

Pruebas de laberintos 2D y 3D con propósitos de uso en rehabilitación neuropsicológica

Víctor Tomás Tomás Mariano¹, Felipe de Jesús Núñez Cárdenas¹,
Raúl Hernández Palacios¹ y Mariano Pozas Cárdenas²

¹Universidad Autónoma del Estado de Hidalgo, Escuela Superior de Huejutla,
Huejutla de Reyes, Hidalgo, México
{victor_tomasm, felipe.huejutla, rpalacios}@hotmail.com

²Universidad Autónoma del Estado de Hidalgo, Centro de Investigación en Tecnologías de la
Información y Sistemas (CITIS),
Pachuca de Soto, Hidalgo, México
mpozas@hotmail.com

Resumen En el presente trabajo se muestran los resultados obtenidos del test de laberinto, al permitir recorrer un entorno virtual, los laberintos son ampliados según la configuración del especialista, estos laberintos permiten tener una mayor movilidad y direccionalidad al recorrerlos. Los laberintos generados son graficados en 2D y 3D; que satisfacen los indicadores valorados por especialistas médicos psicólogos para pruebas psicométricas. Una de las ventajas del sistema es que permite visualizar la solución del laberinto en 2D, se utilizan técnicas de búsqueda para encontrar la ruta-conexión entre dos celdas.

Palabras clave: laberintos ampliados, rehabilitación, algoritmos de búsqueda, graficación 2D y 3D.

1. Introducción

Los laberintos rectangulares son de los más utilizados en diferentes campos como lo es la robótica, educación, medicina y entretenimiento [1, 3, 17]. Existen varios algoritmos para generarlos [2], en este documento se realiza una descripción de los algoritmos más utilizados para la construcción de laberintos de conexión simple (LCS). También se generan LCS con pasillos ampliados en 2D y 3D, estos laberintos brindan una mayor movilidad y direccionalidad al recorrerlos, permitiendo tener laberintos de diferente complejidad, además de ser los que se utilizan en pruebas de rehabilitación. Se muestran los resultados obtenidos en la prueba de test del laberinto. El presente artículo está organizado de la siguiente manera: en el apartado 2, se resalta el uso de la realidad virtual en el proceso de rehabilitación y tipos de pruebas de laberintos; en el apartado 3 y 4 se habla de los tipos de laberintos y de los algoritmos para construirlos; en el apartado 5 se introduce a los laberintos ampliados y el problema que implica resolverlos mediante el algoritmo A*; en las secciones 6 y 7, se muestran los resultados obtenidos en laberintos 2D y 3D respectivamente; en el

apartado 8 y 9 se habla de las aplicaciones posibles y conclusiones generales del proyecto.

2. Laberintos y rehabilitación médica

La manera más común de construir laberintos es mediante un programa de computadora, y se ha puesto mucho énfasis sobre el potencial para el uso de ambientes virtuales en contextos médicos. Esta área se ha enfocado principalmente, en el tratamiento y rehabilitación de pacientes con alguna discapacidad psicomotriz [15, 16, 17].

El desarrollo de aplicación de los ambientes virtuales ha sido usado en la evaluación neuropsicológica (*Neural Assessment*) y rehabilitación cognitiva (*Cognitive Rehabilitation*) de componentes en procesos cognitivos incluyendo: *atención, memoria, habilidades espaciales y disfunciones del sistema nervioso central (Central Nervous System)* [9, 19, 24].

La evaluación neuropsicológica es una ciencia aplicada que evalúa cómo actividades específicas en el cerebro, son expresados en comportamientos observables, usa herramientas de evaluación psicométricas para el diagnóstico de disfunciones [15, 19]. La tecnología virtual brinda el potencial para desarrollar ambientes de prueba que pueden complementar los procedimientos de evaluación neuropsicológica existentes que dependen *principalmente de lápiz, papel y observación conductual*. Usado de esta manera, los ambientes virtuales para rehabilitación pueden mejorar la confiabilidad psicométrica y la validez puede causar una mejor detección y diagnóstico de las diferentes formas de disfunción del Sistema Nervioso Central [9]. Los ambientes de realidad virtual, constituyen una oportunidad interesante para la evaluación de la desorientación topográfica, proporcionando una representación de naturaleza dinámica e interactiva [18]. Entre los trabajos que se pueden mencionar está el de Nair et al, en donde la prueba del laberinto virtual, permite visualizar el recorrido realizado por un sujeto con discapacidad [22]. El procedimiento es comparado con un sujeto normal. Se puede observar que el medio por donde se realiza el recorrido, no necesariamente tiene la forma de un laberinto con paredes rectangulares, en vez de ello, se colocan obstáculos de diferentes formas dentro del ambiente [16].

Otro trabajo es el que presenta Bardorfer A. en donde el principal objetivo es hacer estudios funcionales de los miembros superiores (*Upper Limbs*) en pacientes con enfermedades neurológicas (Neurological Diseases). Los laberintos de prueba que se utilizan son 3D, de tamaño y ampliación de pasillo fijo. Entre los pacientes que desarrollaron la prueba se encuentran aquellos que tiene alguna disfunción psicomotriz, como lo es la enfermedad de Parkinson, Ataxia y Esclerosis Múltiple, para recorrer el laberinto, se utiliza una “palanca” que controla el movimiento de un “apuntador” (ball) usando una interfaz háptica PHANTOM [1, 16]. Un trabajo similar es el realizado por el Domínguez et al. [5] en lo referente a interfaces y sistemas de control aplicados en laberintos, utiliza un volante como interfaz háptica

para controlar el movimiento realizado por un operador humano con discapacidad motriz en un laberinto virtual 3D.

Los trabajos antes mencionados consideran el uso de laberintos para rehabilitación en ambientes virtuales, estos utilizan laberintos prefijados con distintos niveles de complejidad en su estructura, pero no generan una solución automática del laberinto, en vez de ello, la solución se realiza por una persona sana o por el especialista de manera manual.

2.1. Pruebas psicométricas

Test Laberintos de Porteus (TLP): Fue diseñado en 1914 por S.D.Porteus, siendo propuesto originalmente como una prueba capaz de medir la inteligencia general de una persona en términos de edades mentales. La prueba consiste en la resolución de laberintos ordenados en un modelo de dificultad creciente, debiendo la persona trazar con un lápiz el camino desde la entrada hasta la salida cumpliendo consignas que permiten ubicarlo entre las pruebas de funciones de planificación. El TLP es empleado actualmente como un test asociado a la activación de la región frontal del cerebro, implicado en el factor de planeamiento y capaz de detectar errores de tipo perseverantico [12].

Laberintos de Foster. El laberinto de Foster se utiliza para evaluar destrezas manuales, memoria y aprendizaje. La tarea consiste en terminar el recorrido del laberinto según los criterios de la investigación: utilizando el menor tiempo posible, en transferencia de aprendizaje entre las manos, cometiendo el menor número de errores por ensayo, con o sin retroalimentación. Este ejercicio puede evaluar la memoria espacial y la solución de problemas. Las áreas en que se emplea son: aprendizaje, procesos cognoscitivos y senso-percepción. Generalmente, tanto la velocidad del trazado como los callejones de recorrido, son registrados. Típicamente se obtiene una curva de aprendizaje negativa, y la tasa de aprendizaje a través de varios ensayos, puede ser analizada. El laberinto ha sido diseñado para que pueda contabilizar automáticamente el número de errores y el tiempo total para completar el recorrido [12].

3. Tipos de laberintos

Laberinto de Conexión Simple (LCS). Son aquellos que tienen puntos muertos o callejones sin salida, por lo tanto, solo tienen una solución entre la entrada y la salida, en el proceso de buscar la solución, en ocasiones, se recorre gran parte del laberinto, causando frustración en el espectador [2].

Laberinto de Conexión Múltiple (LCM). Estos laberintos tienen más de una solución desde la entrada a la salida. En este tipo de laberintos, además de encontrar una solución, se requiere que sea la más corta. Tienen ciclos internos, lo que dificulta aún más el proceso de buscar la solución, ya que se debe tomar en cuenta en que secciones del laberinto ya se ha transitado, evitando dar giros en un mismo segmento del laberinto [2].

4. Construcción de laberintos

La construcción al azar permite generar gran cantidad de laberintos conteniendo corredores como las ramas de un árbol, persiguiendo la desorientación del explorador, siendo a su vez ésta la forma más fácil de implementarse en un programa de computadora. Los algoritmos explicados a continuación, generan laberintos LCS completamente conectados [2].

La construcción al azar se basa en dos enfoques, los cuales se enuncian a continuación:

- **Cavar túneles.** Como su nombre lo indica, se refiere a cavar túneles a lo largo y ancho del laberinto hasta cubrirlo en su totalidad, como la exploración de una mina. A este tipo de construcción, pertenecen:
 - Algoritmo de construcción Kruscal.
 - Algoritmo de construcción Aldous-Broder.
 - Algoritmo de construcción Prim's.
 - Algoritmo de construcción Recursivo Bactracker.
- **Levantar muros.** Consiste en elevar los muros de los corredores por todo el cuerpo del laberinto, como en la construcción de un edificio.

4.1. Algoritmo de Kruscal

Este algoritmo construye laberintos de conexión simple completamente conectados, levanta muros.

1. Especificar el tamaño o las dimensiones del laberinto de acuerdo con las habitaciones que se desea que contenga.
2. Dividir el laberinto en habitaciones en base a sus dimensiones y etiquetarlas con una identificación única.
3. Seleccionar una habitación al azar y cavar un túnel hacia una habitación adyacente (en caso de existir más de una, elegir al azar) sólo si tiene diferente etiqueta, y etiquetar la habitación o habitaciones que se unieron con la identificación de la habitación inicial; repitiendo este proceso hasta que todas las habitaciones tengan la misma etiqueta.

Marcar la ubicación de la entrada o del punto inicial de la exploración y la ubicación de la salida.

4.2. Algoritmo Aldous-Brother

Es una técnica simple, pero puede llevar gran tiempo la construcción de un laberinto, esto es porque se va moviendo al azar dentro del laberinto sin discriminar las habitaciones con las que ya se cavaron túneles. Los pasos se enuncian a continuación:

1. Especificar el tamaño o las dimensiones del laberinto de acuerdo con las habitaciones que se desea que contenga.
2. Dividir el laberinto en habitaciones en base a sus dimensiones.
3. Elegir al azar una habitación dentro del laberinto.
4. Moverse a una habitación vecina adyacente al azar y cavar un túnel hacia la habitación anterior en el caso de que la nueva habitación no sea parte de otro túnel, continuando con este paso hasta que todas las habitaciones sean parte del laberinto.
5. Marcar la ubicación de la entrada o del punto inicial de la exploración y la ubicación de la salida.

Por motivos de espacio no se describen los demás algoritmos, pero el resultado de estos siempre es un laberinto de conexión simple con una sola pista

5. Desarrollo

5.1. Introducción a laberintos ampliados

La representación matricial en “0” y “1”, del laberinto final de cada algoritmo de construcción; permite representar las celdas libres en blanco, celdas ocupadas de color negro, para indicar la posición (i, j) dentro del laberinto se utiliza en su totalidad la celda respectiva de la matriz, en la fig. 1 se indica con un color distinto la entrada E y salida S del laberinto.

Se puede aumentar la ampliación, para obtener nuevos laberintos ampliados, independientemente del algoritmo de construcción aplicado, lo que es una ventaja, ya que permite conocer la estructura del laberinto y por lo tanto la orientación en la ubicación de la salida, además, los laberintos ampliados tienen mayor movilidad permitiendo tener más direccionalidad respecto a los laberintos de una sola vía, ver fig. 1.

5.2. Búsqueda de ruta solución en laberintos ampliados

El problema de búsqueda de rutas, también conocido como *maze problem*, se ha discutido en varios trabajos de investigación [6, 7, 10, 23] para el caso de estudio en particular, se tiene un laberinto con múltiples vías o laberinto ampliado, en el que, dependiendo de las dimensiones y complejidad, el número de rutas posibles a encontrar dentro del laberinto se incrementa, ver fig. 1.

Un factor importante de las rutas posibles, depende de la ampliación k que tiene el laberinto, además, una vez abandonado el punto de entrada, el número de combinaciones posibles en los movimientos puede llegar a ser muy grande, lo que provoca que el explorador visite una mayor cantidad de posiciones en el laberinto, ya que por cada celda expandida pueden generarse otras más según la cantidad de vecinos transitables que ésta tenga, tal como se muestra en fig. 1, por lo que es necesario de algún modo “recordar” que secciones del laberinto ya han sido exploradas, y así, tomar la mejor decisión de movimiento en el espacio de búsqueda.

Visualizar un laberinto como un grafo, permite retomar el “comportamiento” de los algoritmos principales: Depth First Search (DFS) [4], Breadth First Search (BFS) [4], Dijkstra [11] y su posible implementación para encontrar una solución, de los que, el segundo y el último son óptimos pero el espacio en memoria necesario puede volverse crítico, mientras que el primero de ellos no promete una trayectoria con costo mínimo.

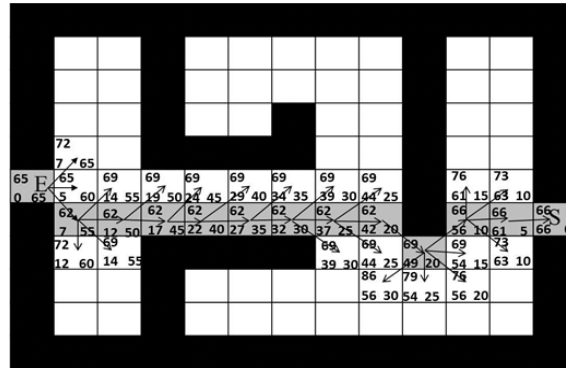


Fig. 1. Trayectoria óptima que enlaza la entrada y la salida del laberinto utilizando A*.

Otros algoritmos permiten modelar el problema en entornos tipo rejilla, y utilizar una matriz A de $m \times n$, en los que se requiere procesar datos dentro de una matriz binaria, ceros y unos, con la finalidad de identificar áreas libres y obstáculos en el laberinto. Entre los algoritmos analizados están principalmente: Algoritmo de Lee [13] [21] y Moore [14], Algoritmo de Hadlock [20], Algoritmo de Soukup [20], Algoritmo A* [8], siendo este último el elegido para ser implementado en un laberinto ampliado ya que la cantidad de nodos que explora es menor que cualquiera de los anteriores, y por garantizar la ruta más corta.

A modo de ejemplo, ver fig. 1, se utiliza un laberinto de ampliación $k=2$, con movimientos en horizontal, vertical y diagonal, donde éste último genera un costo más alto que los dos primeros, con fundamento en el teorema de Pitágoras. En el que para un movimiento en horizontal o vertical se asigna una magnitud de 5 unidades, generando entonces que el costo en diagonal se aproxime a 7 unidades, los cuales son tomados para trazar la ruta más corta en el laberinto, donde la celda E representa el punto de partida, la celda S el punto objetivo o meta, las celdas libres representan caminos posibles y las celdas de color negro representan los obstáculos.

6. Resultados en 2D

La representación de cada celda al graficar en pantalla, varía desde el valor mínimo de 1 pixel hasta m pixeles. Una de las características del sistema desarrollado es que permite almacenar los laberintos, dando la posibilidad de volver a “reproducir” una prueba de interés y ver el comportamiento del usuario durante el recorrido, ver fig. 2.

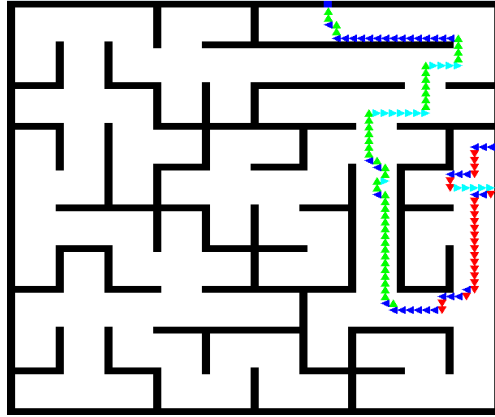


Fig. 2. Laberinto con $k=5$, celda = 10 píxeles.

En la fig. 2, se visualiza el recorrido realizado por el usuario desde el punto inicial con un cuadro de color rojo, y con un cuadro de color azul el punto final, los movimientos se visualizan de la siguiente manera: derecha de color azul, abajo de color rojo, a la izquierda de color cian, y arriba de color verde. Se muestra la posición actual del usuario, el movimiento realizado se resalta al cambiar el color del indicador, observe que se puede girar en una sección muy pequeña del laberinto ampliado.

Al finalizar el recorrido del laberinto, se muestra un mensaje informativo del tiempo de duración en llegar al final, una gráfica de latencias, que mide el tiempo que tarda el explorador en elegir el movimiento a realizar, total de movimientos efectuados y los choques realizados por el explorador en el recorrido del laberinto.

7. Resultados 3D

Una de las ventajas del recorrido de un laberinto en 2D, es que permite visualizar la salida del laberinto, que otorga al explorador la posibilidad de orientar sus movimientos de búsqueda hacia la salida. Sin embargo, para el recorrido en 3D, el proceso es más complicado, debido a que la vista del laberinto, se hace en una sección muy pequeña del mismo, esto desorienta al usuario.

En la interfaz 3D se tiene una pantalla principal con 2 cámaras: una delantera, parte izquierda y una trasera, parte superior derecha, además para ayudar al usuario, se muestra el laberinto en 2D, parte inferior derecha a fin de posibilitar su ubicación en el laberinto, ver fig. 3. Una vez abandonado el punto inicial, el explorador se interna en el laberinto teniendo una vista pequeña del mismo, ver fig. 3, esto hace que se pueda perder en una sección limitada del mismo.

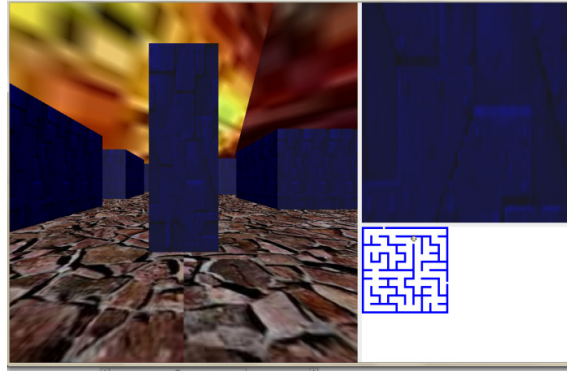


Fig. 3. Vista Interna al Recorrer el Laberinto en 3D.

8. Aplicaciones

Una de las aplicaciones de los laberintos es en la rehabilitación neuropsicológica, los laberintos son pruebas de gran utilidad para la detección de déficit de atención y síndrome frontal. Permiten evaluar diferentes niveles de integración nerviosa y trastornos específicos como secuela de una lesión cerebral [5], [9], [15].

La robótica es una de las áreas en donde el problema de planeación de trayectorias en ambientes o espacios abiertos dinámicos tiene gran aplicación, en ocasiones se cuenta con obstáculos extras en el camino, en estos casos, el robot se auxilia de sensores para su detección y con la implementación de algoritmos que brindan información local, se puede encontrar una solución [3], [6], [7], [10].

El conocimiento de la estructura del laberinto por donde se mueve el robot es de gran importancia ya que se debe tomar en cuenta la forma del propio robot para la planeación de trayectorias. En este sentido, se profundizó en los algoritmos que se aplican a laberintos de estructura conocida [2], [20]. Una posible mejora es abarcar situaciones que permitan modelar el problema cuando la salida y estructura del laberinto es desconocido, así mismo, se pretende llevar un “aprendizaje” del recorrido realizado.

Para la Graficación en 3D se hace uso del código implementado en el libro *Killer Game Programming in Java* By Andrew Davison, capítulo 25.

9. Conclusiones

Los resultados arrojados por el sistema permiten generar laberintos ampliados de diferentes tamaños y complejidades, totalmente conectados y perfectos. Las pruebas se guardan en un sistema de archivos para recuperar información.

El proceso de buscar una solución en un laberinto ampliado es más complejo, se tienen diferentes rutas que conectan la entrada y salida; de los algoritmos analizados

se optó por implementar el algoritmo A* para encontrar la ruta optima en el laberinto ampliado. Garantizando que la solución dada sea segura, es decir, libre de colisiones.

Como trabajo futuro, se pretende medir la diferencia entre el recorrido realizado por el usuario y la solución generada por el sistema con el objetivo de medir el grado de error presentado en secciones del laberinto.

Referencias

1. Bardorfer A., Munih M. Zupan A. and Primožic A., "Upper Limb Motion Analysis Using Haptic Interface" ", IEEE/ASME Transactions on Mechatronics, Vol. 6, No. 3, September 2001.
2. Bernabe S. E. "Construcción y Recorrido de Laberintos", Pachuca de Soto, Hidalgo, México, Julio 2004.
3. Buckley S. J., "Fast Motion Planning for Multiple Moving Robots", IBM T. J. Watson Research Center, 1989.
4. Cormen T. H., Charles E. L., Rivest R. L., and Clifford S. "Introduction to Algorithms", Second Edition. MIT Press and McGraw-Hill, Section 23.2: The Algorithms of Kruskal and Prim, pp.567-574, 2001.
5. Domínguez R. O. A., López M. V., Samperio L. R., "Resultados Preliminares sobre Interacción Háptica en Laberintos Virtuales, con Propósitos de Diagnóstico en Pacientes con Discapacidades Neuropsicológicas", Tesis, UAEH, CITIS, Hidalgo México, 2005.
6. Dracopoulos. D. C., "Robot Path Planning for Maze Navigation", Brunel University, Department of Computer Science, IEEE, 1998.
7. Gordon V. S., Matley Z., "Envolving Sparse Direction Maps for Maze Pathfinding", California State University, Sacramento and Digital Eclipse Software, Inc. Vancouver, 1992.
8. Hart, P. E., Nilsson, N. J., & Raphael, B. (1964). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics, SSC-4 (2).
9. Huiyu Z. & Huosheng H., "Inertial Motion Tracking of Human Arm Movements in Stroke Rehabilitation", Proceedings of the IEEE International Conference on Mechatronics & Automation, Niagara Falls, Canada, July 2005.
10. Jan, G. E., & Chang K-Y, P. I. (2003). A New Maze Routing Approach for Path Planning of Mobile Robot. IEEE International Conference on Advanced Intelligent Mechatronics.
11. Johnsonbaugh, R. (1999). Teoría de Gráficas: Un algoritmo para la ruta más corta. En Matemáticas Discretas. México: Prentice Hall.
12. Kaplan, R. M., & Saccuzzo, D. P. "Psychological Testing: Principles, Applications, and Issues". Thomson Wadsworth. 2005.
13. Lee C. Y., "An Algorithm for Path Connections and its Applications", IRE Transactions on Electronic Computers, vol. EC-10, pp. 346-365, September 1961.
14. Moore E. F., "The Shortest Path Through a Maze", Annals of Computation Laboratory of Harvard University, Harvard University Press, vol. 30, pp. 285-292, 1959.
15. Morgante F., Gaggioli A. Strambi L., Rusconi M.L. and Riva G., "Computer-enhanced Route and Survey Spatial Knowledge Assessment in Clinical Neuropsychology", IEEE Technology for Neuro-Psychology Lab, Istituto Auxologico Italiano, 2006.
16. Nair P., Jadhav C., Krivi V., "Development and Testing of a Low-Cost Diagnostic Tool for Upper Limb Dysfunction", Department of Mechanical and Aerospace Engineering, State University of New York at Buffalo, Buffalo, NY.

17. Reinkensmeyer D. J. Pang C. T., Painter C. C., "Web-Based Telerehabilitation for the Upper Extremity After Stroke", IEEE Transactions of Neural Systems and Rehabilitation Engineering, Vol. 10, No. 2, June 2002.
18. Rizzo A., Buckwalter J.G., van des Zaag C., Neumann U., "Virtual Environment Applications in Clinical Neuropsychology", IEEE, University of Southern California, Fuller Graduate School of Psychology, Technology for Neuro-Psychology Lab, 2005.
19. Rose F.D., Attree E.A., Brooks B.M., Johnson D.A., "Virtual Environments in Brain Damage Rehabilitation: Rationale from Basic Neuroscience", Department of Psychology, University of East London, London, UK., 1998.
20. Sait, S. M., & Youssef, H. (1999). En VLSI Physical Design Automation: Theory and Practice (págs. 233-276). Singapore: World Scientific Publishing Co.
21. Sheldon A. B., "A Modification of Lee's Algorithm", IEEE, Transactions on Electronic Computers, Febrero, 1967.
22. Stefani O. Mager R., "Cognitive Ergonomics in Virtual Environments: Development of an Intuitive and Appropriate Input Device for Navigating in a Virtual Maze", Applied Psychophysiology and Biofeedback, Vol. 30, No. 3, September 2005.
23. Sutherland E. I., "A Method for Solving Arbitrary-Wall Mazes by Computer", IEEE Transactions on Computers, vol. c-18, no. 12, December 1969.
24. Wann J. P., Rushton K. S., Smyth M. and Jones D., "Rehabilitative Environments for Attention and Movement Disorders", Communications of the ACM, vol. 40, no. 8, Agosto 1997.