Capítulo 5

Diseño e Implementación

Para verificar el desempeño de nuestra implementación del método PRM, orientado a solucionar problemas de planeación de rutas para un actor digital, se desarrolló la aplicación que se describe a lo largo de este capítulo.

Esta aplicación utiliza el conjunto de clases descritas en capítulos anteriores, y para su diseño se tomaron en consideración los requerimientos que se presentan a continuación.

5.1 Requerimientos

Uno de nuestros principales objetivos en esta tesis, es desarrollar una herramienta que permita visualizar el ambiente en el que ha de desenvolverse un actor digital, generar mapas de ese ambiente y solucionar problemas de planeación de rutas para el mismo.

De esta manera, se desea que mediante nuestra herramienta, el usuario pueda visualizar los elementos que conforman el espacio de trabajo (actor digital y obstáculos), y la información generada por el PRM (mapa y rutas). Con respecto al actor digital y los obstáculos, ambos se especificarán en archivos bajo el formato Milkshape 3D y podrán visualizarse en modo sólido, como una malla, o bien, segmentados en cajas (bounding boxes). En el caso del actor digital, también es deseable poder visualizar su esqueleto. En cuanto al mapa y las rutas generadas por el PRM, sus nodos y aristas se mostrarán en pantalla como un conjunto de puntos y líneas respectivamente, agrupados de acuerdo al componente del mapa al que pertenecen. También, se desea que la herramienta permita al usuario reconstruir un mapa, o bien, mejorar su conectividad agregando más nodos al mismo.

Además, el usuario podrá interactuar con el ambiente mediante la manipulación de una cámara que le permitirá acercar, alejar y rotar la escena, y navegar en el ambiente virtual. Desde la herramienta, el usuario podrá realizar consultas sobre problemas de planeación de rutas utilizando el formato definido por la clase Query descrita en el capítulo 4, o bien, generando una consulta mediante la selección de una serie de configuraciones en el ambiente virtual. De esta manera, el usuario podrá visualizar el resultado de su consulta y animar al actor digital a lo largo de una ruta.

Por último, para poder comparar los resultados obtenidos por el PRM bajo diferentes parámetros de construcción, se desea que la herramienta permita almacenar y cargar desde un archivo los diferentes elementos del espacio de trabajo, es decir, el actor digital, los obstáculos, el mapa, y las rutas generadas.

5.2 Diseño de la Aplicación

Para el diseño de la aplicación se siguió el modelo espiral descrito en [33]. El modelo espiral define un proceso iterativo que permite desarrollar de forma rápida, versiones incrementales de software. En las primeras iteraciones se produce una versión simple del sistema, misma que va desarrollándose en cada iteración hasta obtener un sistema más complejo. Este modelo para el desarrollo de software inicia con el análisis de requerimientos, lo que permite dividir el sistema en varios componentes. Posterior a esto, se continúa con el diseño y la implementación de los componentes, a los que se realizan pruebas por separado y finalmente se unen para formar un sistema más complejo, al que también se realizan pruebas.

Siguiendo el modelo espiral y tomando en consideración los requerimientos presentados anteriormente, se diseñó la herramienta que permite analizar el desempeño de las clases desarrolladas a lo largo de esta tesis para solucionar el problema de planeación de rutas. Los componentes de esta herramienta se muestran en la figura 5.1.

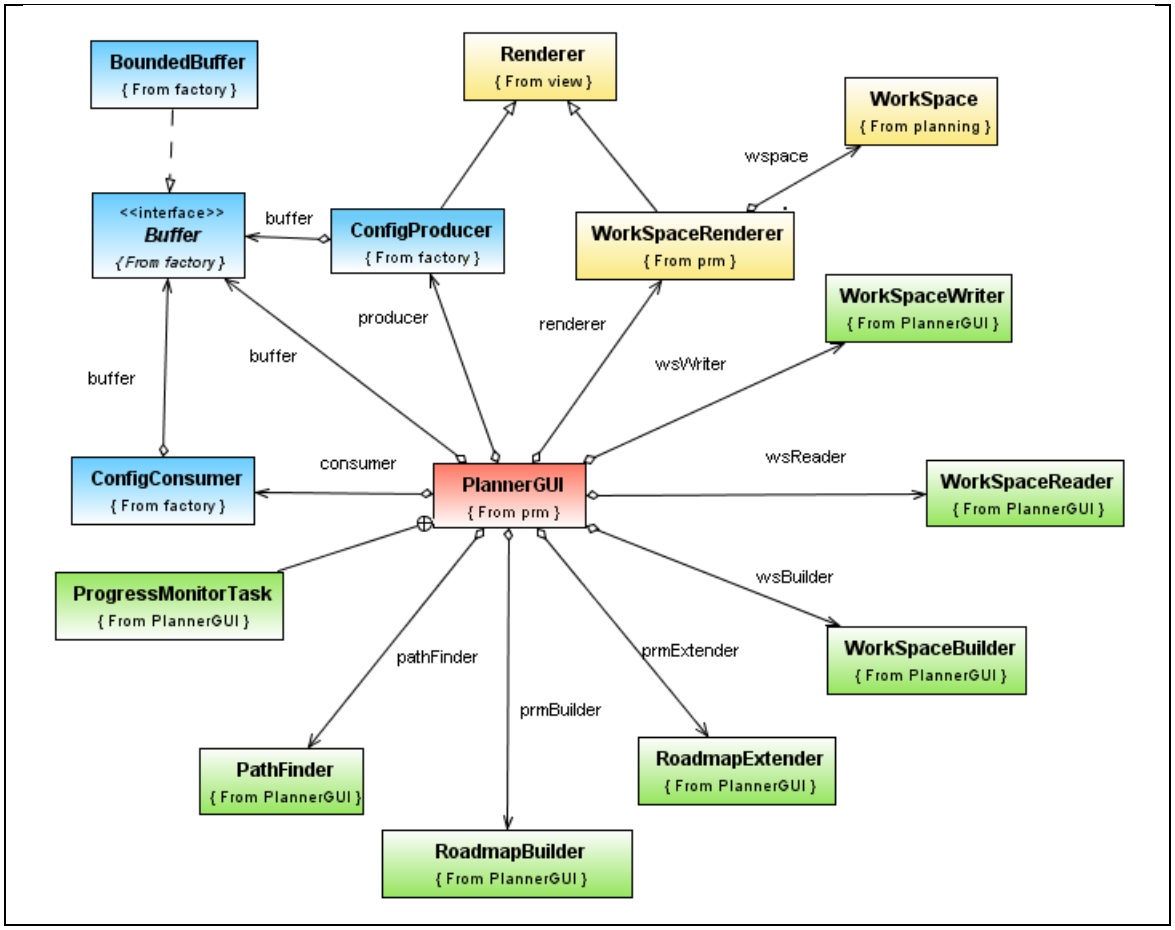

Figura 5.1: Componentes de la herramienta PlannerGUI.

La clase principal en este diagrama es la clase PlannerGUI, que define la interfaz que permite al usuario realizar consultas de planeación de rutas para el actor digital. Para la implementación de la interfaz se utilizó la librería Swing de Java. La elección de esta librería se debe a que proporciona un conjunto muy variado de componentes gráficos como ventanas, menús, barras de herramientas, botones, entre otros, que son fácilmente configurables y portables. El diagrama de la figura 5.1 muestran los principales componentes del sistema, que de acuerdo a su función, clasificamos de la forma que se describe a continuación.

Esclavos

Llamamos *esclavos* al conjunto de clases que realizan una tarea específica para la interfaz de usuario. Estas clases se muestran en la figura 5.1 en color verde. Las clases

esclavo son invocadas por la interfaz de usuario en respuesta a un evento generado por el usuario. El resultado del trabajo de estas clases es utilizado por la interfaz de usuario para dar respuesta a una solicitud del usuario. De esta manera, las clases *esclavo* trabajan única y exclusivamente para la interfaz de usuario cuando ésta lo solicita.

Las tareas realizadas por los esclavos se caracterizan por ser muy demandantes en cuanto al tiempo que requieren para llevarse a cabo, razón por la que se ejecutan en segundo plano. Esto nos permite mantener una interfaz capaz de responder a las solicitudes del usuario en todo momento, ya que cuando el usuario genera un evento, la interfaz delega el trabajo al esclavo correspondiente y regresa de inmediato a esperar por otro evento. De otra manera, si la interfaz de usuario se encargara de todo el trabajo, su capacidad de respuesta se vería afectada de forma considerable.

Las clases *esclavo* son una extensión de la clase *SwingWorker* de la librería Swing. *SwingWorker* es una clase abstracta diseñada para llevar a cabo tareas que requieren demasiado tiempo para una interfaz de usuario en un thread dedicado. Para mayores detalles sobre el uso de la librería Swing de Java, véase [34].

La función de las clases esclavo de nuestra aplicación es la siguiente:

- *WorkSpaceBuilder*: Construye un ambiente de trabajo de acuerdo a los parámetros especificados por el usuario. El espacio de trabajo incluye al actor digital, los obstáculos, y el mapa generado por el método PRM.

- *WorkSpaceReader*: Se encarga de leer y cargar desde un archivo un espacio de trabajo previamente creado mediante la interfaz de usuario. Esta clase se encarga de crear los modelos del espacio de trabajo (actor digital y obstáculos) y de cargar el mapa y las rutas que se hayan generado.

- *WorkSpaceWriter*: Se encarga de guardar en un archivo los elementos de un espacio de trabajo. Esta clase almacena en el nombre de los archivos que se

usaron para definir al actor digital y los obstáculos, así como el mapa y las rutas generadas.

- *RoadmapBuilder*: Construye un mapa del espacio de trabajo utilizando el método PRM, de acuerdo a los parámetros de construcción definidos por el usuario.

- *RoadmapExtender*: Agrega configuraciones al mapa con el fin de mejorar su conectividad. Esta clase es muy útil cuando el mapa generado es insuficiente para responder a una consulta.

- *PathFinder*: Se encarga de consultar el mapa generado por PRM en busca de una ruta para una consulta determinada.

- *ProgressMonitorTak*: Se encarga de monitorear el grado de avance de las tareas realizadas por el resto de los esclavos, y su propósito es mostrar al usuario el grado de avance de estas tareas.

Clases Auxiliares

Las clases auxiliares se muestran en la figura 5.1 en color amarillo. Estas clases proporcionan servicios que son utilizados por la interfaz de usuario y por sus esclavos.

- *Renderer*: Implementación del framework para graficar en 3D utilizando Java y OpenGL descrito en el apéndice A.

- *WorkSpaceRenderer*: Muestra en pantalla una vista en 3D de los elementos del espacio de trabajo (actor digital, obstáculos, mapa y rutas). Además, esta clase se encarga de animar al actor digital a lo largo de la ruta seleccionada. *WorkSpaceRenderer* es una extensión de la clase *Renderer*.

- *WorkSpace*: Encapsula los elementos del espacio de trabajo. Esta clase almacena referencias a los archivos que definen al actor digital y los obstáculos, además del mapa y las rutas generadas por el método PRM para dicho espacio de trabajo. La clase WorkSpace define métodos para leer y almacenar los elementos del espacio de trabajo en un archivo. Estos métodos son utilizados por los esclavos correspondientes.

Interfaz de Consulta

La función más importante de la interfaz de usuario es realizar consultas sobre problemas de planeación de rutas para el actor digital, ya que a través de éstas podemos comprobar que tan bien se desempeña nuestra implementación del método PRM en un espacio de trabajo determinado.

Como se mencionó anteriormente, las consultas se representan mediante objetos de la clase Query, que define el formato al que debe ajustarse una consulta válida. Un ejemplo de consulta válido es *path (0 0 0) (10 0 10)*. Este tipo de consultas pueden realizarse de forma directa a través de la interfaz de usuario. Sin embargo, realizar este tipo de consultas puede llegar a ser tedioso para el usuario, pues es necesario especificar las posiciones exactas para el actor digital. Por esta razón, se diseñó una interfaz de consulta que permite al usuario construir consultas válidas mediante la selección de una serie de posiciones con la ayuda del mouse.

Las clases que conforman la interfaz de consulta se muestran en la figura 5.1 en color azul. El diseño de estas clases se basa en el problema de comunicación entre procesos conocido como *bounded buffer problem* descrito en [35]. En este problema, dos procesos, el productor y el consumidor, se comunican entre sí a través de un buffer. El productor genera objetos que deposita en el buffer y el consumidor los retira del mismo conforme se van generando.

En nuestra adaptación, el productor se encarga de determinar la posición seleccionada por el usuario al presionar el botón izquierdo del mouse sobre una vista aérea del espacio de trabajo. La posición seleccionada se almacena en el buffer, de donde es retirada por el consumidor. El consumidor se encarga de construir una consulta válida utilizando las posiciones que retira del buffer. Como se recordará, para que una consulta sea válida, además de seguir el formato especificado en la clase Query, es necesario indicar al menos dos posiciones entre las que habrá de buscarse una ruta. De esta manera, el consumidor construye una consulta sólo cuando el usuario ha seleccionado al menos dos posiciones.

Esta interfaz es de gran utilidad, pues permite al usuario construir consultas con un número ilimitado de configuraciones intermedias entre las posiciones inicial y final. En este punto es importante resaltar que al hablar de un número ilimitado de configuraciones intermedias, nos referimos a que una consulta es válida si se especifican dos o más posiciones. De esta manera, se puede especificar una consulta con 100 posiciones y ésta seguirá siendo válida. Sin embargo, para fines prácticos es suficiente con especificar consultas con pocas posiciones intermedias.

Las clases que conforman la interfaz de consulta son las siguientes:

- *ConfigProducer*: Determinar las posiciones seleccionada por el usuario y las agrega al buffer.

- *ConfigConsumer*: Toma las posiciones almacenadas en el buffer y construye una consulta válida a partir de ellas.

- *Buffer*: Interfaz que define los métodos para insertar y remover elementos de un buffer.

- *BoundedBuffer*: Implementación de capacidad limitada de la interfaz buffer.

5.3 Implementación de la Aplicación

Utilizando el conjunto de clases hasta aquí descritas, se implementó la aplicación de esta tesis que se muestra en la figura 5.2.

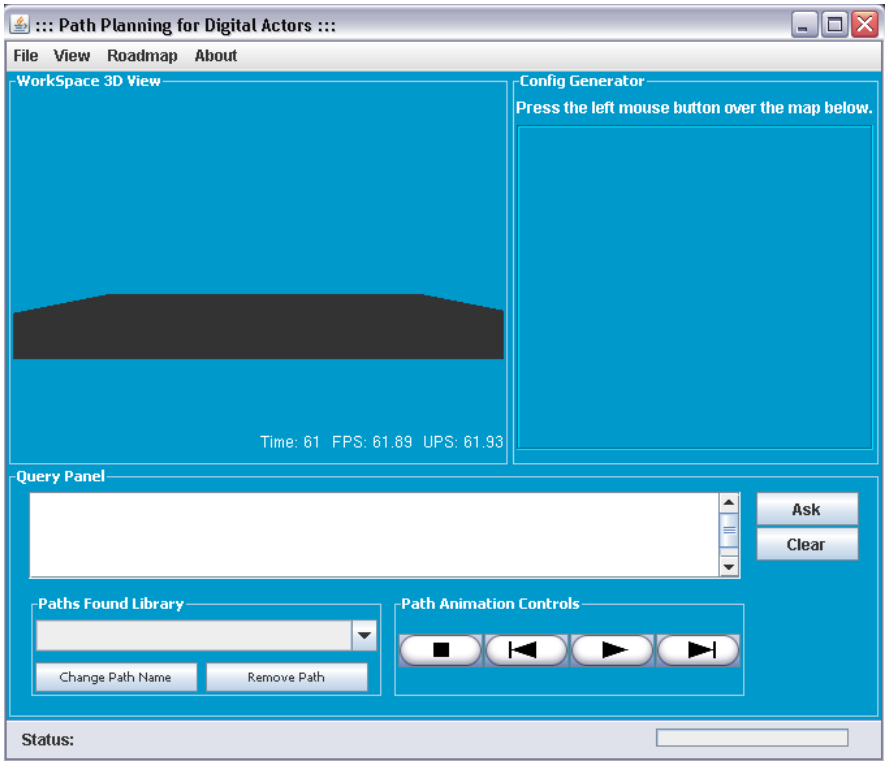

Figura 5.2: Interfaz de usuario.

Esta aplicación se divide en 4 áreas principales: barra de menús, panel de visualización 3D, panel de consulta y panel generador de configuraciones, cuya función se describe a continuación.

Barra de menús

La barra de menús agrupa el conjunto de acciones que el usuario puede realizar desde la aplicación de acuerdo a su tipo. Esta barra consta de los siguientes menús.

Menú File: Este menú es el punto de partida de nuestra aplicación, pues agrupa las opciones para crear, leer desde un archivo, y guardar un espacio de trabajo, además de la

opción para salir de la aplicación. La figura 5.3 muestra las opciones de este menú, y su descripción es la siguiente.

Figura 5.3: Opciones del menú File.

- *New Work Space*: Esta opción crea un espacio de trabajo utilizando los parámetros y archivos especificados por el usuario.

- *Open Work Space*: Esta opción permite al usuario leer un archivo que contiene un espacio de trabajo previamente creado. Los archivos que utiliza tienen la extensión *.wsp, y son archivos binarios en los que se almacena una versión serializada de los archivos que definen al actor digital y los obstáculos, además del mapa y las rutas generadas por el método PRM.

- *Save Work Space*: Esta opción permite al usuario guardar en un archivo una descripción del espacio de trabajo actual, que posteriormente puede ser utilizada para su reconstrucción. Es de gran utilidad para comparar el desempeño del método PRM bajo diferentes parámetros de construcción, ya que almacena el mapa y las rutas generadas.

- *Exit*: Permite al usuario salir de la aplicación.

Menú View: En este menú se agrupan las opciones para mostrar u ocultar algún elemento del espacio de trabajo. Las opciones de este menú se muestran en la figura 5.4, y su descripción es la siguiente.

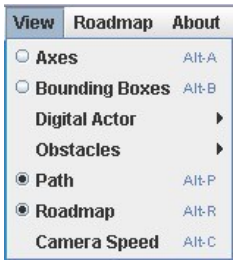

Figura 5.4: Opciones del menú View.

- *Axes*: Muestra u oculta los ejes del espacio de trabajo.

- *Bounding Boxes*: Muestra u oculta los obstáculos segmentados en cajas. Esta representación es la que se utiliza para la detección de colisiones en el método PRM.

- *Digital Actor*: Este submenú contiene las opciones para visualizar al actor digital en modo sólido o como una malla de alambre, además de su esqueleto. La imagen de este submenú se muestra en la figura 5.5.

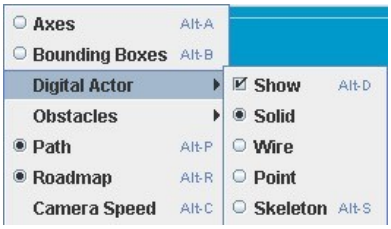

Figura 5.5: Opciones del submenú Digital Actor.

- *Obstacles*: Este submenú muestra las opciones para visualizar los obstáculos del espacio de trabajo en modo sólido o como una malla de alambre, además de la opción para ocultarlos. La imagen de este submenú se muestra en la figura 5.6.

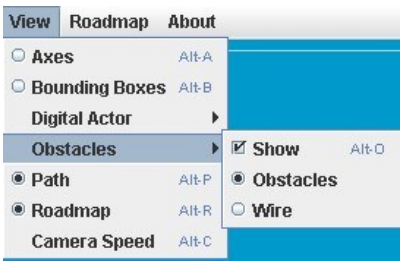

Figura 5.6: Opciones del submenú Obstacles.

- *Path*: Muestra u oculta la ruta generada por el método PRM.

- *Roadmap*: Muestra u oculta el mapa generado por el método PRM.

- *Camera*: Permite al usuario ajustar la velocidad de la cámara para navegar en el ambiente virtual.

Menú Roadmap: Este menú agrupa las opciones de reconstrucción, extensión y estadísticas del mapa utilizado para la planeación de rutas. Sus opciones se muestran en la figura 5.7.

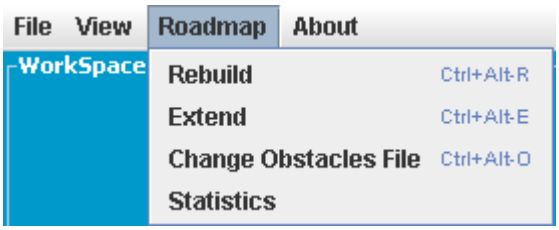

Figura 5.7: Opciones del menú Roadmap.

- *Rebuild*: Reconstruye el mapa de acuerdo a los parámetros de construcción especificados por el usuario.

- *Extend*: Mejora la conectividad del mapa agregando más configuraciones, siguiendo la estrategia de conexión seleccionada por el usuario. Esta opción permite combinar las dos estrategias de conexión entre nodos utilizadas por nuestra implementación del método PRM.

- *Change Obstacles File*: Este opción permite al usuario cambia el archivo que define los obstáculos, y es útil cuando se tiene representaciones de los mismos obstáculos de diferente resolución, por ejemplo, el número de triángulos, la apariencia, entre otros.

- *Statistics*: Esta opción muestra al usuario las estadísticas generadas durante la construcción del mapa, tales como el número de nodos, aristas y componentes en el mapa, número total de configuraciones generadas, y el número de configuraciones que satisfacen la condición de ser libres de colisión. También se muestra el tiempo que llevó la construcción del mapa, el tiempo dedicado a la detección de colisiones, el tiempo dedicado a la última consulta y el tiempo promedio dedicado a todas las consultas realizadas en el mapa. La figura 5.8 muestra un ejemplo de las estadísticas de un mapa. Esta opción es de gran utilidad para el usuario, pues del número de componentes se puede deducir si el mapa generado podrá responder a cualquier consulta.

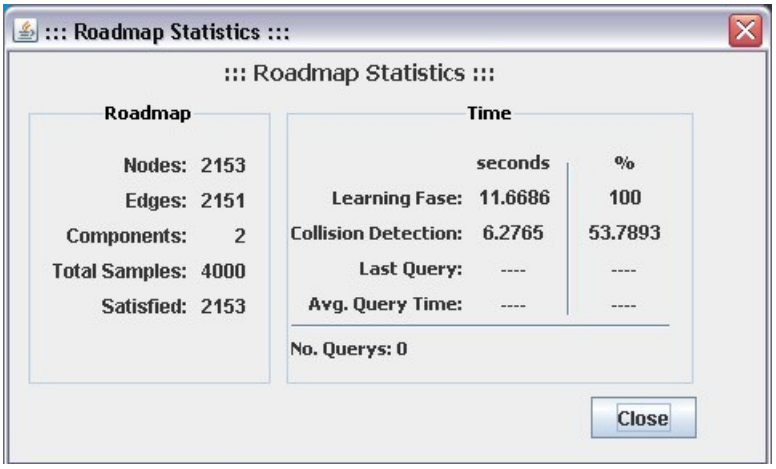

Figura 5.8: Estadísticas del mapa.

Menú About: Este menú muestra información sobre el título de la tesis, así como la versión de la aplicación y los datos del autor. La figura 5.9 muestra la información desplegada por este menú.

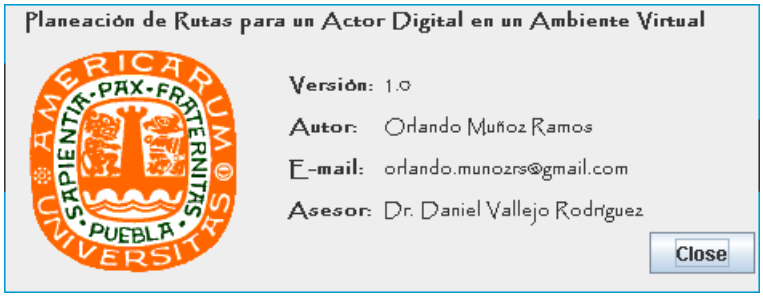

Figura 5.9: Menú About.

Panel de Visualización 3D

El panel de visualización 3D muestra una vista global del espacio de trabajo. En este panel se muestra al actor digital, los obstáculos, el mapa y las rutas generadas por el método PRM. Este panel permite al usuario interactuar con el ambiente virtual mediante la manipulación de la cámara a través del teclado y el mouse. Las acciones de la cámara de describen en el apéndice A. Algunas imágenes de este panel se muestran en la figura 5.10.

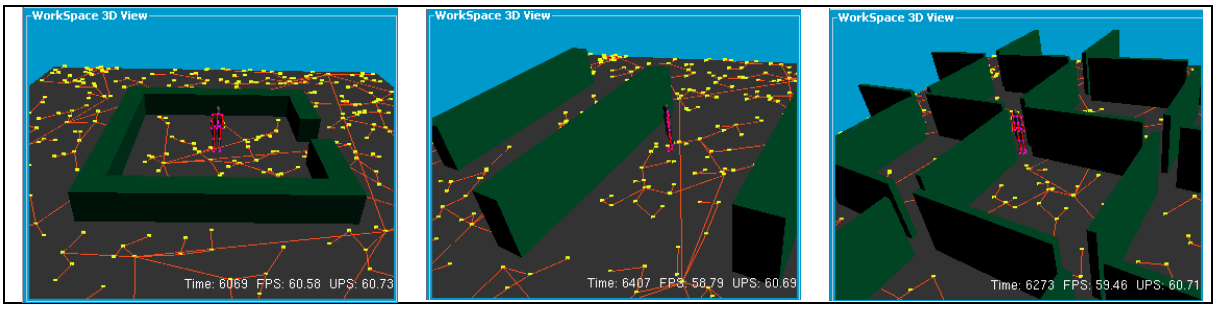

Figura 5.10: Panel de visualización 3D.

Panel del Consulta

El panel de consulta permite al usuario interrogar al mapa sobre la existencia de una ruta para el actor digital. Este panel se divide en tres partes: el editor de texto, la librería de rutas encontradas y los controles de animación del actor digital. La figura 5.11 muestra el panel de consulta. La función de los diferentes componentes que conforman el panel de consulta es la siguiente:

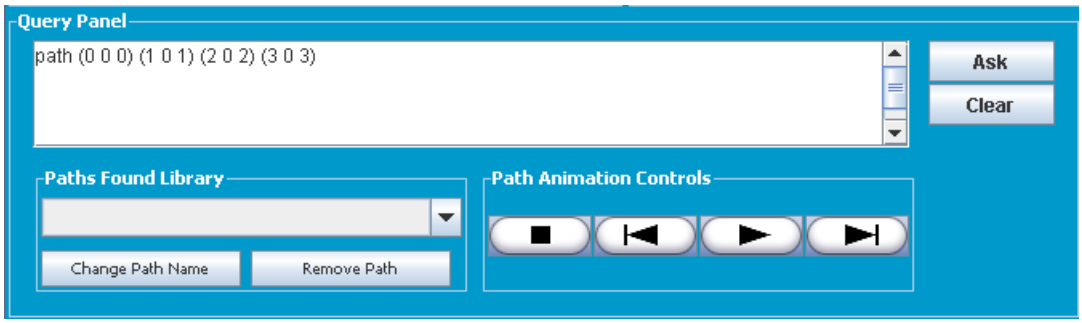

Figura 5.11: Panel de Consulta.

Editor de texto: permite al usuario especificar una consulta para el mapa. Si la consulta es válida, al presionar el botón Ask, la consulta se envía a uno de los esclavos para determinar su solución.

Librería de rutas: su función es almacenar las rutas generadas por el método PRM. Cada vez que se encuentra una ruta, ésta es agregada a la librería de rutas en donde se le identifica con un nombre creado de forma secuencial.

Controles de Animación

Los controles de animación se utilizan para animar al actor digital a lo largo de una ruta. Estos controles funcionan de forma similar a los controles de cualquier reproductor multimedia, pues permiten iniciar, pausar, detener, adelantar y regresar la animación del actor digital.

Generador de Configuraciones

El generador de configuraciones tiene la función de auxiliar al usuario en la generación de consultas. Como ya se mencionó, al momento de especificar una consulta el usuario debe indicar dos o más posiciones entre las que habrá de buscarse una ruta para el actor digital. Las posiciones deben especificarse de forma exacta, y por esta razón, realizar consulta más o menos extensas en cuanto al número de posiciones puede resultar una tarea tediosa y un tanto complicada para el usuario. Además, en muchas ocasiones necesitamos que el actor digital pase por una región determinada del espacio de trabajo, por ejemplo, un corredor, que se mueva alrededor de un obstáculo, que vaya de un extremo del espacio de trabajo a otro, etc. En estos casos, la posición inicial y final no es tan importante. Lo realmente importante es determinar si nuestra implementación del método PRM es capaz de determinar una ruta le permita al actor digital realizar esos movimientos. En estos casos, el generador de configuraciones es de gran utilidad.

El generador de configuraciones es básicamente un mapa donde el usuario puede seleccionar una serie de posiciones al presionar el botón izquierdo del mouse sobre él. Este mapa muestra una vista aérea del espacio de trabajo en donde se identifican fácilmente las regiones ocupadas por los obstáculos. De esta manera, el usuario puede construir consultas de forma fácil y rápida, ya que con las posiciones seleccionadas en el mapa, se construye una consulta válida que se muestra en el editor de texto y que puede utilizarse para interrogar al método PRM sobre la existencia de una ruta. La figura 5.12 muestra el generador de configuraciones.

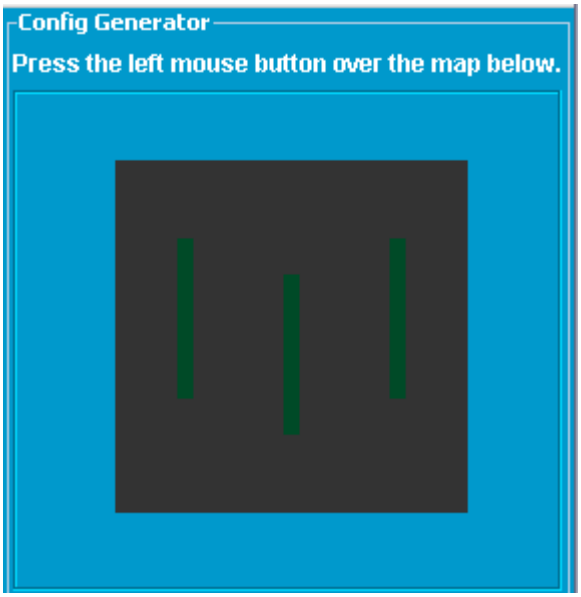

Figura 5.12: Generador de configuraciones.

5.4 Creación de un espacio de trabajo

Crear un espacio de trabajo es una tarea muy sencilla, pues únicamente se necesita que el usuario especifique los archivos que definen al actor digital y los obstáculos, y que ajuste los parámetros de construcción para el PRM descritos en el capítulo 4. El procedimiento a seguir se describe a continuación.

El primer paso es seleccionar la opción *New Work Space* del menú *File*, o bien utilizar el atajo definido para esta opción: *Ctrl + N*. Al seleccionar esta opción aparece en pantalla el dialogo que se muestra en la figura 5.13.

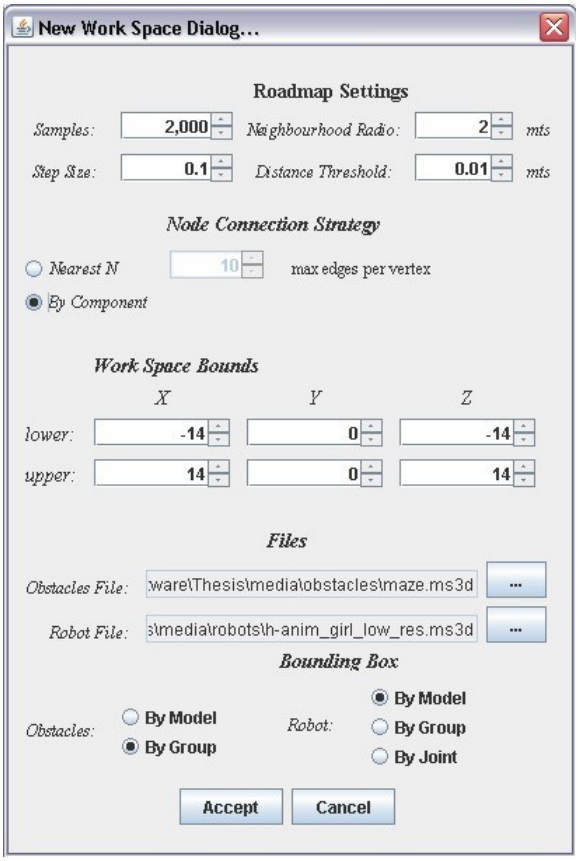

Figura 5.13: Dialogo para la creación de un espacio de trabajo.

En este dialogo se muestran los parámetros que serán utilizados para la construcción del PRM. Estos parámetros deben ajustarse para que el mapa capture la conectividad del espacio de trabajo y su descripción es la siguiente:

- *Samples*: Número de configuraciones que serán generadas durante la construcción del mapa. En la figura 5.13, el valor correspondiente a este parámetro indica que se generarán 2000 configuraciones.

- *Neighbourhood Radio*: Indica la distancia que será utilizada para definir el vecindario de un nodo. En la figura 5.13, se indica una distancia de 2 metros.

- *Step Size*: Frecuencia con la que se realizaran pruebas de detección de colisiones al tratar de agregar una arista. En la figura 5.13, el valor 0.1 indica que la prueba de detección de colisiones se realizará cada 10 centímetros a lo largo de una arista.

- *Distance Threshold*: Este parámetro indica la separación mínima que debe existir entre el actor digital y los obstáculos para considerar que una configuración es libre de colisión. En la figura 5.13 se considera una separación de 1 centimetro.

Después de ajustar estos parámetros debe elegirse una estrategia para la conexión de los nodos del mapa. La aplicación permite elegir entre las dos opciones descritas en el capítulo 4: *Nearest N* y *By Component*. Si se elige la estrategia *Nearest N*, el usuario debe ajustar el número máximo de conexiones por nodo. Posterior a esto, se especifican los límites del espacio de trabajo que definen el área en la que se generarán las configuraciones que serán agregadas al mapa. Por último, es necesario que el usuario seleccione los archivos que definen al actor digital y los obstáculos. Estos archivos se seleccionan mediante un dialogo como el que se muestra en la figura 5.14.

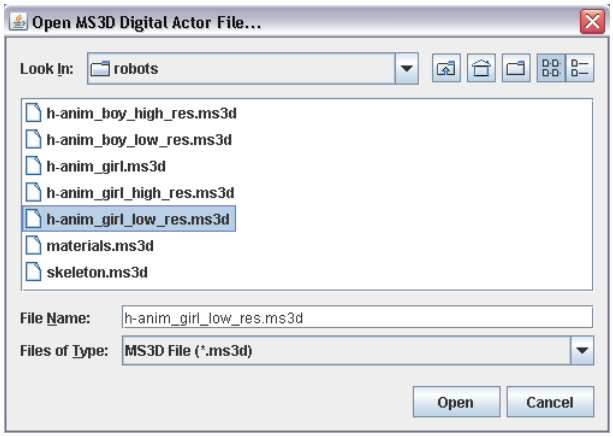

Figura 5.14: Selección de los archivos que definen al actor digital y a los obstáculos.

Como se recordará, el actor digital y los obstáculos se definen en archivos bajo el formato Milkshape 3D. Por esta razón, el dialogo para la selección de archivos utiliza un filtro que permite mostrar al usuario sólo los archivos de este tipo. Finalmente, el resultado que obtenemos es similar al que se muestra en la figura 5.15.

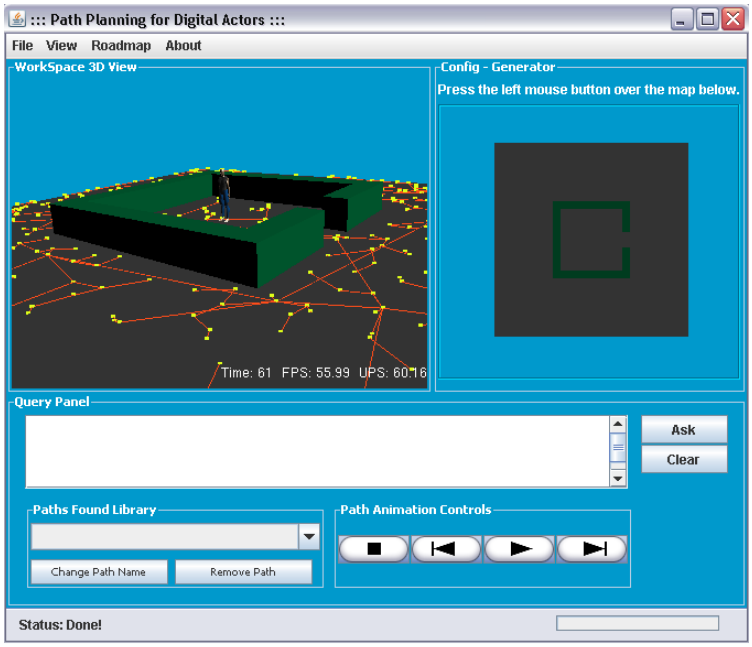

Figura 5.15: Creación de un espacio de trabajo.