

## APRENDIZAJE BASADO EN JUEGOS

**Marco A. Gómez-Martín**  
**Pedro P. Gómez-Martín**  
**Pedro A. González-Calero**

# Aprendizaje basado en juegos<sup>1</sup>

Marco A. Gómez-Martín, Pedro P. Gómez-Martín and Pedro A. González-Calero

Dep. Sistemas Informáticos y Programación  
Universidad Complutense de Madrid, Spain  
{marcoa, pedrop, pedro}@sip.ucm.es

**Resumen.** Gracias al incremento de potencia de los ordenadores, gran cantidad de personas dedican horas y horas a aprovechar su aspecto más lúdico, los videojuegos. Por otro lado, existen programas educativos que aprovechan la infinita paciencia de los ordenadores que les hacen capaces de explicar conceptos una y otra vez hasta que los alumnos lo entiendan. En este artículo mostramos qué cosas pueden aportar las aplicaciones de enseñanza a los videojuegos y viceversa. Terminamos describiendo JV<sup>2</sup>M, como un ejemplo de sistema de aprendizaje basado en juegos.

## 1 Introducción

La llegada de los ordenadores ha propiciado numerosos intentos de desarrollar sistemas que sustituyan al profesor. Estos sistemas pueden categorizarse en animaciones simples, simulaciones, entrenamiento asistido, aprendizaje basado en Web, tutores inteligentes (del inglés *intelligent tutoring systems* o ITS) y agentes pedagógicos.

En dominios donde es necesario poner en práctica lo aprendido, la aproximación natural son los simuladores. Estos sistemas se apoyan en el constructivismo y las teorías psicológicas que afirman que el aprendizaje es la construcción de estructuras cognitivas. Sin embargo, una simulación perfecta sin algún tipo de guía no sirve para nada. Se necesita una realimentación que indique qué se está haciendo mal y cómo mejorar. Por esto, ciertos ITSs son en realidad simuladores con inteligencia añadida que supervisa al alumno mientras está interactuando con el sistema. La simulación es el entorno de aprendizaje, y la aplicación ofrece una guía al final de cada iteración. En estos sistemas, el estudiante trabaja en un ejercicio elegido utilizando algún criterio pedagógico. Para eso, propone al alumno problemas de complejidad creciente, dándole las explicaciones oportunas cuando se necesitan. Cuando el sistema advierte que el estudiante tiene el conocimiento suficiente para pasar al nivel siguiente, le presenta un ejercicio de mayor dificultad. Por último, éstos sistemas se pueden mejorar incorporando un agente pedagógico que habita en el propio entorno de aprendizaje y que proporciona ayuda contextualizada.

---

<sup>1</sup> Subvencionado por el comité interministerial de Ciencia y Tecnología (TIC2002-01961)



**Fig. 1.** Sistema Design-a-plant con HERMAN-THE-BUG

Sin embargo, a pesar de todos los esfuerzos depositados en la implementación de estos sistemas, la mayor parte de ellos son aplicaciones aburridas que el estudiante utiliza porque tiene la obligación de hacerlo. Por otro lado, el incremento de la potencia de los ordenadores de sobremesa ha impulsado el crecimiento del software de entretenimiento, y ha dado un acceso masivo a los juegos. Los jugadores están inmersos en entornos virtuales, sin ninguna intención de aprender ningún dominio particular, sino simplemente convirtiéndose en parte de ese entorno y participando en él. El juego nos muestra que tareas difíciles y tediosas pueden ser lo suficientemente divertidas y motivantes cuando forman parte de una buena historia. Si conseguimos que el contenido de aprendizaje se difumine con la historia de un buen juego, nace el concepto de aprendizaje basado en juegos y el término “enseñanza basada en juegos” se utiliza para describir la aplicación de los juegos en la enseñanza.

En los apartados siguientes nos centramos en la combinación de videojuegos con tutores inteligentes. En particular, en la sección siguiente mostramos qué tipo de cosas pueden proporcionar los sistemas de enseñanza a los juegos, para, en la sección posterior, hacer el análisis inverso. Posteriormente describimos  $JV^2M$ , como un ejemplo de sistema de enseñanza basado en juegos, para instruir en el proceso de compilación de programas escritos en el lenguaje Java.

## **2 Lo que los sistemas de enseñanza pueden dar a los juegos**

Los sistemas de enseñanza típicos van proponiendo al estudiante ejercicios a resolver según su nivel de maestría, complicándolos cuando el usuario va adquiriendo más destrezas.



Fig. 2. Sistema Internet advisor con COSMO

La interacción con uno de estos sistemas suele comenzar con la creación de un modelo del usuario; es en ese perfil del usuario donde el sistema almacena los conocimientos del alumno, qué conceptos sabe y cuales no.

Con esa información, el sistema elige qué ejercicio es el más adecuado para que el usuario intente resolver en ese momento, y le enfrenta a él. El alumno lo trata de resolver, mientras el sistema ofrece ayuda cuando lo cree necesario.

Cuando el estudiante ha resuelto el ejercicio, el perfil del usuario se actualiza, y el sistema se plantea qué es lo más conveniente, dependiendo de cómo ha sido resuelto el ejercicio anterior. En algunos casos, si el alumno respondió de manera torpe o necesitó demasiada ayuda, se puede repetir el mismo ejercicio. En casos no tan extremos, se puede plantear otro problema similar al anterior, que practique los mismos conceptos que el recién terminado. Por último, si el usuario resolvió el problema con la destreza suficiente, el sistema puede decidir practicar conceptos nuevos. En ciertos sistemas [1], el alumno puede intervenir en esta decisión, cuando él mismo estima que debería repetir algún ejercicio.

Una capacidad bienvenida en los entornos interactivos es la existencia de un avatar o personaje que, compartiendo tu propio entorno de aprendizaje, te ayude a resolver las tareas encomendadas. Estos avatares suelen llamarse agentes pedagógicos, en tanto en cuanto aprovechan los momentos más adecuados para dar explicaciones contextualizadas que maximizan las posibilidades de comprensión y asimilación por parte del alumno. Por ejemplo el sistema Design-a-plant [2] (ver Figura 1), enseña la fisiología de las plantas ayudados por el agente HERMAN THE BUG, que te ayuda en la resolución de ejercicios. COSMO hace las veces de profesor en el sistema *Internet Advisor* [3] (ver Figura 2), en el que el usuario aprende el funcionamiento interno de las redes de ordenadores.

Las partes de las que se compone un sistema de aprendizaje es:



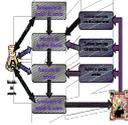
**Fig. 3.** Príncipe de Persia: el alma del guerrero

- Conjunto de ejercicios: guarda todos los problemas a los que se puede ver enfrentado el alumno.
- Modelo de usuario: almacena la lista de conceptos que el alumno parece tener.
- Módulo pedagógico: es el responsable de decidir qué ejercicios resolver en cada momento, en base a los conocimientos del alumno.
- Entorno interactivo de aprendizaje: en el que, normalmente de forma gráfica, se presentan los problemas y el usuario los resuelve. Puede estar habitado por algún agente pedagógico.

Por su parte, en los juegos mucha gente termina echando de menos en algún momento algún tipo de ayuda. Todo jugador ha estado alguna vez inevitablemente atascado en una parte de un juego, y al final ha terminado abandonándolo. La prueba es la gran cantidad de revistas y sitios en Internet dedicados a las guías, trucos, pistas e instrucciones casi paso a paso de cómo pasarse juegos. Esta necesidad debería ser vista como un error de diseño del juego, ya que fuerza al usuario (comprador) a buscar en la Web o gastar (invertir) más dinero para poder disfrutar del juego completo que ha pagado. Porque la otra opción es rendirse, y olvidarlo, pero en ese caso, es más que probable que no vuelva a comprar otro título de la misma compañía o género.

Una solución a este problema es proporcionar al usuario algún tipo de mecanismo para que pueda preguntar qué tiene que hacer a continuación para poder seguir avanzando.

Algunos juegos incorporan en cierta medida este tipo de ayuda para recordar/enseñar a los jugadores por ejemplo las combinaciones de teclas necesarias en ciertas situaciones (por ejemplo *Príncipe de Persia: el alma del guerrero*) pero no ayudan cuando el usuario no sabe dónde ir o qué hacer para resolver el siguiente puzzle.



**Fig. 4.** Ciclo de un sistema de enseñanza

Como ya hemos dicho, la idea que utilizan los sistemas de aprendizaje son los agentes pedagógicos. Éstos podrían incluirse en juegos para ayudar a los usuarios en los momentos en los que no saben seguir. No pensamos en avatares que ayudan a completar una misión luchando a tu lado contra los enemigos (como en *Call of Duty* o *Halo*), sino alguien que acompaña durante todo el juego (como hace *Globox* en *Rayman 3*) y que da consejos cuando detecta que no se progresa. Esto haría que el jugador se sintiera más seguro de poder terminarse el juego, es decir, de tener la certeza de que va a poder disfrutar del juego completo por el que paga, y por lo tanto, podría hacer subir las ventas. Para que esta incorporación no decremente la satisfacción de los usuarios avanzados, debería poder ser desactivada.

Otra idea que los juegos van incorporando cada vez más es la posibilidad de intervenir en la decisión de qué niveles jugar, algo que tradicionalmente han proporcionado los entornos de aprendizaje. En éstos, el estudiante puede decidir repetir un ejercicio cuando siente que lo ha terminado sin comprenderlo como debería. Por ejemplo en *Crash Bandicoot* o *Blinx: The time sweeper*, los niveles ya superados siguen abiertos al jugador para darle la posibilidad de realizarlos en menos tiempo o consiguiendo mayor puntuación.

### **3 Lo que los juegos pueden dar a los sistemas de enseñanza**

Malcolm Gladwell, antiguo periodista del *The Washington Post*, comentaba que *Barrio Sésamo* estaba basado en una simpleza: si puedes mantener la atención de los niños, puedes educarlos. [4]

Para que un estudiante aprenda todo lo que un determinado software educativo puede enseñarle, el alumno debe hacer uso del sistema durante un número de horas que puede llegar a ser considerable. Si el usuario no tiene una razón impuesta que le obligue a usar un determinado programa educativo, el sistema ha de ser capaz de



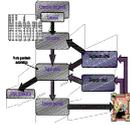
**Fig. 5.** Rayman 3 mirándose en el espejo al lado de Globox

atraer por sí mismo el interés del usuario potencial creándole una motivación interna que le lleve a pasar las horas delante del ordenador, igual que hacía Barrio Sésamo con los pequeños. Este tipo de motivación es beneficiosa para el alumno porque aumenta la velocidad de aprendizaje, a la vez que incrementa el número de horas que invierte en usar el programa, metiéndole en una espiral positiva que le lleva a aprender cada vez más.

Lograr que un sistema educativo sea atrayente a sus usuarios no es una tarea sencilla. Se cree que los agentes pedagógicos aumentan la motivación del usuario al dar la sensación de que hay un ente que se preocupa por sus avances [5]. Sin embargo, esto puede no ser suficiente. Se pueden obtener ideas sobre cómo motivar al usuario a continuar usando la aplicación en el área de los videojuegos.

Por ejemplo, una fuente de motivación puede ser la existencia de final. Aunque algunos juegos como *Tetris* están pensado para no poder ser terminados, la mayoría posee uno o varios finales. Cuando el usuario alcanza alguno de ellos, el juego le recompensa mostrándole algunos efectos impresionantes que compensan las horas de esfuerzo del jugador. El usuario está motivado esperando ver hasta dónde le conduce el juego. Algo equivalente ocurre en los programas educativos donde los conceptos del dominio a enseñar están claramente delimitados, pues la naturaleza del dominio mismo fuerza a la aplicación a terminar cuando el modelo del usuario indica que el usuario sabe todo lo que el sistema es capaz de enseñar.

Como hemos dicho, la recompensa de un juego es llegar al final; sin embargo al empezarlo, este objetivo está bastante lejos. Por eso, muchos juegos añaden pequeños premios al final de ciertos niveles, como el clásico baile de los personajes en *Tetris* o los videos mostrados entre niveles en muchas aventuras gráficas. Aunque *Tetris* era lo suficientemente atrayente como para no necesitar de estos trucos, la experiencia nos enseña que esas recompensas pueden ayudar a mantener la atención de los usuarios. Hoy en día, incluso se utilizan para aumentar las ventas. Por ejemplo, en la parte trasera de la carátula de *Rayman 3*, se anuncia que el jugador puede desbloquear



**Fig. 6.** Ciclo de un videojuego típico

varios mini-juegos, videos secretos y bonus al ir terminando niveles, y el *Príncipe de Persia: las arenas del tiempo* permite jugar al juego original de 1989. Incluso algunos juegos se basan únicamente en esta idea de ir desbloqueando más y más capacidades, como *Kung Fu Chaos*.

Quitando estas recompensas, los juegos mantienen también la atención del usuario utilizando una historia de fondo. En los productos de entretenimiento suele existir un hilo conductor a lo largo de todo el juego que le da la idea de unidad. Los videos que se muestran entre niveles son parte de la historia, que muestran al usuario el siguiente reto en el que se va a ver envuelto.

Por tanto, creemos que para un sistema de enseñanza consiga ser tan divertido como un juego, es importante romper la sensación de que la aplicación es un conjunto independiente de ejercicios; en vez de eso, debería haber algún tipo de relación entre ellos, para que el estudiante perciba cierta coherencia en la aplicación. Algo similar ocurre en algunos juegos de éxito. Por ejemplo, en *Call of Duty*, el jugador es un soldado de la Segunda Guerra Mundial, y si bien los niveles tienen poco que ver unos con otros, en el sentido de que hay un gran salto entre ellos, el video previo a cada fase explica la siguiente misión dentro del ámbito global de la Guerra, justificando el por qué de atacar un puente, o ascender por una colina. Parece pues viable llevar la idea de historia a los tutores inteligentes. Además, dado que toda historia tiene que terminar, se puede sincronizar con el final de la parte pedagógica en la que se ha enseñado todo lo posible. La idea es que el propio sistema imponga una meta más clara que la obvia "has aprendido todo lo que podía enseñarte".

Para lograrlo, la idea es construir un guión entretenido que disponga de un final que esté claramente definido. La historia creada por ese guión esté dividida en fases o niveles, siendo cada uno de ellos uno de los ejercicios planteados por el sistema. El guión hace las veces de "pegamento" entre todos los ejercicios. Cuando el sistema considera, basándose en el modelo del usuario, que el alumno ha aprendido lo

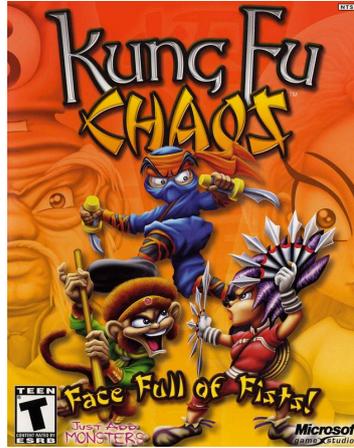


Fig. 7. Kung Fu Chaos

suficiente, lleva la historia a su final, lo que le indica al usuario que la partida ha terminado y su proceso de aprendizaje también.

Aunque la historia construida sea más o menos fija, cada una de las fases continúa siendo un ejercicio seleccionado por el módulo pedagógico en función de los avances del usuario. Es decir, la historia genera una excusa lúdica, más o menos cableada que incita al usuario a afrontar una nueva fase, pero el nivel en sí (lo que en los juegos se llaman puzzles o quizzes) se elige dinámicamente para adaptarlo al estudiante. En sistemas más sofisticados [6], la historia podría no ser tan fija, y disponer de algún tipo de "director" automático e inteligente que adaptara la historia de acuerdo a cómo se van desarrollando las fases.

Una cosa adicional que podría resultar interesante es añadir finales intermedios satisfactorios. Es decir, que no haya que llegar al final del todo del juego para tener más o menos la sensación de haber llegado a algún final honroso. Es una "solicitud" de [7] en su análisis de los programas educativos para poder utilizarlos directamente en clase. Si el juego es muy largo, es inviable terminarlo en clase y deja mal sabor de boca abandonarlos a mitad. Si se ponen finales intermedios, se puede parar ahí, y dejar como "ejercicio para el alumno" terminar el juego entero para aprender más. Esto puede no tener sentido en muchos ITS, dependiendo del tipo de alumnado al que se enfoque y de la importancia de lo que se enseñe para el currículum.

En cualquier caso, sea lo sofisticado que sea la generación del guión y la historia en sí, el ciclo del sistema de enseñanza sigue existiendo, pero no es tan explícito al usuario:

- Se muestra al usuario la motivación del guión para afrontar el siguiente ejercicio. Consiste en los objetivos de la historia que conseguirá si resuelve la fase.
- El módulo pedagógico decide qué ejercicio poner en práctica.
- Se lanza el entorno de aprendizaje. Si hay agente pedagógico, éste recibe la información sobre el nuevo ejercicio.



**Fig. 8.** Escape From Monkey Island, Lucas Arts

- El alumno intenta resolver la fase, y el agente pedagógico le ayuda cuando proceda. El modelo del usuario se actualiza convenientemente.

Una desventaja potencial de ocultar el ciclo es que se puede llegar a impedir que el usuario intervenga en el proceso de elección de ejercicios, que, como hemos dicho, admiten algunos sistemas. Habría que utilizar las artimañas que utilizan los juegos que hoy en día van incluyendo la posibilidad de repetir niveles explicada antes.

#### **4 Relación entre juegos y sistemas educativos**

Usando todas estas ideas, surgen una serie de relaciones entre las características que están disponibles en la mayor parte de los juegos existentes hoy en día y en los sistemas educativos. A continuación se listan las acciones que se pueden realizar en un juego con su análogo en una aplicación de enseñanza.

- Comenzar una partida: en una aplicación de enseñanza sería equivalente a crear el modelo del nuevo usuario, que indicará que no dispone de conocimientos en el dominio enseñado.
- Tutorial del juego: algunos juegos contienen fases iniciales sencillas de familiarización con el entorno e interfaz. Extrapolando esta idea, se podría realizar algo semejante para evaluar los conocimientos previos que tiene el usuario del dominio que vamos a enseñar.
- Superar un nivel (puzzle o quizzle): es equivalente a resolver un ejercicio planteado por el módulo pedagógico.
- Grabar una partida: cada vez que se supera un nivel el usuario tiene la posibilidad de almacenar su estado para hacerlo persistente y volver a ese



**Fig. 9.** Agente pedagógico Javy (al fondo) y el usuario (frente)

punto más adelante. En un sistema educativo eso se traduce en almacenar el modelo del usuario y el estado de la historia que hace las veces de guión. La partida guardada almacenaría también información sobre qué ejercicios ha resuelto ya, para dar la posibilidad de repetir uno anterior.

- Cargar una partida: realiza la acción inversa, recuperando el modelo del usuario y los niveles superados. La carga de una partida da la opción de continuar por el siguiente nivel, de repetir uno anterior, o uno parecido a alguno ya resuelto.
- Fin de la partida: se llega a este estado cuando se han superado todos los niveles del juego. En la aplicación de enseñanza supone el reconocimiento por parte del programa de que el usuario sabe todo lo que se le puede enseñar. La historia es llevada a su final y el usuario es consciente de haber recorrido todo el juego.
- El usuario no consigue superar un nivel ("le matan"): simboliza que el usuario no ha resuelto correctamente el ejercicio, y debe volver a intentarlo, al igual que ocurre con los niveles en los juegos. El sistema debe proporcionar algún tipo de información al estudiante que le enseñe lo necesario para superarlo la próxima vez.
- Puntos automáticos de guardado automático: en algunas ocasiones, en diferentes momentos de una fase, el juego almacena temporalmente el estado. Si el jugador no consigue superar el nivel vuelve automáticamente al punto anterior. No es posible volver a ese punto si se cierra el programa. En el ITS se convierte en lugares a los que volver si el usuario comete demasiados errores en la resolución del ejercicio que hacen inútil cualquier tipo de intento de arreglo.

## 5 JV<sup>2</sup>M

Actualmente, estamos desarrollando JV<sup>2</sup>M, un sistema educativo donde estamos poniendo en práctica las ideas previas de relacionar juegos con ITS. El sistema enseña los procesos que lleva a cabo el compilador de un programa de ordenador escrito en el lenguaje Java para traducir el código fuente que escribe el programador en el fichero ejecutable que se interpreta cuando el usuario lo arranca.

Para poder aprender cómo realizar esa traducción (lo que se llama *compilación* del programa), el alumno necesita primero saber la estructura de la máquina donde se ejecutará ese programa (llamada JVM, del inglés Máquina Virtual de Java o Java Virtual Machine). Una posibilidad para hacer que el estudiante conozca su estructura sería darle a leer su especificación oficial descrita por sus creadores (más de 400 páginas en inglés) [8]. La forma por la que nosotros nos hemos decantado es utilizar el propio sistema para enseñárselo. De esta forma, nuestra aplicación es capaz de instruir sobre los aspectos internos de la JVM, así como sobre cómo compilar el código fuente de Java, utilizando un entorno unificado de aprendizaje.

Más concretamente, el programa presenta un interfaz gráfico bastante similar al utilizado en muchas aventuras gráficas. El estudiante se encuentra inmerso en ese entorno, donde puede manipular objetos e interactuar con otros personajes.

El entorno es una ciudad virtual que simula la JVM. El usuario está representado por un avatar que puede utilizar los objetos que hay en el entorno. Además, posee un inventario donde almacena los elementos que ha ido recogiendo para su uso posterior.

La máquina virtual metafórica también está habitada por JAVY un agente pedagógico que proporciona pistas cuando el estudiante lo necesita y que puede incluso llegar a resolver el ejercicio por él mismo.

En el entorno existen distintos objetos interactivos relacionados con las diferentes estructuras de la máquina virtual. Todos ellos se manipulan a través de cuatro acciones básicas utilizadas tradicionalmente en las aventuras gráficas<sup>2</sup>, que son “mirar”, “coger”, “usar” y “usar con”.

Cada uno de los ejercicios, convertidos ahora en niveles o puzzles del juego, consiste en el código fuente de un programa en Java que el estudiante debe compilar. Para eso, tiene que *ejecutar* el código compilado en la máquina virtual que representa el entorno. De esta forma, evitamos la necesidad de que el estudiante necesite *escribir* la traducción del programa original, ya que para que sea capaz de ejecutar el programa ha tenido que traducirlo mentalmente.

Para maquillar el sistema de aprendizaje le dotamos de una historia que proporcione un hilo de unión entre todos los ejercicios. En la primera toma de contacto con la aplicación, el usuario descubre que el personaje al que controla era un humano normal, de carne y hueso, que vivía en el mundo real y que de repente ha sido empujado a un entorno virtual del que inmediatamente desea salir. Para poder hacerlo, tiene que comprender las reglas de ese lugar imaginario, que no es otro que una metafórica máquina virtual de Java, por lo que entender ese mundo supone ser capaz de ejecutar programas escritos en Java sobre la JVM simulada.

Pronto comprobará que el mundo está poblado por otros personajes en su misma situación, encerrados en un lugar al que tampoco pertenecen. Son campesinos

---

<sup>2</sup> Como por ejemplo en “Escape from Monkey Island” de Lucas Arts

oprimidos que no tienen ni el tiempo ni el conocimiento suficiente como para liderar un movimiento de autoliberación. Y ese es el papel de nuestro alumno. Estudiar el mundo y ejecutar programas más y más complicados para liberar a más y más campesinos y, finalmente a sí mismo.

El usuario cuenta con la ayuda de un habitante especial conocido como JAVY, que es el único que conoce todos los secretos del funcionamiento del mundo. Tiempo atrás intentó escapar, pero por desgracia fue apresado. Hoy en día se ha convertido en alguien con los conocimientos suficientes para ayudar, pero sin la valentía necesaria para intentar otra vez la fuga.

## 6 Conclusiones

En este artículo hemos analizado la relación entre los videojuegos y los sistemas de enseñanza por ordenador, comprobando que hay mucha relación en su funcionamiento básico. Con esa idea en mente, es fácil encontrar elementos en los videojuegos que pueden pasarse a los tutores inteligentes y viceversa.

Creemos que la inclusión de agentes pedagógicos en los videojuegos puede conseguir que el número de usuarios que abandonan, dejando el juego a la mitad, decrezca, al ser estos agentes una ayuda para los momentos en los que el jugador no es capaz de continuar. Esto no tiene por qué suponer un descenso de la diversión percibida, sino muy al contrario, ya que el usuario sabe cuando se compra el juego que podrá disfrutar de él por completo, aunque en algunas partes requiera alguna ayuda.

Por otro lado, los sistemas de enseñanza pueden también incorporar algunos elementos que hacen de los videojuegos sistemas atrayentes y lúdicos que consiguen *enganchar* a los usuarios y mantener su atención durante horas. En particular creemos que incluir una historia que una los diferentes ejercicios que el estudiante tiene que resolver puede ser especialmente motivador.

Con todo lo anterior, estamos desarrollando un sistema de enseñanza basado en juegos llamado JV<sup>2</sup>M, con el que los alumnos pueden aprender los mecanismos que utiliza el compilador del lenguaje Java para traducir el código fuente a programas directamente ejecutables por la máquina virtual.

## Bibliografía

1. R. H. Stottler. Tactical action officer intelligent tutoring system (TAO ITS). En Proceedings of the Industry/Interservice, Training, Simulation & Education Conference (I/ITSEC 2000), November 2000.
2. Stone B. Lester J. C. Dynamically sequencing an animated pedagogical agent , En Proc. of the Thirteenth National Conference on Artificial Intelligence, Agosto 1996.
3. Lester J. C., Voerman J. L., Towns S., Callaway C. B., Cosmo: A life-like animated pedagogical agent with deictic believability , En Working Notes of the IJ-CAI '97 Workshop on Animated Interface Agents: Making Them Intelligent.
4. Prensky, M. Digital Game-Based Learning McGraw-Hill Trade. Diciembre, 2000

5. Lester, J. C.; Converse, S. A.; Kahler, S. E.; Barlow, S. T.; Stone, B. A. and Bhoga, R. S. The Persona Effect: Affective Impact of Animated Pedagogical Agents. In *Proceedings of the Conference on Human Factors in Computing Systems*, 359-366. Atlanta, USA. 1997
6. Fairclough, C. and Cunningham, P. A Multiplayer Case Based Story Engine. In Proc. Of the 4th International Conference on Intelligent Games and Simulation (GAME-ON), Noviembre 2003.
7. McFarlane, A., Sparrowhawk, A. y Heald, Y. Report on the educational use of games. TEEM: Teachers Evaluating Educational Multimedia, 2002
8. Lindholm, T. and Yellin, F. eds. The Java Virtual Machine Specification. 2nd Edition. Oxford: Addison-Wesley. 1999