

Accesibilidad de edificios por medio de modelado y simulación con aplicaciones 3D.

Nicole Denon¹, Santiago Doti¹, Lucas Olivera¹, Martín Morales^{1,2}, Diego Encinas^{1,3}

¹Programa TIC en aplicaciones de interés social - Instituto de Ingeniería y Agronomía - UNAJ

²Unidad CodApli - Facultad Regional La Plata - UTN

³Instituto de Investigación en Informática LIDI (III-LIDI) - Facultad de Informática - UNLP – Centro Asociado CIC

{nicole.a.denon, luksolivera10, santiagoandresdoti}@gmail.com, {martin.morales, dencinas}@unaj.edu.ar

Resumen. En esta publicación se presenta el desarrollo de una aplicación 3D con la finalidad de simular la estructura interna de un edificio público, concretamente la estructura de una universidad, permitiendo a los usuarios interesados recorrerla. El objetivo principal de la aplicación es ofrecer a aquellas personas, que posean movilidad reducida, recorrer el edificio a través de la aplicación y poder conocer su infraestructura antes de acudir al lugar.

Palabras Clave: Modelado y Simulación, Aplicaciones 3D, Accesibilidad.

1 Introducción

A lo largo de la historia el ser humano ha creado diversos dispositivos con el objetivo de ayudar, a personas que poseen algún tipo de discapacidad, a que su vida se realice de manera más simple y digna. Basándose en esta visión, se desarrolló una aplicación de entorno 3D con el objetivo principal de brindar una herramienta a las personas que sufrieron la pérdida parcial o total del movimiento y que particularmente dependen, para su traslado personal, de una silla de ruedas.

La problemática planteada para la confección de este trabajo fue que muchas personas con movilidad reducida deben asistir a diversos edificios públicos (juzgados, municipalidades, universidades, etc.) y estos, en ciertos casos, no cuentan con la infraestructura adecuada para permitir un correcto desplazamiento considerando las limitaciones mencionadas. Por lo tanto, surgió la necesidad de brindar a estas personas la posibilidad de conocer el lugar antes asistir.

Se propuso una solución, la cual fue generar una aplicación que ofrezca la posibilidad de conocer el lugar. La aplicación cuenta con el modelado de la estructura de un edificio público, en este caso la Universidad Nacional Arturo Jauretche, y permite de manera virtual recorrerla. El usuario puede elegir un lugar de interés dentro del edificio y la aplicación le mostrará en tiempo real el camino más corto para llegar.

En otros edificios públicos se han implementado soluciones parecidas a la propuesta. La Universidad Nacional de La Plata (UNLP) tiene actualmente habilitado en su página web un recorrido virtual de la Facultad de Informática [1] [2]. Por otro lado, la Universidad Nacional Abierta y a Distancia (UNAD) de Bogotá también ha desarrollado su propio recorrido virtual de la Sede Principal [3].

La principal diferencia entre la implementación presentada en este trabajo y los anteriores mencionados, es el enfoque de los mismos. El objetivo del trabajo de Bogotá es más publicitario, su intención es mostrar un recorrido virtual de una forma atractiva e interactiva. Por otro lado, el proyecto de La Plata propone guiar a los estudiantes noveles y posee los puntos de interés más importantes para los ingresantes. En cambio, el objetivo del desarrollo presentado es guiar a aquellas personas con movilidad reducida, por lo que se enfoca en los caminos ideales para ellos, como las trayectorias con rampas o sin escaleras.

2 Implementación

En este apartado se detallarán las herramientas, como así también las metodologías que se utilizaron para el desarrollo de la aplicación.

Herramientas:

- Unity: plataforma para la creación de contenidos interactivos en 2D y 3D. Incluye un entorno de desarrollo, una tienda de assets, un motor gráfico, un repositorio de trabajo y una amplia gama de herramientas que facilitan el proceso de creación del proyecto [4].
- Sweet Home 3D: Es un editor CAD de ingeniería, arquitectura y construcción bajo licencia General Public License (GNU) para el diseño de muebles de una vivienda en un plano 2D que permite generarlo y exportarlo en un plano 3D [5].

3 Desarrollo

En este apartado se detalla, la implementación de las herramientas además de las decisiones tomadas para el desarrollo.

Cuando se determinó el alcance que tendría el software se planteó que para su correcto funcionamiento eran necesarios tres módulos fundamentales:

En primer lugar, un mundo diseñado en tres dimensiones en el que se desarrolle toda la acción, es decir, un espacio en donde se muestre al usuario por dónde es más accesible el ingreso a un edificio público. Y que ésto suceda de una forma virtual pero que sea lo más parecido a la realidad de dicho inmueble. En segundo lugar, se precisó que todo lo que suceda en dicho entorno se presente de una manera real, en otras palabras, que respete las leyes de la física para contribuir a la inmersión y credibilidad del usuario final cuando necesite las indicaciones para guiarse dentro del edificio público. Finalmente, se requirió del algoritmo de búsqueda capaz de calcular cuál iba a ser el camino más corto para llegar al destino que el usuario solicitaba. Analizando estos pasos como partes de un proceso para lograr el producto final se puede expresar como se ve en la Figura 1.



Figura 1. Diagrama de los procesos que componen el software

Se puede apreciar en dicho diagrama que cada proceso es cuasi dependiente del otro, en otras palabras, cada uno necesita una característica del otro para poder funcionar con normalidad. Si no existiese un mundo virtual 3D el algoritmo de búsqueda no tendría donde mostrarse y las físicas no actuarían sobre ningún tipo de objeto, y así sucedería lo mismo con los restantes procesos.

A continuación se detalla cada proceso por separado:

3.1 Diseño entorno 3D

Haciendo uso de la herramienta Sweet Home se diseñó uno de los edificios de la Universidad [6]. Se escogió este edificio ya que es uno de los más frecuentados por los estudiantes y posee una especie de rampa en su hall permitiendo que una persona con silla de ruedas puede subir al primer piso como se puede ver en la Figura 2. Una vez elaborado el diseño se exporta a formato OBJ (Wavefront 3D Object File), el cual es una extensión de archivo usado para un objeto tridimensional que contiene las

coordenadas 3D. Esto permite hacer uso de este diseño dentro de la plataforma de Unity. En la Figura 3 se visualiza el diseño 3D desde Unity una vez agregada la iluminación y los colores de los objetos.



Figura 2. Diseño 3D del primer piso en SweetHome 3D



Figura 3. Vista en primera persona de la aplicación

3.2 Simulación Física

Como el proyecto se basó en una simulación del mundo real se necesitaban de las reglas físicas que lo rigen. Para esto se hizo uso del motor físico integrado que provee el mismo Unity, el cual es el encargado de dar los correspondientes comportamientos físicos a cada objeto (GameObjects dentro del sistema) que se encuentra en el entorno virtual.

Los comportamientos de los GameObject son controlados por los Components, éstos no son sólo funciones físicas, sino que pueden ir desde funciones de renderizado, de navegación, de sonido hasta estadísticos, entre muchos otros. Inclusive se pueden llegar a crear los propios en forma de Scripts, éstos pueden ser desarrollados en lenguaje C# o UnityScript.

Centrándose en los Components de físicas más importantes que fueron utilizados en este proyecto, se encuentran:

- **Rigidbody:** componente básico que le permite a cada GameObject actuar bajo el control de la física. Gracias a éste cada uno puede recibir fuerza y torque para moverse de una manera realista, en el proyecto fue aplicado a la silla de ruedas para que cuando detecte colisión detenga su posición y no traspase al objeto en cuestión.
- **Capsule Collider:** como su nombre lo indica, crea una cápsula de dos semi-esferas unidas por un cilindro alrededor de cada GameObject para que detecte colisiones con otros objetos en el entorno (se

puede ver en la Figura 4). Cuando detecta colisiones se lo indica al Rigidbody para que éste actúe según lo programado.

- **Mesh Collider:** funciona al igual que el anterior componente pero en este caso funciona como una malla, la cual se acopla casi perfectamente al objeto asignado por lo que es mucho más precisa pero consume muchos recursos.
- **Movimiento del personaje:** en este caso se trata de un script desarrollado en este trabajo para controlar la silla de ruedas en el entorno virtual. Fue desarrollado en C# y básicamente lo que hace es capturar cuando se presiona la tecla W mientras se recorre el entorno virtual. Cuando detecta dicha acción se mueve al protagonista hacia delante, dependiendo de hacia donde esté enfocada la vista (manejada por el mouse de la computadora), ya que se trata de una silla de ruedas y la única dirección de movimiento permitida es esa.



Figura 4. Personaje y el componente Capsule Collider

3.3 Algoritmo de Búsqueda [7]

Para este desarrollo se analizaron diferentes algoritmos de búsqueda, los cuales se describirán brevemente en este apartado resaltando sus virtudes y falencias.

Algoritmo de Dijkstra: Es un algoritmo que permite recorrer un grafo a través de sus vértices y conocer el camino más óptimo entre dos nodos. La ventaja de este algoritmo es que permite conocer cuál es el camino más corto o de menor costo para moverse de un punto a otro. La desventaja principal es que recorre el grafo por amplitud, es decir, visita a todos los nodos vecinos, esto provoca mayores iteraciones. La Figura 5 muestra cómo trabaja el algoritmo. El punto rosa representa el origen y el violeta el destino, los cuadrados pintados de celeste son los que el algoritmo visitó hasta llegar al destino.

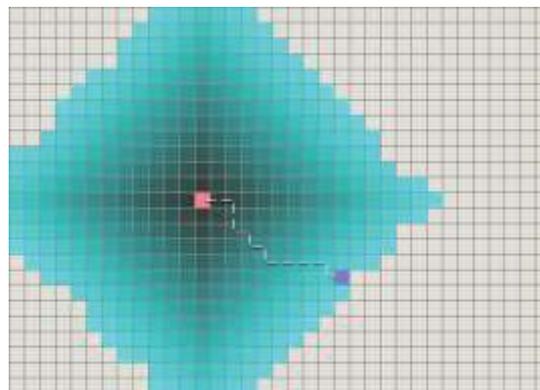


Figura 5. Algoritmo de Dijkstra.

Algoritmo Greedy Best-First-Search (GBFS): Este algoritmo es un ajuste al método tradicional de BFS, la principal diferencia es que utiliza la Búsqueda Heurística (BH) para recorrer el grafo. La gran ventaja de este algoritmo es que gracias a la BH permite en menos interacciones encontrar el nodo destino. Pero a diferencia del algoritmo anterior, como se mencionó es un ajuste de BFS por lo tanto el recorrido que devuelve no es el más óptimo por lo tanto esta es la desventaja al usar este algoritmo. Como puede verse en la Figura 6 el algoritmo visita menos nodos que el algoritmo anterior.

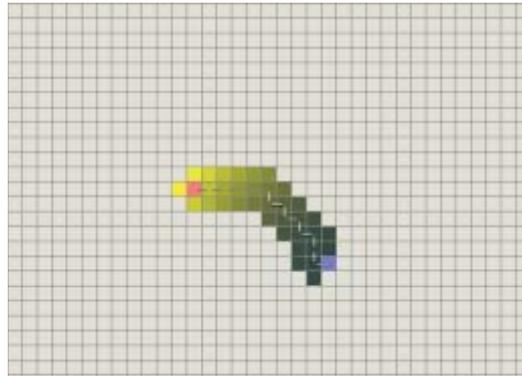


Figura 6. Algoritmo de GBFS.

En las Figura 7 y Figura 8 se puede ver una comparativa entre los dos algoritmos en donde el algoritmo de Dijkstra itera más pero devuelve un camino óptimo. Y como el GBFS itera menos pero su camino no es muy óptimo.

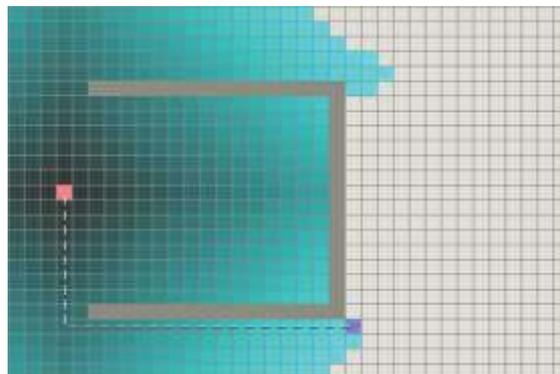


Figura 7. Algoritmo de Dijkstra con obstáculo.

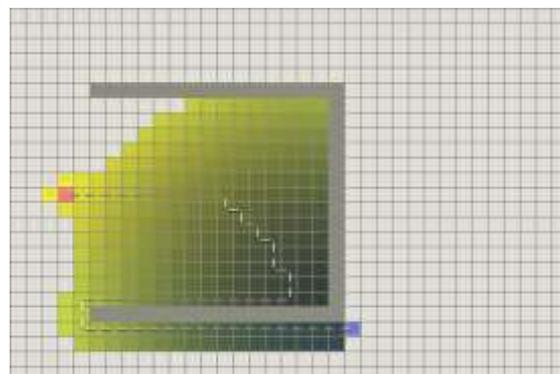


Figura 8. Algoritmo de GBFS con obstáculo.

Algoritmo Estrella o Star (A*): Este es el algoritmo que se seleccionó para realizar el trabajo. A* resuelve los problemas causados por los algoritmos anteriores ya que vincula la búsqueda óptima de Dijkstra y la BH de GBFS.

En las Figura 9 y Figura 10 puede verse cómo se comporta este algoritmo en las mismas condiciones que los casos anteriores y se puede notar que el algoritmo A* itera en menor medida y devuelve el camino óptimo.

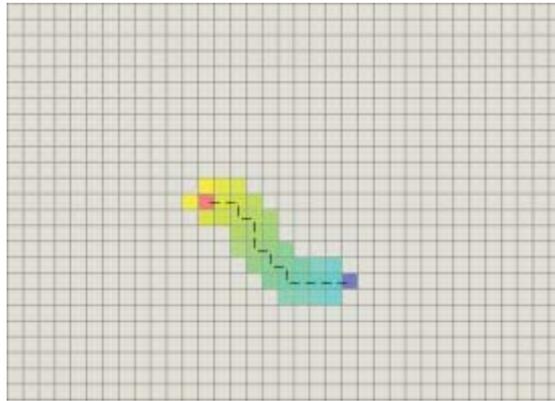


Figura 9. Algoritmo A*

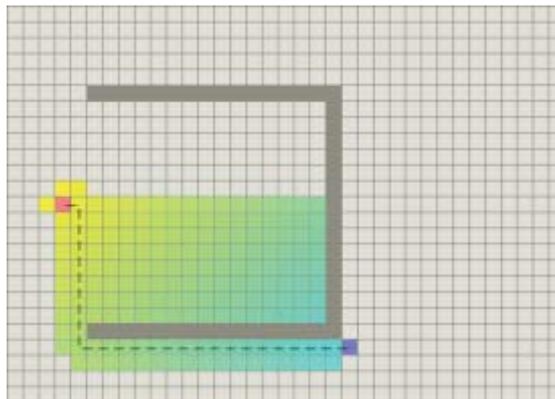


Figura 10. Algoritmo A* con obstáculo.

Búsqueda Heurística

Es una función que se utiliza para la estimación de costos desde cualquier vértice n hasta el destino. Hay diversas funciones dependiendo el problema que se desee solucionar, para este problema en particular se utilizaron funciones que trabajan sobre grillas o cuadrículas:

1. Distancia Manhattan: Este algoritmo sólo permite desplazarse en 4 direcciones, es decir no permite desplazamientos en diagonal por lo tanto el camino devuelto no es del todo el óptimo si en el sistema real existe la posibilidad de moverse diagonalmente. Este funcionamiento puede visualizarse en la Figura 11 en donde el camino solo se traza de forma horizontal o vertical.

Cálculo realizado: sea I y D nodos de la cuadrícula; x, y sus posiciones en los ejes, P un valor de movimiento y C el camino devuelto:

$$Dx = |I_x - D_x| \quad (1)$$

$$Dy = |I_y - D_y| \quad (2)$$

$$C = P * (Dx + Dy) \quad (3)$$

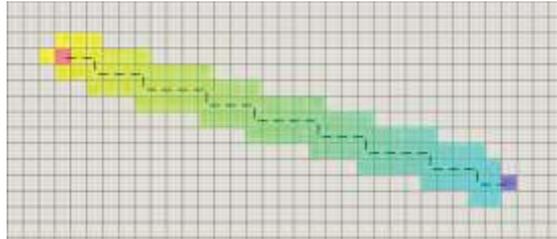


Figura 11. Camino con distancia Manhattan.

2. Distancia Diagonal: Este algoritmo ofrece la posibilidad de desplazamiento en diagonal, es decir se tiene 8 direcciones para poder moverse, y se consigue simplemente agregando unos cambios a la ecuación de la distancia Manhattan. Esto se aprecia fácilmente en la Figura 12. Gracias a las ventajas que provee este algoritmo fue posible el desarrollo en este trabajo.

Tomando (1) y (2), se aplica un ajuste a (3), donde P_2 es un valor de movimiento.

$$C = P * (Dx + Dy) + (P_2 - 2 * P) * \min(Dx, Dy) \quad (4)$$

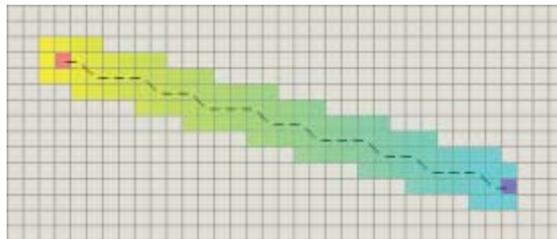


Figura 12. Camino con distancia Diagonal.

3. Distancia Euclidiana: Al igual que la anterior la euclidiana permite movimiento por las diagonales, pero también se puede utilizar si se necesita moverse por cualquier ángulo de la cuadrícula y su función es similar a Pitágoras. Tomando (1) y (2)

$$C = P * \sqrt{Dx^2 + Dy^2} \quad (5)$$

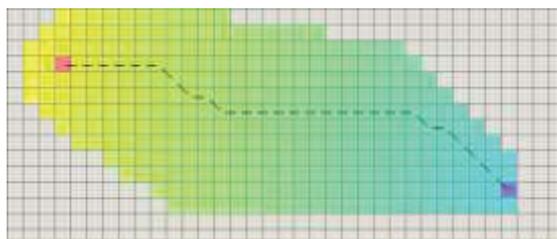


Figura 13. Camino con distancia Euclidiana.

El problema al aplicar este algoritmo es que aumenta la cantidad de iteraciones que realiza el Algoritmo de Búsqueda como se puede observar en la Figura 13.

Vinculación de Componentes:

Un desafío grande para el desarrollo fue intentar que el algoritmo de búsqueda pueda ser implementado por Unity independientemente del modelo 3D creado por Sweet Home 3D con el objetivo de poder incluir diversas arquitecturas de edificio. Para eso se desarrolló el algoritmo con la habilidad de reconocer,

independientemente de la estructura, los objetos a los cuales no se puede atravesar (paredes, mesas, etc). Gracias a ello, y con la debida configuración a los correspondientes modelos 3D, se puede diseñar cualquier edificio y obtener el mejor camino.

4 Resultados

Al iniciar con la aplicación el usuario se encuentra un menú principal en donde puede seleccionar la habitación a la cual quiere ir. Estas habitaciones son en su mayoría aulas u oficinas de administración. Una vez elegido el lugar se pasa a la escena principal del edificio, para este ejemplo se eligió un aula del piso superior. En la Figura 14 se puede observar que al iniciar movimiento por el mapa aparecen círculos verdes que indican el mejor camino, en este caso el camino indica la rampa ya que el lugar de destino se encuentra en el primer piso.



Figura 14. Señalización del mejor camino en ejecución

Una vez que se sube por la rampa y se llega al piso superior, el mejor camino vuelve a marcarse como se muestra en la Figura 15. Ahora el camino indica el aula de destino que se ha ingresado en el menú principal.



Figura 15. Señalización hacia el aula de destino

Por último en la Figura 16 se muestra la llegada al destino final del software, el cual está marcado con un cuadrado rojo para poder visualizarlo. Puede volverse a cambiarse el destino si se presiona "ESC" y se vuelve a seleccionar un nuevo destino en el menú principal. Si esto se hace se repetirá el mismo proceso especificado anteriormente



Fig. 16. Señalización hacia el aula de destino

La aplicación puede verse en funcionamiento en el siguiente link: <https://youtu.be/4CgL2okrR68>

5 Trabajos futuros

El proyecto ha sido diseñado para que sea escalable y que permita implementar de forma sencilla con otros edificios. Aun así existen muchas mejoras que pueden aplicarse, estas son las siguientes:

- Implementación de la aplicación para dispositivos móviles: El smartphone se ha convertido con el paso del tiempo en el medio de comunicación masivo más utilizado por las personas [8]. Esto se debe en gran parte a la movilidad y comodidad que brinda un teléfono celular. Por lo tanto la aplicación desarrollada podría tener un muy buen uso social si contará con una implementación móvil, ya que una persona con capacidad de movimiento reducida podría necesitar una visualización de un edificio justo antes de adentrarse al mismo. Aunque un punto muy importante que hay que tener en cuenta para poder lograrlo es el rendimiento que tendrá la aplicación actual corriendo en un dispositivo móvil. No será una tarea fácil puesto que se necesitarán celulares muy potentes para que funcione de una manera fluida. Si se tienen en cuenta los requisitos que necesita un smartphone para ejecutar aplicaciones no tan complejas desarrolladas en Unity [9] se requiere de Android 4.1 o posterior, arquitectura ARMv7 CPU con soporte NEON o CPU Atom y soporte para OpenGL ES 2.0 o posterior. En cuanto a celulares iPhone se tendrá que contar con iOS 9.0 o posterior.
- Modelado de más edificios de interés: Como se ha mencionado antes, el algoritmo de búsqueda desarrollado se adapta a cualquier edificio que se quiera modelar. Por lo que sólo es necesario importar un diseño 3D de una infraestructura al programa para poder representarla.
- Sumar obstáculos, tales como personas en movimiento o muebles si se quiere simular el edificio tal cual es, es necesario implementar más objetos, como mesas o computadoras, y personas. Esto agrega más obstáculos y dificulta mínimamente el cálculo de la búsqueda del mejor camino.
- Agregar sonidos ambientales: Los efectos de sonido permiten dar una mejor inmersión en el mundo virtual desarrollado.

6 Conclusiones

Se desarrolló una aplicación que cumple con los objetivos planteados inicialmente, inclusive con una alta escalabilidad, es decir, aplicando las mejoras mencionadas previamente se podrá conseguir una aplicación mucho mayor, más integral y de mayor valor si así se desea.

Con el fin de implementar algunas de las mejoras planteadas en los trabajos futuros se debería hacer uso de Android Studio para el desarrollo de una aplicación móvil orientada a los smartphones que cuenten con sistema operativo Android, como así también contar con los planos de los edificios correspondientes que se quieren modelar para lograr un resultado lo más fiel al establecimiento real.

Para finalizar, se entiende que la implementación y uso de esta aplicación podría ayudar a muchas personas que requieran tener esta información previamente y actualmente no poseen acceso a una tecnología de esta índole.

7 Referencias

1. Thomas P., Cristina F., Dapoto S., Pesado P. “Desarrollo de Aplicaciones Móviles 3D Orientadas a Educación.”, XVIII Workshop de Investigadores en Ciencias de la Computación. 2016.
2. Recorrido virtual de la Universidad Nacional de La Plata <https://www.info.unlp.edu.ar/recorrido-virtual-3d-facultad-informatica/> - Fecha de consulta: 29/07/2019
3. Nieto Acevedo Y., López Quintero J., González Clavijo C. “Recorrido Virtual en Tercera Dimensión de la Sede Principal en una Universidad de Bogotá”. Corporación Unificada Nacional (CUN). Universidad Nacional Abierta y a Distancia (UNAD). Bogota, Colombia. 2015.
4. Página oficial de Unity3D, <http://www.unity3d.com/es> - Fecha de consulta 24/11/2018.
5. Página oficial de Sweet Home 3D <http://www.sweethome3d.com/es/> - Fecha de consulta: 24/11/2018.
6. Edificio diseñado para la aplicación, <https://www.unaj.edu.ar/edificio-manuel-savio/> - Fecha de consulta 24/11/2018.
7. Algoritmos de búsqueda, <http://theory.stanford.edu/~amitp/GameProgramming/> - Fecha de consulta 19/11/2018.
8. La venta de celulares vuelve a romper el record en Argentina, <https://www.eldia.com/nota/2017-9-18-2-37-2-la-venta-de-celulares-vuelve-a-romper-el-record-en-argentina-informacion-general> - Fecha de consulta 24/11/2018.
9. Requisitos del sistema para la versión de Unity, <https://unity3d.com/es/unity/system-requirements> - Fecha de consulta 20/02/2019