

# Una mirada a los algoritmos de IA mas comunes: Una clasificación de algoritmos para potenciar la experiencia de un videojuego

Carlos Astengo-Noguez  
Departamento de Ciencias  
Tec de Monterrey  
Monterrey, México  
[castengo@tec.mx](mailto:castengo@tec.mx)

Lorena B. Martinez Elizalde  
Departamento de computación  
Tec de Monterrey  
Monterrey, México  
[lorenamtze@tec.mx](mailto:lorenamtze@tec.mx)

**Abstract—** Desde el surgimiento de la Inteligencia Artificial (IA) a mediados del siglo XX hasta nuestros días, la cantidad de disciplinas, técnicas y métodos que se añaden constantemente a esta disciplina ha tenido un crecimiento acelerado. Hoy en día con la aparición de la IA generativa el interés por estos temas se ha exponenciado.

**Pero surge la pregunta básica, ¿por donde iniciar? Este análisis ofrece un punto de partida para las nuevas generaciones interesadas en el excitante campo de los videojuegos, destacando las técnicas y métodos de la IA que han impulsado su desarrollo.**

**Keywords—**IA, Videogame, ML

## I. INTRODUCTION (HEADING 1)

Los orígenes de los videojuegos se remontan a la década de 1950, cuando los primeros programas informáticos interactivos comenzaron a emerger en laboratorios de investigación y universidades. Uno de los primeros ejemplos fue "Spacewar!", desarrollado por estudiantes del Instituto Tecnológico de Massachusetts (MIT) en 1962, que permitía a dos jugadores controlar naves espaciales y luchar entre sí en pantalla. Desde esos primeros momentos la idea de jugar con y contra las máquinas se volvió tema central.

En esos mismos años el campo de la Inteligencia Artificial (IA) emergía como un campo de estudio y desarrollo que capturaba la imaginación de científicos, empresarios, y la sociedad en general.

La inteligencia Artificial desde 1956 ha escapado a ser encasillada en una sola definición simple y aceptada por todos. En vez de ello es y ha sido amalgama de disciplinas, como la matemática, la biología, las ciencias computacionales, la economía y la filosofía solo por mencionar algunas.

Diversas técnicas y métodos se han desarrollado bajo el paraguas de la IA y poco a poco han encontrado su nicho en áreas específicas. Un ejemplo claro son las técnicas de búsqueda y ordenamiento que hoy en día se encuentran en libros universitarios de algoritmos básicos pero que en algún momento fueron los elementos angulares de la IA. Otras disciplinas como el aprendizaje automático, también llamado lenguaje máquina o aprendizaje máquina por el anglicismo de machine learning nacieron dentro de la inteligencia artificial y hoy en día buscan su propio nicho y reconocimiento.

Este pequeño artículo busca mencionar simplemente algunas de las técnicas y métodos de la inteligencia artificial que hicieron desarrollar la industria de los videojuegos como punto de partida para las nuevas generaciones que desean trabajar en este emocionante campo.

## II. AGENTES COMO PUNTO DE PARTIDA

Existen diversos puntos de vista sobre como introducir y definir Inteligencia Artificial, para este pequeño artículo nos basaremos en las definiciones de Russell y Norvig [1]

Gartner define la Inteligencia Artificial como la aplicación de análisis avanzados y técnicas basadas en la lógica, incluido el machine learning (ML), para interpretar eventos, respaldar y automatizar decisiones y llevar a cabo acciones [1].

Marvin Minsky describió la IA como "la construcción de programas de computadora que se dedican a tareas que, por el momento, son realizadas de manera más satisfactoria por seres humanos, ya que requieren procesos mentales de alto nivel como el aprendizaje perceptual, la organización de la memoria y el razonamiento crítico"[7]

Finalmente Stuart Russell y Peter Norvig describen la IA desde el punto de vista de Agentes, Un agente es cualquier entidad que percibe su entorno y responde ante él. Concluyendo así que "La inteligencia artificial se ocupa del diseño de agentes racionales"[1]

Un agente es una entidad que percibe su entorno a través de sensores y actúa en ese entorno mediante actuadores. Así, podemos identificar agentes biológicos (organismos vivos), agentes de software (como soft bots) y agentes de hardware (como robots).

Para construir agentes podemos basarnos en su arquitectura:

- a) Agentes reactivos simples: Actúan basándose únicamente en la percepción actual del entorno, sin mantener una representación interna del estado del mismo.
- b) Agentes basados en modelos: Mantienen una representación interna del estado del entorno y utilizan esa representación para tomar decisiones.

- c) Agentes basados en objetivos: Tienen un objetivo definido y toman decisiones para alcanzar ese objetivo.
- d) Agentes basados en utilidad: Toman decisiones que maximizan una función de utilidad, que puede estar relacionada con la satisfacción de objetivos o con otros criterios.
- e) Agentes multiagentes: Son sistemas en los que múltiples agentes trabajan juntos para lograr un objetivo común o compiten entre sí.

Existen diferentes arquitecturas de Agentes que van desde aquellos únicamente reactivos, hasta los racionales pasando por agentes de utilidad, agentes de metas y agentes que aprenden por señalar algunos cuantos.

La idea de Agente esta muy relacionada con los videojuegos desde el jugador, el NPC (del inglés non playable character) e incluso el sistema con el que juega compitiendo o colaborando de manera automática.

### III. SETUP DEL JUEGO

En el diseño y desarrollo de videojuegos el diseño de niveles del juego tiene uno de los papeles más cruciales para el éxito comercial del producto.

En el diseño de los juegos un punto importante es la configuración o por su anglicismo el setup del juego. Este consta tanto del espacio de interacción como las relaciones de interacción entre el sistema y el o los jugadores.

Podemos iniciar describiendo el mundo donde se llevará a cabo el juego e identificamos varios puntos de partida:

- f) El mundo es completo y conocido. La mayoría de los juegos como el ajedrez o el monopolio permite tener una visión completa del mundo. En el Ajedrez las 64 casillas y sus disposiciones de cuadros blancos y negros con la regla de que la casilla de la derecha es blanca es conocido desde antes de iniciar el juego.
- g) El mundo es parcialmente conocido. Juegos como The Legend of Zelda: Breath of the Wild, inician con un pedazo del mapa del mundo conocido por el heroe y conforme uno avanza en el juego nuevas secciones del mapa se van revelando.
- h) El mundo es determinista, Si se puede representar como una máquina de estados finitos.
- i) El mundo es aleatorio o pseudoaleatorio si los eventos que suceden obedecen un comportamiento que el jugador puede identificar como debido al azar.

Autores como Rusell y Norvig proponen otras divisiones de los mundos en IA. Como episódico vs secuencial, estático vs dinámico, discreto vs continuos o de simple agente vs multiagente.

Para nosotros estas son subcategorías del tipo de juego que se desea diseñar. Por ejemplo en juegos de Rol como D&D, podemos tener el setup de un mundo multiagente parcialmente conocido con eventos discretos en una mecánica por episodios.

Otro aspecto de la configuración del juego esta en la decisión de:

- a) Personalización del Jugador, desde sus atributos o estadísticas básicas como suelen ser: fuerza, destreza, carisma, inteligencia, constitución, sabiduría, puntos de

daño etc. Hasta aspectos estéticos como género, raza, clase, etc.

- b) Personajes no jugables o NPC por su anglicismo. Son entidades con las que el jugador interactua ya sean amigos o enemigos. Existen diversas técnicas de IA que permiten generarlos. Estas técnicas van desde la creación aleatoria de sus estadísticas hasta por ejemplo crear hordas de NPC a través de algoritmos genéticos [2] con características similares pero a su vez diferenciadas en cada individuo. Esto contribuye a tener enemigos más desafiantes

### IV. ALGORITMOS CLÁSICOS PARA VIDEOJUEGOS

Iniciemos la revisión de algoritmos con juegos clásicos como el llamado Gato o Tic-Tac-Toe donde el setup del mundo es simplemente un arreglo de 3 x 3 donde cada jugador tiene un símbolo y por turnos colocan su simbolo en un espacio vacío del arreglo buscando conectar en fila, columna o diagonal tres de sus simbolos para ganar.

Este simple juego se puede representar con un diagrama de árbol, representado en la figura 1, y simplemente buscar el siguiente movimiento dentro de las ramas donde esto es posible. Asi los algoritmos de Búsqueda en profundidad y búsqueda por anchura son los algoritmos más simples para iniciarse en IA.

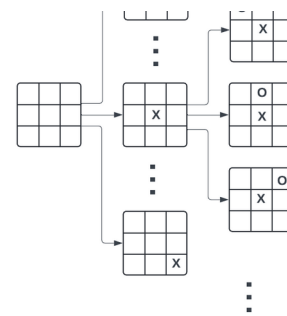


Fig. 1. Árbol generado por las posibles tiradas de ambos jugadores.

Poco a poco la experiencia nos dice que iniciar en cualquier esquina es básicamente el mismo espacio de solución solo con una perspectiva diferente por lo que podemos cortar varias de las ramas. Igual sucede con las casillas que quedan entre las esquinas, así podemos introducir la búsqueda con poda en árboles de solución.

Cuando el juego es más complicado y se desea por ejemplo hacer una búsqueda del camino de un personaje en un mundo con obstáculos estas técnicas no resultan ser óptimas ya que desperdician recursos al calcular todas las rutas posibles incluso con podas. Aquí nacen los algoritmos con heurísticas siendo A\* el algoritmo preferido.

Cabe mencionar que aunque A\* es un algoritmo bastante conocido existen muchas forma de implementarlo. Vease por ejemplo [3].

Dentro de los algoritmos de navegación clásicos estan los diagramas de Voronoi o los waypoints por mencionar algunos.

Otra técnica bastante utilizada por ejemplo para los comportamientos, interacciones de lenguaje o el simple cambio de sprites para animaciones son las máquinas de estados finitos. Una simple representación de procesos

markovianos. Ejemplos clásicos como los juegos de administración Harvest moon, etc.

## V. TÉCNICAS BÁSICAS DE IA PARA VIDEOJUEGOS

Una vez definido los elementos de agentes y configuración del juego podemos hablar de algunas técnicas clásicas que se han utilizado por décadas en videojuegos.

Para la representación del mundo podemos partir del uso de lógica de primer orden. Así las percepciones tanto de los NPC como el sistema de interacción para el jugador se resuelven con un lenguaje de programación orientado a Reglas. Vease por ejemplo CLIPS [4]

Para interacciones más complicadas o reglas en el mundo que puedan variar de forma aleatoria pueden utilizarse algoritmos de redes bayesianas.

Para lograr mejores interacciones entre NPC y jugadores o mundo-jugador podemos añadir elementos de aprendizaje. De estos algoritmos quizá el aprendizaje Máquina es de los más utilizados.

Otras técnicas como el uso de Parvadas que se conocen desde 1986 por Craig Reynolds han abierto la posibilidad de utilizar algoritmos inspirados en la naturaleza como el Swarm, Ants y Boids [5].

Para problemas de Planificación los algoritmos clásicos que podemos encontrar fácilmente en la literatura son: formaciones militares, Sistemas de navegación, Sistemas Presa-Depredador, Búsqueda basada en reglas, Planificación de Tareas, PERT, entre otros

## VI. TÉCNICAS MODERNAS DE IA PARA VIDEOJUEGOS

Aunque es claro que debemos mencionar la IA generativa basada en el procesamiento de Lenguaje Natural en específico en sistemas de lenguaje amplio por su extensa penetración en los últimos años en particular en las industrias creativas. No queremos dejar pasar primeramente otros métodos y técnicas modernas:

- a) Algoritmos que aprovechen las capacidades del hardware por ejemplo para juegos de realidad aumentada, virtual o mixta o de sistemas hápticos que permitan interactuar de formas no tradicionales.
- b) Algoritmos de reconocimiento de voz, que permitan una interacción más natural entre jugadores y el sistema
- c) Algoritmos basados en las preferencias del usuario. Son algoritmos de aprendizaje, más allá de los tradicionales donde el sistema aprende sobre las preferencias, usos y costumbres del jugador personalizando su experiencia. Por ejemplo *The Legend of Zelda*, *Tears of the Kingdom* en el cual el mundo sube de dificultad de acuerdo con las habilidades que va desarrollando el jugador.
- d) Existen diversos métodos para generar el código necesario y así reducir los tiempos de producción del juego. Desde "snippets", que son pequeños conjuntos de líneas de código predefinidas, hasta el "visual scripting", que implica programar utilizando bloques visuales, estos algoritmos son herramientas clave en el proceso de desarrollo.
- e) Desde Noviembre de 2022 las Inteligencias Artificiales Generativas, basadas en redes neuronales y Deep learning, han permeado en la vida cotidiana y en especial en varias industrias. Entre ellas las industrias

creativas se han visto beneficiadas desde varias perspectivas con nuevos productos y servicios. Sin embargo para estas industrias existen plataformas conocidas desde varios años atrás, por ejemplo ADOBE SENSEI desde 2016.

## VII. PROPUESTA DE CLASIFICACIÓN

Con el fin de hacer más sencillo el estudio de los algoritmos de IA para la industria de los videojuegos, proponemos revisar la siguiente clasificación, basada en las aplicaciones más comunes de la industria.

La IA a fin de cuentas son los algoritmos que te ayudan a resolver problemas, proponemos los siguientes problemas y los algoritmos que los solucionan:

### A. Movimiento en el mundo

Uno de los problemas más importantes dentro del área de los videojuegos es movilizar unidades en el mundo. Y no solo eso, si no que el movimiento parezca inteligente o con propósito. Los siguientes algoritmos nos pueden ayudar a solucionar este problema.

- a. Pathfinding clásico: Estos algoritmos buscan encontrar caminos de manera automática con el propósito de apoyar el movimiento de personajes y objetos en el juego.
  - i. Dijkstra: Busca el camino más corto entre dos nodos en un grafo ponderado, es decir, las aristas tienen pesos. Este grafo puede ser dirigido o no dirigido. Este algoritmo visita todos los nodos en el camino y luego determina cual fue el menos costoso.
  - ii. A\*: Es una mejora al algoritmo de Dijkstra, ya que implementa una heurística para estimar el costo restante del camino, en lugar de averiguar todas las posibles opciones.
- b. Parvadas: Conjunto de objetos que se mueven juntos. Estos pueden representar una gran variedad de cosas en los videojuegos.
  - i. Boids: Este algoritmo modela a la parvada nombrando a cada uno de los elementos o agentes que la integran con el nombre de Boid. En general se implementan tres simples reglas: evitar colisionar, mantenerse dentro del grupo y seguir la dirección del grupo. Basándose en estas reglas se calcula la posición y velocidad relativa del Boid con el resto de los objetos en la parvada.
- c. Comportamiento emergente: Este concepto describe como el comportamiento colectivo complejo y ordenado puede surgir de la interacción entre agentes simples.
  - i. Algoritmos de enjambre: Son algoritmos inspirados por la naturaleza se basan en el principio de que las acciones individuales de los miembros, siguiendo reglas simples, conducen a comportamientos colectivos complejos. Algunos ejemplos pueden ser, colonias de hormigas, enjambres de partículas.
  - ii. Sistemas multiagente: Sistemas compuestos de muchos agentes que se comunican y colaboran.
- d. Formaciones militares: Estos algoritmos pueden utilizar algunos de los algoritmos descritos

anteriormente, enfocándose en organizar unidades en formaciones militares.

### B. Toma de decisiones

Otro problema común es lograr que los NPCs tomen decisiones. Y no solo eso, si no que las decisiones parezcan inteligentes.

- a. Árboles de Decisión: Para utilizar esta técnica, primero debemos generar un modelo del problema como en la figura 1. Después podemos realizar búsquedas dentro de la estructura utilizando parámetros para tomar la decisión.
- b. Algoritmos genéticos: Estos algoritmos están inspirados en la biología y utilizan una población de soluciones candidatas expresadas como cromosomas binarios, la cual se evoluciona mediante reglas seleccionadas a lo largo de generaciones utilizando operadores genéticos como selección, cruce y mutación. Son muy útiles para encontrar soluciones óptimas a problemas complejos y multidimensionales.
- c. Redes neuronales: Otro algoritmo inspirado en la biología, el cual se basa en la simplificación de una neurona, en la cual se procesan operaciones matemáticas simples. Estas redes surgen de conectar muchas de estas neuronas, y así se pueden resolver problemas de clasificación, reconocimiento de patrones, procesamiento de lenguaje y regresión entre otras.
- d. Algoritmos de optimización: Son técnicas utilizadas para encontrar la mejor solución a un problema, generalmente maximizando o minimizando una función objetivo tradicionalmente sujeto a restricciones.
- e. Redes Bayesianas: Son modelos gráficos probabilísticos que representan relaciones entre variables a través de un grafo dirigido acíclico. Cada nodo de este grafo representa una variable y los arcos representan las dependencias probabilísticas entre ellas. Se pueden usar para modelar incertidumbre y realizar inferencias probabilísticas, lo que se hace útil para tomar decisiones.

### C. Representación del mundo

Modelar el mundo del videojuego con líneas de código es un reto interesante e importante. Estos modelos nos permiten identificar cambios en las condiciones del mundo en el que se desempeña nuestro jugador.

- a. Mapas de influencia: Son una técnica de representación espacial que se utiliza para modelar y visualizar el impacto que tienen diferentes factores en un entorno determinado. Estos mapas asignan valores a cada ubicación en función de factores como la accesibilidad, la visibilidad, la peligrosidad, etc.
- b. Redes Bayesianas: Ver la definición en la sección B inciso e.
- c. Grafos de conocimiento: Son estructuras de datos que representan el conocimiento mediante nodos y relaciones entre ellos. Cada nodo representa un concepto o entidad, y las relaciones representan conexiones semánticas entre los conceptos.
- d. Modelos de creencias, deseos e intenciones (BDI): Una forma de representar la cognición humana en

sistemas inteligentes. Dividen el comportamiento de un agente en tres componentes principales: creencias (lo que el agente cree sobre el mundo), deseos (los objetivos que el agente quiere alcanzar) e intenciones (los planes que el agente desarrolla para alcanzar sus objetivos).

- e. Modelos de estado acción recompensa: Estos modelos son comúnmente utilizados en el aprendizaje por refuerzo. Representan la interacción entre un agente y su entorno mediante la observación de estados, la selección de acciones y la recepción de recompensas. El agente aprende a tomar decisiones óptimas maximizando la recompensa esperada a largo plazo.
- f. Sistemas de reglas basados en lógica: Son una forma de representar el conocimiento mediante reglas condicionales del tipo "si-entonces". Estas reglas permiten inferir conclusiones lógicas a partir de condiciones dadas.
- g. Máquinas de estados basadas en pilas: Las máquinas de estados basadas en pilas son un modelo computacional que describe el comportamiento de sistemas que pueden estar en diferentes estados y cambiar de estado en respuesta a eventos externos o internos. Utilizan una estructura de datos tipo pila para gestionar la transición entre estados.
- h. Análisis de terrenos: Es una técnica que se utiliza para representar y analizar la topografía y las características físicas de un área geográfica determinada. Se utilizan modelos digitales de elevación (DEM) para representar la elevación del terreno y se aplican algoritmos para realizar análisis espaciales, como la identificación de cuencas hidrográficas, la planificación de rutas y la detección de cambios en el paisaje.

### D. Algoritmos por turnos

Administrar los turnos en las partidas de juego puede ser algo complicado, estos algoritmos nos ayudan a darle solución.

- a. Ciclo de eventos por turnos: Este es un enfoque de procesamiento donde las acciones o eventos se manejan en un orden secuencial y predefinido. En este ciclo, cada entidad o componente del sistema tiene un turno o momento designado para realizar su acción. Después de que todas las entidades hayan completado sus acciones en un turno, el ciclo pasa al siguiente turno y se repite el proceso.
- b. Sistema de líneas de espera de eventos: Este sistema gestiona eventos mediante una cola o línea de espera. Los eventos se encolan en el orden en que ocurren y se procesan uno por uno de acuerdo con su prioridad o tiempo de llegada.
- c. Algoritmos de planificación: Los algoritmos de planificación son métodos computacionales utilizados para generar secuencias de acciones que permiten alcanzar un objetivo específico en un entorno dado.
- d. Simulación de eventos discretos: es un tipo de modelo de simulación donde el tiempo avanza en incrementos discretos y los eventos ocurren en momentos específicos. Cada evento cambia el estado del sistema y puede desencadenar eventos futuros.

- e. Sistema de estados y transiciones: es un modelo que describe el comportamiento de un sistema como una serie de estados y las transiciones entre ellos. Cada estado representa una condición o situación en el sistema, y las transiciones representan los cambios de un estado a otro en respuesta a eventos o condiciones específicas.

#### E. Toma de decisiones

- a. Árboles de decisión: Ver la definición en la sección B inciso a.
- b. Algoritmos genéticos: Ver la definición en la sección B inciso b.
- c. Redes Neuronales: Ver la definición en la sección B inciso c.
- d. Redes Bayesianas: Ver la definición en la sección B inciso e.
- e. Lógica difusa: técnica de procesamiento de información que permite manejar la incertidumbre y la imprecisión en los datos. A diferencia de la lógica booleana tradicional, que solo admite valores de verdad binarios (verdadero o falso), la lógica difusa permite que una variable tenga un grado de pertenencia a un conjunto difuso. Es decir, podemos decir que es "más fácil" o "medio difícil" y esto se representa con un número entre 0 y 1 que indica que tan fuertemente pertenece al conjunto fácil o difícil.

Los problemas aquí expresados no deben tomarse como un estudio exhaustivo de los problemas relacionados con la IA en la industria de los videojuegos. Deben ser entendidos como los problemas básicos para iniciarse en este fascinante mundo.

### VIII. CONCLUSIONES

Este artículo ofrece una visión general de varios algoritmos, técnicas y métodos clave en el campo de la Inteligencia Artificial. Desde su surgimiento en los últimos años de la Segunda Guerra Mundial, la IA ha evolucionado constantemente, presentando desafíos continuos en la actualidad. Para aquellos que se inician en este campo, la diversidad de algoritmos y disciplinas puede resultar abrumadora.

Sin embargo, comprender que la Inteligencia Artificial es una herramienta poderosa para abordar problemas complejos es fundamental en nuestra era tecnológica. Este artículo busca proporcionar un punto de partida para aquellos interesados en explorar más a fondo estos métodos, especialmente en el contexto de los videojuegos. A medida que la industria continúa avanzando, el conocimiento y la aplicación de estos algoritmos son esenciales para impulsar la innovación y el progreso en el desarrollo de los videojuegos.

### IX. ACKNOWLEDGMENT

Nos gustaría agradecer a la REDINDVJ por la oportunidad de esta publicación, así como al IIDIV y al Tec de Monterrey por el apoyo y recursos para la investigación.

### X. REFERENCES

[1] S. J. 1. N. P. & D. E. Russell, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2010

[2] "https://www.gartner.es/temas/inteligencia-artificial#:~:text=En%20Gartner%20definimos%20la%20inteligencia,y%20llevar%20a%20cabo%20acciones.," Gartner, 2023. [Online]. [Accessed 2024].

[3] F. D. Laramée, *Genetic Algorithms: Evolving the Perfect Troll*, vol. AI Game Programming Wisdom, S. Rabin, Ed., Hingham, Massachusetts: Charles River Media, 2002, pp. 629-639.

[4] S. Rabin, *AI Game Programming Wisdom (Game Development Series)*, 2002.

[5] L. Secret Society Software, "CLIPS, A Tool for Building Expert Systems," 2023. [Online]. Available: <https://www.clipsrules.net/>. [Accessed 2024].

[6] C. Reynolds, "Flocks, Herds, and Schools:," 1984. [Online]. Available: <https://www.cs.toronto.edu/~dt/siggraph97-course/cwr87/>. [Accessed 2024].

[7] M. Minsky, "Artificial Intelligence," Cambridge, MA: MIT Press, 1968.

