolowska, N., & Spits, T. (2020). Looking into the Educational Mirror: Why Computation Is Hardly Being Taught in the Social Sciences, and What to Do About It. En H. Verhagen, M. Borit, G. Bravo, & N. Wijermans (Eds.), *Advances in Social Simulation* (pp. 239–245). Springer International Publishing. https://doi.org/10.1007/978-3-030-34127-5_22
- Johnson-Bey, S., Nelson, M. J., & Mateas, M. (2022a). Exploring the Design Space of Social Physics Engines in Games. En M. Vosmeer & L. Holloway-Attaway (Eds.), *Interactive Storytelling* (Vol. 13762, pp. 559–576). Springer International Publishing. https://doi.org/10.1007/978-3-031-22298-6_36
- Majewski, J. (2003). *Theorising Videogame Narrative*. Australia: Bond University.
- McGonigal, J. (2011). *Reality is broken: Why games make us better and how they can change the world* (Ed. with a new, 2. appendix). Penguin Press.

- Morales, A.; Cabrera, V. (2017). Debate teórico-metodológico sobre diseño gráfico: de la linealidad a la complejidad. *Intersticios sociales*, (13) Recuperado en 01 de abril de 2024, de http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S2007-49642017000100002&lng=es&tlng=es
- Nigel, G., & Klaus, T. (2005). *Simulation For The Social Scientist*. McGraw-Hill Education (UK).
- Origlia, A., Cutugno, F., Rodà, A., Cosi, P., & Zmarich, C. (2019). FANTASIA: A framework for advanced natural tools and applications in social, interactive approaches. *Multimedia Tools and Applications*, 78(10), 13613–13648. <https://doi.org/10.1007/s11042-019-7362-5>
- Picucci, M. (2014). *When Video Games Tell Stories: A Model of Video Game Narrative Architectures*. <https://gredos.usal.es/handle/10366/124953>
- Poole, S. (2007). *Trigger Happy, Videogames and the Entertainment Revolution*. USA: Self Published.
- Samuel, B., Reed, A. A., Maddaloni, P., Mateas, M., & Wardrip-Fruin, N. (2015). *The Scored Rule Engine: Next-Generation Social Physics*. International Conference on Foundations of Digital Games. <https://www.semanticscholar.org/paper/The-Scored-Rule-Engine%3A-Next-Generation-Social-Samuel-Reed/94b7e4a02b33740d8d589ec726c6dd28219c65f7>
- Shirvani, A., Ware, S. G., & Baker, L. J. (2023). Personality and Emotion in Strong-Story Narrative Planning. *IEEE Transactions on Games*, 15(4), 669–682. <https://doi.org/10.1109/TG.2022.3227220>
- Starke, S., Zhang, H., Komura, T., & Saito, J. (2019). Neural state machine for character-scene interactions. *ACM Transactions on Graphics*, 38(6), 209:1-209:14. <https://doi.org/10.1145/3355089.3356505>
- Thakara, J. (2010). *Diseñando para un mundo complejo. Acciones para lograr la sustentabilidad*. México: Editorial Designio.

Abstract: The complexity present in our current environment is usually reflected in cultural products, including those produced by design and recreational media, such as cinema and video games, in the latter it is possible to observe the transition of social systems and its complexity in open world games, such as *Second Life* and in various RPG (Role Playing Game) type games, in which complex narratives typical of our social environment are replicated. The construction of narratives and game forms, carried out through story-generating software that simulates social complexity with a certain veracity, as well as the models to achieve it, are increasingly complex. This can be exemplified through the figure of the Non playable character (NPC), a character that commonly appears in a video game and who fulfills a certain task of supporting the gameplay of the game, whether informing, interacting, fighting or solving a task with the player. A current premise, in this example, is that this character is intended to be indistinguishable from a normal person, that is, that, even though it is an AI, it has the ability to interact and react at a given moment, like a human being, tools such as behavior trees, systems that establish various reactions that

an NPC can carry out and social simulators, systems that aim to simulate behaviors when people interact and relate, together with new technologies, seek to generate a relationship of apophenia between the player and the NPC's, that is, the generation of lines of interpretation and interaction according to people's ideas and thoughts.

This essay presents a reflection on the evolution of the processes through which NPCs respond and interact with the player and the generative models that are behind it, the complexity involved in the construction of said narratives, as well as the current scope and the possible paths to follow in this regard.

Keywords: Complexity - Video games - Non-linear narratives - RPG - NPC

Resumo: A complexidade presente no nosso ambiente atual geralmente se reflete nos produtos culturais, incluindo aqueles produzidos pelo design e pelas mídias recreativas, como o cinema e os videogames, nestes últimos é possível observar a transição dos sistemas sociais e sua complexidade em mundo aberto. jogos, como Second Life e em diversos jogos do tipo RPG (Role Playing Game), nos quais se replicam narrativas complexas típicas do nosso meio social. A construção de narrativas e formas de jogo, realizadas através de softwares geradores de histórias que simulam a complexidade social com certa veracidade, bem como os modelos para alcançá-la, são cada vez mais complexos. Isso pode ser exemplificado através da figura do personagem não jogável (NPC), personagem que comumente aparece em um videogame e que cumpre uma determinada tarefa de apoiar a jogabilidade do jogo, seja informando, interagindo, lutando ou resolvendo uma tarefa com o jogador.. Uma premissa atual, neste exemplo, é que este personagem pretende ser indistinguível de uma pessoa normal, ou seja, que, mesmo sendo uma IA, tem a capacidade de interagir e reagir em um determinado momento como um ser humano, ferramentas como árvores de comportamento, sistemas que estabelecem diversas reações que um NPC pode realizar e simuladores sociais, sistemas que visam simular comportamentos quando as pessoas interagem e se relacionam, em conjunto com as novas tecnologias, buscam gerar uma relação de apofenia entre o jogador e os NPC's, ou seja, a geração de linhas de interpretação e interação de acordo com as ideias e pensamentos das pessoas.

Este ensaio apresenta uma reflexão sobre a evolução dos processos através dos quais os NPCs respondem e interagem com o jogador e os modelos generativos que estão por trás disso, a complexidade envolvida na construção de tais narrativas, bem como o escopo atual e os caminhos possíveis para siga nesse sentido.

Palavras-chave: Complexidade - Videogames - Narrativas não lineares - RPG - NPC

[Las traducciones de los abstracts fueron supervisadas por el autor de cada artículo.]
