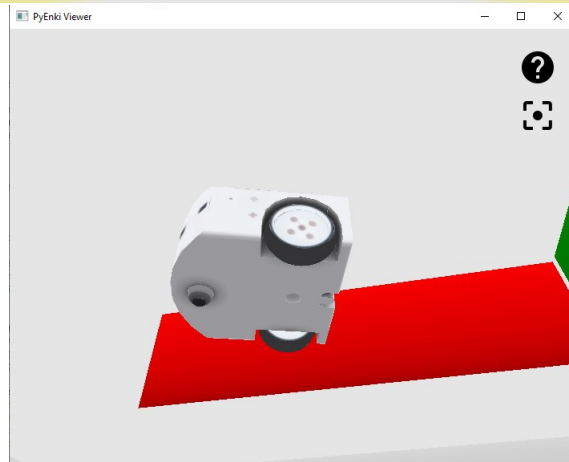
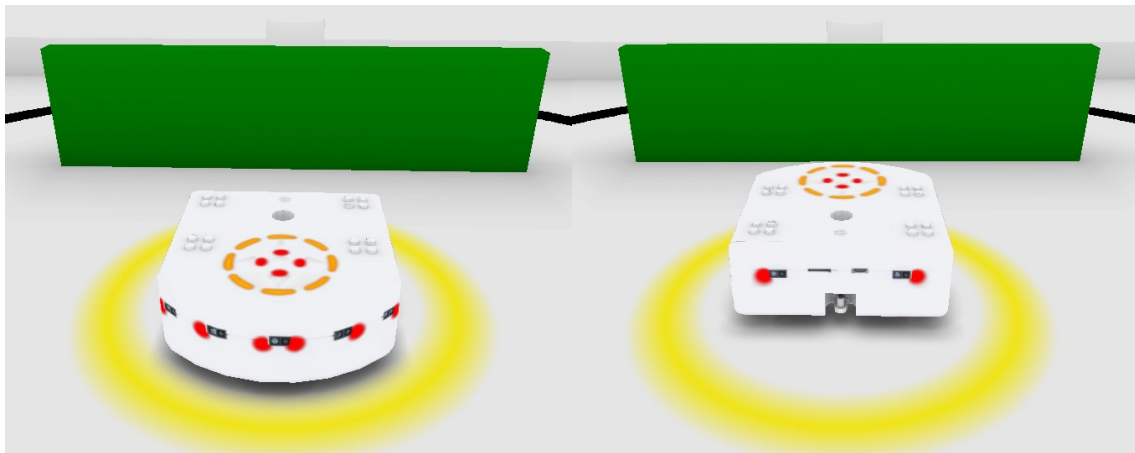


## Robot Thymio2

<b>Sensores de Proximidad: 7</b> 0: frontal izquierdo izquierdo 1: frontal izquierdo 2: frontal frontal 3: frontal derecho 4: frontal derecho derecho 5: trasero izquierdo 6: trasero derecho	<b>Sensores de línea: 2</b> 0: izquierdo 1: derecho
	<b>Leds: 23</b>

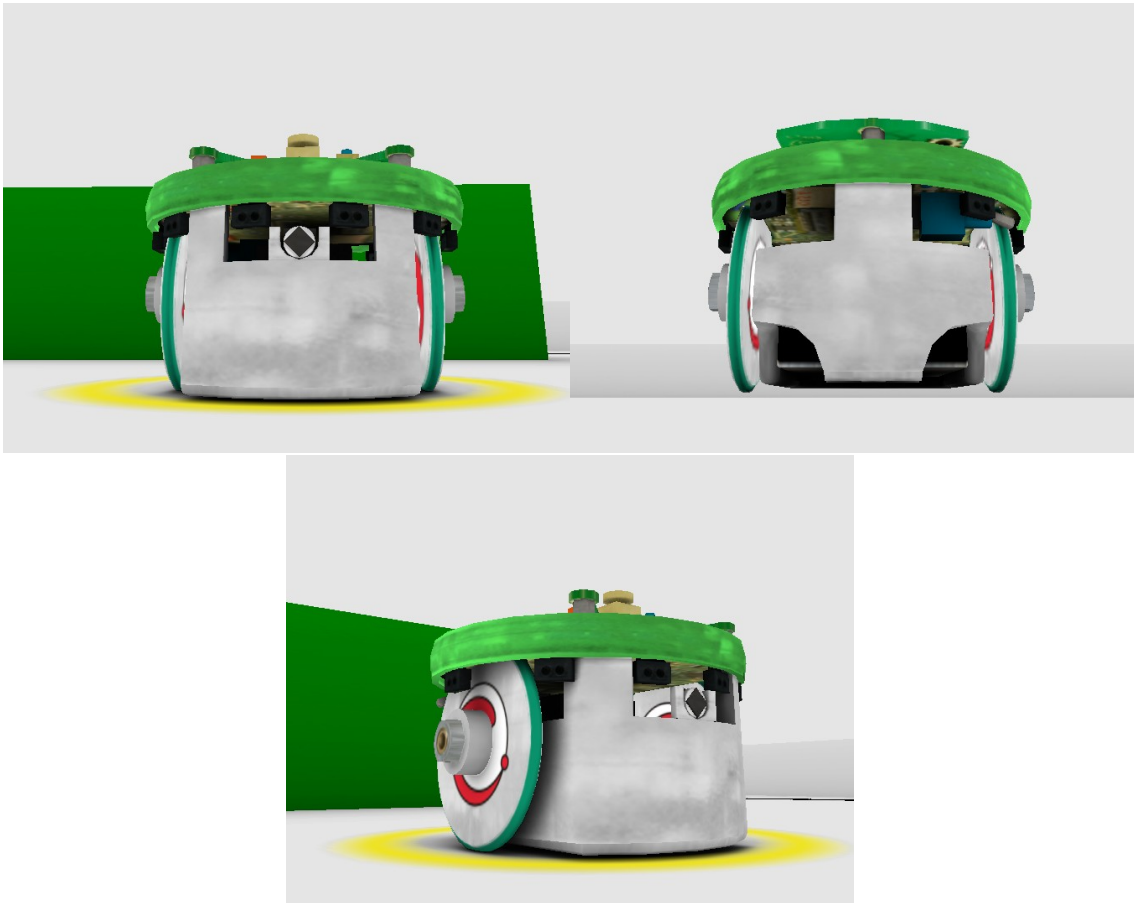


<b>Constructor Java</b>	
RobotThymio2	<b>r = new RobotThymio2(</b> String nombre, String host, int puerta) Conecta a un robot específico del simulador
<b>Métodos Java</b>	
String	<b>r.getName()</b> El nombre de robot
double[2]	<b>r.getPos()</b> Su posición actual (en cms) – x, y
double[2]	<b>r.getSpeed</b> Su velocidad actual de los motores – izquierda, derecha
void	<b>r.getSensors()</b> Actualiza el valor de todos sus sensores
double [2]	<b>r.getGroundSensorValues()</b> El valor de sus sensores de línea (0 negro a 1000 blanco)
double [7]	<b>r.getProximitySensorDistances()</b> La distancia (en cms) de sus sensores de proximidad
double [7]	<b>r.getProximitySensorValues()</b> El valor de sus sensores de proximidad
void	<b>r.setSpeed(</b> int izquierdo, int derecho) Establece velocidad de los motores izquierdo y derecho
void	<b>r.setLedsIntensity(</b> double leds[23] ) Cambia la intensidad de los 23 leds (0.0 a 1.0) según valores del arreglo recibido. El índice del arreglo identifica al led

## Robot EPuck

**Sensores de Proximidad: 8**  
0: frontal frontal derecho  
1: frontal derecho  
2: derecho  
3: trasero derecho  
4: trasero izquierdo  
5: izquierdo  
6: frontal izquierdo  
7: frontal frontal izquierdo

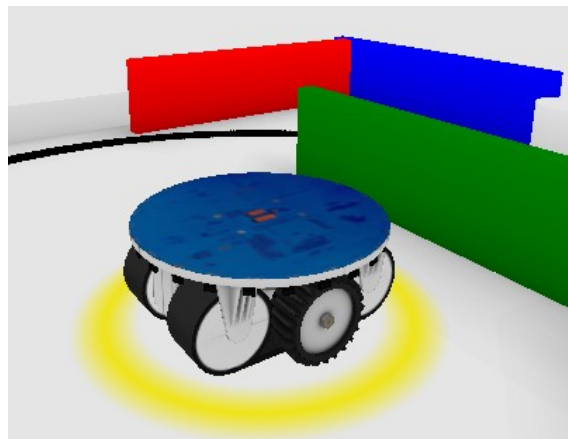
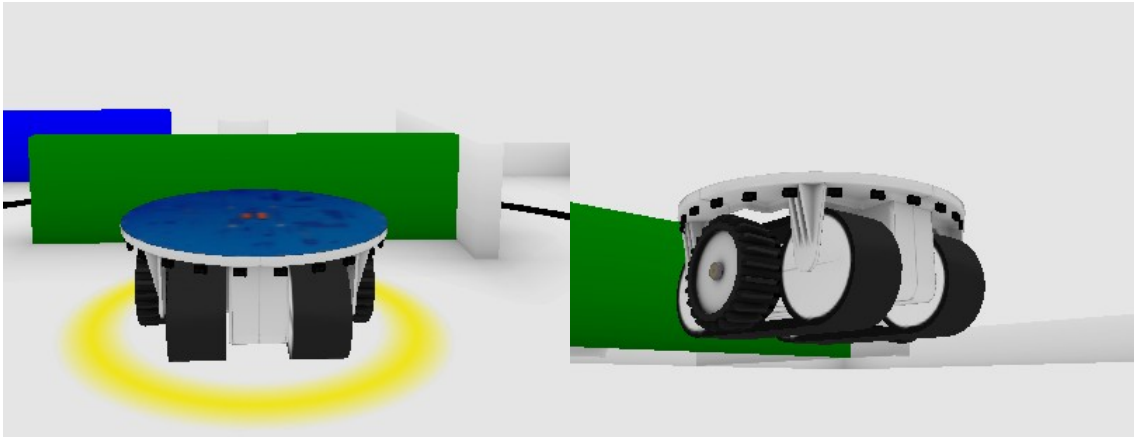
**Leds: 1 anillo circular**



<b>Constructor Java</b>	
RobotEPuck	<b>r = new RobotEPuck</b> ( String nombre, String host, int puerta) Conecta a un robot específico del simulador
<b>Métodos Java</b>	
String	<b>r.getName()</b> El nombre de robot
double[2]	<b>r.getPos()</b> Su posición actual (en cms) – x, y
double[2]	<b>r.getSpeed</b> Su velocidad actual de los motores – izquierda, derecha
void	<b>r.getSensors()</b> Actualiza el valor de todos sus sensores
double [8]	<b>r.getProximitySensorDistances()</b> La distancia (en cms) de sus sensores de proximidad
double [8]	<b>r.getProximitySensorValues()</b> El valor de sus sensores de proximidad
List<int[4]>	<b>r.getCameraImage()</b> Retorna la imagen lineal de la camera. La imagen es una línea de 60 píxeles y cada elemento de la lista es un arreglo de 4 elementos con los colores RGB en los 3 primeros y el cuarto es la transparencia
void	<b>r.setSpeed</b> ( int izquierdo, int derecho) Establece velocidad de los motores izquierdo y derecho
void	<b>r.setLedRing( boolean on)</b> Apaga (true) o enciende (false) el anillo que lo rodea

## Robot Marxbot

<b>Sensores de Presión:</b> 24 alrededor del cuerpo	
---	--



<b>Constructor Java</b>	
RobotMarxbot	<b>r = new RobotMarxbot</b> ( String nombre, String host, int puerta) Conecta a un robot específico del simulador
<b>Métodos Java</b>	
String	<b>r.getName()</b> El nombre de robot
double[2]	<b>r.getPos()</b> Su posición actual (en cms) – x, y
double[2]	<b>r.getSpeed</b> Su velocidad actual de los motores – izquierda, derecha
void	<b>r.getSensors()</b> Actualiza el valor de todos sus sensores
double [24]	<b>r. getVirtualBumpers ()</b> El valor de los sensores de presión que lo rodean
void	<b>r.setSpeed</b> ( int izquierdo, int derecho) Establece velocidad de los motores izquierdo y derecho