

## **Proyecto CoppeliaSimu**

### **Robot detector de obstáculos**

*Laurys Grimaldo, Jean Piere Adames, Brayan Moreno*

*Ingeniería Mecatrónica, tercer año, Facultad de Informática, Electrónica y  
Comunicación, Universidad de Panamá*

#### **Objetivos**

- Reconocer los diferentes comandos del software
- Aprender a manipular el software
- Analizar el funcionamiento del robot

#### **Introducción**

El uso del robot actualmente ya no se centra en una parte industrial y profesional de diversas áreas en una empresa, los robots ya están siendo utilizado como asistentes en diferentes áreas cotidianos. Realizaremos un robot inteligente el cual es capaz de detectar objetos o personas en su rango de movimiento utilizando sensores de proximidad.

Para este proyecto utilizaremos el entorno de simulación de CoppeliaSim, más específicamente la versión 4.0.0 educativa la cual es gratuita diseñada para ser utilizada en escuelas y universidades.

#### **Procedimiento**

Lo primero que debemos hacer es dar clic derecho al ratón el computador, seguidos debemos ir al apartador de "Add", luego "Primitive Shape" y por último "Cuboid". Estos pasos nos dirigen a la sección para agregar una figura geométrica como cuerpo.

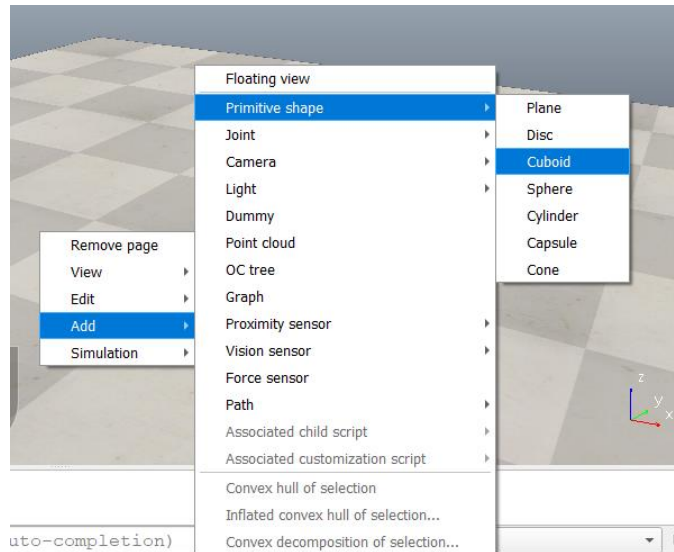


Figura 1. Selección de la base del robot.

Luego de dar clic en “Cuboid” nos debe aparecer la siguiente tabla de propiedades, esta tabla tiene como objetivo modificar las dimensiones y propiedades de nuestro cuerpo a colocar. Debemos dar clic en “OK” y nos aparecerá un rectángulo el cual representa nuestro cuerpo principal.

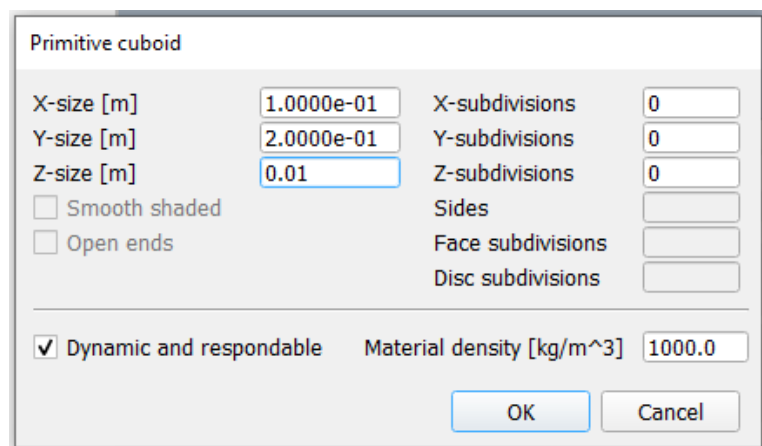


Figura 2. Configurar las dimensiones de la base del robot.

Ahora le cambiamos el nombre a nuestro objeto colocado, vamos a la esquina superior izquierda y damos doble clic sobre la pieza, en nuestro caso le colocamos el nombre “Base”.

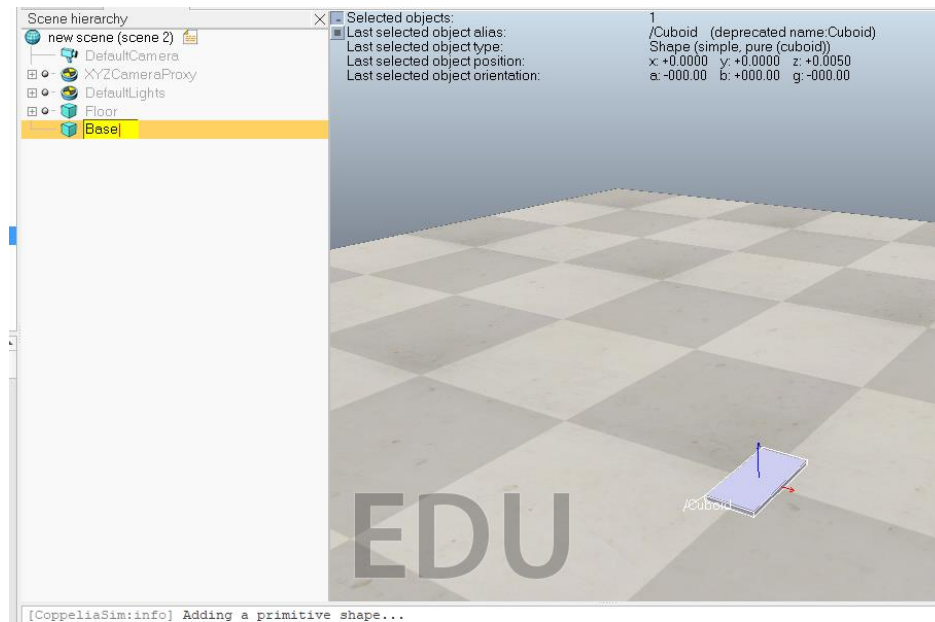


Figura 3. Modificamos el nombre de la pieza.

Seguido modificaremos las propiedades del objeto. Seleccionamos “Scene Object Properties” esta sección se encuentra a la esquina izquierda debajo del logo de una tuerca, luego que damos clic nos aparece la siguiente ventana la cual debemos cambiar a la opción de “common” y por último marcamos todas las propiedades especiales con el nombre de:

- Collidable: esta propiedad tiene como objetivo detectar si existe colisiones entre 2 cuerpos.
- Detectable: con esta propiedad el cuerpo seleccionado tiene la habilidad de detectar factores externos.
- Measurable: Esta propiedad especial nos indica que el objeto puede ser medido.

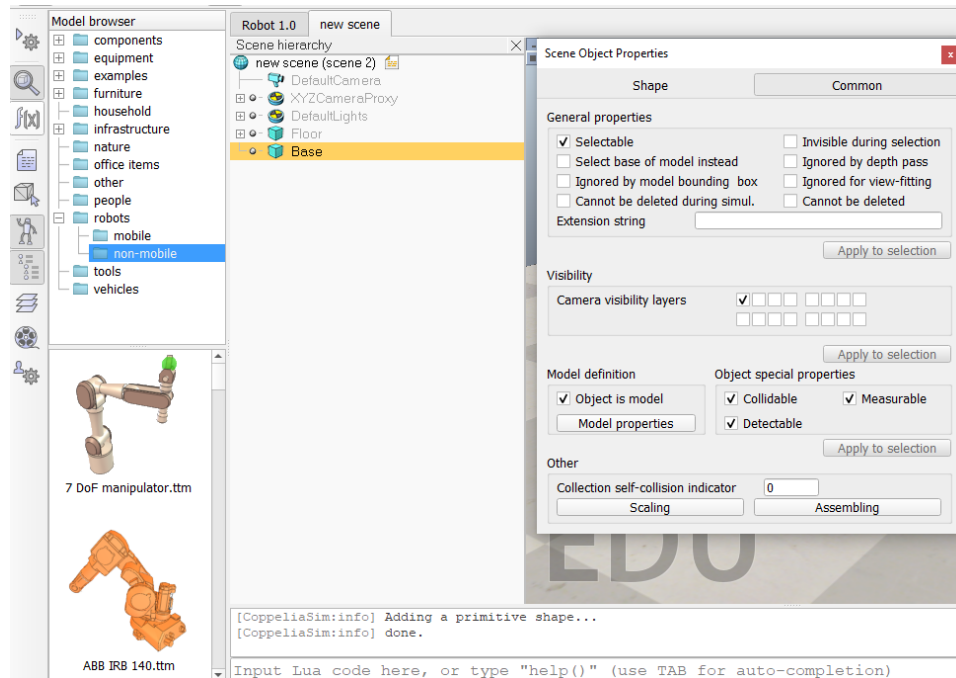


Figura 4. Representación de las propiedades del objeto a modificar.

Realizamos los mismos pasos marcados en el punto 1 pero en este caso utilizaremos un “Cylinder”.

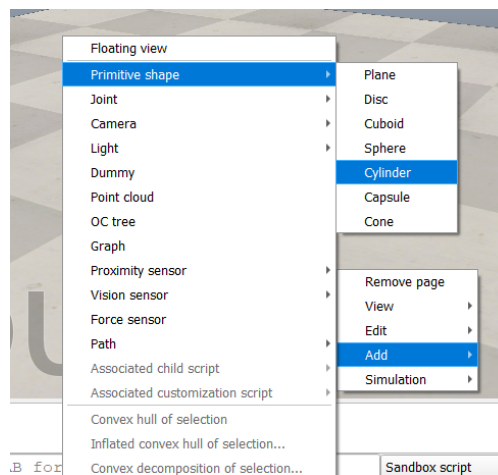


Figura 5. Selección de figura geométrica para las ruedas.

Cuando nos aparece esta ventana, debemos modificar los parámetros de X y Z. Realizamos los mismos pasos marcados en el apartado 3 solo que en este caso colocamos el nombre “roda\_Kanan” (rueda derecha) y también realizamos los mismos datos del párrafo 4.

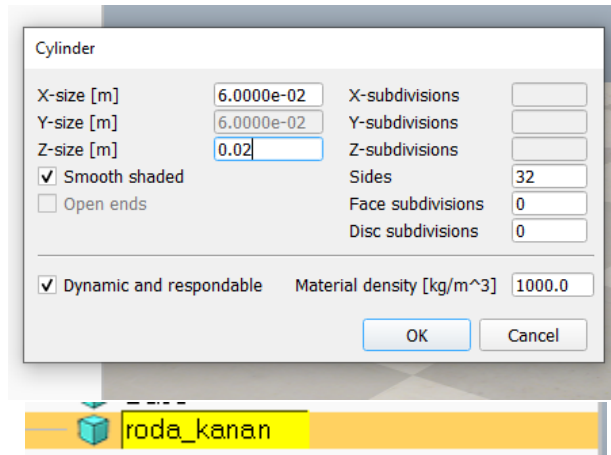


Figura 6. Modificar dimensiones de la pieza cylinder y cambiamos el nombre.

Ahora nos dirigimos al apartado que se encuentra en la parte de arriba, debemos seleccionar el objeto a modificar y luego nos aparecerá la siguiente imagen la cual debemos dar clic en “Position” y modificar los parámetros los cuales nos permitirá cambiar de posición la pieza.

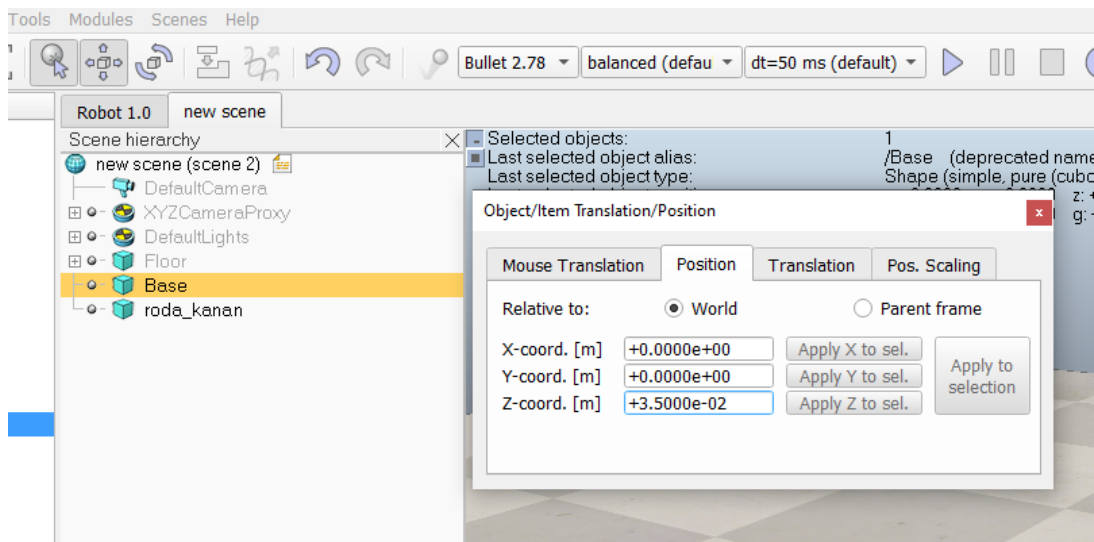


Figura 7. Modificar la posición de la pieza “Base” en el plano tridimensional.

Para este paso hacemos exactamente lo mismo que hicimos en el paso anterior, la diferencia esta en los valores utilizados para la propiedad de “Position” este cambio se le introduce al objeto con nombre” roda\_kanan” (rueda derecha).

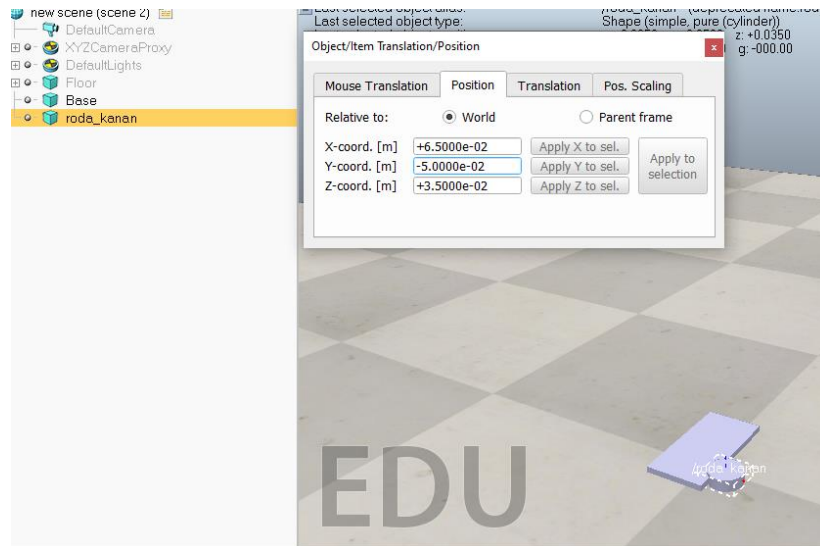


Figura 8. Modificar la posición de la pieza “roda\_kanan” en el plano tridimensional.

El siguiente paso consta de dar clic a la herramienta que esta al lado de “Mouse Rotation”, esta nos permite rotar una pieza. Al dar clic nos aparece una ventana la cual debemos escoger la opción “Orientation”.

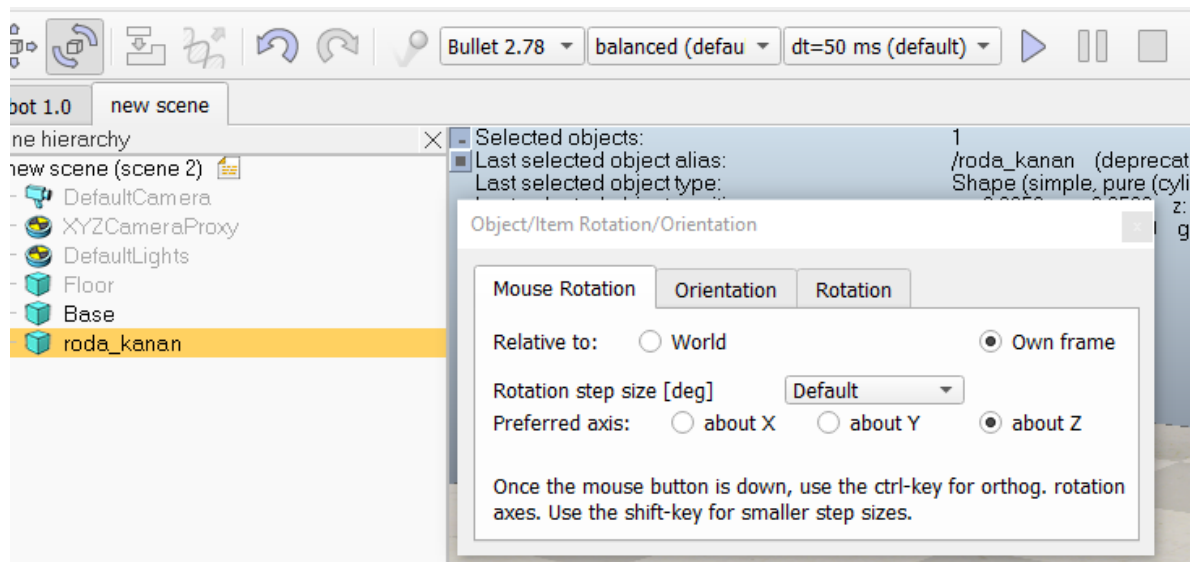


Figura 9. Ventana Object/Item.

Seleccionamos “Orientation”, modificamos la propiedad Beta y colocamos “90”. Esto hace que la pieza gire 90 °.

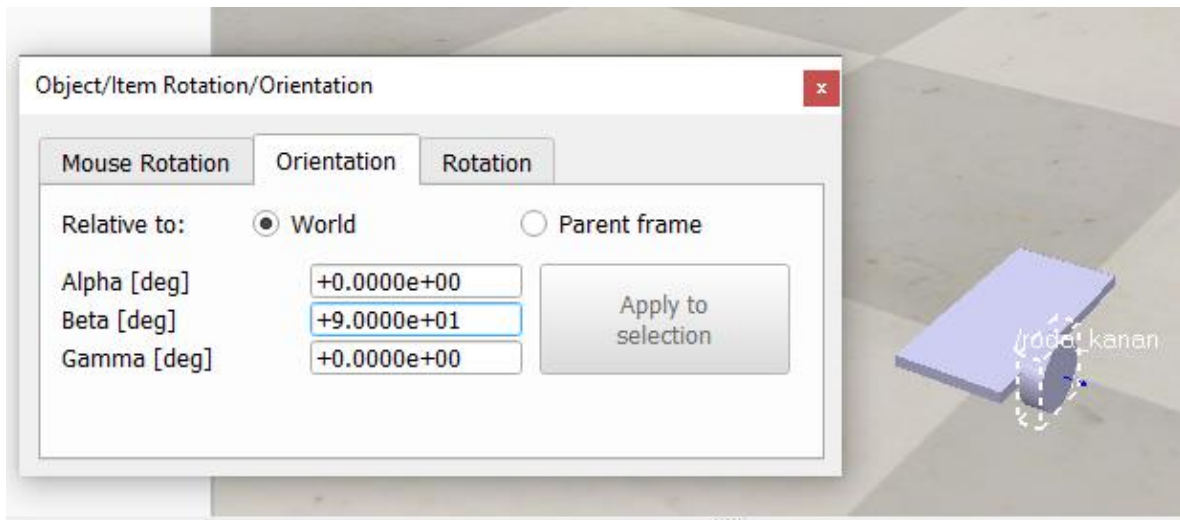


Figura 10. Ventana para modificar la orientación de la pieza que representa una de las ruedas.

Ahora vamos a dar clic derecho sobre la pieza “rode\_kanan”, nos dirigimos al apartado “edit” y por último “Copy selected objects”.

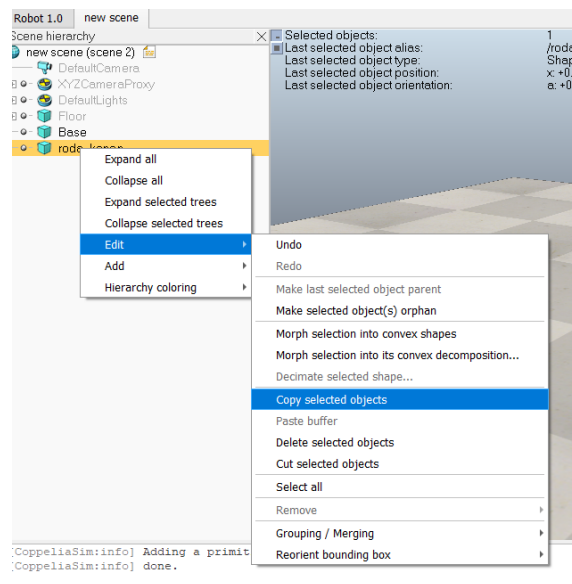


Figura 11. Representación del paso para copiar la pieza “rode\_kanan”.

Le colocamos el nombre “roda\_kiri” (rueda izquierda), le cambiamos la posición y quedara de la siguiente forma.

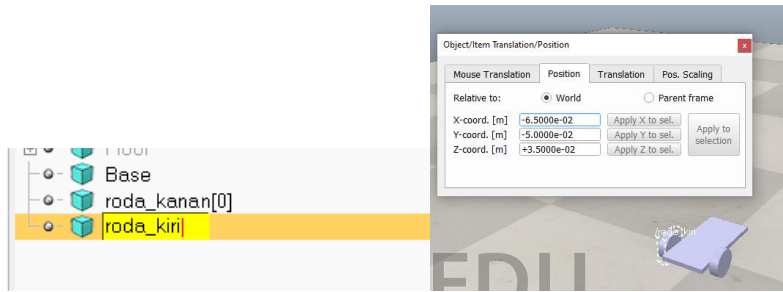


Figura 12 y 13. modificación del nombre y la posición de la pieza que copiamos anteriormente.

Volvemos a realizar los mismos pasos anteriores para agregar un cuerpo nuevo, en este caso escogemos el llamado "Sphere". Modificamos las dimensiones del objeto cilindrito y damos clic en ok.

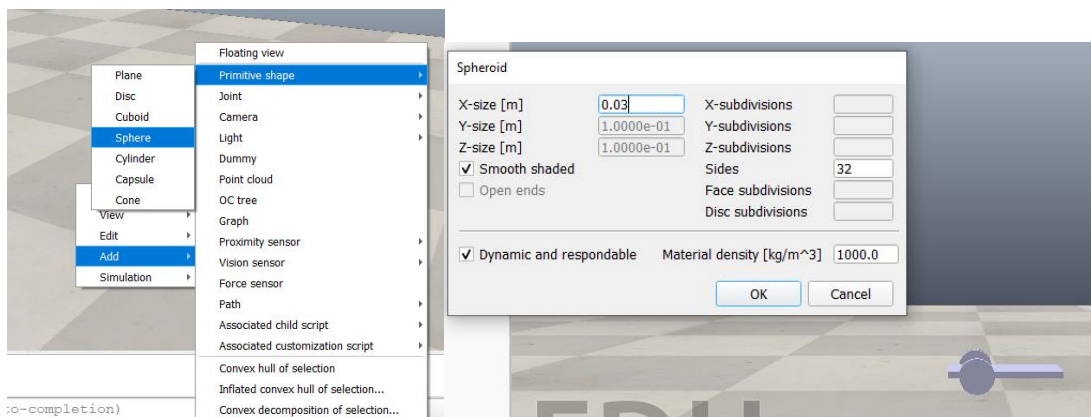


Figura 14 y 15. Agregamos cuerpo nuevo "Sosphere" representa la rueda delantera y modificamos las dimensiones.

Cambiamos el nombre a la pieza agregada "Ballcaster" y luego modificamos las propiedades de posición.

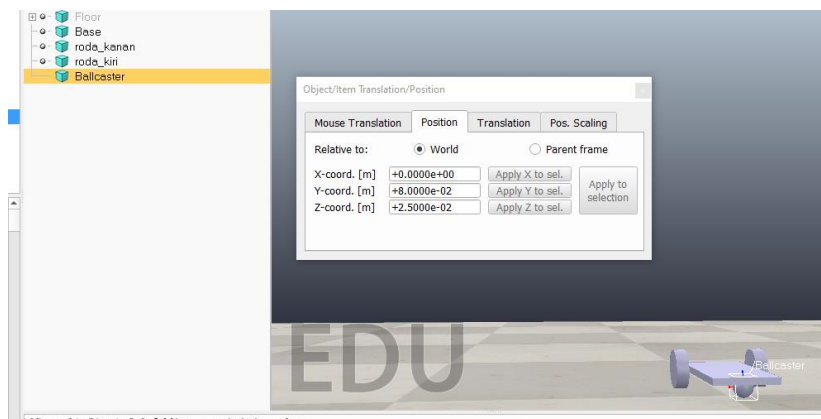


Figura 16. Cambiamos el nombre de la pieza agregada y modificamos la posición en el plano tridimensional.



Ahora vamos a utilizar la herramienta “joint” esto nos permite colocar un eje de tipo “Revolote” en el cuerpo seleccionado.

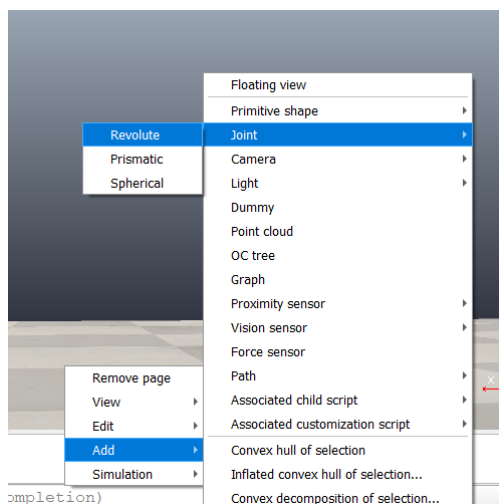


Figura 17. Seleccionar el motor “Joint”.

Cambiamos el nombre de la revoluta y damos clic en la herramienta de propiedades, cambiamos los valores de “Length” y “Diameter” para modifica la revoluta.

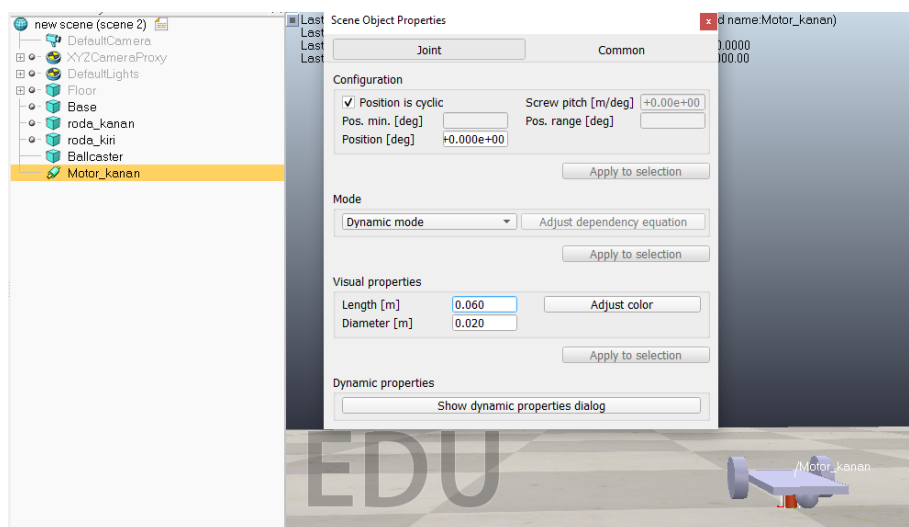


Figura 18. Modificar las dimensiones del eje revoluta agregado en el paso anterior.

Esta herramienta trata sobre darle clic a “show Dynamic Properties dialog” y nos aparece la siguiente tabla de propiedades dinámicas, debemos marcar las 2 primeras y le damos “Apply to selection”.

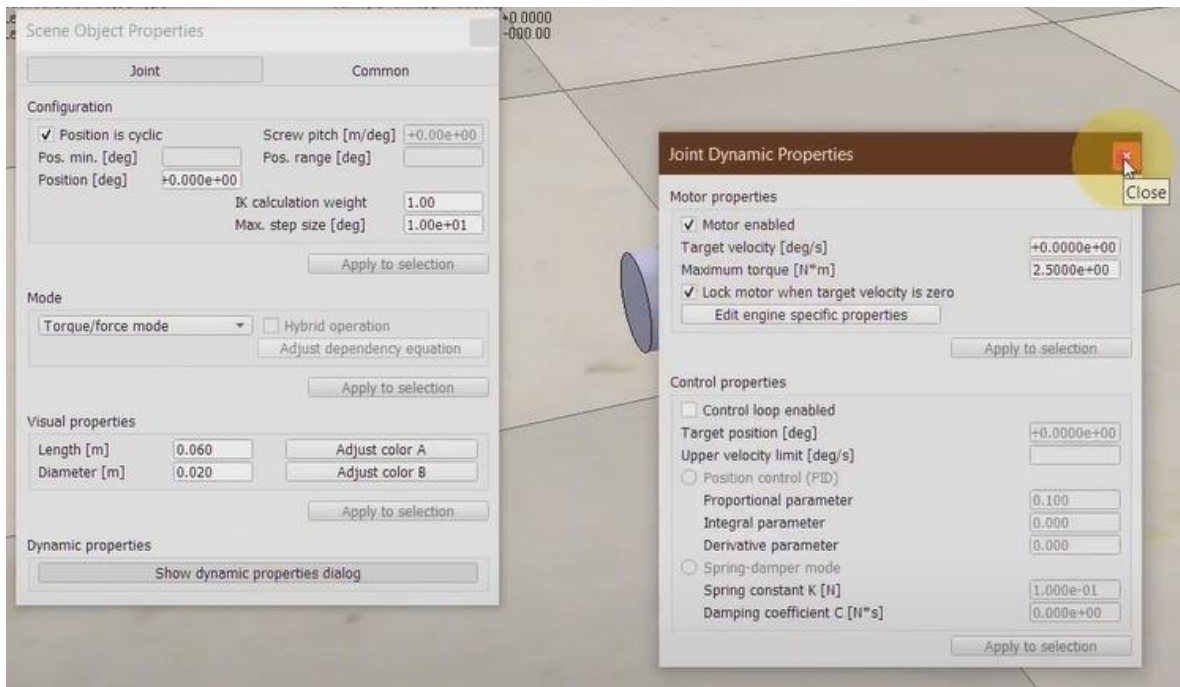


Figura 19. Modificar las propiedades dinámicas del eje revoluta agregado anteriormente.

Como siguiente punto nos aseguramos de tener seleccionado el “motor\_kanan” y luego clicamos “roda\_kanan”, teniendo eso seleccionado le damos a “object” luego “position” y “aply to selection”.

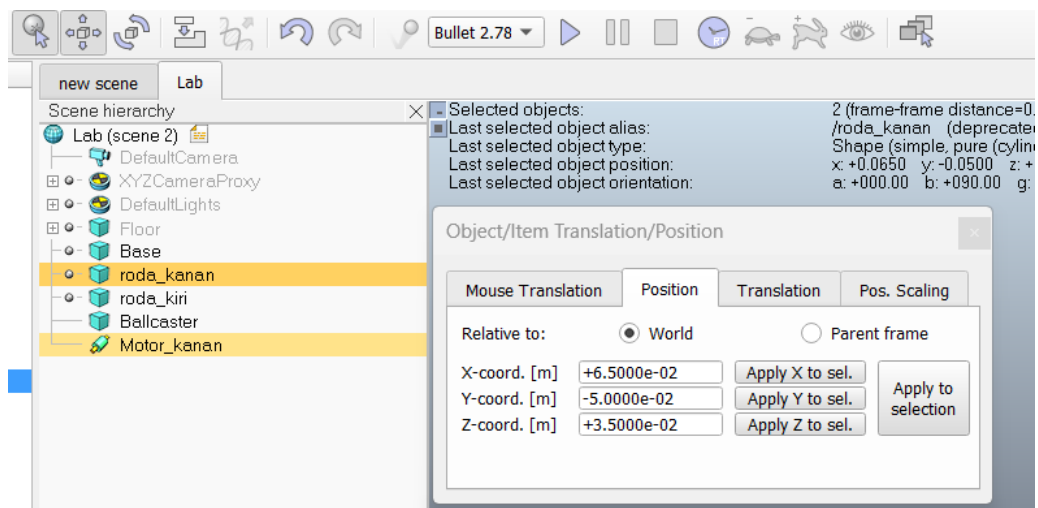


Figura 20. Enlazamos el motor agregado con la pieza “roda\_kanan.

Ya teniendo lo anterior le damos click al motor derecho luego rotación del objeto y cambiamos los grados de rotación del motor derecho que en este caso es 90°.

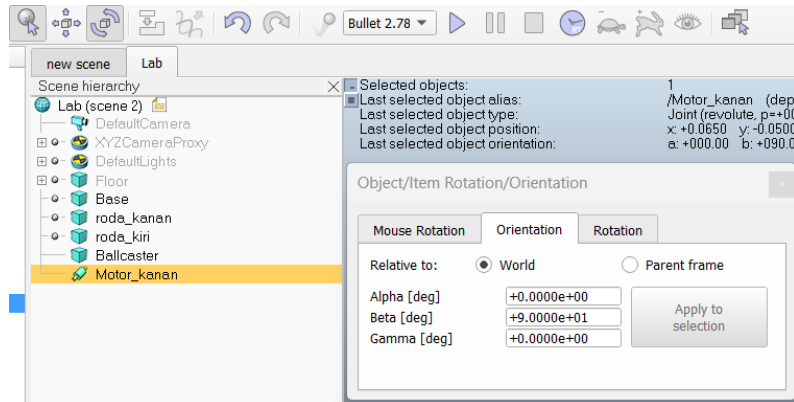


Figura 21. modificación del motor en 90 °.

Ahora para el motor izquierdo lo que hacemos es que le damos click derecho al "moto\_kanan", " editamos, copiamos el objeto seleccionado y lo pegamos en el panel, de esta manera:

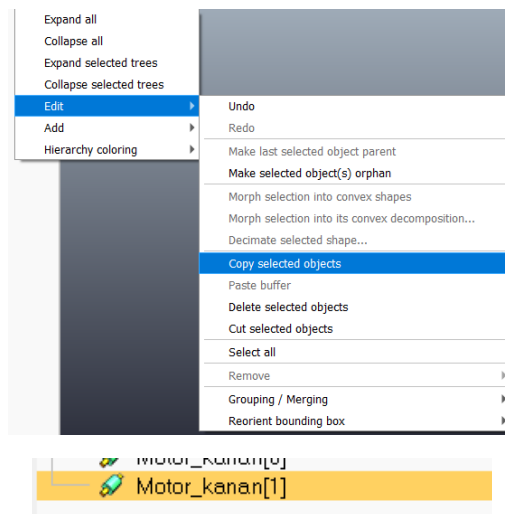


Figura 22. Copiar la pieza "Motor\_Kanan".

modificamos la posición del motor izquierdo de la misma forma que se hizo con la rueda derecha "roda\_kanan" y de esta manera se añadió el motor y la rueda izquierda

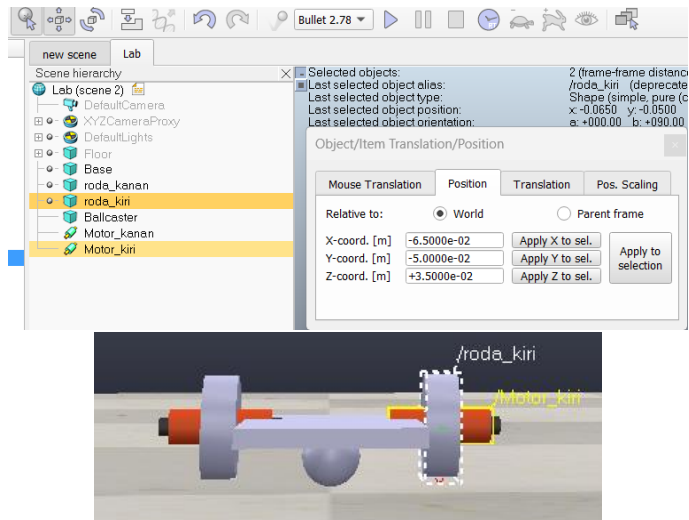


Figura 23 y 24. Modificar posición del nuevo motor agregado "Motor\_kiri" y colocamos en la rueda izquierda.

En esta parte agregamos un sensor de fuerza, seguimos manteniendo presionado el motor y la rueda izquierda, le damos click a la pantalla del software y nos aparece lo siguiente:

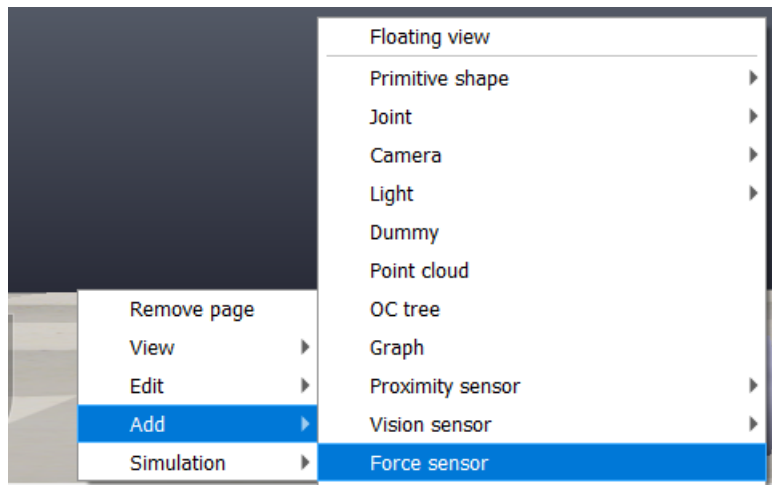


Figura 25. Instrucciones para agregar un sensor de fuerza a nuestros motores.

Damos clic en sensor de fuerza y una vez que tenemos el sensor, le cambiamos el nombre a "casterSensor" sensor de giro y nos vamos a propiedades del objeto. Le cambiamos el tamaño al objeto a 0.03. Y nos queda de la siguiente manera.

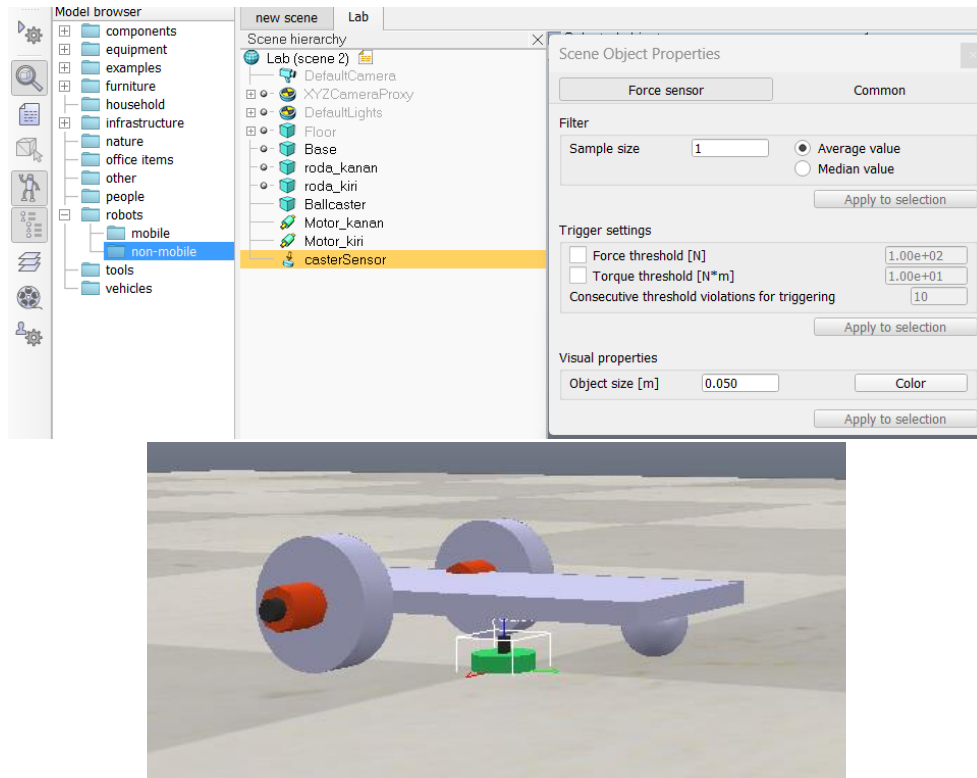


Figura 26 y 27. Colocar el sensor y modificamos las propiedades y el nombre del objeto.

Como siguiente punto mantenemos presionados el sensor de giro/fuerza, teniendo eso nos vamos a posición del objeto y "apply to selection" Quedaría así y modificamos la posición de este objeto en el plano tridimensional.

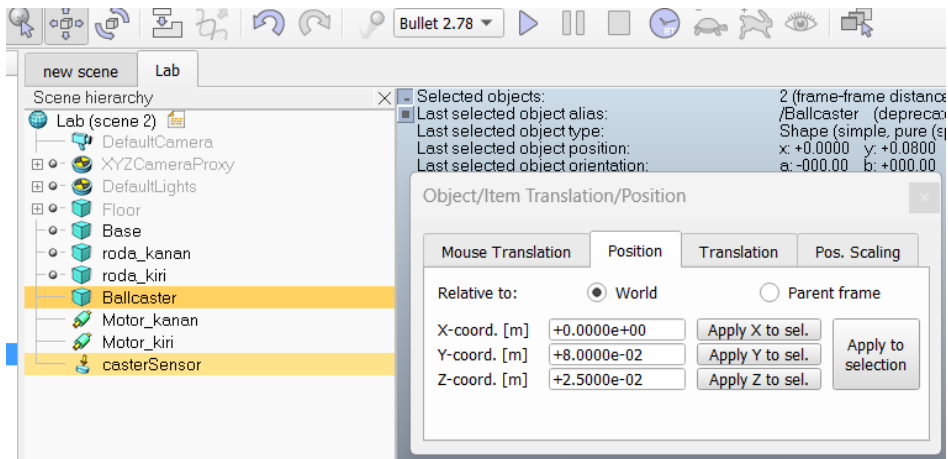


Figura 28. Modificar las posiciones del sensor agregado.

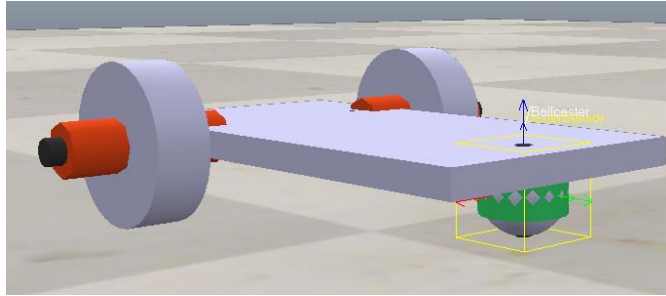


Figura 29. Representación del sensor agregado

Le agregamos un sensor de proximidad, para esto seguimos manteniendo presionados el sensor de giro y el lanzador de pelota. Le damos click derecho al espacio de trabajo en sensor de proximidad en forma de cono.

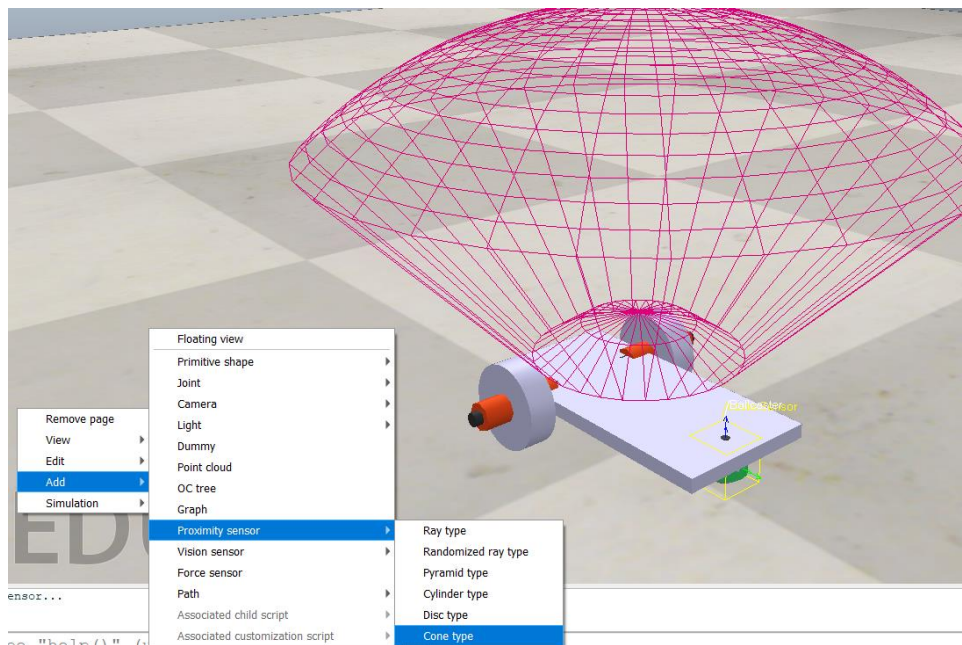


Figura 30. Representación del área de acción de un sensor de proximidad.

Le cambiamos el nombre al sensor de proximidad por "front\_prox", vamos sus propiedades y hacemos los cambios. Se logra de la siguiente manera: en parámetros de volumen.

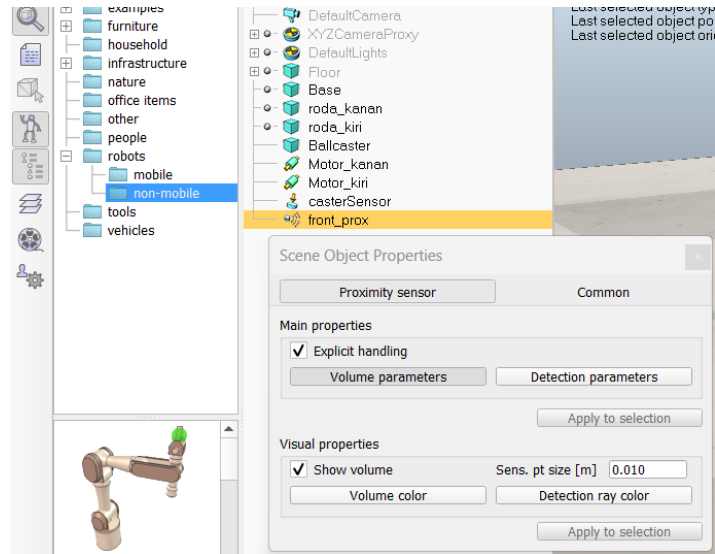


Figura 31. Modificamos el nombre del sensor de proximidad y algunas propiedades

Nos sale la siguiente pestaña, le cambiamos las características del rango y el ángulo del sensor y queda de la siguiente manera:

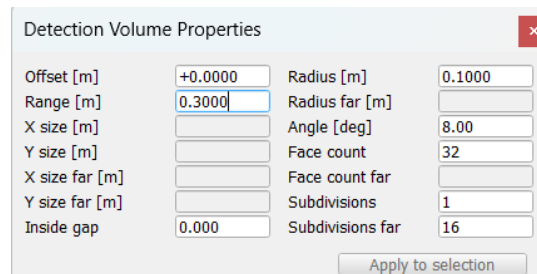


Figura 32. Modificar características del sensor.



Figura 33. Rango del sensor de proximidad modificado.

Nos vamos a la orientación del sensor de proximidad y configuramos su orientación y posición, lo podemos hacer en “mouse orientation” en las dos. Quedaría de la siguiente manera:

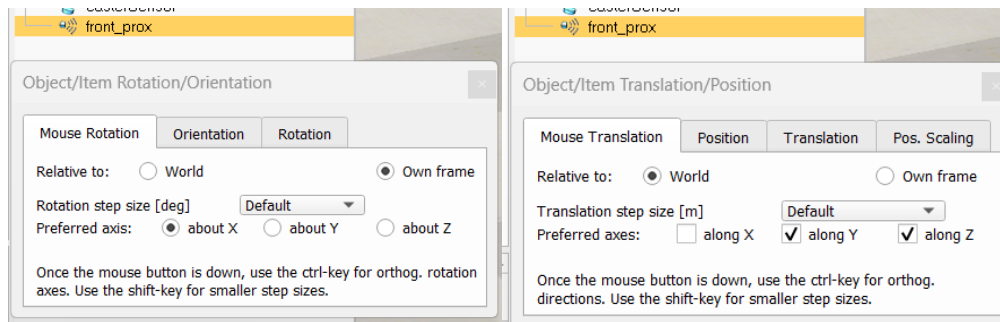


Figura 34 y 35. modificación de la posición y orientación del sensor de proximidad.

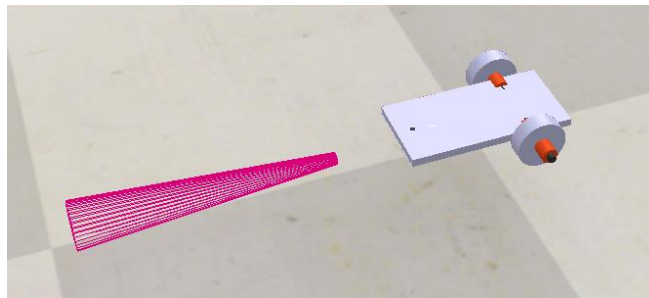


Figura 36. Representación del sensor de proximidad a utilizar.

Para el siguiente paso le agregaremos un script al robot de la siguiente manera y esto nos permitirá programar.

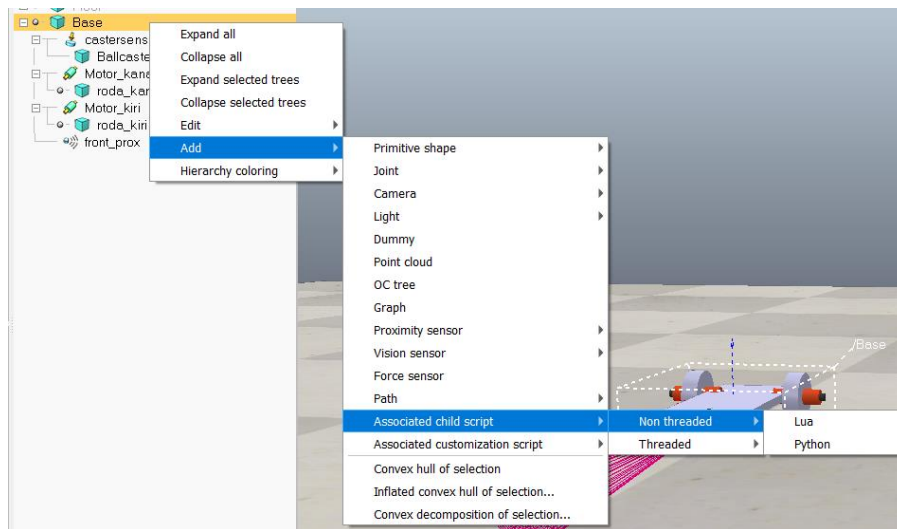


Figura 37. Agregamos la opción que nos permite introducir códigos de programación.



Para programar esto abrimos el script que esta al lado del nombre del “Base” y empezamos a dar las instrucciones de los comandos.

```

1 function sysCall_init()
2     -- do some initialization here
3     RobotBase=sim.getObjectHandle(sim.handle_self)
4     leftMotor=sim.getObjectHandle("motor_kiri")
5     rightMotor=sim.getObjectHandle("motor_kanan")
6     noseSensor=sim.getObjectHandle("front_prox")
7     minMaxSpeed={50*math.pi/180,300*math.pi/180}
8     backUntilTime=-1
9
10    xml = '<ui title=""..sim.getObjectHandle(RobotBase)..\'speed\' closeable="false" resizable="false" activate="false">\'..
11    <slider minimum="3" maximum="100" on-change="speedChange_callback" id="1"/>
12    <label text="" style="" {margin-left: 300px;}/>
13    </ui>
14
15    ui=simUI.create(xml)
16    speed=(minMaxSpeed[1]+minMaxSpeed[2])*0.3
17    simUI.setSliderValue(ui,1,100*(speed-minMaxSpeed[1])/(minMaxSpeed[2]-minMaxSpeed[1]))
18 end
19
20 function speedChange_callback(ui,id,newVal)
21     speed=minMaxSpeed[1]+(minMaxSpeed[2]-minMaxSpeed[1])*newVal/100
22 end
23
24 function sysCall_actuation()
25     -- put your actuation code here
26     result=sim.readProximitySensor(noseSensor)
27     if(result>0) then backUntilTime=sim.getSimulationTime()+4 end
28
29     if(backUntilTime<sim.getSimulationTime()) then
30         sim.setJointTargetVelocity(leftMotor,-speed)
31         sim.setJointTargetVelocity(rightMotor,-speed)
32     else
33         sim.setJointTargetVelocity(leftMotor,speed/5)
34         sim.setJointTargetVelocity(rightMotor,speed/7)
35     end
36 end
37
38 function sysCall_sensing()
39     -- put your sensing code here
40 end
41
42 function sysCall_cleanup()
43     -- do some clean-up here
44     simUI.destroy(ui)
45 end
46
47 -- See the user manual or the available code snippets for additional callback functions and details

```

Figura 38. programación para introducir y que el robot funcione correctamente.

Como primera prueba le colocamos al robot unos vasos para ver si este los detecta y no choque con ninguno de ellos. Estos vasos estarán en filas de la siguiente manera

Cabe destacar que podemos cambiar las características del sensor en sus propiedades y podemos aumentar o disminuir su ángulo.

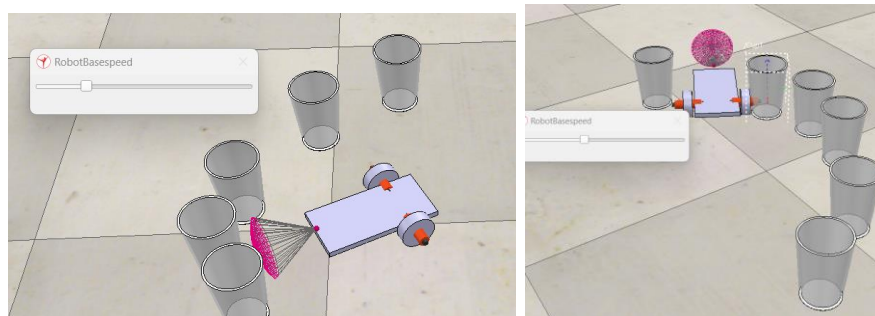


Figura 39 y 40. Primera prueba del robot en funcionamiento.

Para concluir esta experiencia, realizamos una ultima prueba, en donde colocamos unas paredes para comprobar de que el robot siga el camino establecido y no choque con ningun obstaculo.

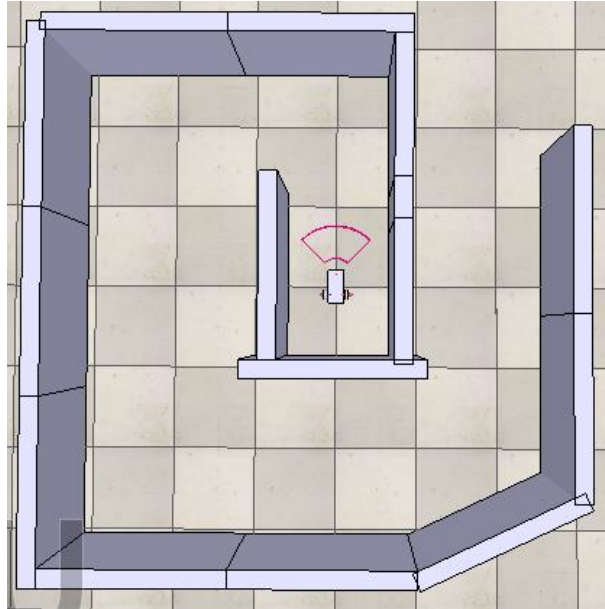


Figura 41. Representacion final del robot en funcionamiento.

## Conclusion

El robot no pasará por los objetos si el sensor detecta que está en frente si no hay nada en frente este se moverá por ese lado. También podemos aumentar o disminuir la velocidad del robot.

De esta manera pudimos completar la experiencia, podemos reconocer que se pueden hacer proyectos sencillos hasta más complejos que ya este tiempo una variedad de herramientas. Solo nos queda seguir aprendiendo a usar el programa.

Este tipo de robot tiene aplicaciones en su uso cotidiano en el área de la domótica, con uso como asistente de limpieza ya que este robot inteligente se mueve por el área de limpieza y su sensor de proximidad evita chocar con personas y objetos.

Otra aplicación está en el área de la industria automatizada con robot automatizado los cuales se encargan de mover un gran peso de un lugar a cierto lugar evitando cualquier objeto en medio del trayecto a realizar.

## Bibliografía

Todos los derechos reservados. "CoppeliaSim - Coppelia Robotics". Robot simulator CoppeliaSim: create, compose, simulate, any robot - Coppelia Robotics. <https://coppeliarobotics.com/coppeliaSim> (accedido el 2 de diciembre de 2022).

Ha ves. CoppeliaSim Tutorial #1 || Obstacle Avoidance Robot || Robot Penghindar Halangan. (30 de marzo de 2020). Accedido el 2 de diciembre de 2022. [Video en línea]. Disponible: <https://www.youtube.com/watch?v=oKOx25SmX58>