



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
FACULTAD 5

MÓDULO DE INTELIGENCIA ARTIFICIAL  
PARA EL VIDEOJUEGO RÁPIDO Y CURIOSO.  
INFORME DE LOS APORTES PERSONALES AL PROYECTO  
JUEGOS CONSOLA

**Presentado en opción al título de  
Máster en Informática Aplicada**

Autor: Ing.Marvyn Amado Márquez Rodríguez.

Tutor: MSc.Yuniesky Coca Bergolla

**Ciudad de la Habana**

**Junio de 2010**

## Declaración jurada de autoría

Yo, Marvyn Amado Márquez Rodríguez, con carné de identidad 84071814286, declaro que soy el autor principal del resultado que expongo en la presente memoria titulada “Módulo de Inteligencia Artificial para el videojuego Rápido y Curioso”, para optar por el título de Máster en Informática Aplicada.

Autorizo a la Universidad de las Ciencias Informáticas a hacer el uso que estime pertinente de los resultados aquí presentados, como propietaria de los derechos legales de este proyecto.

Finalmente declaro que todo lo anteriormente expuesto se ajusta a la verdad, y asumo la responsabilidad moral y jurídica que se derive de este juramento profesional.

Y para que así conste, firmo la presente declaración jurada de autoría en Ciudad de la Habana a los \_ días del mes de \_ del año \_.

---

Marvyn Amado Márquez Rodríguez

## Agradecimientos

Quisiera agradecer a la Revolución por haberme dejado participar en este proyecto maravilloso.

Agradecer en especial a mis padres que han sido siempre los motores impulsores de mi vida profesional. Mi madre, la mejor de las mujeres, siempre supo como guiar mis pasos en todo momento; agradecerte el haberme inculcado el sentimiento de superación constante. Mi padre, la persona de mayor cultura que he conocido, gracias por todo el sacrificio que has hecho para que yo culminara mis estudios, gracias por tener ese ingenio que te caracteriza y saber responder siempre todas mis dudas con las respuestas adecuadas. A Melvin: agradecerte por ser un hermano incondicional, darte las gracias por ser mi inspiración y por plantearme el reto profesional.

Agradecer a mi novia, Diannys, que estuvo siempre a mi lado sin importar cuán difícil fuera el momento y darme las fuerzas y el amor necesario para salir adelante.

A mi tutor, Yuniesky Coca, por sus sabios consejos que me sirvieron para enfocar de la mejor manera mi investigación. A dailyn por enseñarme latex.

Agradecer a Rolando, por haber compartido conmigo la confección de la memoria colectiva de un producto maravilloso. A mis compañeros del proyecto: Hassan, Ernesto Mileidy, Yalina, Sandor, Ismael, Yulien, Pacheco, Carlos, Yordanis, quienes siempre han sido ejemplo de abnegación y de constancia en el trabajo.

Quisiera agradecer a mi compañero de trabajo y de grandes batallas: Andy, por su sabiduría y por compartir conmigo mis logros como informáticos. . . .

## Síntesis

En este trabajo se combinan técnicas de inteligencia artificial deterministas y no deterministas para definir el comportamiento de carros autónomos en un videojuego de carreras. Para desarrollar un modelo de oponente competitivo se aprovecha la capacidad de generalización de un perceptrón multicapa, así como la eficiencia y sencillez de combinar una máquina de estados finitos con comportamientos de locomoción. El proceso de aprendizaje de la red neuronal artificial cuenta con dos etapas fundamentales: en la primera se obtiene una red con un conocimiento básico de las acciones que debe realizar el carro autónomo, y en la segunda etapa se mejora su comportamiento entrenándola con registros del desempeño de varios jugadores reales. Las técnicas deterministas permiten lidiar con situaciones específicas que evitan aumentar el número de neuronas de la red y mantienen los niveles de respuesta en tiempo real.

**Palabras Claves:** Inteligencia artificial, técnicas no deterministas, técnicas deterministas, red neuronal artificial, perceptrón multicapa, carro autónomo, aprendizaje, entorno virtual, máquina de estado, comportamiento de locomoción.

# Índice de contenidos

<b>1. Introducción</b>	<b>1</b>
<b>2. Marco Teórico</b>	<b>4</b>
2.1. Inteligencia Artificial en videojuegos . . . . .	4
2.1.1. Técnicas Deterministas . . . . .	5
2.1.2. Técnicas No Deterministas . . . . .	6
<b>3. Contribución personal.</b>	<b>8</b>
3.1. Comportamientos de Locomoción. . . . .	8
3.2. Máquina de estado y comportamiento de locomoción. . . . .	8
3.3. Redes neuronales. . . . .	9
3.4. Combinación de técnicas deterministas y no deterministas. . . . .	10
3.4.1. Fuera de competencia. . . . .	11
3.4.2. Adelantando Contrarios. . . . .	12
3.4.3. Siguiendo Trayectoria. . . . .	13
3.4.4. Redes Neuronales Artificiales. . . . .	13
3.4.5. Entorno del videojuego. . . . .	14
3.4.6. Sistema Sensorial. . . . .	14
3.4.7. Distancia al centro de la pista. . . . .	14
3.4.8. Ángulo entre la dirección del carro y la pista. . . . .	15
3.4.9. Ángulo de la próxima curva en la pista. . . . .	16
3.4.10. Distancia del carro a la próxima curva. . . . .	16
3.5. Interfaz para controlar el auto. . . . .	17
3.6. Diseño de la RNA. . . . .	17
3.7. Interpretación matemática de las variables empleadas. . . . .	18
3.7.1. Distancia al centro de la pista. . . . .	19
3.7.2. Ángulo entre la dirección del carro y la pista. . . . .	20
3.7.3. Ángulo de la próxima curva. . . . .	21
3.7.4. Distancia del carro a la próxima curva. . . . .	21
3.7.5. Ángulo de giro del timón. . . . .	22
3.8. Velocidad deseada. . . . .	22
3.9. Proceso de entrenamiento. . . . .	23
<b>4. Conclusiones</b>	<b>25</b>
<b>A. Glosario de términos</b>	<b>26</b>
<b>Referencias</b>	<b>28</b>

## 1. Introducción

Con el incremento del poder computacional y la experiencia acumulada, los juegos han evolucionado considerablemente en los últimos años. Han incorporado detallados modelos de miles de polígonos que interactúan entre ellos, respondiendo a complejas leyes físicas. Los agentes controlados por computadora en los ambientes que recrean los videojuegos poseen personalidades adaptativas y son capaces de crear sus propias estrategias de juego.

Diferentes géneros han sido creados para satisfacer los diversos gustos de los jugadores. Se destacan los de deportes, estrategias, combates, juegos de roles, de mesa y los de motor. Dentro de esta última categoría encontramos los juegos de carreras de automóviles, que cuentan con gran aceptación del público. En ellos los jugadores adoptan el rol de pilotos, y tratan de superar los límites de velocidad para mejorar sus marcas de tiempo en carreras contrarreloj, o para vencer a sus oponentes humanos y artificiales.

Los desarrolladores buscan satisfacer las expectativas de los usuarios ofreciéndole una experiencia única con cada nuevo juego. Para lograr este objetivo es sumamente importante el desafío ofrecido por los elementos inteligentes.

En la mayoría de los videojuegos de carreras de automóviles, los carros autónomos <sup>1</sup> son controlados por acciones y reacciones predeterminadas por el desarrollador y no poseen capacidad de aprendizaje o adaptación ante situaciones desconocidas, por lo que su comportamiento es restringido, esquemático y se torna predecible; cuando esto ocurre, los jugadores vencen con facilidad a los oponentes artificiales, perdiendo el interés por un juego que deja de ser entretenido y útil.

---

<sup>1</sup>Carros que son controlados por la computadora y desarrollan un comportamiento inteligente.

Por esta razón los desarrolladores buscan satisfacer las expectativas de los usuarios ofreciéndole una experiencia única con cada nuevo juego. Para lograr este objetivo es sumamente importante el desafío ofrecido por los elementos inteligentes.

Con la gran cantidad de juegos que existen hoy en el mundo, es posible afirmar que la clave del éxito entre dos juegos de similares características visuales, se encuentra precisamente en la capacidad que tenga de entretener al usuario, presentándole rivales capaces de retar sus habilidades con un adecuado nivel de inteligencia.

En la facultad 5 de la Universidad de las Ciencias Informáticas se desarrolló el videojuego de conducción de automóviles “*Rápido y Curioso*”, el cual solamente contaba con el modo de carreras individuales contra reloj, donde el usuario adoptaba el rol de un conductor y competía para mejorar su propio tiempo. Con el objetivo de brindar mayor diversión a los usuarios y alargar su vida útil, se hacía necesario incorporar nuevos modos de juego que le permitieran competir contra oponentes controlados por la máquina u otros jugadores.

Por este motivo se hacía necesario crear carros autónomos capaces de competir y que brindaran emoción al jugador, que no presentaran un comportamiento esquemático, para lograr que el videojuego “*Rápido y Curioso*” tuviera un mayor nivel de entretenimiento y vida útil.

La problemática anteriormente expuesta motivó al autor a resolver el problema científico de ¿Cómo lograr que los carros controlados por la computadora (NPC) desempeñen un comportamiento inteligente en el videojuego de carreras “*Rápido y Curioso*”? Teniendo como objeto de estudio las técnicas de inteligencia artificial. Por tanto el objetivo general de este trabajo será: Definir el comportamiento de los carros controlados por la computadora en el videojuego

de carreras “*Rápido y Curioso*”. El área de conocimiento o campo de acción de la investigación es: las técnicas de inteligencia artificial aplicadas en videojuegos de conducción de automóviles.

El desarrollo de esta investigación reviste especial importancia para el juego “*Rápido y Curioso*” porque le permitirá incorporar a su entorno elementos inteligentes, que serán controlados por técnicas de inteligencia artificial.

## 2. Marco Teórico

### 2.1. Inteligencia Artificial en videojuegos

La inteligencia artificial generalmente se utiliza para resolver problemas difíciles, en los que no se conoce de antemano el mejor método para resolverlo. Tiene aplicaciones en diversos campos, como el militar, la medicina, los negocios, los servicios, la ciencia, la industria del entretenimiento, entre otros.

La industria de los videojuegos ha servido como base para probar y desarrollar muchas técnicas de inteligencia artificial; beneficiándose a la vez con realistas modelos de inteligencia en sus productos de entretenimiento, que los hacen más atractivos para los usuarios.

En la industria de los videojuegos, el término Non-Player Character (NPC) se refiere a agentes autónomos controlados por la computadora [2]. Para que estos agentes se comporten de manera inteligente, se utilizan varias técnicas de la inteligencia artificial que se pueden clasificar en:

- **Deterministas:** El comportamiento del agente inteligente es especificado por completo; o sea, los programadores tienen que codificar todas las acciones explícitamente, dificultando y retrasando el proceso de desarrollo del juego. Las más usadas son las máquinas de estado finito, los árboles de decisión, y los sistemas de reglas de producción.
- **No deterministas:** Es todo lo contrario, el grado de incertidumbre es mayor, por lo que no se puede predecir el comportamiento que seguirá el agente. Generalmente usan técnicas como redes neuronales, algoritmos

evolutivos o redes bayesianas, que facilitan el aprendizaje y la adaptación al entorno de los elementos inteligentes. No hay que codificar explícitamente todas las posibles situaciones en el juego, ya que los elementos inteligentes pueden incluso extrapolar sus comportamientos y desarrollar otros nuevos o emergentes.

### 2.1.1. Técnicas Deterministas

Los desarrolladores de videojuegos han experimentado con la mayoría de las técnicas de inteligencia artificial, pero sin dudas las más sencillas, eficientes, fáciles de implementar, entender y depurar son las deterministas, por lo que han sido las más usadas en este campo.

El empleo de trampas (del inglés cheating) es común en los videojuegos. Esta técnica le atribuye a los elementos controlados por la computadora capacidades o información extra que les permiten hacer mayor resistencia a los jugadores [16]. Por ejemplo, pueden tener información de la ubicación, cantidad y el tipo de las unidades del usuario en un juego de estrategia; pueden incluso tener mayor resistencia al daño, ser más veloces o tener armas más potentes. Sin embargo estas técnicas deben utilizarse con medida, ya que si son muy evidentes o el jugador las descubre puede perder el interés en un juego que “le hace trampas”.

En el juego “*The Sims*”, la inteligencia artificial es implementada con máquinas de estado finito difusas, además utiliza técnicas específicas de *A-Life*<sup>2</sup> para simular el comportamiento de organismos vivos (en este caso, personas que viven en familia). La base de la inteligencia en este juego es su motor de

---

<sup>2</sup>Se refiere a sistemas multi-agentes que intentan aplicar algunas de las propiedades universales de sistemas vivientes a agentes inteligentes en un entorno virtual.

comportamiento, el cual asocia acciones posibles a cada objeto. [2]

Un NPC debe ser capaz de reconocer y comprender el espacio que le rodea, empleando para ello técnicas de percepción. Además debe ser capaz de buscar caminos para navegar, por lo que se le presta una especial atención a los algoritmos de búsqueda como Dijkstra o A\*. Los juegos de tablero como el ajedrez y el backgammon, han usado árboles de búsqueda heurística con formidables aciertos. [2]

Tradicionalmente en los videojuegos se han utilizado técnicas de planificación o guiones (del inglés script) para definir la conducta de los agentes, al igual que los sistemas basados en reglas y los comportamientos grupales o de manadas. Los sistemas expertos son un tipo de sistema basado en reglas que definen el conocimiento para que los agentes autónomos se comporten de manera similar a un jugador experto.

### **2.1.2. Técnicas No Deterministas**

El reto actual de los desarrolladores es crear juegos novedosos, divertidos, realistas y con mayor vida útil. Un buen paso en ese sentido es lograr que los NPC aprendan, evolucionen, se adapten a nuevas situaciones y exhiban un comportamiento genuino. Estos resultados son posibles de alcanzar con métodos no deterministas, que algunos desarrolladores investigan con gran interés a pesar de su complejidad.

Se han alcanzado excelentes resultados en juegos que usan redes neuronales, algoritmos genéticos o métodos probabilísticos como *Dirt Track Racing*, *Creatures*, *Black & White*, *Battlecruiser 3000AD*, *Fields of Battle*, y *Heavy Gear*. Generalmente se combinan con métodos deterministas más tradicionales y es-

tudiados, conformando una especie de sistemas híbridos. [1]

*Unreal Tournament* es conocido y elogiado por su inteligencia artificial. El jugador puede elegir un nivel de dificultad (desde "novato" hasta "semidiós"), y más adelante el juego implementa una opción que ajusta automáticamente la dificultad al nivel del jugador humano.

Los juegos de carreras de autos también han experimentado con estas técnicas. El Colin McRae Rally 2.0 es posiblemente el que primero las incorporó a juegos de este tipo, y ha servido como referencia a otros desarrolladores. La red neuronal que controla la inteligencia de los autos logra que estos tomen las curvas en superficies de arena o nieve con el nivel de precisión que lo hacen los jugadores; se mueven por toda la pista inteligentemente, bloquean a los contrarios, toman desvíos para acortar el camino, en fin, son dignos oponentes [11].

Por otra parte, Julian Togelius ha estado trabajando en un proyecto que vincula redes neuronales artificiales y algoritmos evolutivos para "hacer los juegos de carreras más divertidos" como él mismo dice en su artículo [17]. La idea es modelar el estilo de conducción de un jugador con una red neuronal y a partir de ahí, generar con un algoritmo evolutivo una pista de carreras que se adapte a las habilidades personales de dicho jugador.

### **3. Contribución personal.**

#### **3.1. Comportamientos de Locomoción.**

A partir del estudio de la teoría existente en la comunidad de desarrollo de videojuegos, para enfrentar el problema de la presente investigación, se comenzó a elaborar un módulo de inteligencia artificial controlado por comportamientos de locomoción (del inglés Steering Behaviors), lo que evidenciaba ser la variante más sencilla de implementar.

Esta solución incluyó 3 comportamientos básicos, encargados de calcular los valores de ángulo de giro del auto con respecto a la dirección y sentido del circuito, el valor de aceleración y de frenado necesarios para controlar la velocidad del carro en cualquier instante de la competencia [15]. Como resultado se obtuvo un auto capaz de recorrer la pista rápidamente. Sin embargo una de las principales dificultades de esta variante fue que el comportamiento obtenido era rígido y esquemático, por lo que el juego se tornaba monótono y poco divertido, además los oponentes artificiales no eran capaces de reaccionar adecuadamente ante situaciones de emergencia.

#### **3.2. Máquina de estado y comportamiento de locomoción.**

La segunda variante empleada combinaba máquinas de estado finitos (FSM) y comportamientos de locomoción, *para mayor información ver capítulo 10 del informe final del proyecto*. La máquina de estados controlaba las transi-

ciones entre los diferentes estados en los que se podía encontrar un carro: en una línea recta de trayectoria, adelantando contrarios, evitando obstáculos, tomando una curva de la trayectoria o en una situación de estancamiento. Los comportamientos de locomoción se encargaban de controlar la dirección, aceleración y freno del auto en cada uno de los estados descritos.

Con esta solución se obtuvo un carro capaz de recorrer la pista en un tiempo razonable, y de reaccionar correctamente ante diversas situaciones del juego; pero resultó complejo desde el punto de vista de desarrollo identificar todos los estados y transiciones entre ellos, lo que hacía la solución poco escalable.

### **3.3. Redes neuronales.**

Con el objetivo de obtener un comportamiento más auténtico y menos determinista en los autos inteligentes, se decidió utilizar una red neuronal artificial (RNA), *para mayor información ver capítulo 10 del informe final del proyecto*. Básicamente la red funciona como el “cerebro” de uno de los conductores a derrotar. Se puede definir como un neurocontrolador <sup>3</sup> para el auto, que recibe información del entorno para determinar qué acción debe ejecutar en cada instante de la simulación.

El objetivo de la red es desempeñar un comportamiento capaz de emular un jugador humano. El competidor y el agente autónomo estarán en igualdad de condiciones en cuanto al acceso a información y acciones para maniobrar se refiere. Se evita el empleo de “trampas”, una efectiva fórmula utilizada a menudo por los desarrolladores de inteligencia artificial para videojuegos pero a la vez aborrecida por los jugadores, que pueden quedar muy decepcionados al

---

<sup>3</sup>neurocontrolador: Auto controlado por una Red Neuronal Artificial

descubrir las.

Con esta solución se obtuvo un carro capaz de recorrer la pista rápidamente, que respondía ante situaciones nuevas para las cuales no fue explícitamente adiestrada la red neuronal artificial. Se incorporó de esta manera capacidad de aprendizaje a los autos controlados por la computadora; mejorando el realismo del juego al conseguir oponentes artificiales que no exhiben un comportamiento rígido y esquemático.

Con el objetivo de mantener bajo el número de neuronas y no sobrecargar el procesamiento de la aplicación, se decidió no brindarle información a la RNA sobre los contrarios u obstáculos en la pista, aunque esto significara que el carro resultante no fuese lo suficientemente competitivo.

### **3.4. Combinación de técnicas deterministas y no deterministas.**

Como se explicó anteriormente las técnicas utilizadas ofrecían potencialidades y limitaciones para controlar los autos artificiales del videojuego; por lo que el siguiente paso fue combinar las mejores cualidades de cada una, en una solución robusta y eficiente. El comportamiento de los carros autónomos controlados por la RNA demostró ser la variante más natural y creíble, capaz de recorrer la trayectoria eficientemente y responder correctamente ante situaciones desconocidas. Para incorporarle a este auto la habilidad de adelantar y evitar colisiones con los contrarios, así como controlar las situaciones de estancamiento; se decidió utilizar máquinas de estado finito y comportamientos de locomoción, técnicas menos costosas, más sencillas de programar y depurar, y que además arrojaban un comportamiento adecuado, como se describió en el

*epígrafe 3.2.*

Así, se definieron los estados en los que un auto inteligente puede encontrarse durante la carrera:

- *SIGUIENDO TRAYECTORIA.*
- *ADELANTANDO CONTRARIOS.*
- *FUERA DE COMPETENCIA* (Inmovilizado, atorado, invertido).

En dependencia del estado en que se encuentre el agente autónomo, la FSM decide quién controlará sus acciones: el neurocontrolador o los comportamientos de locomoción predefinidos.

#### **3.4.1. Fuera de competencia.**

##### **Situación:**

Si el carro se encuentra obstruido por algún objeto (una valla u otro auto) o volcado en la pista producto de una colisión a alta velocidad, se puede afirmar que está en el estado *FUERA DE COMPETENCIA*. Por lo tanto es necesario colocarlo nuevamente en la dirección correcta de la trayectoria. Para determinar que el auto se encuentra *FUERA DE COMPETENCIA* existen dos posibilidades:

1. El auto ha estado colisionando durante un período de tiempo determinado, y su velocidad promedio ha disminuido a valores cercanos o iguales a cero.

2. El auto no está apoyado sobre sus 4 gomas, razón suficiente para afirmar que se ha volcado.

**Respuesta del sistema:**

La FSM otorga el control del carro al comportamiento “reincorporarse a la competencia” quien a su vez ejecuta las siguientes acciones:

1. Cambiar la marcha a retroceso para separarse del objeto que obstruye el paso y luego continuar la trayectoria.
2. Restaurar la dirección y posición del carro hacia un punto del recorrido cercano a la posición donde se encontraba.

**3.4.2. Adelantando Contrarios.**

**Situación:**

Cuando se tiene en el rango de visión del carro a un contrario cuya velocidad promedio es inferior a la suya el agente está en el estado *ADELANTANDO CONTRARIOS*.

**Respuesta del sistema:**

La FSM otorga el control del carro al comportamiento “adelantar\_ponente” que realiza las acciones siguientes:

1. Se determina en qué momento y por qué flanco acelerar.
2. Se evitan colisiones con el oponente y con otros obstáculos.
3. Se evita perder el control del carro sobre la trayectoria del circuito.

### 3.4.3. Siguiendo Trayectoria.

#### **Situación:**

En cualquier otro caso el agente está en el estado *SIGUIENDO TRAYECTORIA*.

#### **Respuesta del sistema:**

La FSM otorga el control del carro al neurocontrolador.

### 3.4.4. Redes Neuronales Artificiales.

Debido a la importancia que reviste el uso de redes neuronales artificiales para la solución presentada en este documento, se detallará a continuación su funcionamiento.

Para controlar la dirección y velocidad del auto en la pista se utilizó la red neuronal artificial perceptrón multicapa (PMC) con flujo de datos hacia adelante y algoritmo de aprendizaje back-propagation. Las conexiones son inter-capas, en las que cada neurona se conecta con la capa adyacente. [3]

Se decidió utilizar el PMC debido a su eficacia para resolver problemas complejos, principalmente de clasificación, en los que las variables no son linealmente separables. Una de sus principales virtudes es su habilidad para generalizar, que le permite reconocer patrones en el conjunto de entrenamiento [14]. Una vez entrenada esta red, si se le presentara una situación nueva, podrá reaccionar satisfactoriamente, aunque no haya sido “adiestrada” específicamente para ello. Otro aspecto a destacar de los PMC es la rapidez en el tiempo de respuesta, condición que lo ratifica como el candidato más apropiado para ser empleado en un videojuego, donde la eficiencia es fundamental.

#### **3.4.5. Entorno del videojuego.**

Para que un auto recorra la pista rápidamente evidenciando un comportamiento inteligente, debe poder reconocer su entorno.

Utilizando el editor de pistas, se ubican nodos circulares en el mapa del juego, procurando que su centro coincida con el de la pista y que su diámetro sea similar al ancho de la misma. Los nodos se asocian mediante aristas de un sólo sentido, conformando un grafo dirigido que representa el circuito de carreras. Por otra parte, cada nodo sólo contiene información sobre su posición, radio y la lista de sus vecinos, *para mayor información ver capítulo 10 del informe final del proyecto.*

#### **3.4.6. Sistema Sensorial.**

Cada auto tiene acceso a parámetros que son utilizados intuitivamente por las personas cuando conducen, como su ubicación espacial, a qué velocidad y en qué dirección se mueve, qué obstáculos tiene en el campo de visión, cuán próximo está del borde de la carretera, qué grado de peligrosidad presenta la próxima curva y cuál es el trazado del circuito.

#### **3.4.7. Distancia al centro de la pista.**

La distancia al centro de la pista fue la variante escogida para saber cuándo el carro se está alejando de su centro. Su utilización en lugar de la distancia a cada uno de los bordes, se debe en gran medida a la limitada información contenida en los modelos del juego, que no permite diferenciar los objetos sóli-

dos impidiendo conocer por ejemplo, cuáles son vallas, cuáles son señales de tránsito o cuál es el piso. Curiosamente la solución empleada resultó ser más eficiente, ya que evita realizar continuos chequeos de colisión con cada objeto sólido en la escena. En su lugar se calcula la distancia del carro al segmento de pista más cercano, valiéndose de la ecuación de la distancia de un punto a una recta (Ecuación 1).

El segmento más cercano al carro, se determina a partir de los tres segmentos de interés para el agente autónomo (actual, siguiente y anterior).

$$D(P, S) = \frac{|A_X + B_Y + C|}{\sqrt{A^2 + B^2}}$$

*Ecuación 1: Distancia de un punto a una recta.*

Donde  $P$  es la posición del carro y  $S$  el segmento definido por dos nodos consecutivos, que al estar ubicados aproximadamente en el medio de la pista, hacen que el segmento coincida con su centro, *ver capítulo 10 del informe final del proyecto.*

#### **3.4.8. Ángulo entre la dirección del carro y la pista.**

Al conducir el automóvil, es necesario conocer el sentido del movimiento del carro para compararlo con la trayectoria a seguir, y corregirlo en caso necesario.

Mediante la Ecuación 2 se obtiene el coseno del ángulo formado por el vector dirección del carro ( $V1$ ) y un vector ( $V2$ ) que parte del centro del carro y tiene la misma dirección que su segmento más cercano, por lo que es paralelo al mismo.

$$\cos(x) = \frac{V_1 * V_2}{|V_1| * |V_2|}$$

*Ecuación 2: Coseno del ángulo entre dos vectores.*

Si el ángulo tiene valores cercanos a cero, significa que el carro se mueve en la dirección correcta por todo el trazado, *ver capítulo 10 del informe final del proyecto.*

#### **3.4.9. Ángulo de la próxima curva en la pista.**

Utilizando la *Ecuación 2* se puede conocer la curvatura de la pista, que influye en el modo de controlar el carro. La amplitud de la próxima curva coincide con el ángulo formado entre los segmento actual ( $V_1$ ) y siguiente ( $V_2$ ), *ver capítulo 10 del informe final del proyecto.*

#### **3.4.10. Distancia del carro a la próxima curva.**

Otro aspecto que se tiene en cuenta a la hora de conducir, es la distancia a la que se encuentra el carro de una curva. Las curvas en el juego se identifican con un nodo, o sea: dondequiera que haya un nodo existe una variación en el sentido de la pista, aunque sea mínima. Por lo tanto, si se halla la distancia al próximo nodo se podrá conocer la distancia a la próxima curva:

$$D = |\text{próximo nodo} - \text{posición carro}|$$

*Ecuación 3: Distancia entre 2 puntos.*

### **3.5. Interfaz para controlar el auto.**

Para controlar un auto en el juego, se necesitan las posiciones del acelerador y el freno, así como el ángulo de giro del timón. A partir de las posiciones del acelerador y el freno, se calculan las revoluciones por minuto del motor, que generan un torque que actúa sobre los ejes del carro para alcanzar una velocidad deseada. Abstrayendo este proceso, se puede concluir que el objetivo de la aceleración y el frenado es alcanzar una velocidad deseada (que puede ser baja en las curvas y alta en las rectas).

### **3.6. Diseño de la RNA.**

La arquitectura de la RNA depende del problema, por lo que no existe un consenso entre los investigadores referente a la cantidad óptima de capas y neuronas por cada una de ellas (4). De acuerdo con esto, se experimentó con varias arquitecturas, comenzando por la más simple: una capa de entrada, que recibirá los parámetros de retroalimentación del entorno virtual; una oculta, en la que tendrá lugar la mayor parte del procesamiento de la red; y otra de salida, que decidirá cómo controlar el auto.

Es necesario determinar cuántas neuronas habrá en cada capa y qué representa cada una de ellas. Se trató de minimizar su cantidad para evitar que la red reciba datos incorrectos, ineficaces o redundantes; reduciendo además el procesamiento y aumentando la eficiencia. Esto se logra a partir de un análisis cuidadoso del contexto del problema en que se utilizará la red. Las neuronas de la capa de entrada se deducen del sistema sensorial descrito con anterioridad, que contiene la información necesaria y suficiente para producir valores

correctos en las neuronas de la capa de salida, con los que se controla el auto. Las neuronas de salida representan el ángulo de giro del timón y la velocidad deseada. Tratando de minimizar el número de variables de la red, en lugar de las posiciones del freno y el acelerador se prefirió emplear la velocidad deseada, porque como se explicó anteriormente, sintetiza su función, *ver capítulo 10 del informe final del proyecto*

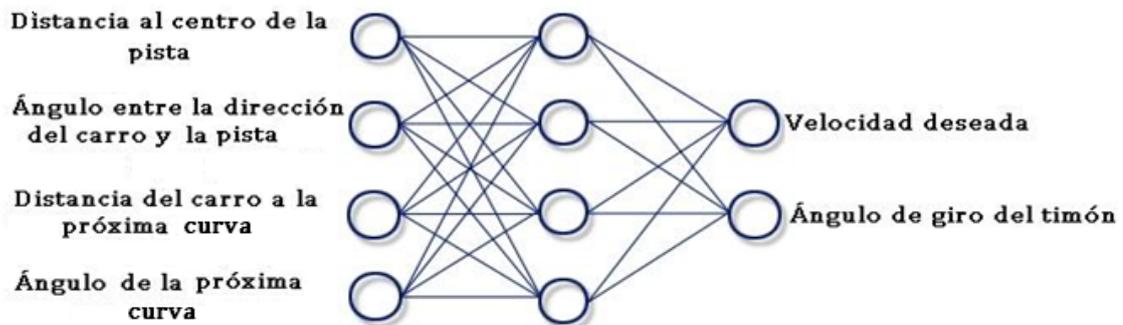


Figura 1: Modelo de Red utilizado.

### 3.7. Interpretación matemática de las variables empleadas.

Las redes neuronales artificiales trabajan con números reales, por lo que hay que traducir a este tipo de dato, los valores del dominio de las variables seleccionadas (booleanos, enumerados, valores continuos que pueden representar formatos de imágenes, música, entre otros) para que puedan ser comprendidas por la red.

**Entradas.**

Es recomendable que los valores de entrada a la red estén aproximadamente en el mismo orden de magnitud, para evitar que algunos tengan mayor impacto que otros sobre los pesos de la red. En caso contrario, durante el proceso de entrenamiento se escalan a un mismo rango, comúnmente  $[0; 1]$  ó  $[-1; 1]$ . Desde luego, una vez que esté funcionando, los datos que recibe como entrada tienen que ser escalados en la misma forma que se hizo durante el proceso de entrenamiento.

**3.7.1. Distancia al centro de la pista.**

La información que brinda esta variable es empleada por el neuro-controlador para evitar que el carro se salga de la pista.

Se calcula la distancia al centro de la pista como se explicó con anterioridad. Su valor puede ser positivo si está a la derecha del segmento más cercano, o negativo si está a la izquierda. Para conocer en qué lado de la pista se encuentra el carro, se emplea la *Ecuación 4* donde  $V_1$  y  $V_2$  son las posiciones de los nodos que definen el segmento más cercano a  $V_3$  (posición del carro).

$$\Theta = (V_1x - V_2x) * (V_3z - V_2z) - (V_1z - V_2z) * (V_3x - V_0x)$$

*Ecuación 4.*

Si  $\Theta$  es menor que cero,  $V_3$  está a la izquierda del segmento  $\overline{V_1V_2}$ , en caso contrario está a la derecha.

Los valores de esta variable se escalan al rango  $[-1, 1]$  que recibe como entrada el neuro-controlador. Para lograrlo, se compara la distancia al centro de la pis-

ta ( $DCP$ ) con el menor de los radios de los nodos que conforman el segmento más cercano ( $r$ ):

- Si es mayor que el 80 % del radio, se considera el auto está próximo al borde:  $DCP = \pm 1$
- Si es menor que el 30 % del radio, se considera el auto está en el centro:  $DCP = 0$
- Si no, toma un valor proporcional al radio:  $DCP = \pm \frac{DCP - (0,3r)}{(0,8r - 0,3r)}$



Figura 2: Clasificación de un segmento de pista.

### 3.7.2. Ángulo entre la dirección del carro y la pista.

El ángulo entre la dirección del carro y la pista puede tomar valores positivo o negativo, para lo que se utiliza la *Ecuación 4* donde  $\overline{V_1V_2}$  es el segmento que parte de la posición del carro y es paralelo al segmento más cercano; y  $V_3$  es el vector dirección del carro.

En dependencia del ángulo y el signo, el neuro-controlador determinará en qué medida hay que girar y hacia donde, para mantener la dirección del movimiento del carro en el sentido de la pista.

### **3.7.3. Ángulo de la próxima curva.**

Su valor puede ser positivo o negativo, para lo que se utiliza la *Ecuación 4*, donde  $s$  es el segmento actual; y  $V_3$  es el vector posición del nodo siguiente al siguiente.

La curva puede ser de riesgo, media o mínima; lo que es interpretado por el neuro-controlador a partir del ángulo, permitiéndole determinar la velocidad óptima deseada para tomar la curva.

### **3.7.4. Distancia del carro a la próxima curva.**

Se especifica un rango de visión para el agente autónomo, representado por un valor entero (40 en este caso). Si la distancia del carro a la próxima curva es menor que dicho valor, se considera que la curva definida por los segmentos actual y siguiente se encuentra en el campo de visión del agente, por lo que la puede “ver”.

Esta información es empleada por el neuro-controlador para determinar en qué momento comenzar a reducir o aumentar la velocidad.

**Salidas.**

Un aspecto crucial en el empleo de redes neuronales, es la interpretación que se hace de las salidas, cuyos valores están regidos por la función de activación. La función de activación utilizada para la red neuronal fue la función *Sigmoid*, por lo que los valores salida de cada neurona están en el rango  $[0 - 1]$ .

**3.7.5. Ángulo de giro del timón.**

Los valores arrojados por esta variable deben ser escalados al rango  $[-1, 1]$ , que son los admitidos por la interfaz que controla la dirección del auto en el juego. Esto se puede lograr mediante la *Ecuación*

$$x = 2 * (y - 5)$$

*Ecuación 5.*

Un valor negativo implica girar a la izquierda, uno positivo girar a la derecha y cero no girar.

**3.8. Velocidad deseada.**

Es necesario escalar la salida de esta variable al rango de valores de velocidad admisible por el carro  $[0, Velocidad Máxima]$ , por lo que se multiplica por el valor máximo de velocidad.

### 3.9. Proceso de entrenamiento.

El aprendizaje supervisado se ajusta perfectamente a los requerimientos de este proyecto, ya que se pueden utilizar ejemplos del modo de juego de una persona para que el neuro-controlador aprenda a desempeñarse. Se empleó una estrategia de entrenamiento offline.

Es básicamente un problema de regresión funcional, en el cual a partir de una correspondencia de valores de entrada y salida, se actualizan los pesos de la red mediante el algoritmo back-propagation. Como resultado se obtiene una red que es capaz de responder correctamente ante patrones de entrada desconocidos.

El conjunto de entrenamiento está compuesto por dos vectores: uno contiene los valores de las cuatro variables de entrada, que describen una instantánea del juego; y el otro está formado por los valores de velocidad y ángulo de timón deseados para reaccionar ante dicha situación.

Se utilizaron dos variantes para obtener los datos de adiestramiento: en una se generan automáticamente las entradas y luego se le ofrece la salida deseada correspondiente mediante una aplicación diseñada con ese objetivo; mientras que en la otra se obtienen los datos del comportamiento desempeñado por el jugador en una carrera.

Ambas formas de entrenar reportan utilidad, mediante la primera la red recibe ejemplos que cubren la mayor parte del dominio de cada variable, aumentando las probabilidades de generalización y dotando al neurocontrolador de un conocimiento básico, *ver capítulo 10 del informe final del proyecto*. Cuando se entrena usando la segunda variante, el comportamiento estará más adaptado a las características del jugador. Combinando ambas, el agente autónomo co-

mienza la carrera con el conocimiento básico que le proveen los datos generados automáticamente. A medida que avanza la competencia, se va personalizando su comportamiento para las condiciones específicas de cada pista según el perfil del jugador; aumentando de esta manera el nivel de dificultad y planteándole siempre un reto aceptable al usuario y con él la vida útil del sistema.

## 4. Conclusiones

En esta investigación se obtuvo un carro autónomo capaz de recorrer la trayectoria de la pista, respondiendo correctamente ante situaciones desconocidas. Se demostró que la utilización del híbrido mejora el realismo del videojuego incorporando capacidad de aprendizaje a los elementos controlados por la computadora, permitiéndoles adaptarse a nuevas situaciones.

Se implementaron 2 formas para obtener los datos de entrenamiento de la red neuronal artificial, mediante la primera la red recibe ejemplos que cubren la mayor parte del dominio de cada variable, aumentando las probabilidades de generalización, mientras que en la segunda los datos de entrenamiento están más adaptados a las características del jugador. La combinación de las 2 contribuye a optimizar el proceso de aprendizaje de la misma.

## A. Glosario de términos

**Agente autónomo:** Sistema situado en un entorno, que tiene la capacidad de sentirlo y actuar sobre él, a través del tiempo, persiguiendo sus propios objetivos de forma que afecte lo que siente en el futuro.

**Algoritmo de entrenamiento:** Proceso que le permite a una red neuronal artificial aprender de un conjunto de patrones.

**A-Life (Vida Artificial):** Sistemas multi-agentes que intentan aplicar algunas de propiedades universales de los sistemas vivientes a agentes inteligentes en un entorno virtual.

**Aprendizaje:** Proceso mediante el cual un individuo adquiere conocimiento y experiencia, de forma que le permita modificar su comportamiento y adaptarlo a nuevas condiciones de su entorno.

**Comportamientos de locomoción:** Procedimientos reactivos que toman como entrada información local acerca del entorno produciendo como salida un vector velocidad.

**Elemento inteligente:** Individuo auto-controlado, capaz interactuar con su entorno y conseguir metas.

**Inteligencia artificial:** Rama de la informática que se encarga de crear software y hardware capaces de imitar la inteligencia humana.

**Máquina de estados finitos:** máquinas abstractas que pueden existir en uno o muchos estados diferentes y predefinidos.

**Perceptrón multicapa:** Tipo de red neuronal artificial formada por múltiples capas de perceptrones.

**Red neuronal artificial:** Técnica no determinista que emula el cerebro humano. Está compuesto por un conjunto de neuronas interconectadas que colaboran entre sí para obtener una salida.

**Videojuego:** Programa informático normalmente asociado a un hardware específico, que recrea un ejercicio sometido a reglas, se debe lograr uno o varios objetivos, donde los jugadores pueden interactuar y tomar decisiones.

## Referencias

- [1] <http://aigamedev.com/>, 2007.
- [2] Omar Baqueiro Espinosa Alfredo González González and Emmanuel Meza Cota. Metodologías de la i.a. (agentes autónomos y redes neuronales supervisadas) aplicadas a npcs (non player characters). <http://www.redcientifica.com/doc/doc200401210112.html>, 2004.
- [3] Rafael Bello Pérez. *Curso Introductorio a las Redes Neuronales Artificiales*. 1993.
- [4] Anna Booth. Using neural networks to improve behavioural realism in driving simulation scenarios, 2007.
- [5] David Bourg and Glenn Seeman. *AI for Game Developers*. O'Reilly, 2004.
- [6] Mat Buckland. *AI Techniques for games programming*. Course Technology, 2002.
- [7] Mat Buckland. *Programming Game AI by Example*. Wordware Publishing, 2005.
- [8] Colectivo de autores. *AI game Programming Wisdom*. 2002.
- [9] Faraón Llorens Mar Pujol Ramón Rizo Francisco Gallego. Driving-bots with a neuroevolved brain: Screaming racers., 2005.
- [10] Abdullah Çavusoglu Haldun Göktas and Sen Baha. The use of artificial neural networks in simulation of mobile ground vehicles, 2007.
- [11] Jeff Hannan. [http://www.generation5.org/content/2001/hannan.asp.](http://www.generation5.org/content/2001/hannan.asp), 2001.

- [12] Alexandros Agapitos Simon Lucas Julian Togelius. Evolving controllers for simulated car racing., 2005.
- [13] Simon Lucas Julian Togelius and Pete Burrow. Multi-population competitive co-evolution of car racing controllers, 2007.
- [14] Juan Rabuñal and Julián Dorado. *Artificial Neural Networks in Real Life Applications*. Idea Group., 2005.
- [15] Dalián Sánchez Cruz and Alexey Herrera Pérez. Comportamientos de autos en pistas de carreras basados en steering behaviors para el videojuego rápido y curioso., 2008.
- [16] Julian Togelius. [3http://www.togelius.blogspot.com/2007/08/how-better-ai-can-make-racing-games.html](http://www.togelius.blogspot.com/2007/08/how-better-ai-can-make-racing-games.html).
- [17] Julian Togelius. <http://julian.togelius.com/Togelius2007Towards.pdf>, 2007.
- [18] Julian Togelius and Simon Lucas. Arms races and car races., 2006.

---

Fecha de impresión: 10 de junio de 2010

Generado con L<sup>A</sup>T<sub>E</sub>X