

Optimización en videojuegos: retos para la comunidad científica

Raúl Lara-Cabrera, Mariela Nogueira, Carlos Cotta, Antonio J. Fernández-Leiva

Resumen— Este artículo analiza algunos de los desafíos más interesantes a los cuales los miembros de la comunidad MAEB pueden enfrentarse en el área de la aplicación de técnicas de Inteligencia Artificial/Computacional al diseño y creación de videojuegos. El artículo se centra en tres líneas, que en un futuro cercano, seguramente van a influenciar de forma significativa la industria del desarrollo de videojuegos, en concreto se enfoca en la Generación Automática de Contenido, en la Computación Afectiva aplicada a los videojuegos y en la Generación de Comportamientos que gestionen la toma de decisiones de las entidades no controladas por el jugador humano.

Palabras clave— Videojuegos, Inteligencia Computacional, Optimización, Generación de Contenido, Computación Afectiva, Comportamientos Inteligentes.

I. INTRODUCCIÓN

La industria del videojuego ha tomado una posición dominante dentro de la industria del entretenimiento, con un gasto total por parte de los clientes cifrado en casi 25 mil millones de dólares en 2011 [1] y unos ingresos en 2013 estimados en 70 mil millones de dólares (lo cual representa un incremento anual de un 6%), según el informe de Newzoo [2] sobre el mercado global de los videojuegos. Además, se prevé que el número total de jugadores supere los 1200 millones a final de año. Esta situación ha supuesto una motivación para la investigación aplicada a los videojuegos, la cual ha estado adquiriendo notoriedad a lo largo de estos últimos años, abarcando varias áreas como la psicología y la satisfacción del jugador, el marketing y la gamificación, la inteligencia artificial, los gráficos por computador e incluso la educación y la salud (juegos serios). Actualmente están emergiendo nuevos retos y objetivos dentro del área de los videojuegos, especialmente en el campo de la inteligencia artificial (IA) [3].

Tras realizar un análisis sobre la red de co-autoría dentro del campo de la inteligencia computacional en videojuegos [4], llegamos a la conclusión de que nos encontramos ante un campo activo y vibrante, el cual atrae a nuevos investigadores año a año y genera nuevas publicaciones. Se ha constatado un crecimiento estable en el número de autores, el cual se acentúa a mediados de la década 2000-2010 (ver Figura 1). El número de publicaciones por año de la comunidad ha estado incrementándose desde 2005, apoyando de esta forma el constante crecimiento de

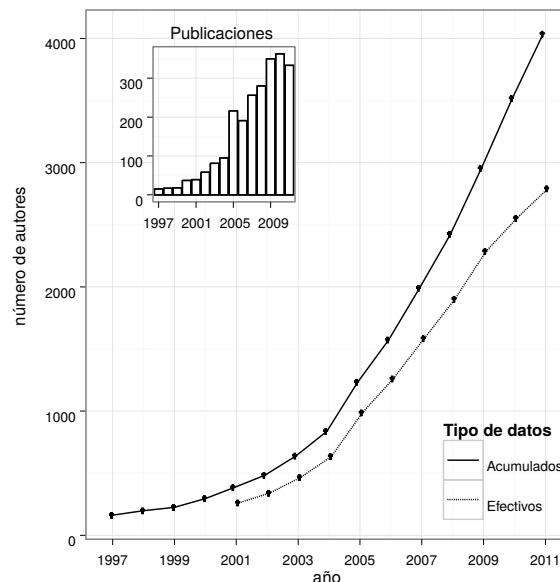


Fig. 1. Evolución del número de autores y publicaciones.

la comunidad. Hemos observado que, además de existir un flujo de entrada constante de nuevos autores, hay otro elevado número de ellos que abandonan este campo de investigación, lo cual podría indicar una breve desaceleración de la comunidad. Sin embargo, la tendencia real de la volatilidad de la comunidad, es decir, el número de autores que la abandonan, es muy similar al número de autores nuevos que entran, por lo que el crecimiento de la misma se mantiene constante. La comunidad de la inteligencia computacional en videojuegos se encuentra en un estado inicial de desarrollo, formando enlaces y mejorando su cohesión, pero no llega a estar completamente formada. De la misma forma, la industria está empezando a adoptar las técnicas y recomendaciones que el mundo académico les ofrece. El lector interesado en el estado actual de las técnicas de inteligencia artificial dentro de la industria puede consultar recursos como el sitio web AiGameDev¹, el *AI Summit* de la *Game Developers Conference*, las colecciones de libros *AI Game Programming Wisdom* y *Game Programming Gems*, o el libro de Ian Millington y John Funge [5].

Los videojuegos ofrecen una gran variedad de problemas fundamentales susceptibles de ser usados en el ámbito de la investigación en inteligencia artificial, tales como la planificación de tareas en tiempo real y

Dpto. Lenguajes y Ciencias de la Computación. Universidad de Málaga.
E-mails: {raul,mnogueira,ccottap,afdez}@lcc.uma.es .

¹<http://aigamedev.com>

teniendo en cuenta al adversario y toma de decisiones con incertidumbre, entre otros. Por ejemplo, sólo los juegos de estrategia en tiempo real, que suponen un género dentro de los videojuegos, han servido como entornos de prueba e investigación para nuevas técnicas de inteligencia artificial, como puede comprobarse en nuestro estudio sobre juegos de estrategia en tiempo real e investigación en inteligencia artificial [6].

Este trabajo tiene como objetivo enfocar la atención en algunas de las principales tendencias que parecen guiar el futuro de los videojuegos, y de los retos que éstas le imponen a la comunidad científica, centrándonos en la perspectiva de la aplicación de técnicas de inteligencia artificial y, en particular, de la inteligencia computacional, entendida esta como el conjunto de técnicas de optimización bio-inspiradas y metaheurísticas [3]. Queremos matizar que el universo de aplicaciones de las técnicas de optimización sobre el desarrollo y diseño de videojuegos es enormemente amplio y no pretendemos realizar un recorrido exhaustivo sobre el mismo en este artículo; de hecho recomendamos al lector interesado en obtener más información, la lectura de otros artículos que ya han sido publicados en la literatura y que sirven como base para conocer el estado del arte del este área, tales como [7], [8]. Nosotros nos centramos en determinadas áreas de investigación que creemos que van a influenciar significativamente la creación de juegos comerciales en la próxima década; en concreto nos referimos a la generación automática de contenido (PCG, por sus siglas en inglés)², a la Computación Afectiva, lo cual afecta directamente a la satisfacción del jugador, y a la generación de comportamientos o estrategias de toma de decisiones asociados a los jugadores virtuales (es decir, no-humanos) o a los llamados NPCs (Non-player characters, es decir, las entidades del juegos no controladas por el jugador humano). Adicionalmente, y para incentivar el trabajo en este área a los miembros de la comunidad de investigadores en técnicas metaheurísticas y técnicas bio-inspiradas, hemos decidido añadir unas secciones en las cuales se presentan herramientas y entornos que pueden servir para aplicar nuevos métodos de inteligencia computacional y que pueden emplearse como marco de prueba para analizar y comparar la eficacia de algoritmos novedosos. De nuevo esta sección mencionada no presenta una lista exhaustiva, lo cual está más allá del objetivo de este trabajo.

Este artículo se ha organizado como sigue. La Sección II muestra un análisis sobre el tema de Generación Automática de Contenidos, luego se presenta un resumen del estado actual de los Videojuegos Afectivos y su relación con el Modelado e Incremento de la Satisfacción del jugador. Le sigue la Sección IV que aborda la programación de Comportamien-

²PCG se corresponde con “Procedural Content Generation”, lo cual se ha traducido literalmente como “Generación Procedimental de Contenido”, lo cual creemos menos general que el término que nosotros usamos aquí.

tos Inteligentes, y después se explican algunos de los principales marcos de trabajos que existen para apoyar las investigaciones relacionadas con IA aplicada a videojuegos. Finalmente se exponen las conclusiones en la Sección VI.

II. GENERACIÓN AUTOMÁTICA DE CONTENIDO

La generación automática de contenido, o generación por procedimientos, se refiere a la creación algorítmica de contenido para videojuegos, ya sea con intervención humana como sin ella, tales como mapas, niveles, texturas, personajes, reglas y misiones, pero excluyendo el comportamiento de los personajes no jugables y el motor de juego en sí mismo. Puede tratarse desde contenido totalmente necesario para avanzar en el desarrollo del videojuego hasta aquel que es puramente decorativo. Eso sí, los algoritmos utilizados deben garantizar la generación de contenido que cumpla ciertos requisitos de calidad, adaptándose al tipo de contenido que se pretendiese crear manualmente.

El hecho de producir contenido de videojuegos mediante algoritmos PCG puede permitirnos reducir sustancialmente el consumo de memoria del juego, algo que hoy en día puede tener un carácter secundario pero que en el pasado motivó el empleo de estas técnicas. Otro motivo importante para el uso de estos algoritmos es reducir el elevado coste que tiene en muchos casos la generación de cierto contenido de los juegos de forma manual e incluso servir como fuente de creatividad a los diseñadores, proponiendo diseños que posiblemente no se les hubiesen ocurrido en primera instancia. Además, si como se ha dicho antes se asegura que el contenido generado por PCG cumple con ciertos criterios, como ajustarse a la habilidad de un jugador, el nuevo contenido puede suponerle un reto constante. Si esta generación de contenido adaptado resulta siempre diversa y se genera en tiempo real, es decir, al mismo tiempo que el juego se ejecuta, pueden conseguirse auténticos juegos infinitos, presentando constantemente a cada tipo de jugador nuevos y distintos retos que ha de superar. Estos beneficios son bien conocidos por la industria, habiéndose usado este tipo de técnicas en el desarrollo de videojuegos comerciales de éxito como la saga *Borderlands*, *Skyrim*, *Minecraft* o *Terraria*.

A la hora de generar contenido automático para videojuegos pueden hacerse muchas distinciones globales en lo que respecta a los procedimientos a seguir. Atendiendo al momento en el que se produce la creación de los contenidos, la generación puede ser tanto online, durante la ejecución del juego (con las ventajas que esto puede tener), como offline, durante la fase de desarrollo del mismo.

Con respecto a la finalidad del contenido generado, éste puede ser considerado como necesario para poder progresar en el juego, en cuyo caso se ha de asegurar que es correcto (en el sentido de no propo-

ner objetivos imposibles al jugador), o bien opcional, como pueden ser los elementos decorativos y que por lo general pueden ser mucho menos restrictivos.

También cabe preguntarse por la naturaleza del algoritmo de generación, es decir, si nos encontramos ante un algoritmo puramente estocástico, mediante el cual se crea el contenido a partir de semillas aleatorias, o, por el contrario, un algoritmo determinista, donde el contenido se genera mediante un vector de parámetros. La tercera posibilidad radica en la hibridación de ambas perspectivas, diseñando un algoritmo con una componente estocástica y otra determinista, funcionando conjuntamente.

Si nos fijamos en los objetivos que se pretenden cumplir, el proceso de creación puede hacerse de forma constructiva, asegurando la validez del contenido durante todo el proceso. La otra opción consiste en seguir un esquema de generación y prueba, donde se genera una gran cantidad de contenido el cual se pasa por una fase de validación y posterior descarte de todo aquel que no cumpla con las restricciones. Este último esquema es el más empleado actualmente por la comunidad, y se basa en la búsqueda de los contenidos en el espacio de posibles soluciones. La validación se realiza asignando uno o varios valores al contenido de forma que quede cuantificada el nivel de calidad, según nuestros objetivos, del contenido creado.

Estas técnicas se utilizan comúnmente para generar mapas y niveles, como demuestra el elevado número de artículos dedicados a ello [9]. Por ejemplo, Mahlmann et al. [10] presentan un generador de mapas para una versión simplificada del juego de estrategia Dune 2, el cual se basa en la transformación de matrices de baja resolución en mapas de mayor resolución utilizando para ello autómatas celulares. Frade et al. [11], [12], [13] introducen el uso de programación genética para evolucionar mapas de videojuegos (denominado por los autores como programación de terrenos), usando tanto la evaluación subjetiva de los humanos como medidas de calidad extraídas de forma automática como la accesibilidad de los mapas o la longitud de aristas. La evaluación humana del contenido generado por procedimientos es tratada por Liapis et al. en [14]. Togelius et al. [15] han diseñado un sistema capaz de generar circuitos de carreras a partir de un vector de parámetros usando una transformación genotipo-fenotipo determinista. Un artículo de Ashlock y McGuinness [16] introduce un autómata que busca en el espacio de soluciones posibles de mapas de altura usando un algoritmo evolutivo. En este caso, el diseñador es capaz de definir puntos del mapa que deben ser mutuamente accesibles para influir en la morfología de los mapas generados. Otro generador de mapas basado en un algoritmo evolutivo se define en [17], pero en este caso la función de aptitud depende de la diferencia entre distintas curvas de dificultad definidas por el diseñador para cada nivel. Como se puede compro-

bar, existe un gran número de artículos dedicados a la generación de mapas y niveles, los cuales tienen algo en común: sus algoritmos buscan cumplir varias restricciones relacionadas con la mecánica del juego, tales como la accesibilidad de ciertas zonas de los niveles, el ajuste de la dificultad o el equilibrio entre el nivel de los jugadores. De esta forma, se incrementa la satisfacción del jugador que, a fin de cuentas, es el objetivo principal de los videojuegos.

Además de mapas y niveles, la generación por procedimientos se utiliza para otro tipo de contenido. Por ejemplo, Font et al. [18] describen los primeros pasos hacia un sistema capaz de generar juegos de cartas, siendo en este caso las reglas del juego lo que se genera de forma automática. Collins [19] hace una introducción a la generación procedimental de música para videojuegos, estudiando las diferentes estrategias de composición y control que se han usado anteriormente. Otros autores [20] han creado un prototipo de una herramienta que crea patrones de diseño para misiones del juego de rol *Neverwinter Nights*. Por su parte, García-Ortega et al., [21] utilizan algoritmos genéticos para la generación automática de historias interesantes en videojuegos.

III. COMPUTACIÓN AFECTIVA Y SATISFACCIÓN DEL JUGADOR

Fue Rosalind Picard quien, en 1995, introdujo el término Computación Afectiva y lo define como el cómputo que relaciona, surge o influye en las emociones [22]. En el contexto de los videojuegos aún se investiga en cómo extrapolar el inmenso campo de las emociones hasta el escenario de un juego, las buenas razones de por qué hacerlo están claras [23], pero los resultados obtenidos hasta el momento son modestos comparado con todo lo que se pretende lograr.

Una de las primeras formas usadas para incorporar emociones en los juegos fue a través de la narrativa, mediante la generación de situaciones que atraen al jugador ya sea por los personajes, la presentación de todo tipo de conflictos e historias fantásticas o de la vida real. En esta caracterización destacan entre las sagas preferidas por los usuarios *Final Fantasy* y *Resident Evil*. También están los videojuegos como *Fable* o *Beyond: Two Souls* que se distinguen por tener un alto grado de realismo en las simulaciones e incorporan emociones en algunas de las actuaciones del protagonista, véase en la Figura 2 una captura de pantalla de estos juegos donde se observa al personaje mostrando emociones. Este tipo de *afectividad centrada en el personaje principal* lleva detrás un meritorio trabajo artístico que permite la simulación realista de las emociones, por ejemplo en el juego *Beyond: Two Souls* se utilizan técnicas de captura de movimiento para animar al personaje virtual y son actores reales (estrellas de cine como Ellen Page y Willem Dafoe) los que interpretan a los protagonistas de la historia. Este juego en particu-

lar tiene un peso narrativo muy grande pues en todo momento el jugador es guiado en la partida por un drama que ha sido predefinido, con lo cual se acotan considerablemente las posibles acciones que puede elegir el usuario mientras juega, y esto hace que sea más fácil para el software controlar el flujo emocional del protagonista ya que los estados emocionales por los que transita son en su mayoría predecibles. En general, este enfoque para la implementación de la afectividad tiene como objetivo principal incrementar la inmersión del usuario en el juego y para eso establece una dinámica emocional entre el jugador humano y el personaje principal; tratando de lograr en todo momento que el jugador se sienta identificado con lo que le transmite su personaje.

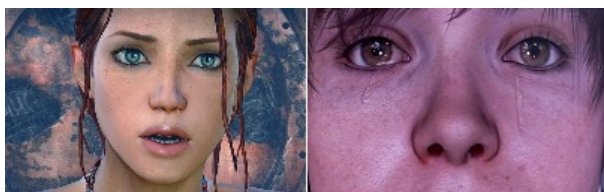


Fig. 2. Personajes de los juegos Fable y Beyond: Two Souls

Esa forma de transmitir afectividad al jugador mediante su personaje virtual es una de las tendencias usadas por los juegos comerciales actuales, suele estar relacionada con los géneros de acción-aventura y los juegos basados en roles donde es típico que el jugador tenga total visibilidad del personaje principal casi todo el tiempo. Además de este, existe otro enfoque cuyo objetivo es que los *bots* que participan en la partida se comporten como individuos emocionales y, por tanto, sus emociones influyan cuando él tome decisiones en la partida. Para lograr esto se hace necesario la existencia de un alto grado de percepción para estos *bots* pues ellos tendrían que reaccionar ante todo suceso que ocurra a su alrededor que pudiera afectar su estado emocional, por ejemplo, si se provoca un ruido cerca del *bot* hay que evaluar si ese sonido podría afectar su estado de ánimo (asustarle, alarmarle, alegrarle, etc.). Precisamente este aumento en el nivel de percepción del entorno demanda un proceso de toma de decisiones más complejo, que implica encontrar una estrategia de juego que dote a esos *bots* de inteligencia para que tomen las decisiones lógicas correctas y a su vez tenga en cuenta sus emociones. Esta pretensión de simular de forma realista el comportamiento humano en *bots* es desde hace mucho un campo de investigación abierto, dentro de él está la rama que se centra en imitar la inteligencia humana y es la que más resultados tiene, muestra de ello son los trabajos referenciados en la Sección IV. Y por otra parte, está la línea que se dedica al estudio de cómo incorporar las emociones a la simulación, en la cual también se han desarrollado propuestas muy interesantes que ofrecen técnicas y modelos para implementar comportamientos emocionales en los agentes virtuales, ejemplos de algunos

de ellos son [24], [25] y [26], sin embargo, los mejores resultados están aún por verse y se sigue echando en falta el factor emocional en la Inteligencia Artificial que exhiben la mayoría de los videojuegos comerciales.

Los dos enfoques afectivos analizados hasta este momento manejan la afectividad desde los jugadores virtuales (ya sean *bots* o personajes controlados por el jugador), centrándose en lo que ocurre dentro del videojuego y en cómo deben reaccionar los agentes virtuales emotivos; sin tener en cuenta las reacciones emocionales que expresa el usuario mientras juega las cuales son necesarias para lograr una verdadera interacción afectiva entre el videojuego y el jugador. De aquí que exista un tercer enfoque, el de la *Auto-adaptabilidad* en videojuegos afectivos, que está estrechamente relacionado con la temática de *Modelado, Evaluación e Incremento de la satisfacción del jugador* [27], [28]. La *Auto adaptabilidad* se refiere a la capacidad que puede tener un juego de auto-adaptar la partida a cada jugador, teniendo en cuenta sus preferencias y características, con el objetivo de hacer que esta resulte una experiencia única para cada usuario usuario e influya directamente en el aumento de su satisfacción. En el contexto de la afectividad, se trataría de auto-ajustar el juego según las emociones que va expresando el usuario durante la partida, haciendo que los lazos afectivos entre humano y personajes virtuales se construyan y expresen en tiempo real. Este sería el enfoque más completo, que incluye también a los dos anteriores, y que actualmente es un anhelo para muchos investigadores. Todavía no existe un videojuego que permita experimentar un nivel de interacción emocional tan alto. Los resultados más interesantes relacionados con este tema son los que se han obtenido en el campo del *Modelado, Evaluación e Incremento de la satisfacción del jugador*, los cuales pueden considerarse parte de la computación afectiva en videojuegos pues el concepto de satisfacción del jugador está fuertemente ligado a las emociones humanas. Los trabajos más interesantes en esta rama se centran en su mayoría en crear un modelo formal que represente el comportamiento del jugador y que permita hacer una evaluación del nivel de satisfacción del mismo, todo esto basado en investigaciones psicológicas sobre la satisfacción [29], [30], [31]; luego viene la parte de usar dicho modelo para determinar qué grado de satisfacción experimenta el jugador y proceder entonces a reajustar el juego para mantener o elevar ese grado. Algunas propuestas que han tenido éxito se puede ver en [32], [33], [34] y [35].

Cada enfoque de los mencionados aquí es un campo abierto de investigación, que tiene dentro varias líneas que también demandan nuevas soluciones, y la optimización es una de ellas. La mayoría de las propuestas exitosas que existen, algunas de las cuales han sido citadas aquí, llevan detrás un proceso de búsqueda y optimización basado en metaheurísticas,

para poder explorar el amplio espacio de búsqueda que se genera de dos contextos intrínsecamente complejos: los videojuegos y las emociones.

IV. COMPORTAMIENTOS

Tradicionalmente la Inteligencia Artificial (IA) de un videojuego ha sido codificada manualmente mediante sistemas predefinidos de reglas dando lugar a menudo a comportamientos englobados dentro de la llamada *estupidez artificial* [36] lo que resulta en un conjunto de inconvenientes conocidos, tales como por ejemplo la sensación de irrealidad, la aparición de comportamientos anormales en situaciones no consideradas o el conocimiento de comportamientos predecibles. Actualmente se emplean técnicas avanzadas para solventar estos problemas y se logran *bots* con comportamientos racionales que toman decisiones lógicas correctas de la misma forma que lo hace un jugador humano. La principal ventaja que ofrecen estas técnicas es que realizan de forma automática el proceso de búsqueda y optimización de esas estrategias de juegos “inteligentes”.



Fig. 3. Captura de pantalla del juego NERO

Los algoritmos bioinspirados son la base de muchos de esos métodos avanzados, pues resultan un enfoque muy apropiado en este sentido, ya que son capaces de producir soluciones de gran complejidad como resultado emergente del proceso de optimización, y su capacidad adaptativa les permite incorporar información proporcionada por el usuario y explotarla de manera no anticipada por el mismo. De aquí que existan muchas propuestas exitosas que siguen esta línea. Por ejemplo, la Coevolución [37] es una de las técnicas basadas en la aplicación de métodos heurísticos, inspirada por los principios de la evolución natural que ha sido ampliamente utilizada en la programación de IA para videojuegos. En [38] se describe una investigación que logra evolucionar mediante coevolución competitiva la morfología y el comportamiento de criaturas virtuales que interactúan en un entorno depredador/presa, otras propuestas interesantes fueron presentadas en [39],

[40], [41] y [42] donde se coevolucionaron estrategias de juego con el objetivo de obtener *bots* competitivos para el Tempo que es un videojuego bélico.

El aprendizaje computacional también se utiliza a la hora de modelar el comportamiento de los jugadores artificiales. A continuación se destacan algunos ejemplos que pertenecen al estado del arte: los autores de [43] han empleado mapas auto-organizados para mejorar el manejo de los batallones de unidades en un juego de estrategia en tiempo real; mediante el análisis de los datos obtenidos por los sensores, los autores de [44] han diseñado un algoritmo para que un piloto artificial se aprenda el circuito en el que posteriormente tiene que pilotar de forma autónoma. También se pueden encontrar ejemplos en los que se combinan redes neuronales con los algoritmos bioinspirados descritos anteriormente, como en el caso de [45], artículo en el cual se presenta un sistema para la evolución en tiempo real de redes neuronales con el objetivo de mejorar los comportamientos de los agentes artificiales del juego NERO (ver Figura 3). Otro posible uso de las técnicas de aprendizaje computacional, en concreto la minería de datos, se puede encontrar en [46], donde se construye un predictor de posibles estrategias tras someter a una base de datos de partidas de Starcraft a un proceso de análisis de datos. En [47], los autores extraen características de los mapas de un juego de estrategia en tiempo real las cuales influyen en el comportamiento de un jugador artificial, que se aprovecha de esta caracterización para obtener una ventaja con respecto a su oponente.

A. Competiciones

A lo largo de los últimos años han ido apareciendo diferentes competiciones donde los investigadores tienen la oportunidad de comparar sus estrategias y algoritmos en escenarios y juegos específicos. A continuación se detallan algunas de las más importantes junto con una breve descripción:

- **BotPrize³**: Competición consistente en desarrollar un jugador artificial para *Unreal Tournament* que engañe al resto de jugadores humanos haciéndoles creer que él también es humano
- **Starcraft AI Competition⁴**: Competición anual de jugadores artificiales para *Starcraft* cuyo objetivo es ser el mejor jugador y derrotar al resto de oponentes
- **Simulated Car Racing Competition⁵**: En esta ocasión el objetivo es diseñar un piloto artificial de carreras que competirá en un conjunto de grandes premios contra otros pilotos artificiales
- **GVG-AI⁶**: La *General Video Game AI Competition* es una competición donde las estrategias de inteligencia artificial, o controladores, deben ser ca-

³<http://botprize.org/>

⁴<http://webdocs.cs.ualberta.ca/~cdavid/starcraftaicomp/>

⁵<http://cig.dei.polimi.it/>

⁶<http://www.gvgai.net/>

paz de jugar a varios tipos de juegos y escenarios, intentando ser lo más genérico posible

Existe un problema con estas competiciones: los retos que se presentan son muy específicos y estrechamente ligados al juego sobre el cual se monta la competición. De esta forma, las estrategias de inteligencia artificial ganadoras se sobre-especializan en explotar las características del juego en sí, pero arrojando un bajo rendimiento cuando se usa ésta estrategia en otro juego. Por tanto, otro posible reto sería diseñar competiciones de estrategias de inteligencia artificial genéricas, que sean capaz de dar buenos resultados no solo en un único juego o entorno, sino en varios de ellos, algo que ya se está empezando a hacer en la competición *GVG-AI* previamente reseñada.

V. MARCOS DE TRABAJO

Esta sección está dedicada a las herramientas o marcos de trabajo que la comunidad científica tiene a su disposición para las pruebas y validación de los resultados obtenidos durante la investigación. Hoy en día existen multitud de herramientas de libre disposición, por lo que a continuación se presenta un compendio de las más usadas junto a sus características principales, para servir como lista de referencia a investigadores del campo de la inteligencia artificial y los videojuegos.



Fig. 4. Open Real-Time Strategy

La herramienta ORTS (Open Real-Time Strategy) [48] es un juego de estrategia en tiempo real diseñado específicamente para su uso en la investigación y liberado bajo la licencia GPL (GNU public license). Su protocolo abierto de intercambio de mensajes y el programa cliente permiten a los investigadores analizar el rendimiento de sus algoritmos jugando partidas en un entorno seguro donde la simulación se lleva a cabo en el lado del servidor. Este marco de trabajo proporciona la funcionalidad básica de un juego de este género, la cual puede ser ampliada con facilidad. Otro juego de estrategia en tiempo real muy popular en este campo de investigación es el Starcraft ⁷ el cual a pesar de no ser software libre, cuenta con

⁷<http://us.blizzard.com/en-us/games/sc/>.

una biblioteca (BWAPI⁸) que facilita la conexión del motor del juego con las posibles estrategias de inteligencia artificial. Por su parte, RoboCode⁹ es una plataforma cuyo objetivo es desarrollar (usando Java o .NET) un robot de combate para luchar contra otros robots similares en tiempo real

Planet Wars¹⁰ y ANTS¹¹ son dos juegos desarrollados en el marco de la competición de inteligencia artificial organizada por Google en sus ediciones de 2010 y 2011, respectivamente. El primero es un juego de conquista espacial para varios jugadores donde el objetivo es conquistar todos los planetas del mapa, mientras que el segundo es un juego también para varios jugadores donde cada uno de ellos representa a un conjunto de hormigas cuyo objetivo es recoger comida y conquistar los hormigueros del resto de jugadores, en unos mapas que incluyen zonas no transitables.

En Vindinium¹² se ofrece la posibilidad de tomar el control de un héroe que luchará contra otros jugadores por el control de unas minas de oro durante un número determinado de turnos. El jugador con mayor cantidad de oro al final de la partida se convierte en el ganador. Lo más destacable de este marco de trabajo para estrategias de inteligencia artificial, además de su aspecto visual, es el elevado número de lenguajes de programación que soporta, pudiendo programar las estrategias en más de 25 lenguajes distintos.

Eryna¹³ [49] es otra herramienta creada para apoyar la investigación sobre IA aplicada a juegos. Ofrece un juego de estrategia en tiempo real y multijugador, que le permite al usuario lanzar partidas entre varios *bots* y evaluar los resultados. Sus componentes fundamentales son: el motor de juego que ha sido desarrollado en java y está basado en una arquitectura de servidor autoritario que maneja de forma concurrente varias conexiones y procesos; un módulo de IA que es completamente editable y le permite al investigador programar sus propios *bots*; y otro de Generación Automática de Contenidos que ofrece la posibilidad de generar nuevos escenarios. La principal ventaja que ofrece esta herramienta es que es muy configurable, el investigador tiene a su alcance la combinación de muchos parámetros que dan lugar a partidas diferentes sobre un mismo escenario de juego.

Por su parte, SpelunkBots¹⁴ es un conjunto de herramientas en C++ construidas sobre el código fuente original del juego de plataformas *Spelunky*, que permite a los investigadores desarrollar estrategias de IA para determinar el comportamiento del personaje principal a lo largo de las diferentes pantallas

⁸<http://bwapi.github.io>

⁹<http://robocode.sourceforge.net/>

¹⁰<http://planetwars.aichallenge.org/>

¹¹<http://ants.aichallenge.org/>

¹²<http://vindinium.org/>

¹³<http://eryna.lcc.uma.es/>

¹⁴<http://t2thompson.com/projects/spelunkbots/>

del juego. La herramienta ha sido desarrollada por Daniel Scales [50].

VI. CONCLUSIONES

Cada enfoque de los mencionados aquí es un campo abierto de investigación, que tiene dentro varias líneas que también demandan nuevas soluciones, y la optimización es una de ellas. La mayoría de las propuestas exitosas que existen, algunas de las cuales han sido citadas aquí, llevan detrás un proceso de búsqueda y optimización basado en metaheurísticas, para poder explorar el amplio espacio de búsqueda que se genera de dos contextos intrínsecamente complejos: los videojuegos y las emociones [8].

Los retos que se presentan en las líneas de investigación que hemos mencionado a lo largo de este artículo son enormes y seguramente afectarán a otras áreas más allá del reino de los videojuegos. Por ejemplo, la generación de comportamientos cuasi-humanos es algo que ya se investiga más allá y que tradicionalmente tienen su germen en el conocido “test de Turing”, y las posibilidades que se abren al aplicar ciencia a los videojuegos son inmensas; si a ello añadimos, por un lado, la integración de sentimientos a los jugadores virtuales y a otras entidades del juego que guien sus comportamientos, y la posibilidad de construir un canal directo entre estos y la percepción sentimental del jugador humano, a través de la llamada Computación Afectiva, pues el desafío aumenta.

En el ámbito de la generación automática de contenido, se ha comprobado que es un campo de creciente interés para la comunidad científica, con un elevado número de publicaciones relacionadas con este tema. Pero no sólo es atractivo para los científicos, ya que la industria del videojuego está incorporando de forma satisfactoria muchos de los avances que se consiguen en el ámbito científico. No obstante, y según queda patente en el presente trabajo, quedan muchos retos abiertos todavía en la generación automática de contenido.

Terminamos este artículo mencionando que existen multitud de áreas asociadas con el empleo de técnicas de Inteligencia Artificial/Computacional, no tratadas de forma específica en este artículo, que suponen otros desafíos para los investigadores y entre las cuales podemos mencionar algunas tales como el modelado del jugador, el proceso de aprendizaje, la definición de lenguajes de modelado para la descripción de videojuegos, métodos de narrativa computacional, creación de jugadores virtuales “creíbles”, diseño de juegos asistidos por IA, y un largo etc. Digamos que nos enfrentamos a retos realmente estimulantes, no sólo para un futuro cercano sino para el actual presente.

AGRADECIMIENTOS

Este trabajo está parcialmente financiado por la Junta de Andalucía dentro del proyecto P10-TIC-

6083 (DNEMESIS¹⁵), por el MICINN dentro del proyecto ANYSELF¹⁶, y la Universidad de Málaga. Campus de Excelencia Internacional Andalucía Tech.

REFERENCIAS

- [1] Entertainment Software Association, “Essential facts about the computer and video game industry,” 2012.
- [2] Thijs Hagoort and Peter Warman, “2013 Global Games Market Report,” Tech. Rep., Newzoo, 2013, Accessed 20 Jan 2014.
- [3] Simon M. Lucas, Michael Mateas, Mike Preuss, Pieter Spronck, and Julian Togelius, Eds., *Artificial and Computational Intelligence in Games*, vol. 6 of *Dagstuhl Follow-Ups*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.
- [4] R. Lara-Cabrera, C. Cotta, and A.J. Fernández-Leiva, “An analysis of the structure and evolution of the scientific collaboration network of computer intelligence in games,” *Physica A: Statistical Mechanics and its Applications*, vol. 395, no. 0, pp. 523 – 536, 2014.
- [5] Ian Millington and John Funge, *Artificial intelligence for games*, CRC Press, 2009.
- [6] Raúl Lara-Cabrera, Carlos Cotta, and Antonio J. Fernández-Leiva, “A Review of Computational Intelligence in RTS Games,” in *2013 IEEE Symposium on Foundations of Computational Intelligence*, M. Ojeda, C. Cotta, and L. Franco, Eds., 2013, pp. 114–121.
- [7] Julian Togelius, Georgios N. Yannakakis, Kenneth O. Stanley, and Cameron Browne, “Search-based procedural content generation: A taxonomy and survey,” *IEEE Trans. Comput. Intellig. and AI in Games*, vol. 3, no. 3, pp. 172–186, 2011.
- [8] G.N. Yannakakis and J. Togelius, “A panorama of artificial and computational intelligence in games,” *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.
- [9] Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup, “Procedural content generation for games: A survey,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 9, no. 1, pp. 1:1–1:22, Feb. 2013.
- [10] Tobias Mahlmann, Julian Togelius, and Georgios N. Yannakakis, “Spicing up map generation,” in *Applications of Evolutionary Computation*, Cecilia Di Chio et al., Eds., Málaga, Spain, 2012, vol. 7248 of *Lecture Notes in Computer Science*, pp. 224–233, Springer-Verlag.
- [11] Miguel Frade, Francisco Fernández de Vega, and Carlos Cotta, “Modelling video games’ landscapes by means of genetic terrain programming - a new approach for improving users’ experience,” in *Applications of Evolutionary Computing*, Mario Giacobini et al., Eds., Berlin Heidelberg, 2008, vol. 4974 of *Lecture Notes in Computer Science*, pp. 485–490, Springer-Verlag.
- [12] Miguel Frade, Francisco Fernández de Vega, and Carlos Cotta, “Breeding terrains with genetic terrain programming: The evolution of terrain generators,” *International Journal of Computer Games Technology*, vol. 2009, 2009.
- [13] Miguel Frade, Francisco Fernández de Vega, and Carlos Cotta, “Evolution of artificial terrains for video games based on obstacles edge length,” in *IEEE Congress on Evolutionary Computation*. 2010, pp. 1–8, IEEE.
- [14] A. Liapis, H.P. Martinez, J. Togelius, and G.N. Yannakakis, “Adaptive game level creation through rank-based interactive evolution,” in *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*, Aug 2013, pp. 1–8.
- [15] J. Togelius, R. De Nardi, and S.M. Lucas, “Towards automatic personalised content creation for racing games,” in *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on*, 2007, pp. 252–259.
- [16] D. Ashlock and C. McGuinness, “Landscape automata for search based procedural content generation,” in

¹⁵<http://dnemesis.lcc.uma.es/wordpress/>

¹⁶<http://anyself.wordpress.com/>

- Computational Intelligence in Games (CIG), 2013 IEEE Conference on*, Aug 2013, pp. 1–8.
- [17] H.A. Diaz-Furlong and A.L. Solis-Gonzalez Cosio, “An approach to level design using procedural content generation and difficulty curves,” in *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*, Aug 2013, pp. 1–8.
- [18] JoseM. Font, Tobias Mählmann, Daniel Manrique, and Julian Togelius, “A card game description language,” in *Applications of Evolutionary Computation*, AnnaL. Esparcia-Alcázar, Ed., vol. 7835 of *Lecture Notes in Computer Science*, pp. 254–263. Springer Berlin Heidelberg, 2013.
- [19] Karen Collins, “An introduction to procedural music in video games,” *Contemporary Music Review*, vol. 28, no. 1, pp. 5–15, 2009.
- [20] Curtis Onuczko, Duane Szafron, Jonathan Schaeffer, Maria Cutumisu, Jeff Siegel, Kevin Waugh, and Allan Schumacher, “Automatic story generation for computer role-playing games,” in *AIIDE*, 2006, pp. 147–148.
- [21] Rubén Héctor García-Ortega, Pablo García-Sánchez, Antonio Mora, and Juan José Merelo, “My life as a sim: evolving unique and engaging life stories using virtual worlds,” in *Proceedings of the fourteenth International Conference on the Synthesis and Simulation of Living Systems*, Sebastian Risi René Doursat Hiroki Sayama, John Rieffel and Hod Lipson, Eds., 2014, pp. 580–587.
- [22] Rosalind W Picard, *Affective computing*, MIT Media Laboratory, 1995.
- [23] David Freeman, “Creating emotion in games: The craft and art of emotioneering™,” *Comput. Entertain.*, vol. 2, no. 3, pp. 15–15, July 2004.
- [24] Luis Peña, José María Peña, and Sascha Ossowski, “Representing emotion and mood states for virtual agents,” in *Multiagent System Technologies - 9th German Conference, MATES 2011, Berlin, Germany, October 6-7, 2011. Proceedings*, Franziska Klügl and Sascha Ossowski, Eds. 2011, vol. 6973 of *Lecture Notes in Computer Science*, pp. 181–188, Springer.
- [25] Luis Peña, Sascha Ossowski, José María Peña, and José Ángel Sánchez, “EEP - A lightweight emotional model: Application to RPG video game characters,” in *2011 IEEE Conference on Computational Intelligence and Games, CIG 2011, Seoul, South Korea, August 31 - September 3, 2011*, Sung-Bae Cho, Simon M. Lucas, and Philip Hingston, Eds. 2011, pp. 142–149, IEEE.
- [26] Anja Johansson and Pierangelo Dell’Acqua, “Emotional behavior trees,” in *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*. IEEE, 2012, pp. 355–362.
- [27] Georgios N Yannakakis, “How to model and augment player satisfaction: a review.,” in *WOCCI*, 2008, p. 21.
- [28] Mariela Nogueira, Carlos Cotta, and Antonio José Fernández Leiva, “On modeling, evaluating and increasing players’ satisfaction quantitatively: Steps towards a taxonomy,” in *Applications of Evolutionary Computation - EvoGAMES*, Cecilia Di Chio et al., Ed. 2012, vol. 7248 of *Lecture Notes in Computer Science*, pp. 245–254, Springer.
- [29] Thomas W Malone, “What makes things fun to learn? heuristics for designing instructional computer games,” in *Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems*. ACM, 1980, pp. 162–169.
- [30] Mihaly Csikszentmihalyi and Mihaly Csikszentmihalyi, *Flow: The psychology of optimal experience*, vol. 41, HarperPerennial New York, 1991.
- [31] Penelope Sweetser and Peta Wyeth, “Gameflow: a model for evaluating player enjoyment in games,” *Computers in Entertainment (CIE)*, vol. 3, no. 3, pp. 3–3, 2005.
- [32] Georgios N Yannakakis and John Hallam, “Towards optimizing entertainment in computer games,” *Applied Artificial Intelligence*, vol. 21, no. 10, pp. 933–971, 2007.
- [33] Zahid Halim, A Rauf Baig, and Hasan Mujtaba, “Measuring entertainment and automatic generation of entertaining games,” *International Journal of Information Technology, Communications and Convergence*, vol. 1, no. 1, pp. 92–107, 2010.
- [34] Georgios N Yannakakis and John Hallam, “Real-time game adaptation for optimizing player satisfaction,” *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 1, no. 2, pp. 121–133, 2009.
- [35] Georgios N Yannakakis and John Hallam, “Towards capturing and enhancing entertainment in computer games,” in *Advances in artificial intelligence*, pp. 432–442. Springer, 2006.
- [36] Lars Lidén, “Artificial stupidity: The art of intentional mistakes,” *AI Game Programming Wisdom*, vol. 2, pp. 41–48, 2003.
- [37] John N Thompson, *The geographic mosaic of coevolution*, University of Chicago Press, 2005.
- [38] Karl Sims, “Evolving 3D Morphology and Behavior by Competition,” *Artificial Life*, vol. 1, no. 4, pp. 353–372, 1994.
- [39] Phillipa M. Avery and Zbigniew Michalewicz, “Static experts and dynamic enemies in coevolutionary games,” in *IEEE Congress on Evolutionary Computation*, 2007, pp. 4035–4042.
- [40] R.W. Johnson, M.E. Melich, Z. Michalewicz, and M. Schmidt, “Coevolutionary Tempo game,” in *Evolutionary Computation. CEC’04. Congress on*, 2004, vol. 2, pp. 1610–1617.
- [41] P. Avery et al., *Coevolving a computer player for resource allocation games: using the game of Tempo as a test space.*, Ph.D. thesis, School of Computer Science University of Adelaide, 2008.
- [42] Mariela Nogueira, Carlos Cotta, and Antonio José Fernández Leiva, “An analysis of hall-of-fame strategies in competitive coevolutionary algorithms for self-learning in RTS games,” in *Learning and Intelligent Optimization - 7th International Conference, LION 7, Catania, Italy, January 7-11, 2013, Revised Selected Papers*, Giuseppe Nicosia and Panos M. Pardalos, Eds. 2013, vol. 7997 of *Lecture Notes in Computer Science*, pp. 174–188, Springer.
- [43] Mike Preuss, Nicola Beume, Holger Danielsiek, Tobias Hein, Boris Naujoks, Nico Piatkowski, Raphael Stür, Andreas Thom, and Simon Wessing, “Towards intelligent team composition and maneuvering in real-time strategy games,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 2, pp. 82–98, 2010.
- [44] Jan Quadflieg, Mike Preuss, Oliver Kramer, and Günter Rudolph, “Learning the track and planning ahead in a car racing controller,” in *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*. IEEE, 2010, pp. 395–402.
- [45] Kenneth O. Stanley, Bobby D. Bryant, and Risto Miikkulainen, “Real-time neuroevolution in the NERO video game,” *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 6, pp. 653–668, 2005.
- [46] Ben George Weber and Michael Mateas, “A data mining approach to strategy prediction,” in *IEEE Symposium on Computational Intelligence and Games*, Pier Luca Lanzi, Ed. 2009, pp. 140–147, IEEE.
- [47] AJ Fernández-Ares, Pablo García-Sánchez, Antonio Miguel Mora, and JJ Merelo, “Adaptive bots for real-time strategy games via map characterization,” in *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*. IEEE, 2012, pp. 417–721.
- [48] Michael Buro, “ORTS: A hack-free RTS game environment,” in *Computers and Games*, Jonathan Schaeffer et al., Eds. 2002, vol. 2883 of *Lecture Notes in Computer Science*, pp. 280–291, Springer.
- [49] Mariela Nogueira Collazo, Antonio José Fernández Leiva, and Carlos Cotta Porras, “Eryna: una herramienta de apoyo a la revolución de los videojuegos,” in *Proceedings 1st Congreso de la Sociedad Española para las Ciencias del Videojuego, CoSECivi 2014, Barcelona, Spain, June 24, 2014.*, David Camacho, Marco Antonio Gómez-Martín, and Pedro Antonio González-Calero, Eds. 2014, vol. 1196 of *CEUR Workshop Proceedings*, pp. 173–184, CEUR-WS.org.
- [50] Daniel Scales and Tommy Thompson, “Spelunkbots api-an ai toolset for spelunky,” in *Computational Intelligence and Games (CIG), 2014 IEEE Conference on*. IEEE, 2014, pp. 1–8.