

UNIVERSIDAD AUTONOMA
METROPOLITANA (Azcapotzalco)
Ingeniería en Computación

Desarrollo de un robot virtual basado en un
modelo de navegación no cartesiana y su
entorno de desplazamiento

Autor:

Juan Manuel Martínez Hernández
204202062

Asesor:

Dr. Georgii Khatchatourov
No.eco. 22664

Índice

Objetivo general.....	3
Objetivos particulares	3
Antecedentes	4
Justificación	6
Descripción técnica	7
Funciones	7
Descripción de bloques funcionales.....	8
Especificaciones Técnicas.....	9
Aplicación.	9
Software	10
Bibliografía	11
Anexo 1. Manual de Usuario	12

Objetivo general

Se desarrolló un robot móvil basado en navegación no cartesiana, el cual se desplaza en un entorno, con base en un lenguaje de instrucciones y será usado en futuras investigaciones de navegación automática.

Objetivos particulares

- Desarrollo del entorno virtual y de los obstáculos usados dentro del entorno, así como la implementación de los algoritmos para la manipulación de obstáculos en el entorno virtual
- Diseño y programación de la sensórica (el sistema de percepción) del robot virtual basada en imágenes unidimensionales.
- Desarrollo del lenguaje del supervisor para la construcción del entorno del robot (espacio con obstáculos).
- Desarrollo del lenguaje de control de movimiento del robot.
- Desarrollo del sistema de visualización del mundo virtual, que muestra en pantalla, tanto lo que percibe internamente el robot mediante sus sensores, como la vista de una cámara virtual controlada por el supervisor con ayuda de la librería OpenGL.
- Diseño e implementación de la interfaz para la integración del robot virtual con el sistema de navegación no cartesiana.

Antecedentes

Numerosos artículos [1] manejan el movimiento de un robot como el siguiente problema Cartesiano: dado un objetivo y un estado inicial del robot como puntos en un espacio Euclidiano, encontrar un camino entre esos dos puntos evadiendo obstáculos. Esta interpretación evita el uso de modelos de entorno complicados, ya que los algoritmos para encontrar una ruta, modifican un camino reaccionando a un nuevo obstáculo. Por otro lado, para la mayoría de los problemas de navegación más naturales el enfoque Cartesiano es inadecuado. Por ejemplo, la tarea “*ir al comedor*” no asume el uso de ningún sistema Cartesiano, además si el robot ya ha estudiado su entorno, no es natural involucrar alguna tarea de exploración.

Las tareas involucradas en la navegación de un robot móvil son: la percepción del entorno a través de sus sensores, de modo que le permita crear una abstracción del mundo; la planificación de una trayectoria libre de obstáculos, para alcanzar el punto destino seleccionado; y la guía del robot a través de la referencia construida. De forma simultánea, el robot puede interactuar con ciertos elementos del entorno.

Dentro de nuestra casa de estudios se ha realizado un proyecto [2] relacionado con la simulación y entrenamiento de un robot virtual mediante el modelo GT.

En términos formales, un modelo GT es definido como un par de grafos $\{G, T\}$ donde los componentes satisfacen las siguientes propiedades:

$G = (V_G, E_G)$ es un grafo dirigido con pesos no-negativos asignados a los extremos.

$T = (V_T, E_T)$ es un grafo dirigido que satisface la siguiente condición:

T tiene un único nodo fuente donde el conjunto de sus nodos terminales coincide con V_G .

En otras palabras, podemos decir que G será el grafo de conocimientos causa/efecto y T será el grafo de conocimientos geográficos del robot.

Uno de los objetivos logrados en este proyecto fue el realizar el lenguaje de instrucciones para el entrenamiento y el respectivo simulador que posteriormente ayudarán en el entrenamiento del robot, así como la creación de los grafos G y T.

En el ámbito industrial existen diversos simuladores, los cuales en su gran mayoría son para robots que no recorren distancias, esto es, robots que solo están formados por un brazo mecánico, un conjunto de efectores y que sus únicas funciones son, entre otras, manipular objetos mediante dichos efectores [3]. Esta característica es precisamente la que hace diferente a nuestro proyecto de este tipo de simuladores, puesto que el robot que desarrollaremos tendrá que desplazarse en un entorno y no se toma en cuenta factores como dispositivos de movimiento (ruedas, efectores), a diferencia de los usados en el ámbito industrial, que se mantienen en una sola posición y si toman en cuenta este tipo de factores.

Justificación

En este proyecto la finalidad es la realización de un componente indispensable para la implementación práctica del enfoque GT y su investigación futura, cuya meta es demostrar que la simulación del robot en un ambiente virtual puede ser manejada adecuadamente por medio de la navegación no cartesiana y así poder aproximar la forma en la que se podría desplazar un robot real en su entorno.

En general, la representación de los entornos reales es muy compleja. (Por ejemplo, a través de la técnicas de realidad virtual [4]) Sin embargo para dicha realización es suficiente limitarse a usar un entorno 2D. Esto cubre los objetivos de estudio de eficiencia del enfoque GT. Lo primordial en dicho enfoque es satisfacer la necesidad de crear un robot que funcione con instrucciones basadas en la relación “causa – efecto”. Esto es, que de un conjunto de instrucciones proporcionadas por el administrador que involucran controles (causa), el robot pueda cambiar su vista del entorno y registrar los cambios correspondientes (efecto).

Actualmente dentro de la universidad se encuentra en desarrollo un proyecto llamado “Investigación y desarrollo sobre eventos en flujo de imágenes mediante Inteligencia Artificial”[5] cuyo objetivo es usar las técnicas de IA para detección de eventos "relevantes" en un flujo de imágenes unidimensionales de un robot virtual para su uso posterior para creación automática de los mapas de geografía implícita.

Dentro de los planes a futuro que se tienen para ambos proyectos está el almacenamiento en una base de datos de los resultados de las acciones del robot. Dicha base de datos deberá contener un par de grafos ya mencionados, G (grafo de conocimientos causa y efecto) y T (grafo de conocimientos geográficos del robot) para cada uno de los resultados obtenidos.

Descripción técnica

Funciones

El proyecto consta de una interfaz grafica, la cual permite a un administrador supervisar los movimientos del robot, en particular, muestra el entorno tanto en la vista del supervisor, como en la vista del robot. Además, la supervisión cuenta con un modulo donde se hace uso del teclado para acceder a los comandos de movimiento que el robot puede realizar tales como:

- Avanzar
- Girar a la derecha o a la izquierda
- Retroceder

La aplicación genera un entorno virtual 2D, el cual está dotado de un conjunto de obstáculos (figuras geométricas primitivas) para el robot, y éste es manipulado por medio de los comandos antes mencionados, de forma tal que el robot percibe cuándo está cerca de un obstáculo.

La interfaz grafica que se ha desarrollado para la supervisión del robot consiste en una vista superior del entorno simulado donde el robot se desplaza para que de esta manera el supervisor pueda ver en conceptos amigables todos los movimientos efectuados por el robot.

Asimismo, para el robot se desarrolló una cámara virtual de imágenes de rango 1D, en la cual podrá visualizar los objetos que se encuentran a su alrededor para medir su distancia respecto a los mismos y así, en conjunto con los comandos proporcionados por el administrador tomar decisiones de los movimientos que puede hacer.

Descripción de bloques funcionales

A continuación se muestra un diagrama a bloques con el funcionamiento de la aplicación:

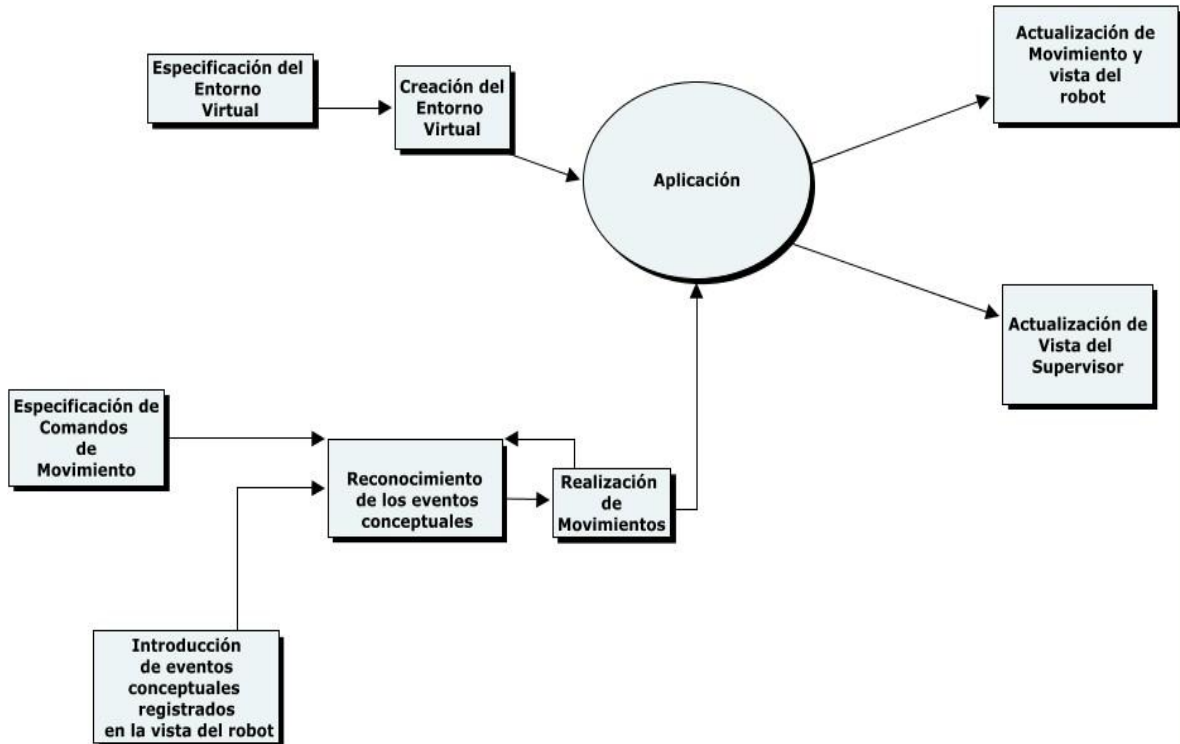


Fig. 1. Descomposición en bloques funcionales.

Especificación del entorno virtual. En este apartado se especifica el conjunto de obstáculos que formarán parte del entorno.

Creación del entorno virtual. Este modulo crea el entorno antes especificado.

Especificación de los comandos de movimiento. Aquí se reciben los comandos de movimiento para el desplazamiento del robot.

Introducción de eventos conceptuales registrados en la vista del robot. Este apartado provee al robot de un conjunto de eventos que han sido registrados en su sensórica para el reconocimiento de obstáculos.

Reconocimiento de los eventos conceptuales. En este apartado el robot interpreta los parámetros de movimiento que le han sido especificados con base a su sensórica.

Realización de movimientos. Una vez interpretados los parámetros de movimiento, el robot comienza a desplazarse dentro del entorno.

Actualización de movimiento y vista del robot. Aquí se modifica la vista del robot con cada movimiento efectuado por este último, así como su posición dentro del entorno.

Actualización de vista del supervisor. Aquí se modifica la vista del supervisor cuando el robot realice algún movimiento.

Especificaciones Técnicas

Aplicación.

En la especificación del entorno virtual se crea un entorno en el que pueden ser agregados los obstáculos a usar, así como la ubicación inicial del robot dentro del entorno. Los obstáculos que son usados en la aplicación son cubos y pirámides.

Para la especificación de los comandos de movimiento se hace uso del teclado para seleccionar los comandos *avanzar*, *girar* y *retroceder*, y de esta manera el robot los interprete y posteriormente los ejecute.

El lenguaje de instrucciones (comandos) para los movimientos del robot es básico porque el entorno modelado así lo requiere. Estos comandos son hasta cierto punto obvios para nosotros, por ejemplo la instrucción *avanzar* nos dice que tenemos que desplazarnos en una dirección pero el robot no lo interpreta de manera intuitiva, entonces, la tarea del robot será interpretar esos comandos para realizar las respectivas funciones. Además de que posteriormente el uso de este tipo de instrucciones facilitará la implementación de la fase de entrenamiento del robot y el correspondiente llenado de la información de los grafos G y T.

La aplicación muestra los cambios que va teniendo la visión del robot, conforme interpreta y ejecuta los comandos ingresados por el supervisor. El robot contará con una visión 1D y un ángulo de visión de 120 grados.

Para una mayor referencia del funcionamiento de la aplicación ejecutable consultar el anexo 1 (manual de usuario) en el cual se describe la forma adecuada de usar la aplicación ejecutable.

Software

Los sistemas operativos en los que fue probado el proyecto son los siguientes:

- Windows XP
- Windows Vista
- Windows 7

El ambiente de desarrollo que se utilizó fue *Microsoft Visual C++* en conjunto con las librerías para el desarrollo de gráficos por computadora *OpenGL*.

Bibliografía

- [1] http://riai.isa.upv.es/CGI-BIN/articulos%20revisados%202006/versiones%20impresas/vol3_num4/articulo3_vol3_num4.pdf
- [2] Netz Romero Durán, Josué Figueroa González y Georgii Khachaturov, "Desarrollo de un simulador para el entrenamiento de un robot virtual y construcción de grafos en el modelo *GT*", 2007 (*proyecto de investigación*)
- [3] <http://www.easy-rob.com/>
- [4] http://macedoniamagazine.frodriq.com/etapa_2001_2002/aula/Flocking/flocking.htm
- [5] Investigación y desarrollo sobre eventos en flujo de imágenes mediante IA. (Proyecto de Investigación MCC UAM Azcapotzalco). José Antonio Espinosa de los Monteros Lizárraga. Aprobación Marzo 2010

Anexo 1. Manual de Usuario

Este archivo puede ser encontrado dentro del cd en el directorio llamado Documentación. El nombre del archivo es Manual de Usuario.pdf

UNIVERSIDAD AUTONOMA
METROPOLITANA (Azcapotzalco)
Ingeniería en Computación

Desarrollo de un robot virtual basado en un
modelo de navegación no cartesiana y su
entorno de desplazamiento

Manual de Usuario

Autor:

Juan Manuel Martínez Hernández
204202062

Asesor:

Dr. Georgii Khatchatourov

No.eco. 22664

Índice

Objetivo.....	3
Requisitos del sistema.....	3
Descripción de sistema de coordenadas usado	4
Ventana Supervisor y Ventana de Robot	4
Ventana Grafica de robot.....	5
Descripción de ventanas	5
Ventana de Supervisor	6
Ventana de Robot	8
Ventana de grafica de robot	9
Modos de Visualización.....	9
Modo 3D.....	10
Modo 2D.....	11
Movimiento Cámara de supervisor.....	12
Teclas de movimiento básicas.....	12
Teclas de movimiento opcionales.....	15
Modo 2D-agregar obstáculos.....	16
Modo 2D-Mover obstáculos	19
Modo 2D-redimensionar obstáculos.....	20
Redimensión ON	21
Redimensión OFF	22
Modo 2D-Rotar obstáculos	22
Rotación ON	23
Rotación OFF	24
Modo 2D-Eliminar obstáculos.....	25
Eliminación ON.....	25
Eliminación OFF.....	26
Modo 3D-Opciones	27
Modo 3D-Alambrico	27
Modo 3D-Solido	28
Índice de figuras	29

Objetivo

El objetivo del presente manual es hacer una detallada descripción de la aplicación ejecutable, así como de cada uno de los módulos que lo componen, para que de esta forma el usuario pueda utilizar de manera adecuada la aplicación.

Requisitos del sistema

Esta aplicación fue probada en los siguientes sistemas operativos:

Windows XP

Windows Vista

Windows 7

Para hacer uso de la aplicación se requiere una computadora con las siguientes características mínimas:

15Mb de espacio libre en disco duro.

Además, el proyecto hace uso de un conjunto de librerías dinámicas (incluidas en el cd) las cuales deben ser colocadas en las siguientes ubicaciones dentro del sistema operativo:

glut32.dll -> C:\WINDOWS\system (Sistema operativo Version 64 bits)

glut32.dll -> C:\WINDOWS\system32 (Sistema operativo Versión 32 bits)

Una vez realizado esto, elegimos ver el contenido del cd, accedemos a la siguiente carpeta:

`PT_Juan_Martinez_Hernandez_204202062\robot_virtual\debug\`

y hacemos doble *click* sobre el icono llamado “*robot_virtual.exe*”. Eso dará como resultado que se abran tres ventanas, de las cuales en los siguientes apartados se da una explicación de su contenido y de las operaciones que se pueden hacer sobre ellas.

Descripción de sistema de coordenadas usado

Ventana Supervisor y Ventana de Robot

En lo que a las ventanas de supervisor y robot se refiere, el sistema de coordenadas usado es el sistema de coordenadas espacial, el cual está formado por 3 rectas perpendiculares entre sí (x , y , z). La siguiente figura muestra la forma en la que están definidas estas rectas.

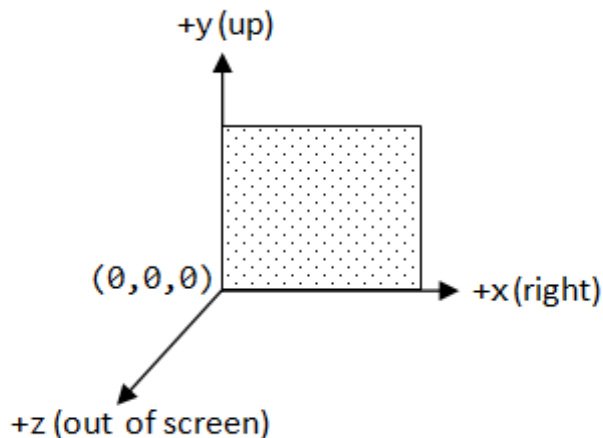


Figura 1. Sistema de coordenadas espacial

Ventana Grafica de robot

Para la interpretación de valores de la ventana de grafica de robot se hace uso del plano cartesiano como sistema de referencia(ver figura 2), en el cual las coordenadas pueden nombrarse mediante dos números: (x,y)

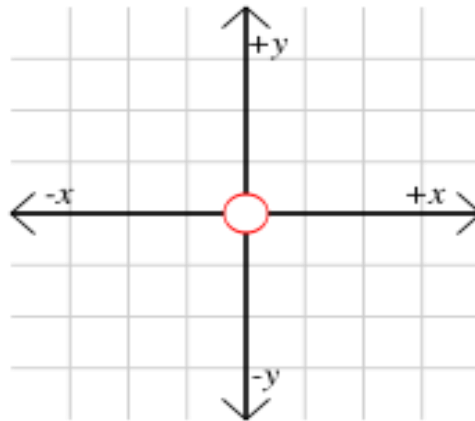


Figura 2. Sistema de coordenadas cartesiano

Descripción de ventanas

La aplicación está compuesta de tres ventanas, en las cuales se puede visualizar lo siguiente:

- **Ventana de Supervisor:** Muestra el escenario visualizado por la cámara del supervisor
- **Ventana de Robot:** Muestra el escenario visualizado por la cámara del robot
- **Ventana de gráfica de robot:** Gráfica del rango de visualización de la cámara del robot

- **Ventana de Estados:** En la que se va mostrando el estado actual de la aplicación (Selección de obstáculos, modos a los que se ingresa etcétera. Los cuales serán explicados posteriormente)

Cada una de las ventanas que componen la aplicación está hecha con la ayuda de las librerías OpenGL.

Ventana de Supervisor

Esta ventana es la que tiene acceso a todas las operaciones de control del entorno (agregar, redimensionar y mover obstáculos). Asimismo, esta ventana cuenta con dos modos de visualización del entorno, cada uno de los cuales permite realizar operaciones distintas. Dichos modos son:

- Modo 2D: manipulación de obstáculos(ver figura 3)
- Modo 3D: recorrido del entorno(ver figura 4)

Hay que mencionar que el uso del teclado está dividido en dos secciones, de las cuales se hablara más adelante:

- Teclas de movimiento básicas
- Teclas de movimiento opcionales

En la ventana del supervisor se tiene acceso a ambas secciones.

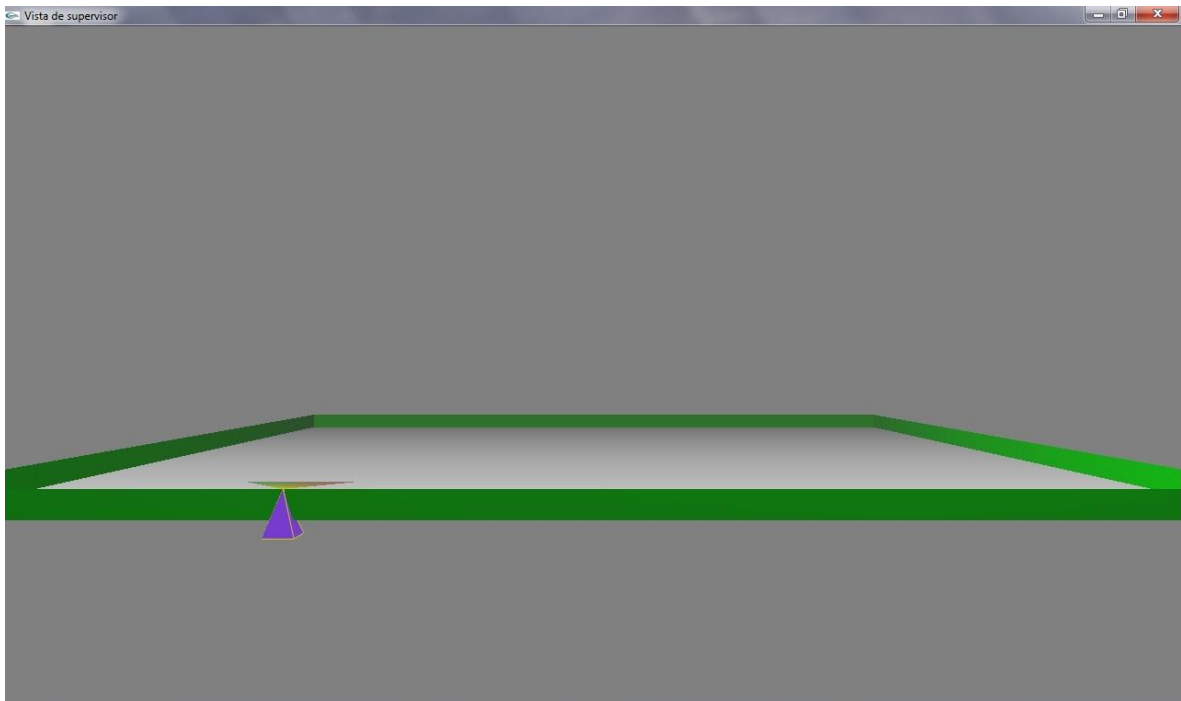


Figura 3. Ventana de Supervisor modo 3D

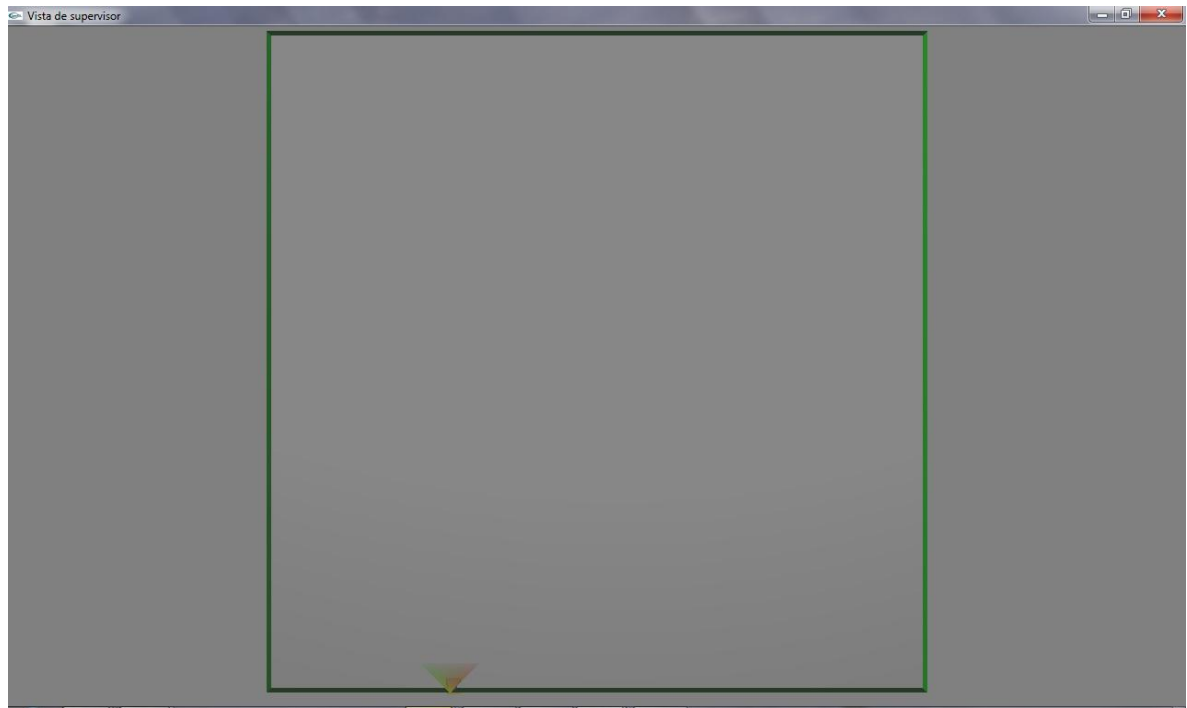


Figura 4. Ventana de Supervisor modo 2D

Ventana de Robot

Esta ventana presenta la visualización de la cámara del robot en su posición inicial. Todo cambio en el entorno se verá reflejado también en esta ventana, siempre y cuando los cambios estén dentro del alcance de la cámara del robot (Ver figura 5).

De igual manera el robot también puede recorrer todo el entorno haciendo uso del teclado, pero sus movimientos son más limitados (solo se pueden usar las teclas de movimiento básicas).

Esta ventana no tiene acceso a los menús de la aplicación.



Figura 5. Ventana de Robot

Ventana de grafica de robot

Esta ventana muestra una grafica en el plano XY, en la que se puede observar el comportamiento de la sensórica del robot en base a la distancia que hay entre la posición de la cámara y algún obstáculo que se encuentre frente a él (Ver figura 6).

La grafica va cambiando conforme se registran cambios relevantes en la sensórica de la cámara del robot (en base a los valores de profundidad obtenidos).

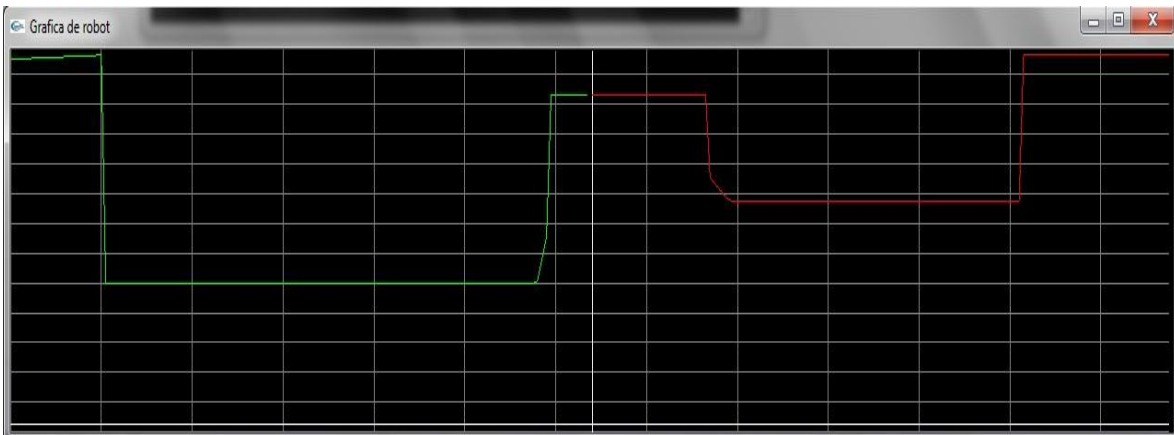


Figura 6. Ventana de gráfica de robot

Modos de Visualización

En la aplicación existen dos modos en los que se puede visualizar el entorno virtual, las cuales permiten realizar distintas funciones. A continuación se explican dichos modos.

Modo 3D

Este es el modo por defecto de la aplicación (ver figura 7). Particularmente este modo está diseñado para navegar libremente en el entorno virtual y así poder visualizar mejor tanto el escenario, el robot y todos los obstáculos. Esto se logra haciendo uso tanto de las teclas de movimiento básicas como de las opcionales (explicadas a detalle más adelante)

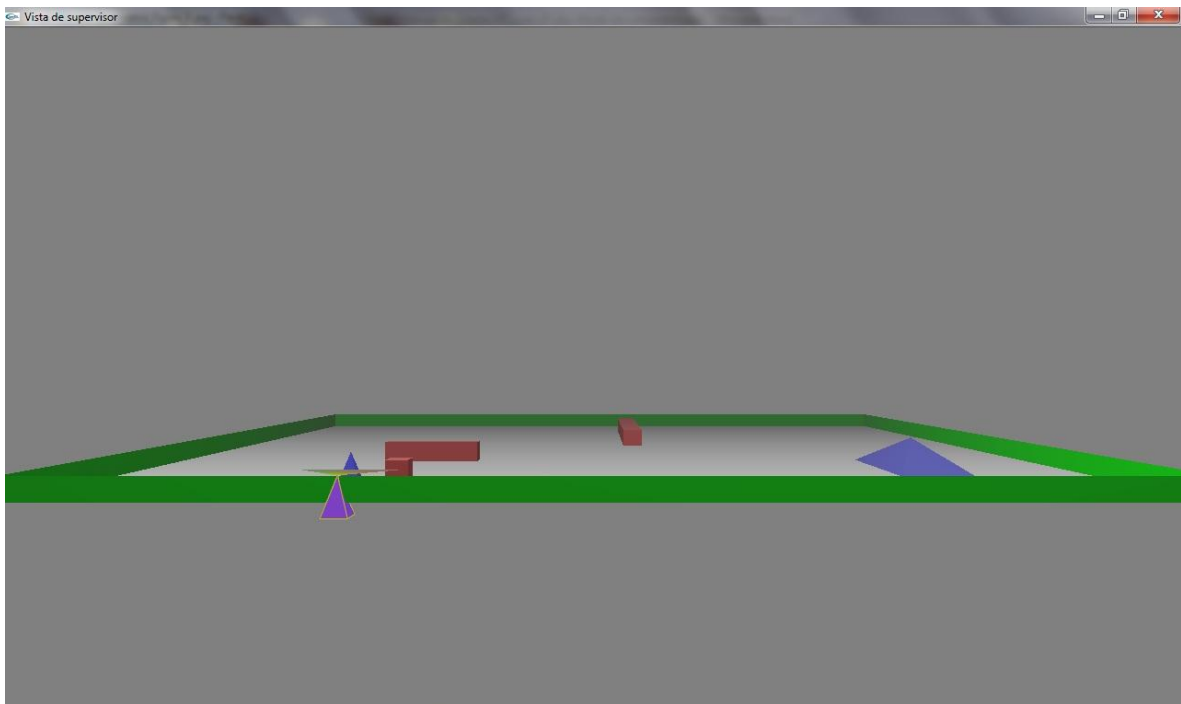


Figura 7. Modo 3D

Modo 2D

Dentro de este modo se puede visualizar completamente el entorno al colocar la cámara del supervisor en la parte superior del escenario (ver figura 8), permitiendo así la fácil manipulación de los obstáculos (movimiento, redimensión, rotación). Para entrar a este modo es necesario oprimir el botón derecho del mouse para desplegar el menú de opciones y seleccionar la opción “modo 2D”

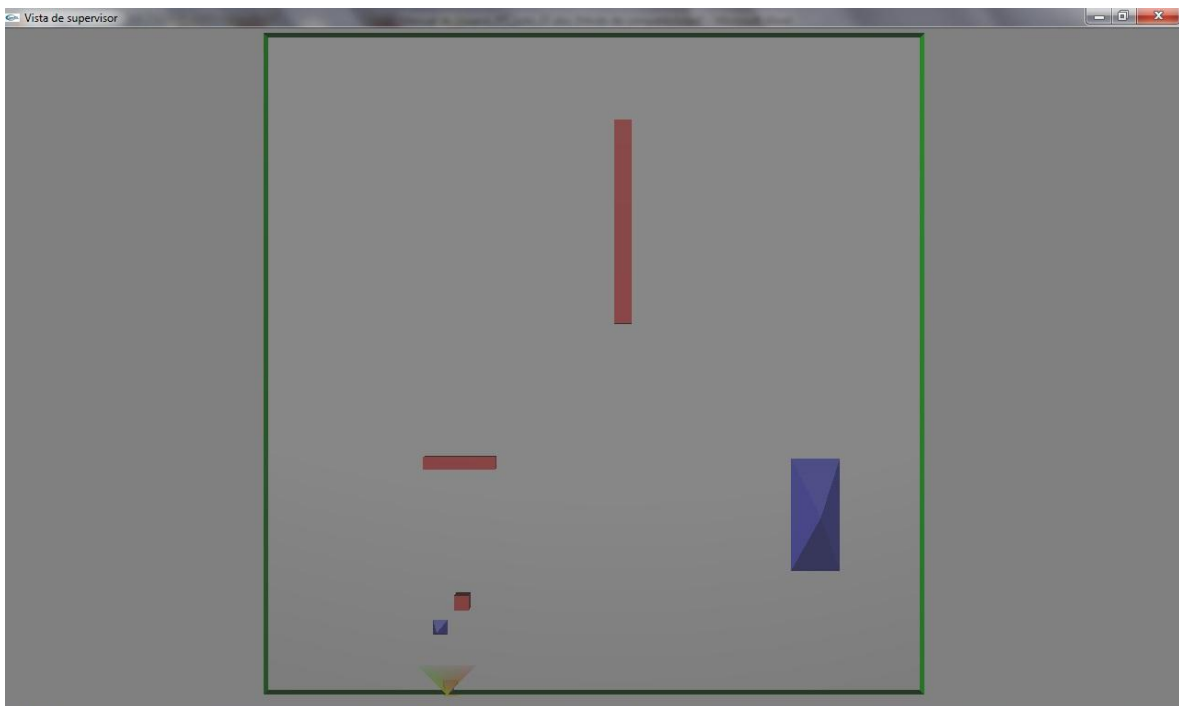


Figura 8. Modo 2D

Nota: Es recomendable estar en este modo para manipular los obstáculos.

Movimiento Cámara de supervisor

Como se menciona anteriormente, la ventana del supervisor cuenta con una cámara, la cual, puede recorrer libremente el entorno virtual haciendo uso del siguiente conjunto de teclas:

- Teclas de movimiento básicas
- Teclas de movimiento opcionales

Teclas de movimiento básicas

Este conjunto de teclas permite a la cámara del supervisor realizar distintos movimientos elementales para desplazarse dentro del entorno. Dichos movimientos se encuentran asociados al siguiente conjunto de teclas:

Tecla ←: Gira la cámara del supervisor a la izquierda (Ver figura 9)

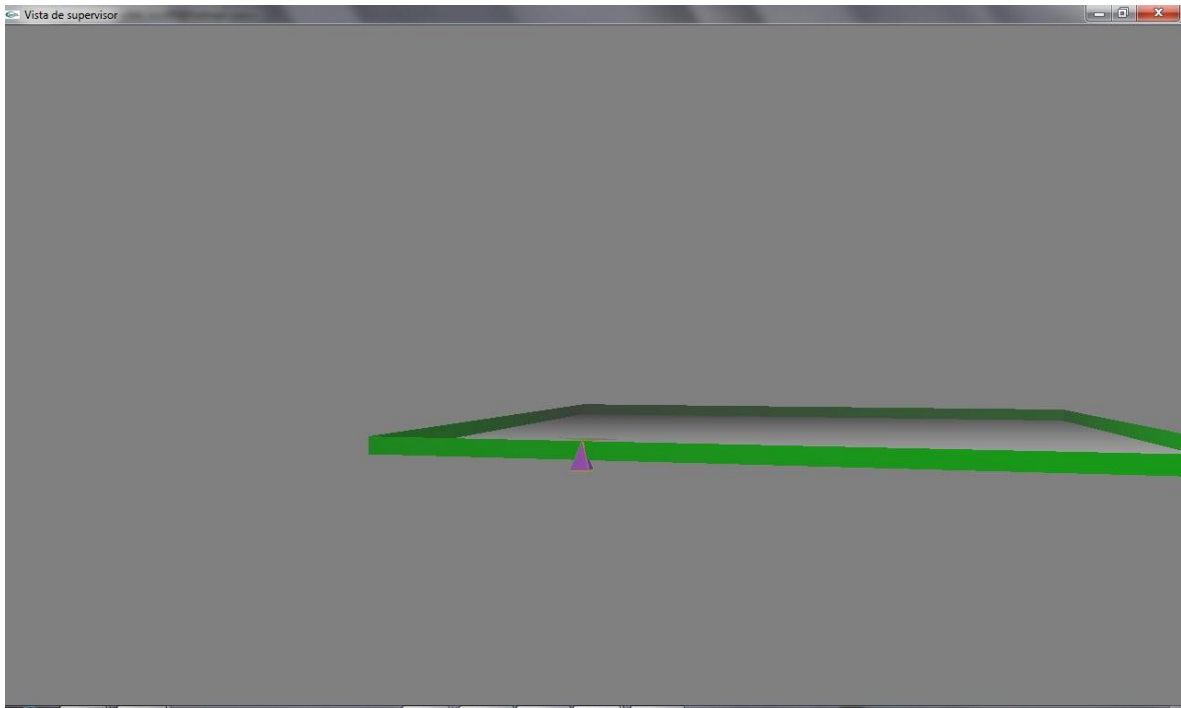


Figura 9. Uso de tecla ←

Tecla →: Gira la cámara del supervisor a la derecha (Ver figura 10)

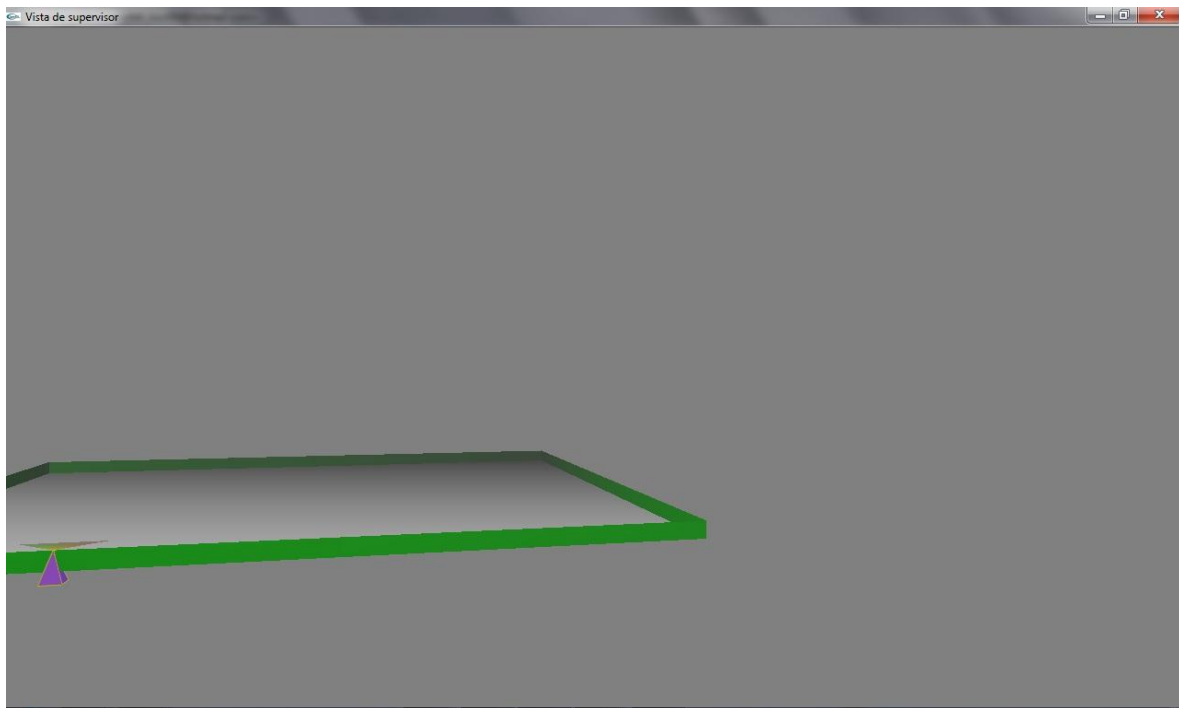


Figura 10. Uso de tecla →

Tecla ↑: Desplaza hacia adelante la cámara del supervisor (Ver figura 11)

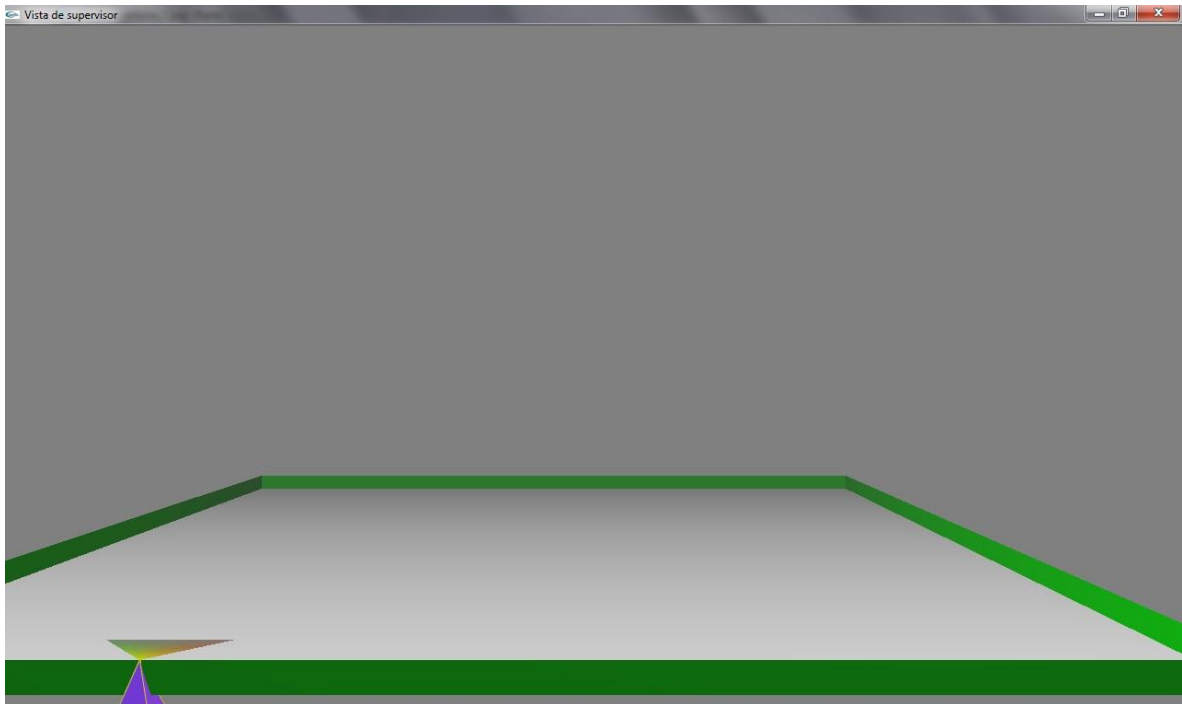


Figura 11. Uso de tecla ↑

Tecla ↓: Desplaza hacia atrás la cámara del supervisor (Ver figura 12)

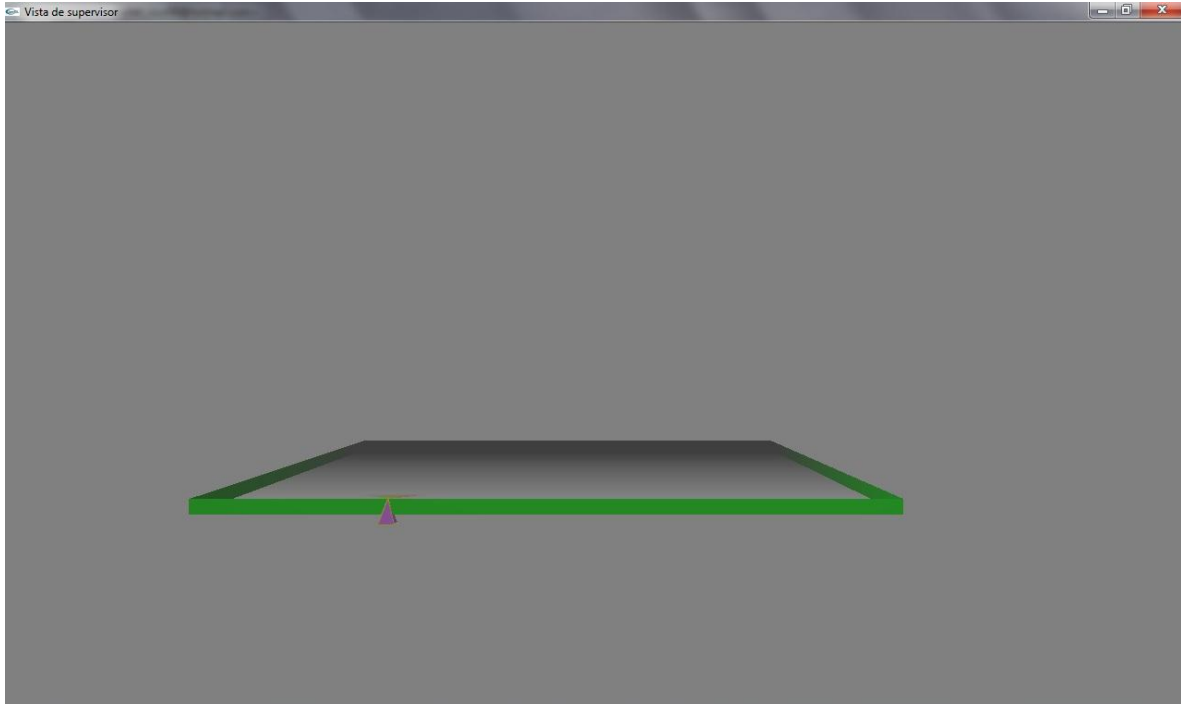


Figura 12. Uso de tecla ↓

Teclas de movimiento opcionales

De la misma manera que los movimientos básicos, el supervisor puede tener acceso a movimientos opcionales para desplazarse en el entorno virtual haciendo uso del siguiente conjunto de teclas:

Tecla F1: Desplaza la cámara hacia arriba, manteniendo el mismo ángulo de visión

Tecla F2: Desplaza la cámara hacia abajo, al mismo tiempo que gira el ángulo de visión hacia arriba

Tecla F3: Desplaza la cámara a la izquierda, manteniendo el mismo ángulo de visión

Tecla F4: Desplaza la cámara a la derecha, manteniendo el mismo ángulo de visión

Tecla F5: Desplaza la cámara hacia abajo, manteniendo la misma posición

Tecla F6: Desplaza la cámara hacia arriba, al mismo tiempo que gira el ángulo de visión hacia abajo

Modo 2D-agregar obstáculos

Para agregar un obstáculo dentro del entorno virtual debe colocar el cursor en la ventana del supervisor y oprimir el botón derecho del mouse para que se despliegue un menú como el de la figura 13:

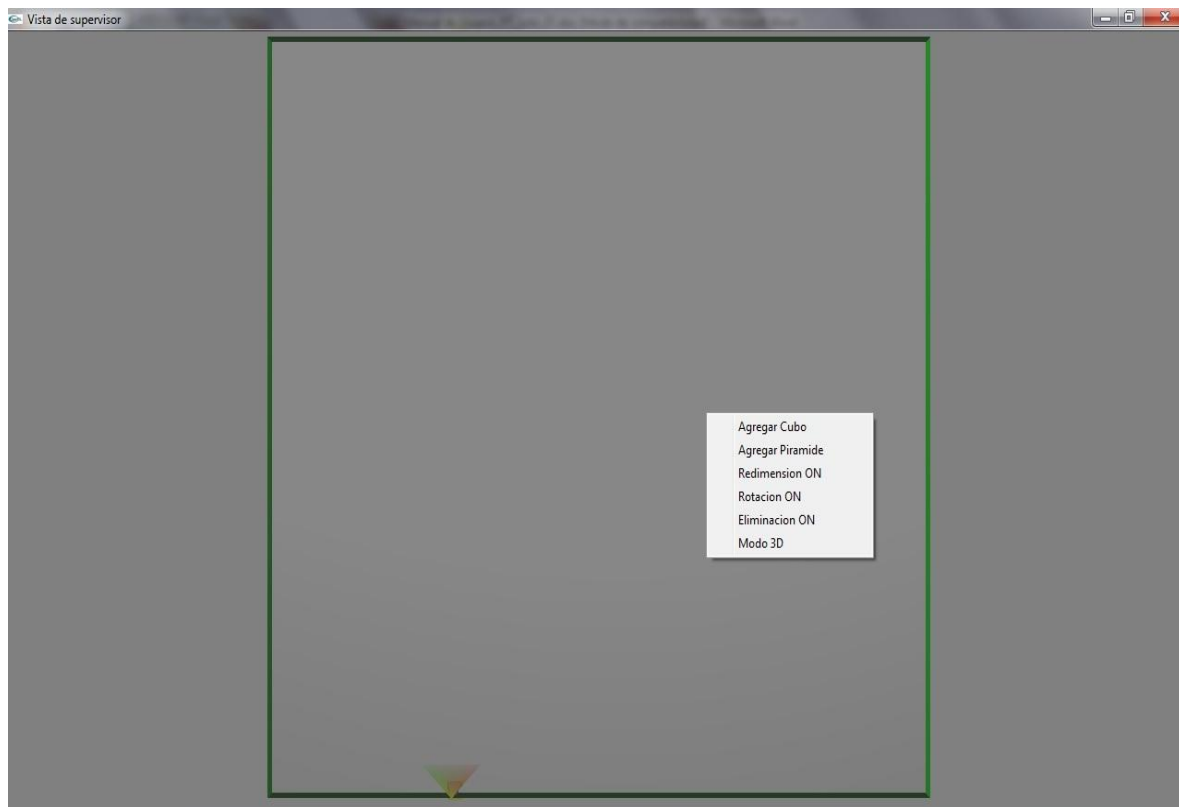


Figura 13. Menú modo 2D

Posteriormente se puede seleccionar alguna de las dos opciones mostradas en dicho menú para agregar un obstáculo, y dependiendo de la figura que deseemos agregar, esta puede ser un cubo (Ver figura 14) o una pirámide (Ver figura 15).

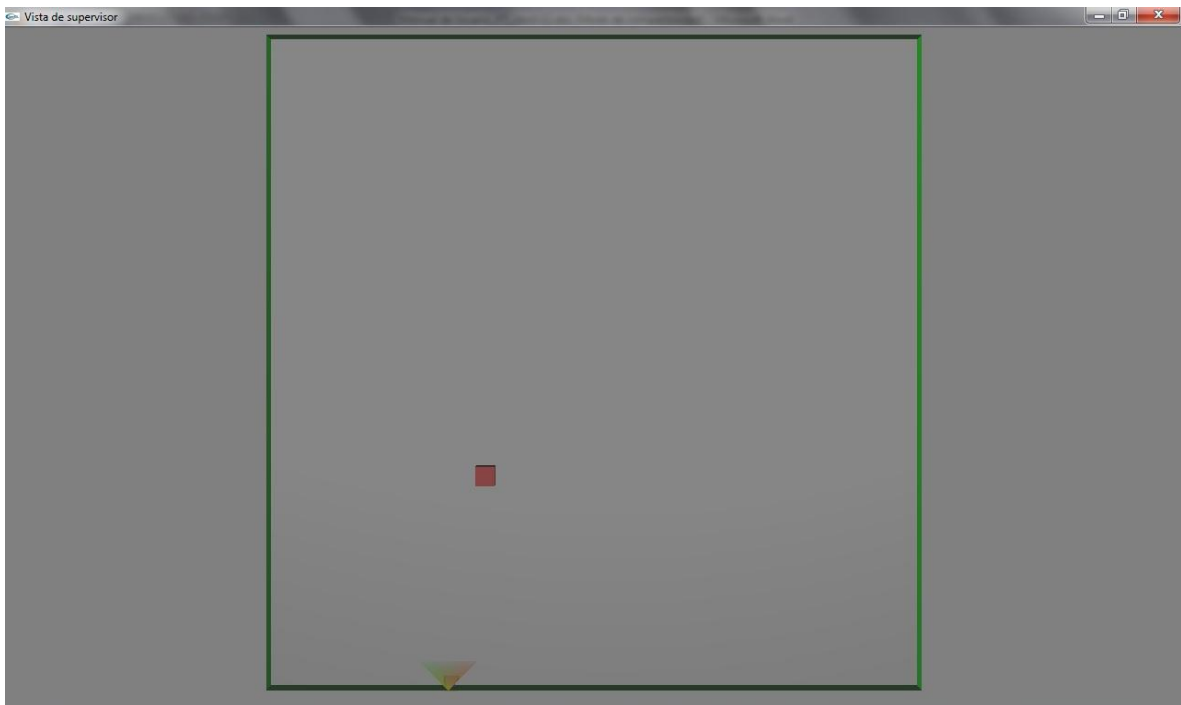


Figura 14. Opción Agregar Cubo

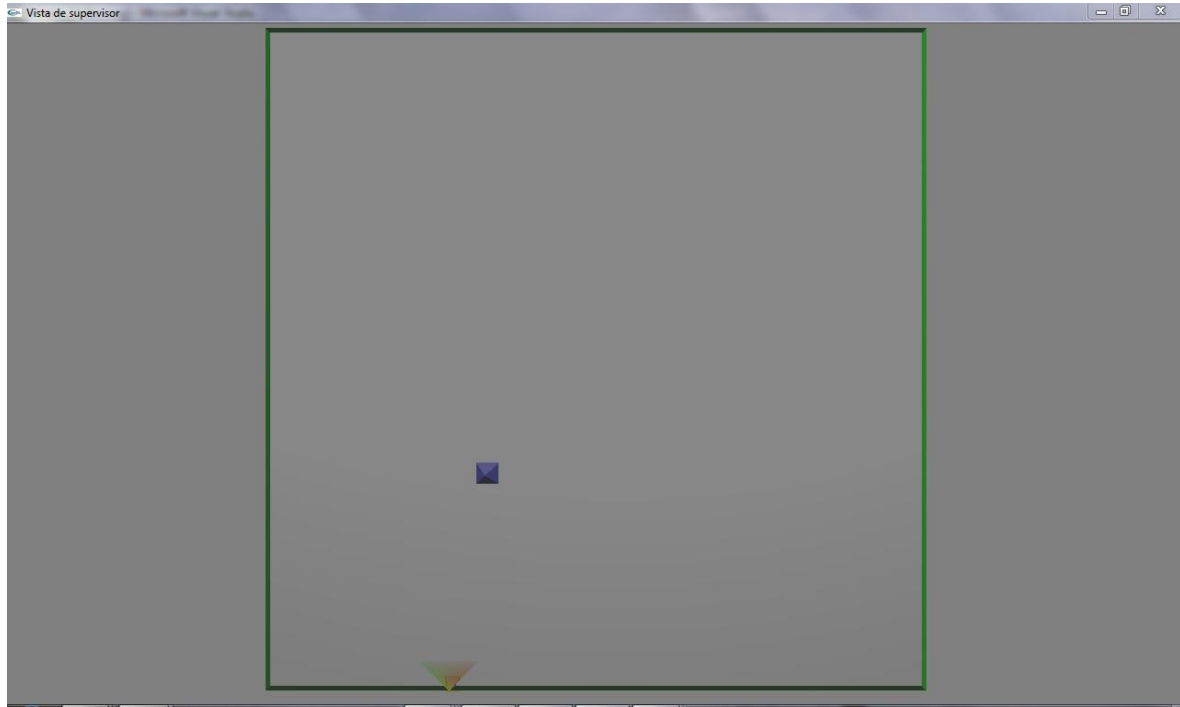


Figura 15. Opción Agregar Pirámide

Nota: La cantidad máxima de obstáculos que pueden ser agregados es de 20, ya sean cubos o pirámides.

Modo 2D-Mover obstáculos

Para cambiar la posición de un obstáculo dentro del entorno virtual basta con posicionar el cursor sobre el obstáculo que se desea mover (Ver Figura 16), oprimir el botón izquierdo del mouse, y sin soltar el botón mover el cursor hasta la posición donde se desea que quede el obstáculo (Ver Figura 17).

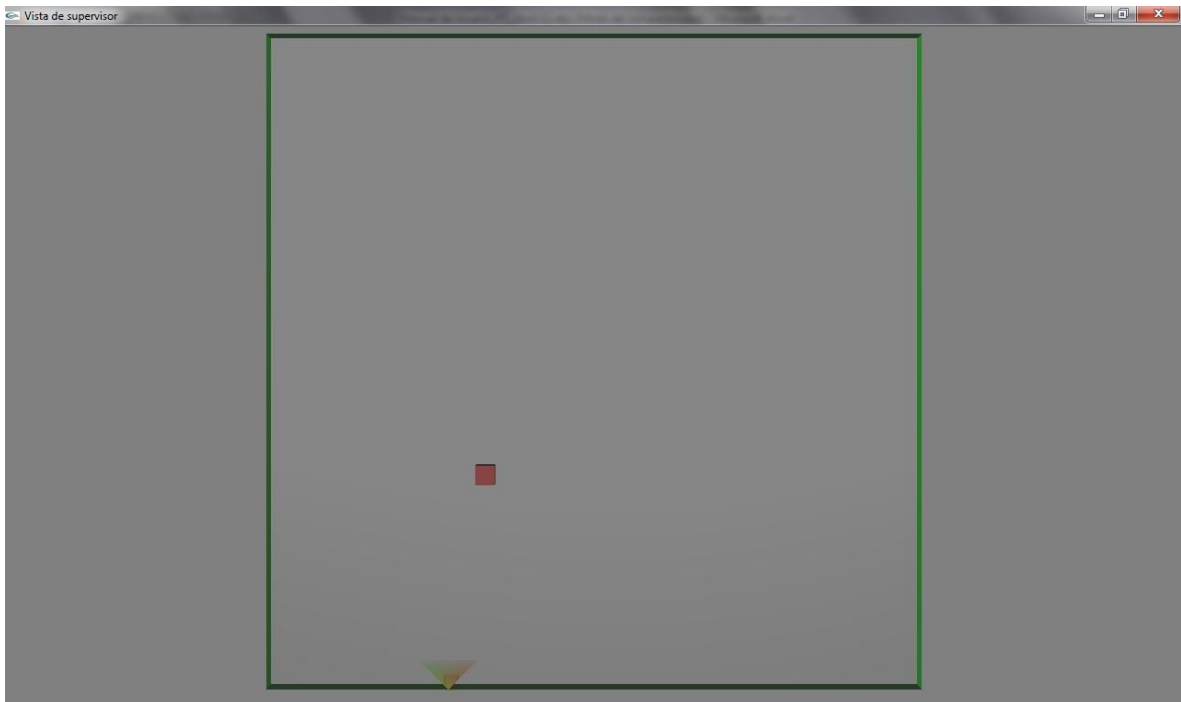


Figura 16. Antes de mover obstáculo

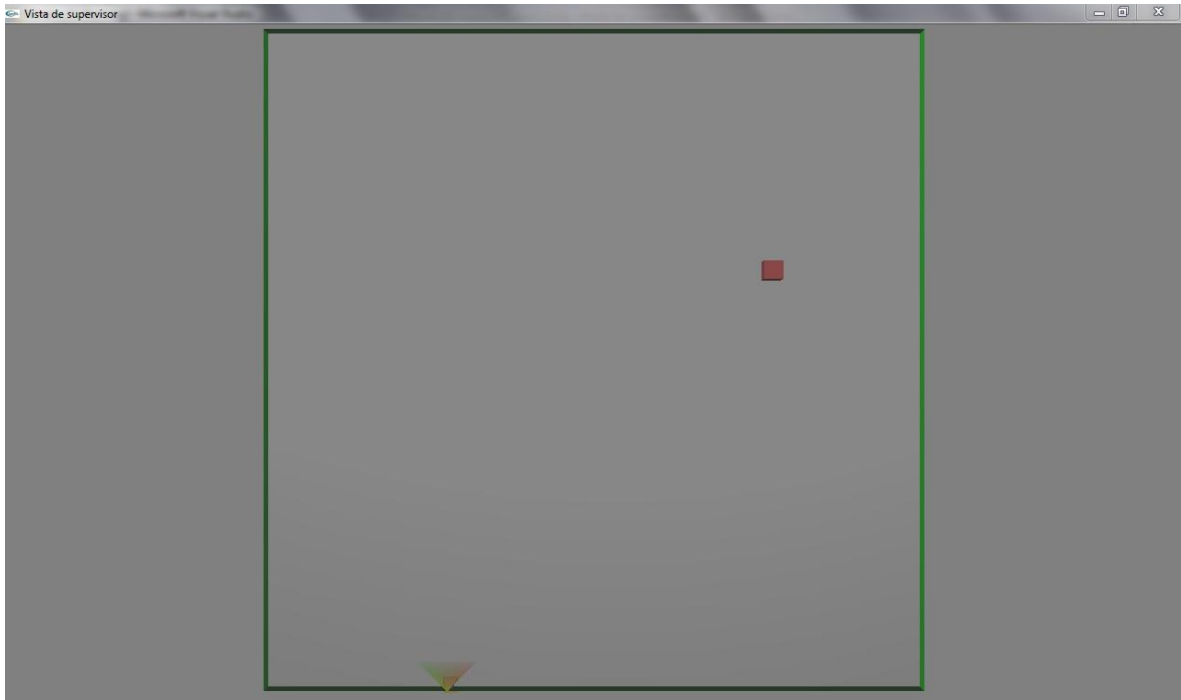


Figura 17. Después de mover obstáculo

Nota: Se recomienda estar en el modo pantalla completa para poder desplazar los obstáculos de manera adecuada

Modo 2D-redimensionar obstáculos

Para manipular los obstáculos en el modo "redimensionar obstáculos" se hace uso de la siguiente opción del menú:

- Redimensión ON

Redimensión ON

Una vez seleccionada esta opción, debe colocar el cursor sobre el objeto que desee redimensionar (Ver figura 18), oprimir el botón izquierdo del mouse, y sin soltar el botón mover el cursor hacia la izquierda para incrementar la dimensión del obstáculo sobre el eje X, o hacia arriba para incrementar la dimension del obstáculo sobre el eje Z según se desee (Ver figura 19).

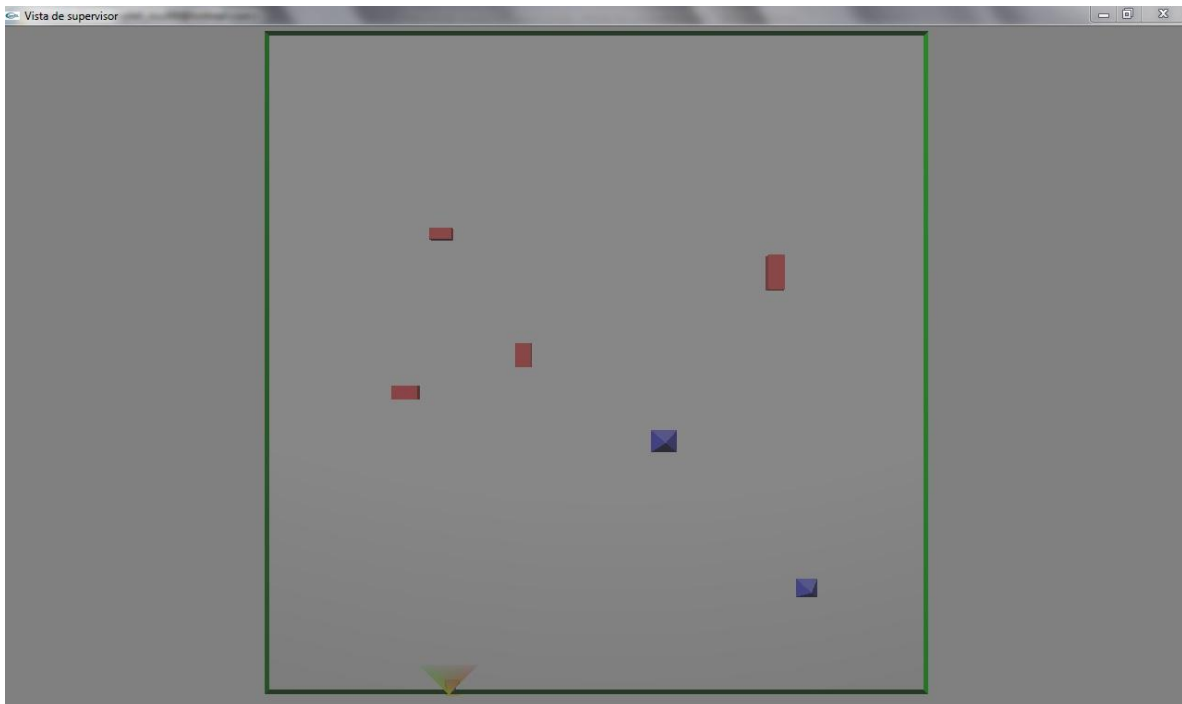


Figura 18. Antes de Redimensionar obstáculos

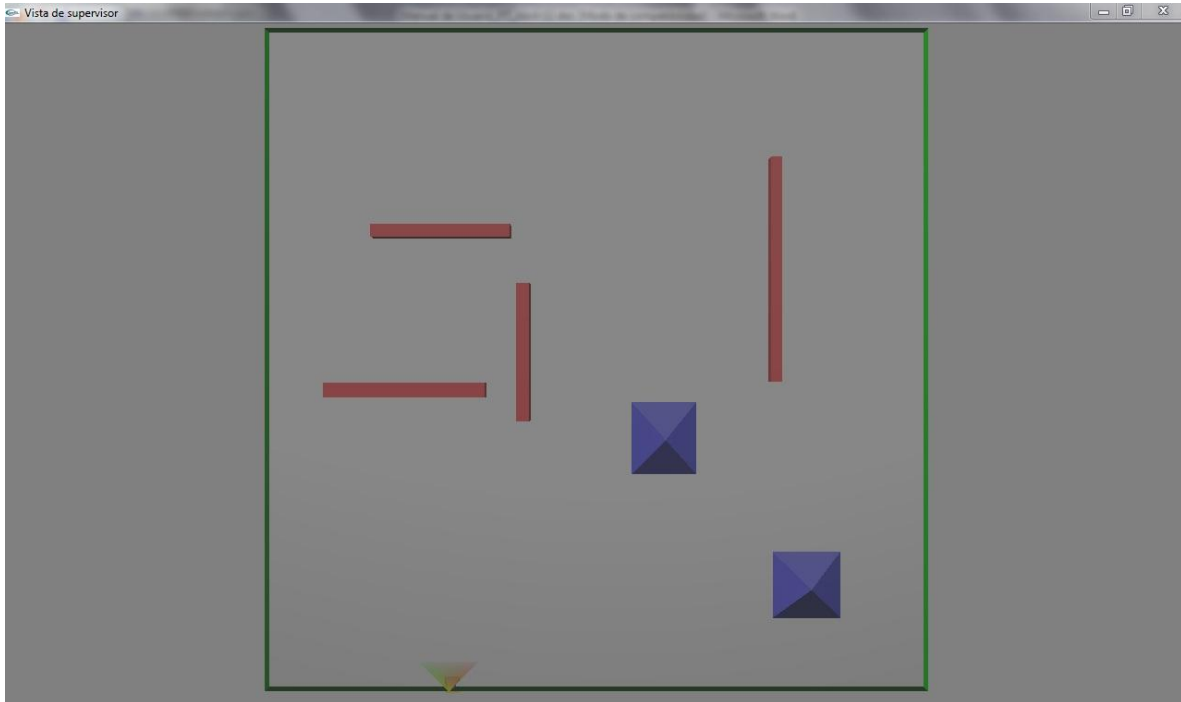


Figura 19. Después de redimensionar obstáculos

Redimensión OFF

Esta opción es usada cuando se ha terminado de redimensionar los objetos seleccionados.

Nota: Se recomienda estar en el modo pantalla completa para poder redimensionar los obstáculos de manera adecuada

Modo 2D-Rotar obstáculos

Para iniciar con la rotación de los obstáculos se necesita ingresar al modo "rotar obstáculos", el cual está formado por la siguiente opción:

- Rotación ON

Rotación ON

Una vez seleccionada esta opción, debe colocar el cursor sobre el objeto que desee rotar (Ver figura 20), oprimir el botón izquierdo del mouse, y sin soltar el botón comenzar a mover el mouse para iniciar la rotación (Ver figura 21).

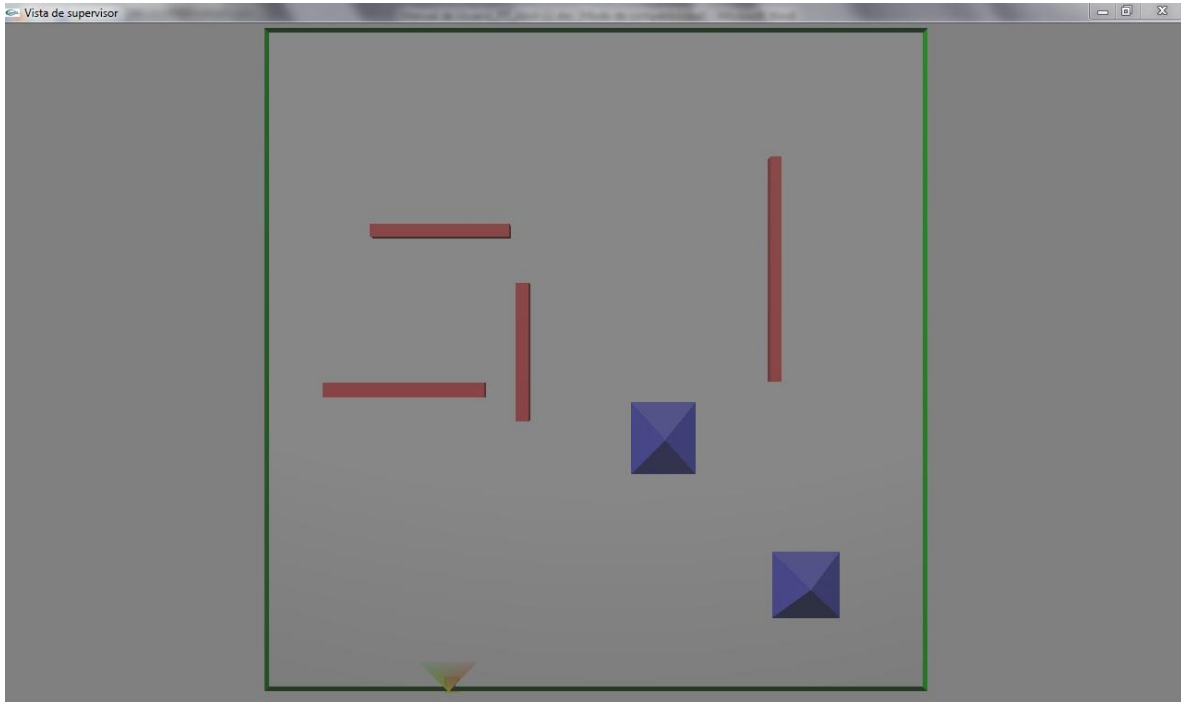


Figura 20. Obstáculos antes de rotar

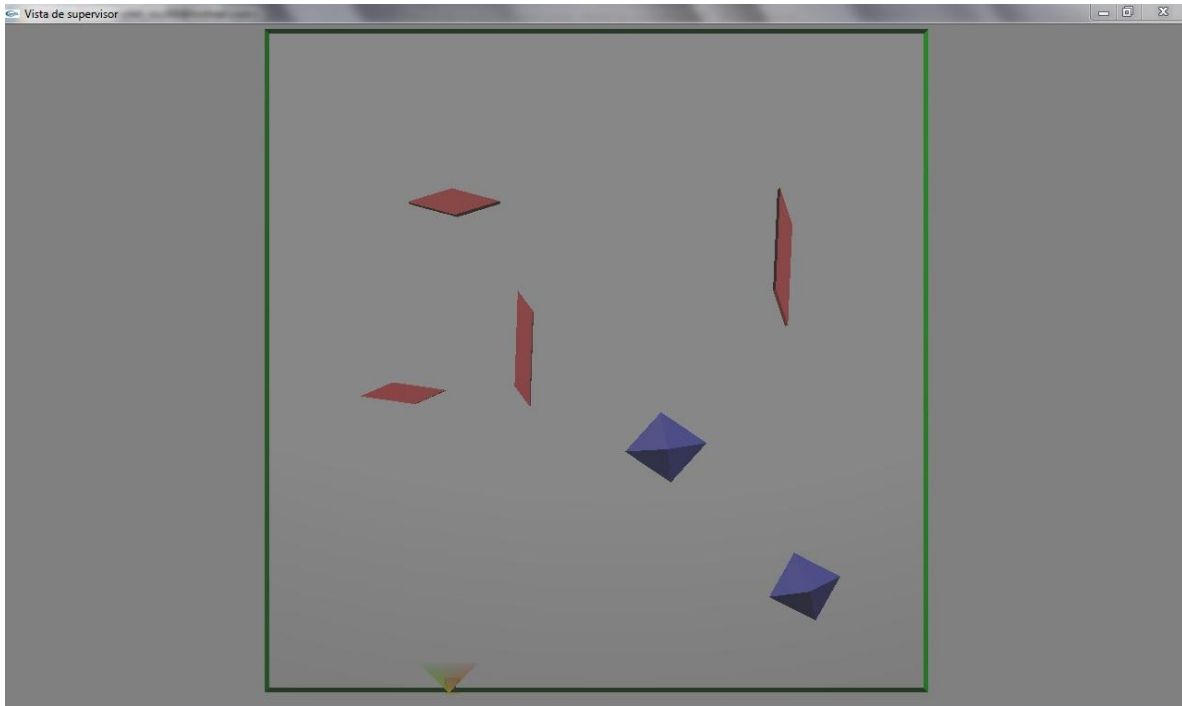


Figura 21. Obstáculos después de rotar

Rotación OFF

Esta opción es usada cuando se ha terminado de rotar los objetos seleccionados.

Nota: Se recomienda estar en el modo pantalla completa para poder rotar los obstáculos de manera adecuada

Modo 2D-Eliminar obstáculos

Para eliminar obstáculos se necesita ingresar al modo “eliminar obstáculos”, esto se logra usando la siguiente opción:

- Eliminación ON

Eliminación ON

Una vez seleccionada esta opción, debe colocar el cursor sobre el objeto que desee eliminar (Ver figura 22) y oprimir el botón izquierdo del mouse para que el obstáculo sea borrado del escenario (ver figura 23).

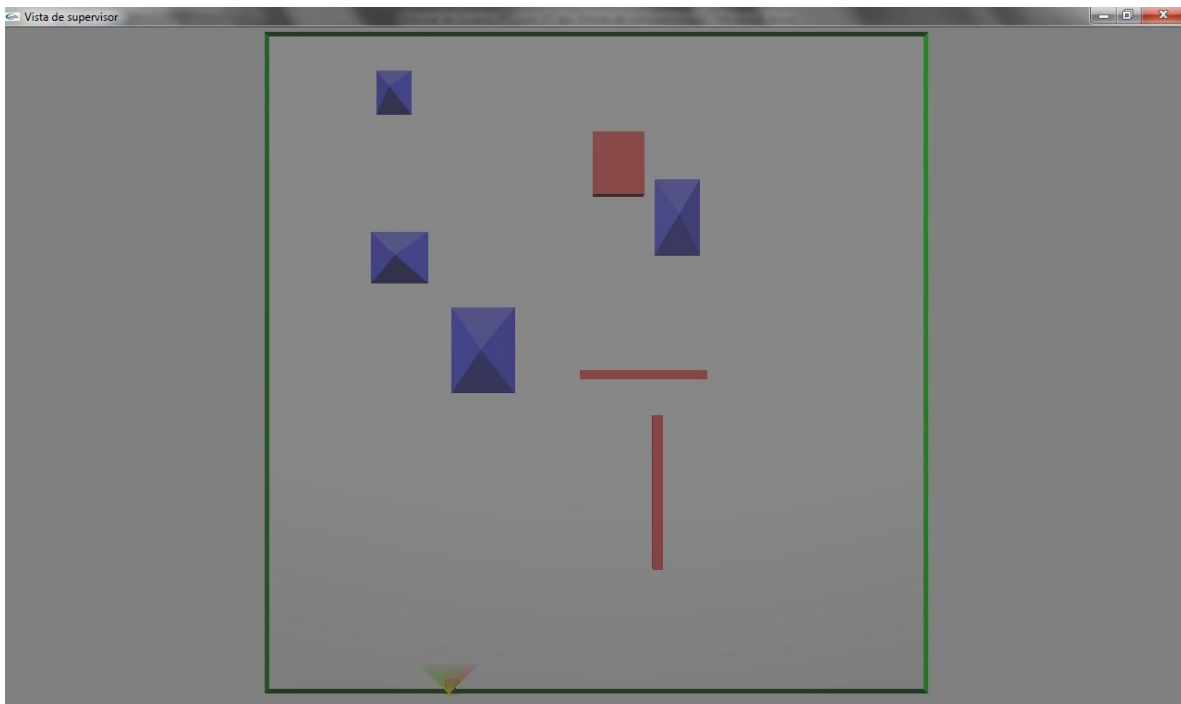


Figura 22. Antes de eliminar obstáculo

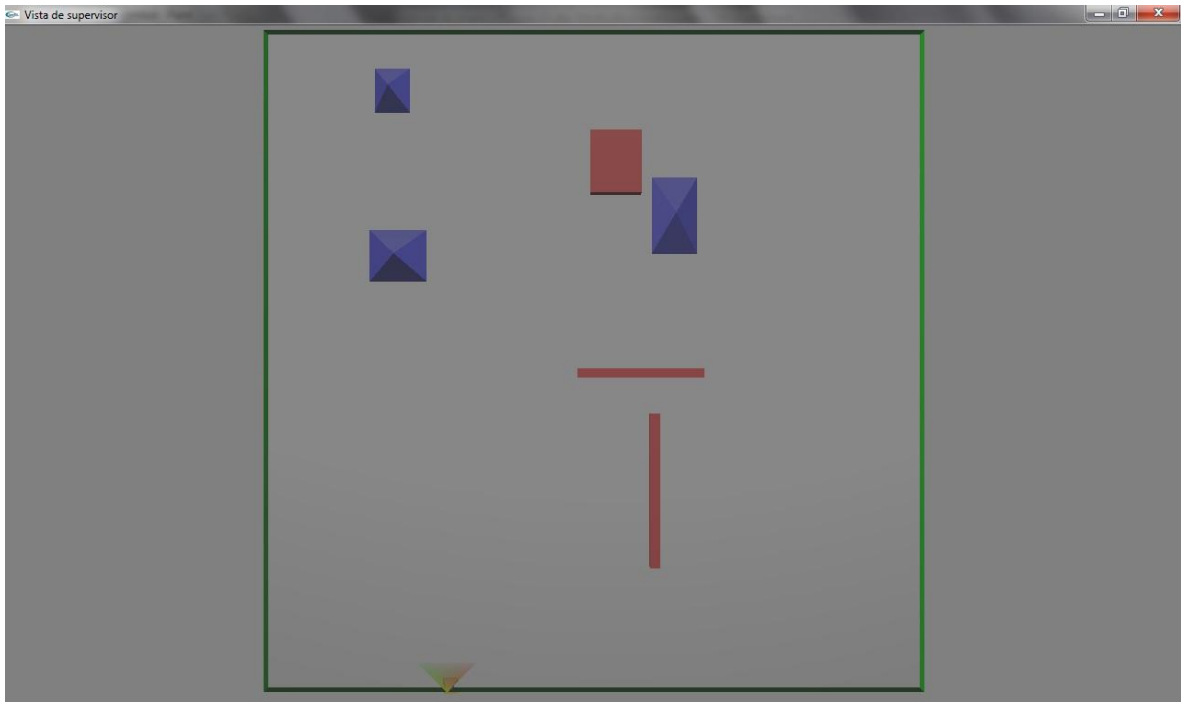


Figura 23. Después de borrar obstáculo

Eliminación OFF

Esta opción es usada cuando se ha terminado de eliminar los objetos seleccionados.

Nota: Se recomienda estar en el modo pantalla completa para poder eliminar los obstáculos de manera adecuada

Modo 3D-Opciones

Esta aplicación cuenta con modos alternativos que permiten visualizar el escenario de dos distintas formas y así poder apreciar los efectos de iluminación añadidos a la aplicación. Estos son:

- Modo alámbrico
- Modo sólido

Modo 3D-Alámbrico

Entrando en este modo se visualiza el escenario construido con polígonos sin color de relleno, esto es, solo se muestran las aristas de los polígonos que forman parte del escenario.

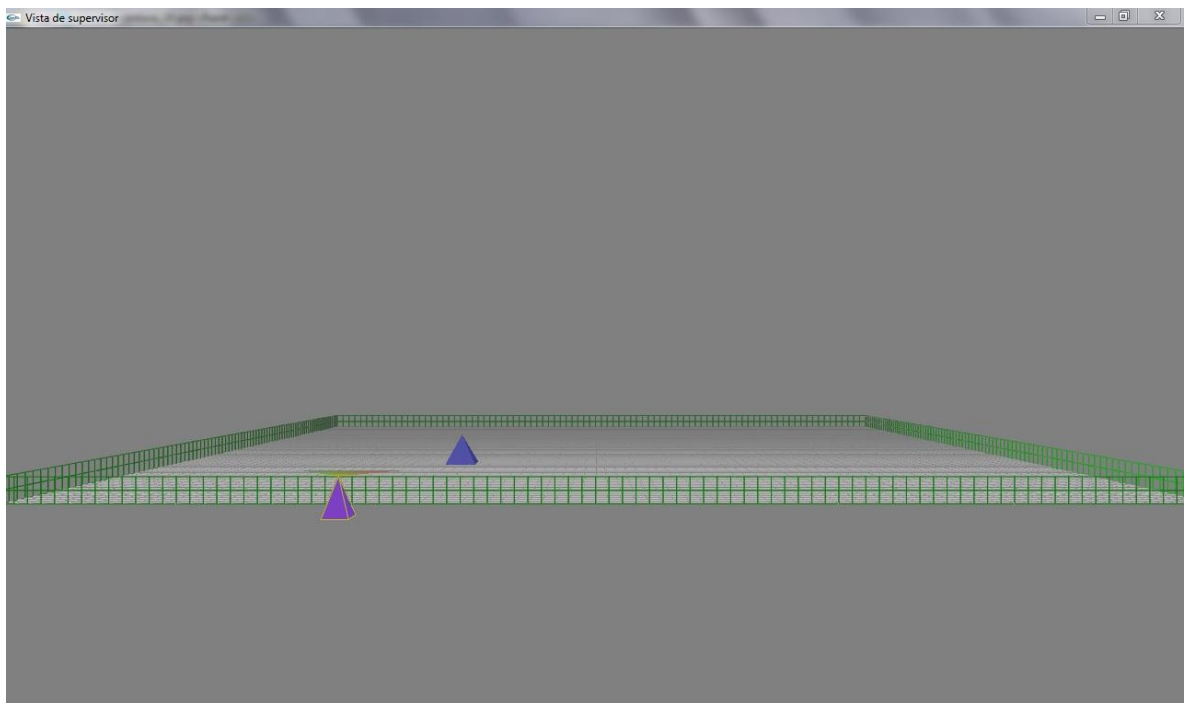


Figura 24. Escenario en modo alámbrico

Modo 3D-Solido

Este es el modo por defecto de la aplicación. En él se pueden ver los polígonos del escenario dibujados con color de relleno. Este modo permite apreciar mejor los efectos de iluminación y niebla que han sido añadidos a la aplicación.

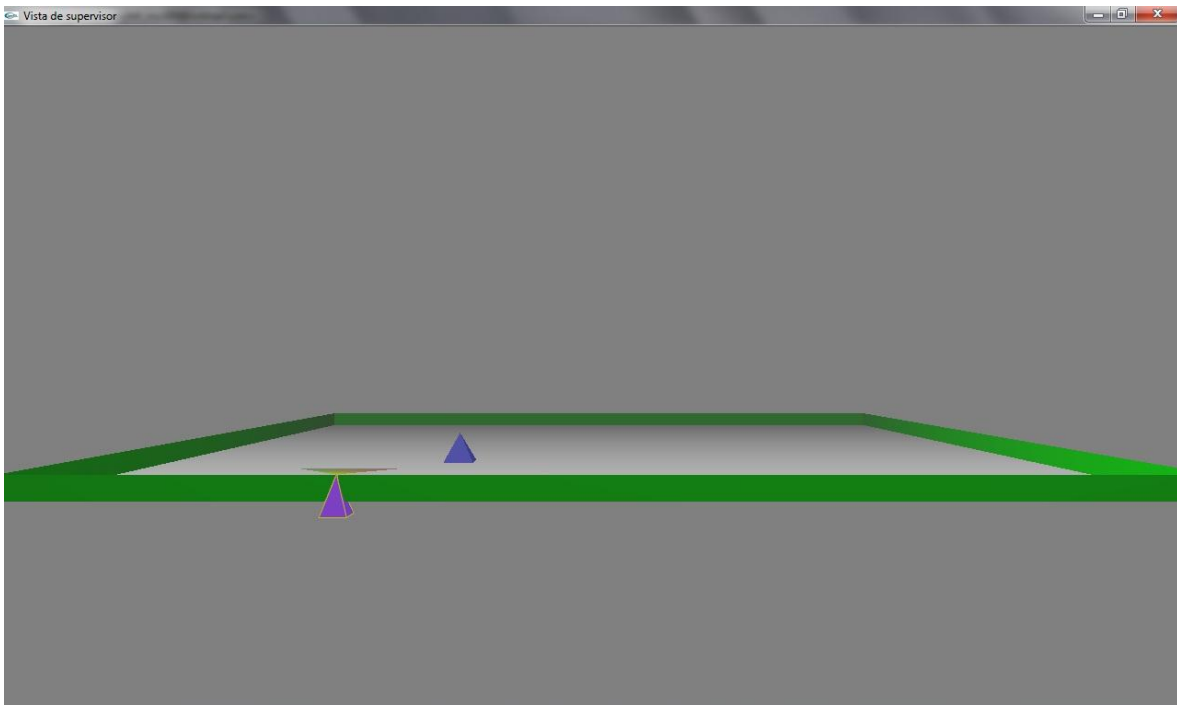


Figura 25. Escenario en modo solido

Nota: Se recomienda estar en este modo para poder visualizar mejor los cambios registrados en la gráfica de la visión del robot (Ventana gráfica del robot)

Índice de figuras

<i>Figura 1. Sistema de coordenadas espacial</i>	4
<i>Figura 2. Sistema de coordenadas cartesiano</i>	5
<i>Figura 3. Ventana de Supervisor modo 3D</i>	7
<i>Figura 4. Ventana de Supervisor modo 2D</i>	7
<i>Figura 5. Ventana de Robot</i>	8
<i>Figura 6. Ventana de gráfica de robot</i>	9
<i>Figura 7. Modo 3D</i>	10
<i>Figura 8. Modo 2D</i>	11
<i>Figura 9. Uso de tecla ←</i>	13
<i>Figura 10. Uso de tecla →</i>	13
<i>Figura 11. Uso de tecla ↑</i>	14
<i>Figura 12. Uso de tecla ↓</i>	15
<i>Figura 13. Menú modo 2D</i>	16
<i>Figura 14. Opción Agregar Cubo</i>	17
<i>Figura 15. Opción Agregar Pirámide</i>	18
<i>Figura 16. Antes de mover obstáculo</i>	19
<i>Figura 17. Después de mover obstáculo</i>	20
<i>Figura 18. Antes de Redimensionar obstáculos</i>	21
<i>Figura 19. Después de redimensionar obstáculos</i>	22
<i>Figura 20. Obstáculos antes de rotar</i>	23
<i>Figura 21. Obstáculos después de rotar</i>	24

<i>Figura 22. Antes de eliminar obstáculo.....</i>	<i>25</i>
<i>Figura 23. Después de borrar obstáculo.....</i>	<i>26</i>
<i>Figura 24. Escenario en modo alámbrico</i>	<i>27</i>
<i>Figura 25. Escenario en modo solido</i>	<i>28</i>