

Trazado del Aprendizaje de las Reglas de un Juego de Ingenio por Parte de Niños con Síndrome de Down

Krause, W.¹, Britos, P.^{1,2}, García-Martínez, R.^{1,2}

¹ Centro de Ingeniería del Software e Ingeniería del Conocimiento. Escuela de Postgrado. ITBA. Buenos Aires. Argentina

² Laboratorio de Sistemas Inteligentes. Facultad de Ingeniería. Universidad de Buenos Aires. Argentina

Resumen

Sabidas las dificultades que poseen los niños con Síndrome de Down para aprender exclusivamente a partir del lenguaje expresivo oral, ¿Se puede a partir de la evaluación de su desempeño ante un juego interactivo inferir la mejor manera de aprender a través de estímulos visuales y a partir de ello, mejorar los métodos de enseñanza para que se adecuen a esta situación? El objetivo del sistema será el de trazar la actuación de un niño con Síndrome de Down ante un problema planteado en modo de juego de ingenio donde fundamentalmente deberá hacer uso de habilidades Lógico-Matemáticas y Espaciales, que ayudando al niño en forma interactiva, lo irá guiando en el aprendizaje de sus reglas. El trazado de la actuación del niño podrá ser analizado posteriormente con otras herramientas de software y dicha información podrá ser de utilidad para los investigadores del área educativa a fin de poder adaptar los métodos de enseñanza centrándose en las fortalezas de los niños con necesidades educativas especiales.

Palabras clave: Aprendizaje, educación, sistema de ayuda .

1. Introducción

Cuando se descubre la atracción que los niños con SD tienen por las computadoras, surgen algunos programas educativos con la finalidad de facilitar su aprendizaje, sin embargo, la mayoría de ellos está destinado a mejorar su mayor déficit, es decir, el lenguaje [1]. También existen otros para el apoyo en operaciones matemáticas básicas [2]. Asimismo, existe gran cantidad de software educativo y lúdico para toda la población infantil orientados a la resolución de problemas de ingenio (puzzles en general), pero que no guardan registro de las decisiones tomadas por el niño, ni tienen métodos cuidados para asistirlo en su aprendizaje. Sería de utilidad entonces para los investigadores de procesos de aprendizaje de niños con Síndrome de Down (SD) disponer de un programa que mientras le enseña al pequeño a jugar, permita el registro de cada decisión

tomada por el niño durante dicho proceso sobre el cual se conoce literatura [3], [4], [5]. Esto posibilitará una revisión posterior que incluye por ejemplo obtener información sobre los aciertos, las clases de errores cometidos, en qué circunstancias, si aprende con las ayudas recibidas. En particular, si el juego es de los tipificados como “juego de ingenio”, rompecabezas o puzzles, el niño podrá mejorar sus habilidades lógico-matemáticas y espaciales [6]. También sería interesante minimizar el uso de recursos lingüísticos, debido a que en esta área se encuentran las mayores dificultades de desarrollo del niño, haciendo que dicha variable no intervenga en el proceso. El objetivo de este proyecto, está orientado a dar apoyo a la investigación sobre procesos de aprendizaje en niños con SD. Se buscarán aportes de la teoría de Inteligencias múltiples [7], con el fin de desarrollar un software multimedia que evite el uso de lo que Gardner denomina inteligencia lingüística. Para ello, los sonidos que emita el programa, deberán manifestar los estados de ánimo del instructor (la computadora en este caso) y serán acompañados por imágenes acordes, sin pronunciar palabra alguna. De igual manera, se evitará la inclusión de cualquier vocablo escrito. El niño interactuará con el programa a través del uso del mouse o del teclado, sin esperar la emisión de vocablos. También podrá utilizarse para que en posteriores análisis, el investigador encuentre similitudes o diferencias con otras poblaciones de niños (hipoacúsicos, con diferentes problemas neurológicos, niños sin NEE (Necesidades Educativas Especiales)). Se considerará exitoso este proyecto, si el investigador que hará uso de la herramienta verifica que un niño testigo esté en condiciones de avanzar en el aprendizaje del juego y disponga del registro de interacciones del niño con el programa.

2. Estado de la Tecnología

La mayoría del software desarrollado para esta población, se orienta a enseñar al niño a hablar, en general, haciendo uso de imágenes, inclusive, animaciones faciales que gesticulan los vocablos. Podemos incluir entre ellos el proyecto Icatiani [8], cuyo principal objetivo fue el desarrollo de un software que apoya la adquisición del lenguaje ajustándose a

algunas de las necesidades de la escuela de educación especial. Suele incluirse simultáneamente el texto escrito, a modo de introducción a la lectura, como lo hace el programa SOALE [9], que “es un método de lenguaje - escritura, es decir, no se propone, como único objetivo la enseñanza de las habilidades lectoras, sino que, mediante estrategias adaptadas a los déficits de los niños con SD, se pretenden desarrollar las habilidades lingüísticas de los niños y, más adelante y si es posible, abordar la enseñanza de la lectura.”. Para el apoyo de las matemáticas, podemos encontrar un Sistema Tutorial Inteligente, independiente del dominio particular de enseñanza, para la ayuda de alumnos con necesidades educativas especiales [2]. En este caso, orientado a trabajar con dos tipos de NEE: Síndrome de Down y dislexia. “La adaptación del sistema se basa en la secuenciación de los conceptos a enseñar, la dinámica de la presentación, la interacción del sujeto con el sistema y la presentación de los estilos de aprendizaje según la secuencia de eventos y tipo de tareas presentadas. A través de esto, el sistema se ajusta a las características individuales de los alumnos.” Se ha acotado el alcance al software desarrollado en lengua española. Sin embargo, lo aquí presentado es bastante descriptivo de una población de similares características en lengua inglesa, donde el material disponible es mucho más vasto. Cabe destacar que no se ha encontrado (en idioma alguno) software que registre los procesos de aprendizajes lógico-matemáticos y espaciales, evitando el uso del lenguaje expresivo oral o escrito.

3. Identificación del problema

Hace casi 200 años se pensaba que una persona ciega no sería capaz de leer. Sin embargo Lois Braille, quien tenía prácticamente perdida su visión por una infección, creó a partir de un sistema de comunicación militar el lenguaje hoy ampliamente difundido. Nadie piensa ya que una persona ciega no puede leer, aunque utilice otros métodos. Para el caso de niños con SD, Chapman y Hesketh observan que el fenotipo conductual en los niños con SD se caracteriza por “retraso mental, acompañado de un déficit específico adicional en el desarrollo del lenguaje...” [8]. Lo que se pretende a través de este trabajo de tesis es aportar una herramienta que ayude a contestar la siguiente pregunta: ¿existirá un proceso que ayude a una persona con SD a adquirir habilidades lógico-matemáticas y espaciales que no haga uso del lenguaje expresivo oral?. El software en general está orientado a personas sin NEE. Cuando vemos un video-juego de acción, que al finalizar un nivel (que puede ser exitoso o no), muestra al protagonista contento, no deja dudas que el nivel fue superado. ¿Qué significado puede tener para un niño que no sabe leer las palabras GANA o PIERDE al finalizar la etapa?

En la investigación de procesos, la observación de una o más sesiones sería suficiente para la observación de niños sin NEE. Pero si por el contrario, el niño no avanza en las pruebas planteadas, se pueden presentar, entre otros, los siguientes inconvenientes:

- El proceso se dilata en varias sesiones
- El proceso se descontextualiza.
- El estado de ánimo en el momento de la observación es variable.

Por el contrario, si el observador es una computadora, que registra cronológicamente las acciones que el programa lleva a cabo en conjunto con las decisiones que toma el usuario, en principio se puede decir que es una tarea viable. Para ello, bastaría con diseñar adecuadamente la información a guardar. La sesión comienza en el momento que el niño quiere jugar y no depende de la hora, del día. Cuando el niño lo decide, deja de jugar y cuando quiere, vuelve a empezar. Este proceso de aprendizaje puede durar días o meses, dependiendo no sólo de las dificultades en el aprendizaje, sino también con la familiaridad que tenga con las computadoras. Asimismo, distintos investigadores podrán intercambiar información sobre estudios suministrados con una contextualización similar. Si el investigador cuenta con una herramienta que mientras guía al niño en el aprendizaje de habilidades evitando el uso del lenguaje (oral o escrito), y mientras el niño aprende, va dejando sus huellas de forma tal que el investigador pueda reconstruir el proceso seguido, el objetivo se habrá alcanzado. A partir de la generación del reporte, los posteriores análisis que hace el investigador con él (comparaciones entre pares dentro de una población o con otras poblaciones en forma individual o en conjunto, estadísticas, etc.) están fuera del alcance de este proyecto.

4. Solución propuesta

Siguiendo la línea de Gardner, recogemos recomendaciones acerca de cómo mejorar las habilidades lógico-matemáticas, fortaleciendo las acciones de “Experimentar, preguntar, resolver rompecabezas lógicos, calcular, etc.” [6]. Tomando como premisa que el juego es un facilitador del aprendizaje (según Piaget y Vigotzky entre tantos), el niño deberá ver al programa de esa manera, como un juego. Si se administra este programa a un niño, será porque un conocimiento básico de uso de computadora tiene, en general, y dicho conocimiento seguramente será a través de un juego. Es un objetivo que el niño no note diferencias entre este programa y un juego de computadora, quedando las tareas de gestión del reporte lejos del alcance del entorno que pueda manejar el niño. El juego es multimediado. El niño va aprendiendo habilidades que le permiten ir incorporando los

conocimientos necesarios para demostrar que ha aprendido las reglas del juego. Para ello, el programa va variando los tiempos de espera (timeout) para mostrar la ayuda al movimiento que debe realizar el niño. Ese tiempo depende de la situación en que se encuentre el desarrollo de la partida, de partidas anteriores y de lo adecuado o no que haya sido el movimiento similar previo. Para gestionar el proyecto se usó como apoyo la metodología Métrica V3 [11] del Ministerio de Administraciones Públicas Español. Para la implementación se planeó el desarrollo en entorno Windows (sin resignar futura portabilidad a Linux - fuera del alcance del proyecto). El entorno de desarrollo soporta librerías 3D, que dada la portabilidad esperada, son orientadas al uso de OpenGL (www.opengl.org). Se estimó a priori que el lenguaje de programación fuera Delphi, sin descartar otros. Se utilizó el paradigma de orientación a objetos para el desarrollo del software. Sin embargo, el desarrollo de software para animación 3D, tiene sus particularidades, en especial, en lo que respecta precisamente a la animación de las escenas que se consideraron tanto en el diseño como en el desarrollo. Los objetos 3D, estáticamente se comportan como cualquier otro objeto en programación clásica, sin embargo, dinámicamente, no alcanza con indicarle por ejemplo a un cubo que se desplace, sino, hay un evento (temporal) que al dispararse le indica que llegó el momento de reubicarse. Cada escena que se ve en pantalla (pueden ser del orden de 220 imágenes por segundo – fps de Frames per Second), muestra a los objetos en su posición actual, para lo cual previamente, cada objeto debió tomar su posición correcta. En este sentido, OpenGL cuenta con “Listas de Despliegue, que son una manera de almacenar comandos de dibujo en una lista para un trazado posterior” [2]. Un entorno gráfico resulta agradable si sus objetos tienen realismo. Para ello, los objetos 3D se diseñaron, no solo para cumplir con colores y dimensiones, sino, también considerando sus texturas y movimientos, como así también la iluminación de la escena y la posición de las cámaras.

5. Descripción del problema

El sistema tiene una interfaz visual que permite al jugador interactuar con las piezas. El registro de las acciones tomadas por el usuario y los eventos generados por el sistema, se realiza en forma automática. La Figura 1 describe la posición inicial del tablero cuando el jugador tiene asignadas piezas blancas:

Una vez iniciada la partida, pueden darse en cuatro eventos de interacción del usuario haciendo uso del clic del mouse sobre el objeto correspondiente, y que se detallan a continuación: Selección de la pieza, Selección del casillero, Rotación de la pieza y Aceptación de la jugada. La regla básica del juego es que los colores de los lados o de las piezas vecinos coincidan. Una vez

aceptada la jugada, el oponente está en condiciones de jugar. En el caso del sistema, realiza los movimientos inmediatamente después que el usuario finaliza su jugada. Los desplazamientos son cadenciados y dan una idea de desplazamiento real. El objetivo del juego es el de colocar en el tablero más fichas que el oponente, sin embargo, el objetivo del sistema es que el niño aprenda las reglas de juego (esperar su turno para jugar, poder realizar los cuatro eventos, seleccionando en cada caso los objetos adecuados. Cuando el niño demora en tomar una decisión adecuada, el sistema lo guía a través de ayudas visuales y sonoras. En caso de selección inadecuada de los objetos, el sistema lo hace saber también a partir de señales visuales y sonoras. Las ayudas del primer tipo, se manifiestan a través de un rayo de luz verde clara que partiendo de la esfera ilumina el objeto a seleccionar. Los errores se verifican a través de la iluminación en color rojizo del objeto. Al finalizar la partida, se manifiesta el resultado de la misma en formas diferenciadas entre el triunfo, empate o derrota. Se ha buscado que los movimientos sean armónicos, con lo cual las piezas se desplazan y no desaparecen de su posición de origen para aparecer en su posición de destino. Asimismo, las rotaciones se realizan a una velocidad angular de 90° / segundo, dando así un sentido de movimiento real. Las manifestaciones visuales de ayudas y de errores sonidos son acompañan por señales sonoras que ayudan a identificar el tipo de evento acontecido.

Desde el punto de vista del diseño, se ha buscado una subdivisión que facilite la evolución y el mantenimiento del producto. Por ejemplo, si dado el caso en futuras versiones, el soporte de OpenGL no es el adecuado para nuevos sistemas operativos, y se decide migrar a otra biblioteca 3D, solamente deberán reemplazarse los subsistemas que hacen referencia a dicha biblioteca y no el resto. Para ello, el patrón de diseño que mejor se aproxima al problema es el Modelo-Vista-Controlador (MVC). En la Figura 2 se presenta el diagrama que subdivide el sistema desde el punto de vista de la arquitectura de diseño del software.

De esta forma, también será posible realizar modificaciones en la forma de “pensar” las jugadas, donde sólo se afectará al modelo o el momento en el cual realizar las ayudas en el cual exclusivamente se verá afectado el controlador. El uso de una fuente común de definiciones externa, permite aislar la definición de clases de intercambio entre los distintos componentes, facilitando el reemplazo completo del componente respetando exclusivamente lo definido en la unidad Common.

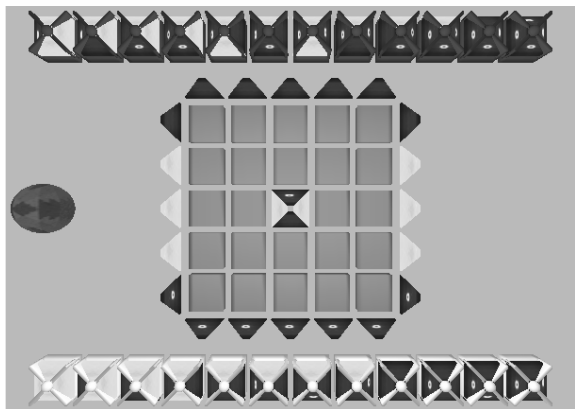


Figura 1. Inicio de Partida con Piezas Blancas

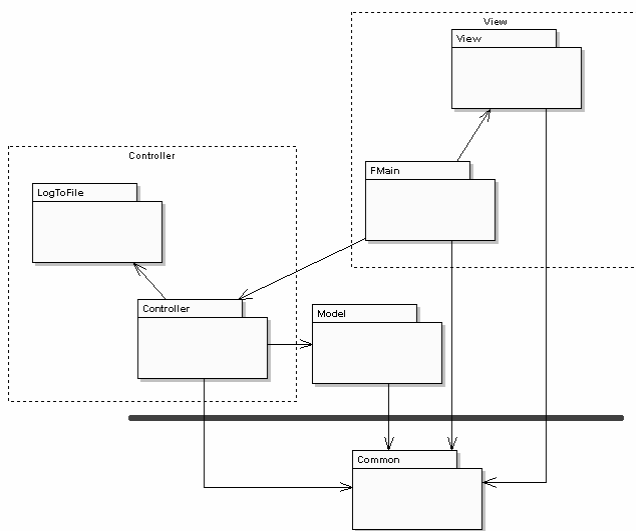


Figura 2. Arquitectura del Sistema

Cada componente de la Arquitectura MVC se implementó de la siguiente forma:

- View:
 - Unidad View: Contiene las clases visuales propias del sistema. Recibe los mensajes de la interfaz de usuario.
 - Unidad Fmain: Ventana principal del sistema que contiene el escenario OpenGL. Envía mensajes a las unidades View y a Controller y registra los eventos de interacción que publica Controller (ayuda, error, fin de juego).
- Controller:
 - Unidad Controller: Contiene las clases de control de la partida, diagrama de estados de la jugada y timeout dinámico para cada estado. Se comunica con la Unit Model para requerir jugadas posibles o verificar cumplimiento de reglas del juego. Se comunica con la

unidad LogToFile para registrar los eventos programados (movimientos, ayudas, errores, etc.).

- Unidad LogToFile: Contiene la clase que permite el registro de eventos.

• Model:

- Unidad Model: Contiene las clases abstractas de los objetos del juego y las reglas del mismo. Publica funciones para acceder a jugadas posibles.

Finalmente, la unidad Common tiene la definición de los valores constantes del juego que son comunes a todas las unidades y que definen los colores de los lado de las piezas, los bordes del tablero, los colores de cada pieza y también funciones que permiten acceder a los datos de las piezas sin necesidad de conocer cómo se llama la clase implementada en otras unidades.

6. Caso de estudio

A continuación se presenta una serie de ilustraciones que ponen de manifiesto el desempeño de distintos casos que se suscitan durante una partida:

- [a] Jugada completa de una pieza sin necesidad de rotarla (Figuras 3 a 6). El usuario elije la pieza que se encuentra en su sector de piezas sin usar. Esto lo hace haciendo clic con el mouse sobre ella. En la Fig.3. se observa la pieza ya seleccionada, es decir desplazada verticalmente desde su posición de reposo y la flecha (no es parte del entorno visual del juego) señala el casillero donde se desea jugar. Luego se indica con el clic del mouse sobre el casillero señalado por la flecha y como se observa en la Fig.4., la pieza se desplaza horizontalmente hacia su destino. En la Fig.5. la pieza se encuentra lista a ser aceptada (o rotada, cosa que no se hizo en este caso). Al hacer clic con el mouse se observa el descenso hasta quedar en la posición vertical de reposo como se ve en la Fig.6.

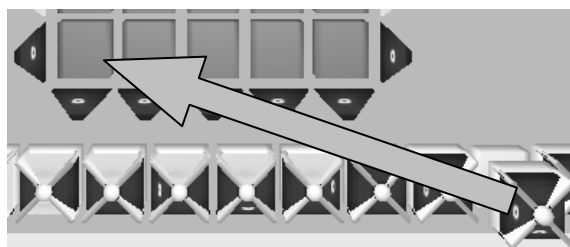


Fig. 3. Selección de Pieza a Jugar (se indica con la flecha el objetivo)

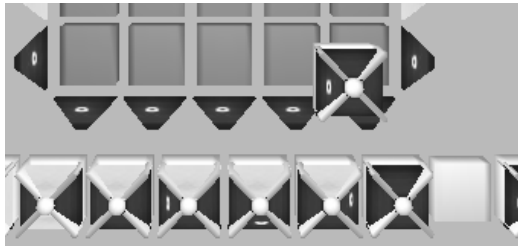


Figura 4. Desplazamiento hacia el Casillero Seleccionado

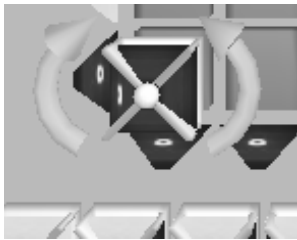


Figura 5. Pieza sobre Casillero Seleccionado

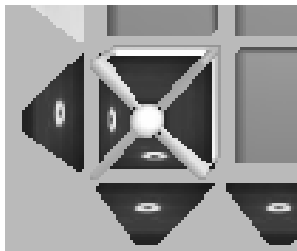


Figura 6. Pieza Aceptada

[b] Ejemplos de manifestaciones de ayuda (uno por cada estado posible de la jugada en Figuras 7 a 10). Las ayudas acontecen al cumplirse el timeout dinámico correspondiente a cada estado y van acompañadas de una señal sonora. En la Fig.6, se observa cómo se ilumina la pieza a modo de ayuda. En la Fig.8, se ve que la pieza fue seleccionada y la ayuda indica un casillero donde puede jugarse. En la Fig.9, se observa otra pieza que necesita rotarse para poder aceptarse en el casillero seleccionado donde la ayuda se manifiesta sobre el indicador de giro correspondiente (sentido horario en este caso). Una vez rotada la pieza, se observa en la Fig.10, como la ayuda se dirige hacia la pieza con la intención de que el usuario haga clic con el mouse sobre ella para que sea aceptada.

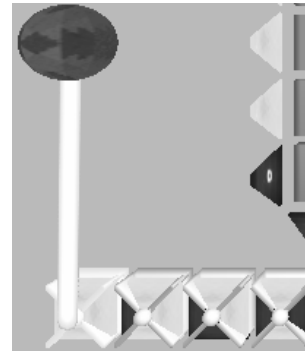


Figura 7. Manifestación de Ayuda sobre una Pieza Posible

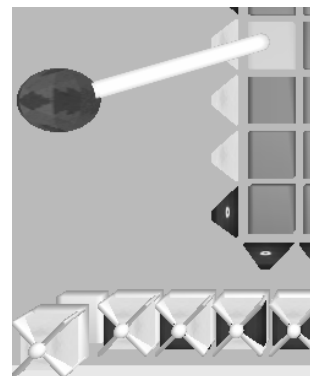


Figura 8. Manifestación de Ayuda sobre Casillero Posible

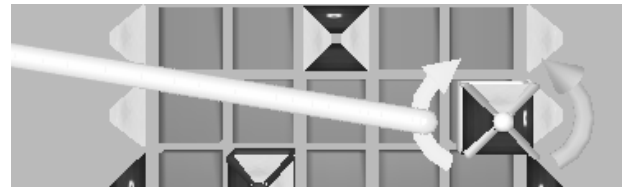


Figura 9. Manifestación de Ayuda para Rotación de pieza en sentido horario

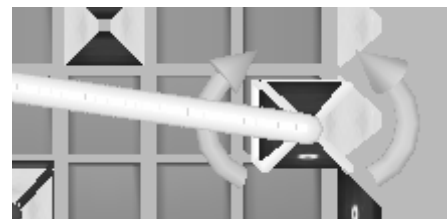


Figura 10. Manifestación de Ayuda para aceptación de Pieza tras ser rotada

[c] Ejemplos de manifestaciones de error (Figuras 11 y 12). Siempre las manifestaciones de error visuales son acompañadas por el sonido de error adoptado. En la Fig.11 se observa que el usuario intentó aceptar una pieza que tal como está orientada no puede aceptarse en el casillero elegido, pudiendo en este caso rotar la pieza o elegir otro casillero apto antes de aceptarla. En la Fig.12, se observa

la manifestación de error al elegir un objeto del juego que no tiene sentido como es un casillero de posición inicial del oponente.



Figura 11. Manifestación de Error al intentar Aceptar la pieza sin rotarla previamente

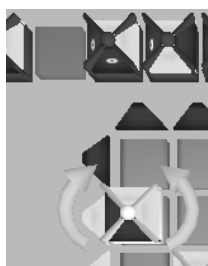


Figura 12. Manifestación de error al elegir un Casillero libre del oponente

[d] Rotación de una pieza, que se realiza en forma cadenciada (Fig.13.). Esto ocurre cuando el usuario ha hecho clic sobre el indicador de giro (en este caso la flecha de la izquierda).



Figura 13. Pieza en rotación en sentido horario

El sistema ha sido informalmente probado en adultos y los comentarios, en general de aprobación, siempre fueron acompañados de sugerencias, algunas de ellas se incluyen en la bibliografía. Curiosamente, los casos de pruebas en niños (sin necesidades educativas especiales), al momento reflejan que se adaptan fácilmente a aprender las reglas y sus sugerencias apuntan hacia la inclusión de distintos niveles de dificultad, escenarios de fondo y todo aquello que al momento “como juego” el sistema no dispone. Al momento, se está evaluando un ámbito adecuado en el que se puedan realizar experiencias a fin de obtener una causística apropiada para realizar los análisis de los

datos registrados a través de técnicas estadísticas o de minería de datos.

Conclusiones

Una de las cuestiones planteadas al inicio del proyecto, fue saber si el software era o no un Sistema Tutorial Inteligente (STI). Una vez finalizado el mismo, se podrá disentir en cuánto se acerca o no a un tutorial inteligente, ya que funcionalmente pareciera serlo. Lo que si se puede corroborar es lo dificultoso que resulta el diseño e implementación de una interfaz gráfica que cumpla con los requisitos que en general tiene un Sistema de estas características. La comunicación visual alcanzada ha sido más que satisfactoria. Las ayudas resultan claras como así los desplazamientos. Para resumir la forma de aprendizaje de las reglas del juego por parte del niño, podemos decir que lo hace en base a prueba y error con ayudas, es decir, encuadra en la línea del Descubrimiento Guiado, uno de las categorías de STIs. Finalmente, el juego tiene ciertas similitudes respecto de otros juegos de mesa como son el ajedrez y las damas. Existen piezas blancas y piezas negras. Las blancas comienzan a jugar. Se juega alternadamente una pieza por jugador. Esto implica que la variable conocimiento de juegos similares sea de interés para el investigador. Se prevén como futuras líneas de investigación y desarrollo:

- [a] Interfaz Multimedia: hacer que las piezas encastran (como piezas de puzzles), es decir, que no sólo relacionen colores sino formas adaptativas, hasta la inclusión de personajes animados que ayuden, guíen, jueguen por el oponente, festejen las jugadas correctas y todo aquello que pueda aportar a mejorar la interacción y agregar música de fondo contextualizada, por ejemplo que aumente el volumen a medida que se acerca el momento de ayudar al usuario. Asimismo, se podrán mejorar los sonidos que indican el error o la ayuda.
- [b] “Inteligencia” del Modelo: la operación “Pensar Jugada” se implementó con una simple heurística consistente en contar la cantidad de posibles jugadas que tiene cada pieza y quedarse con la que menos tiene. El algoritmo funciona adecuadamente, pero adolece del problema que sobre todo en las primeras movidas muchas piezas se equiparan en el número mencionado, con lo que la resolución adoptada se toma en forma aleatoria. Se estudian heurísticas mucho más complejas, pero que permitirán anticipar jugadas y evaluar tanto en qué estado quedan las piezas propias como las del oponente. Con ello, se podrá seleccionar el perfil de personalidad del oponente en agresivo y conservador o aún en que grado de uno y otro perfil se encuadra. Cabe destacar que las combinatorias posibles, exceden las capacidades de cálculo de las PCs para el problema dado, de ahí el hecho de utilizar técnicas alternativas al recorrer todas las soluciones posibles del problema.
- [c] Análisis de los datos obtenidos: En este sentido, se ha hecho lo básico, es decir, el trazado de cada una

de las partidas. Pero el análisis que se puede hacer individualmente podrá profundizarse si dicha información se registra en una base de datos. Dicha base de datos, estando disponible para otra aplicación, permitirá que investigador pueda realizar desde estadísticas complejas hasta el uso de técnicas de Minería de Datos. Para ello, deberán realizarse un par de tareas previamente. Por un lado, el agregado de la información correspondiente al niño al que se evalúa (edad, escolaridad, conocimientos de otros juegos, experiencia en la computadora, etc.). Por el otro, se tratará de sintetizar la información correspondiente a partir del proceso de cada partida (duración, si fue completada o no, cantidad de errores, cantidad de veces que se ayudó en cada estado, etc.) y posteriormente al grupo de partidas evaluadas (sumas, promedios, desvíos de las mencionadas).

- [d] Interacción Investigador – Usuario: No existe al momento una función que permita al padre del niño que instala en su casa el software que exporte los registros cronológicos en forma automática. Lo deseable, sería que el sistema pueda realizar dicha tarea por email, sin más que pulsar una combinación de teclas adecuadas. Otra posibilidad interesante, es que el investigador pueda hacer una evaluación a distancia de los avances del niño, jugando él mismo una partida. Para ello, habría que facilitar la conexión por red entre las aplicaciones de ambos actores.

Referencias

- [1] NewComb, R., *Enseñanza del lenguaje asistida por computador*, web del Laboratorio de Comunicación Oral de la Facultad de Informática de la UPM
- [2] González C, 2000, *Sistema tutorial inteligente para la enseñanza en niños con dificultades intelectuales y cognitivas*. Universidad La Laguna. Pagina web vigente al 08-05-2006. http://sid.usal.es/mostrarficha.asp_Q_ID_E_4786_A_fichero_E_8.4.3
- [3] García Escamilla, S. 1983, *El niño con síndrome de Down*. (Ed. Diana)
- [4] Mosston, M. y Ashworth, S. 1990. *The spectrum of teaching styles from command to discovery*. White Plains, NY: Longman. 327 pages. 0801303508. Location: Dallas SIL Library 371.1 M914. Interest level: specialist. Pagina web vigente al 08-05-2006. <http://www.uned.ac.cr/servicios/global/ensenanza/instruccion/articulos/sistemas.htm>
- [5] Troncoso, M. y Del Cerro, M., 1998, *Síndrome de Down: Lectura y Escritura*. Ed. Mason – Barcelona.
- [6] Gorriz, B., 2001, *Inteligencias Múltiples*. Pagina web vigente al 08-05-2006. <http://www.ilustrados.com/publicaciones/EpZlypuFAIEZqBJhxP.php>.

[7] Gardner H., 1996, *Inteligencia – Múltiples perspectivas*, traducción al español 2000 (Ed. Aique)

[8] Toledo Ma., Kirschning I., 2001, *Icatiani: Un Sistema de Apoyo para la Adquisición del Lenguaje*, Universidad de las Americas, Puebla, Mexico (UDLAP). Pagina web vigente al 08-05-2006. <http://mailweb.udlap.mx/~ingrid/ingrid/Icatiani1.pdf>.

[9] Cana López, D., 1996, SOALE, *Sistema para el aprendizaje de la lectura – escritura para niños con SD*. Pagina web vigente al 08-05-2006. <http://www.geocities.com/Athens/Atrium/5189/>

[10] Chapman, R. y Hesketh, L., 2000, *Fenotipo Conductual de las personas con síndrome de Down*. Universidad de Wisconsin-Madison – Traducción Fundación Síndrome de Down de Cantabria Pagina web vigente al 08-05-2006. <http://empresas.undivia.es/downcan/fenotipo.html>.

[11] Métrica V3, 2000. *Metodología de Planificación, Desarrollo y Mantenimiento de Sistemas de información*. Ministerio de Administraciones Públicas Español.

Dirección de Contacto del Autor/es:

Ing. Walter Otto Krause

M. Ing. Paola Britos

Dr. Ramón García-Martínez

25 de Mayo 444 – 6to piso

Ciudad Autónoma de Buenos Aires

Argentina

e-mail: {capis, pbritos, rgm}@itba.edu.ar

sitio web: <http://www.itba.edu.ar/capis>

Ing. Walter Otto Krause. Es Ing. Electrónico del Instituto Tecnológico de Buenos Aires. Desde 1991 es Jefe de Proyectos de SoftLab a cargo de soporte técnico de clientes, soporte pre-venta, consultoría en modelado de datos, análisis, programación y mantenimiento de sistemas en grandes corporaciones.

MA. Ing. Paola Britos. Es Magister en Ingeniería del Conocimiento por la Universidad Politécnica de Madrid. Es Docente-Investigador Categoría III SPU-MECyT. Es Investigadora del Centro de Ingeniería del Software e Ingeniería del Conocimiento del ITBA. Es Investigadora del Laboratorio de Sistemas Inteligentes de la Facultad de Ingeniería de la UBA.

Dr. Ramón García Martínez. Es Doctor en Informática por la Universidad Politécnica de Madrid. Es Docente-Investigador Categoría I SPU-MECyT. Es Director del Centro de Ingeniería del Software e Ingeniería del Conocimiento del ITBA. Es Director del Laboratorio de Sistemas Inteligentes de la Facultad de Ingeniería de la UBA.
