

Motores de Juegos e Inteligencia Artificial para la Enseñanza

Frittelli, V. – Strub, A. M. – Destéfani, E. – Steffolani, F – Teicher, R. – Tartabini, M. – Bett, G. – Fernández, J. – Serrano, D..

Proyecto: *Diseño de Motores de Juegos y Componentes para Enseñanza y Aplicaciones de la Inteligencia Artificial (Código: UTN1707)*

Carrera de Ingeniería en Sistemas de Información – UTN: Facultad Regional Córdoba

Dirección: M. López eqz. Cruz Roja Argentina – Ciudad Universitaria (5000) – Córdoba (Capital)
Teléfono: (0351) 5986000

{ vfrittelli@gmail.com - anastrub@gmail.com - edestefanis1@gmail.com - fsteffolani@gmail.com - rteicher@gmail.com - marcelatartabini@hotmail.com - gfbett@gmail.com - jujulifer@gmail.com - diegojserrano@gmail.com }

Resumen

En la organización curricular de la carrera de Ingeniería en Sistemas de Información es poco común la integración de asignaturas de diferentes áreas aun cuando existan temáticas, prácticas o incluso docentes en común. Si bien es habitual identificar estrategias de integración entre asignaturas correlativas (integración vertical), no ocurre lo mismo con asignaturas de un mismo nivel (integración horizontal) ni con asignaturas de otras áreas. Distintas materias incluyen el estudio de temas que permitirían al alumno desarrollar software para dominios diversos. Por otro lado, otras asignaturas solicitan el desarrollo de agentes inteligentes que suelen plantearse como “jugadores” virtuales para simulaciones de estrategias frente al accionar de un adversario; y la evaluación de estos trabajos requiere un esfuerzo considerable por parte del docente. Se plantea aquí una propuesta de integración consistente diseñar actividades prácticas que involucren a alumnos de asignaturas distintas,

mediante la cual algunos desarrollen motores para diferentes juegos o escenarios, y alumnos de otras asignaturas desarrollen los jugadores o agentes. El desafío es que en el desarrollo del motor del juego, se plantee un marco de trabajo general, que permita el diseño de nuevos escenarios desde un modelo de reusabilidad, pero que permita incorporar agentes externos con relativa sencillez.

Palabras clave: Algoritmos - Aplicación Educativa - Componentes - Integración - Inteligencia Artificial - Modelos de Juegos -

Contexto

La propuesta se planteó en el proyecto que se designa oficialmente como *Diseño de Motores de Juegos y Componentes para Enseñanza y Aplicaciones de la Inteligencia Artificial*, presentado para su aprobación definitiva en los primeros días de Marzo de 2013. El proyecto está integrado por docentes

de varias Cátedras de la UTN Córdoba con orientación a la programación y/o a los paradigmas de programación, y surgió como una forma de estudiar, poner en práctica y analizar diferentes ideas de integración entre trabajos de distintas cátedras. Estas ideas surgían de reuniones y conversaciones entre pares y notando que muchos docentes sugerían desde hace tiempo la posibilidad del desarrollo de actividades conjuntas, con algún tipo de marco de trabajo diseñado y prefijado, es que se decidió el planteo de un proyecto de investigación que las formalice.

Introducción

Diversas asignaturas de la carrera de Ingeniería en Sistemas de Información (como Algoritmos y Estructuras de Datos (AED), Paradigmas de Programación (PPR), y las electivas Tecnología de Software de Base (TSB) y Diseño de Lenguajes de Consulta (DLC)) incluyen en sus programas el estudio de un amplio abanico de estructuras de datos y sus algoritmos relacionados, permitiendo al alumno desarrollar software para dominios diversos. Esta situación es muy aprovechada en evaluaciones parciales, trabajos prácticos y exámenes finales en las cuales se solicita el desarrollo y entrega de programas aplicables a escenarios muy diferentes entre sí. Por otro lado, en otras asignaturas (como Inteligencia Artificial (IAR)) se pide a los alumnos el desarrollo de agentes inteligentes tales como redes de Hopfield, redes neuronales o agentes de búsqueda mediante heurísticas varias. Entre tales actividades suele solicitarse a los alumnos la creación de "jugadores" para ciertos juegos ampliamente conocidos, o bien el desarrollo de componentes a modo de "agentes"

(Dignum, Bradshaw, Silverman, & van Doesburg, 2009) para simulaciones de estrategias frente al accionar de un adversario, de modo que los jugadores o agentes consisten en alguna pieza de software cuya interfaz o protocolo es definida con exactitud por el cuerpo docente y cuyo comportamiento es desarrollado por cada grupo de alumnos. Para la evaluación de esta actividad práctica el docente puede hacer competir a los "jugadores" de diferentes grupos entre sí. Sin embargo, esta tarea requiere un esfuerzo considerable por parte del cuerpo docente que debe programar un "motor" y/o un "tablero" para el juego, es decir, un software que seleccione la estrategia de dos o más grupos y las ejecute en forma simultánea comunicando a cada una de las partes el estado del juego y el movimiento del adversario. Además el motor debe identificar si un jugador intenta un movimiento o una acción no válida según las reglas o restricciones del juego y debe poder detectar una situación ganadora o que identifique la finalización del juego. El desarrollo del motor del juego es una tarea de programación compleja y debe ser realizada una vez por cada juego, ya que es muy difícil utilizar un mismo motor para juegos con reglas diferentes. Por lo tanto se requiere un esfuerzo importante por parte del plantel docente para programar ese motor, ya sea que lo realicen los mismos docentes o que requieran el auxilio de un ayudante de cátedra o becario. En los lenguajes de programación que se utilizan en la carrera resulta relativamente simple y natural construir un marco de trabajo (o *framework*) (Salen & Zimmerman, 2003) a modo de contenedor, dentro del cual se pueda correr el motor de las reglas del juego para que tome a los jugadores o agentes como

componentes. Además, al mismo tiempo que se programan componentes para dicho framework, se continúa el perfeccionamiento de la programación flexible y extensible aprovechando a fondo un paradigma como el de la Programación Orientada a Objetos. En ese sentido, entonces, la propuesta de integración consiste en la realización de actividades prácticas que involucre a los alumnos de asignaturas diversas. Los alumnos de una u otra asignatura desarrollan motores para diferentes juegos y los alumnos de otras asignaturas desarrollan los jugadores o agentes. El desafío es que en el desarrollo del motor del juego, se llegue al planteo de un framework que permita el diseño de esos motores y escenarios en forma genérica (Salen & Zimmerman, 2003), de forma que pueda reusarse y facilitar el trabajo posterior cuando se pidan nuevos escenarios, pero que permita integrar agentes externos con relativa sencillez.

Líneas de Investigación y Desarrollo

Las principales líneas de acción, investigación y desarrollo previstas para el proyecto son las siguientes:

1. Selección de los juegos o situaciones base y determinación de las reglas: Deben seleccionarse juegos o escenarios de competencia que ofrezcan dificultades razonables y planificadas para el aprendizaje de los alumnos (Schell, 2008), de forma tal que los motores requieran el uso de estructuras de datos y algoritmos que puedan aprender en asignaturas previas y que los jugadores requieran estrategias de juego que puedan desarrollar con los conceptos adquiridos en materias posteriores. Son aplicables

también diversas estrategias, como los algoritmos evolutivos (Ashlock, 2006) o los algoritmos genéticos (Chisholm & Bradbeer, 1997). Esta actividad requiere del análisis del estado del arte actual en la materia, estudiando publicaciones que se hayan realizado al respecto, y experiencias llevadas a cabo por otros grupos de investigación.

2. Selección de las tecnologías de desarrollo: Un aspecto muy importante es que deben seleccionarse tecnologías de desarrollo que sean ampliamente dominadas por todos los alumnos para que dicha elección no resulte en una complejidad adicional. Para ello deben considerarse todas las tecnologías que los alumnos conozcan en profundidad por haberlas aprendido en el transcurso de su carrera (no necesariamente las que hayan adquirido en estas asignaturas específicas).

3. Definición del protocolo de comunicación entre el motor y los jugadores: El protocolo de comunicación seleccionado también influye en esta decisión. Si se decide que todos los grupos utilicen el mismo lenguaje de programación, podría seleccionarse Java por ser el más estudiado en la actualidad en nuestras materias. Si se decide que los jugadores se comuniquen con el motor por medio de la entrada y salida estándar, cada equipo puede elegir prácticamente cualquier lenguaje de programación moderno.

4. Desarrollo del motor: Esta tarea requiere trabajo en equipo y dominio acabado de una plataforma de desarrollo que incluya capacidades gráficas. Aquí se aplicarán algoritmos y estructuras de datos básicas y avanzadas, así como elementos de diseño y análisis de algoritmos. El motor puede plantearse como un componente evaluador y controlador de reglas del

juego. De esta forma, el desarrollo termina enfocándose en un servidor que sea capaz de correr el motor, incorporando agentes a modo de jugadores.

5. Desarrollo de los jugadores: Se requiere el conocimiento y el dominio de elementos de Inteligencia Artificial aplicados al contexto específico de situaciones de juegos y presencia de adversarios. En el contexto de lo expuesto en el punto anterior, los jugadores pueden desarrollarse como componentes que ejecutan las estrategias impuestas por el motor.

6. Ejecución de las competencias: Aquí se integran los desarrollos de motores, escenarios y agentes o jugadores, para llevar a la práctica el juego o competencia diseñado.

7. Emisión de resultados y reportes: Se miden los resultados obtenidos en cuanto a logro de objetivos del juego desarrollado, el cumplimiento de las metas por parte de cada agente, el nivel de eficiencia alcanzado por cada uno (y por lo tanto de las estrategias programadas para ellos) y la posibilidad de realimentación y refinamiento de esas estrategias.

Resultados y Objetivos

El objetivo general del proyecto es el *"desarrollo de un Motor o Servidor de Estrategias de Juego, que permita el planteo de Plataformas para el diseño de Juegos y Escenarios con Adversarios, con el fin de facilitar la integración de conocimientos y prácticas de diversas asignaturas de la Carrera de Ingeniería en Sistemas de Información, y posibilitar el desarrollo futuro de aplicaciones similares en contextos profesionales"*. A nivel de resultados esperados, se apunta a la posible integración de desarrollos y aplicaciones

prácticas entre asignaturas diversas de la carrera de Ingeniería en Sistemas de Información, pero es claramente extrapolable a otras situaciones y aplicaciones no académicas, tales como: simulaciones de estrategias de agentes adversarios en cualquier ámbito de aplicación de la Teoría de Juegos, desarrollo profesional de software para juegos, e incluso aplicaciones en la programación de software de control de componentes de hardware para robótica (software, motores de juego y escenarios para placas programables). En cuanto a posibles dificultades, un proyecto de esta naturaleza requiere considerar que existirán muchos alumnos y grupos trabajando desde cátedras distintas (aunque esto puede manejarse en parte mediante el planteo de estas actividades en forma opcional), además del tiempo de clase ocupado con los desarrollos. Si varias cátedras acuerdan un trabajo conjunto y colaborativo en esta línea de acción, debe preverse adecuadamente el tema en las planificaciones de todas ellas.

Formación de Recursos Humanos

Director del Proyecto: Valerio Frittelli (Magister en Docencia Universitaria, Ingeniero en Sistemas de Información, Director de la Carrera de Licenciatura en Tecnología Educativa de la UTN Córdoba, Docente de grado y posgrado e investigador en la UTN Córdoba)

Codirector: Ana María Strub (Ingeniera en Sistemas de Información, titulada como Profesora Universitaria en Ingeniería por la Universidad Católica de Córdoba), docente e investigadora en la UTN Córdoba.

Investigador Formado y Asesor: Eduardo Destéfanis (Doctor en

Ingeniería por la UTN Córdoba, Ingeniero Electricista Electrónico, Investigador Categoría II en Programa de Incentivos, Profesor Titular Ordinario en la UTN Córdoba)

Investigadores de Apoyo Categorizados: Felipe Steffolani, Romina Teicher (ambos Ingenieros en Sistemas de Información, Categorizados E y F respectivamente en el programa interno UTN)

Investigadores de Apoyo en Formación: Marcela Tartabini, Gustavo Bett, Julieta Fernández y Diego Serrano (todos Ingenieros en Sistemas de Información y docentes en la UTN Córdoba)

Del objetivo del proyecto se desprende un fuerte trabajo en formación de recursos humanos. Los alumnos de las cátedras involucradas serán beneficiarios directos de la experiencia, así como los propios docentes e investigadores. Algunos de estos docentes se incorporan por primera vez a un proyecto de investigación, aportando sus conocimientos y prácticas, pero también buscando experiencia y antecedentes que les posibiliten el acceso futuro a la categorización como investigadores. En esta primera etapa, en que el proyecto se está planteando, todo el equipo se conforma con docentes de diversas cátedras. A medida que el trabajo avance, se prevé incorporar estudiantes de diversos años de cursado de la carrera. Esto permitirá contribuir a su formación como investigadores y a aportar elementos de aplicación profesional. Al diseñar e implementar el Servidor de Estrategias en una arquitectura de dos niveles de componentes (servidor de reglas y estrategias vs. agentes que ejecutan

reglas y estrategias) se contribuye además en la formación profesional de investigadores y alumnos, ya que el grupo estaría en condiciones de extender el alcance a otros proyectos orientados a la construcción de un servidor de aplicaciones amplio (que evalúe y controle reglas de negocio en general, y no solo aplicables al contexto de juegos, simulaciones e integración entre contenidos de asignaturas).

Referencias

- [1.] Ashlock, D. (2006). Evolutionary Computation for Modeling and Optimization. New York: Springer.
- [2.] Chisholm, K., & Bradbeer, P. (1997). Machine Learning Using a Genetic Algorithm to Optimise a Draughts Program Board Evaluation Function. Proceedings of IEEE International Conference on Evolutionary Computation, (ICEC'97. Indianapolis: IEEE Conference Publications.
- [3.] Dignum, F., Bradshaw, J., Silverman, B., & van Doesburg, W. (2009). Agents for Games and Simulations: Trends in Techniques, Concepts and Design. Berlín: Springer.
- [4.] Salen, K., & Zimmerman, E. (2003). Rules of Play: Game Design Fundamentals. Cambridge: MIT Press.
- [5.] Schell, J. (2008). The Art of Game Design: A book of lenses. Burlington: Morgan Kaufmann.