Active Class #4...

Maestría en Inteligencia Artificial Aplicada

Sensores para Vehículos Autónomos

Dr. David Antonio-Torres

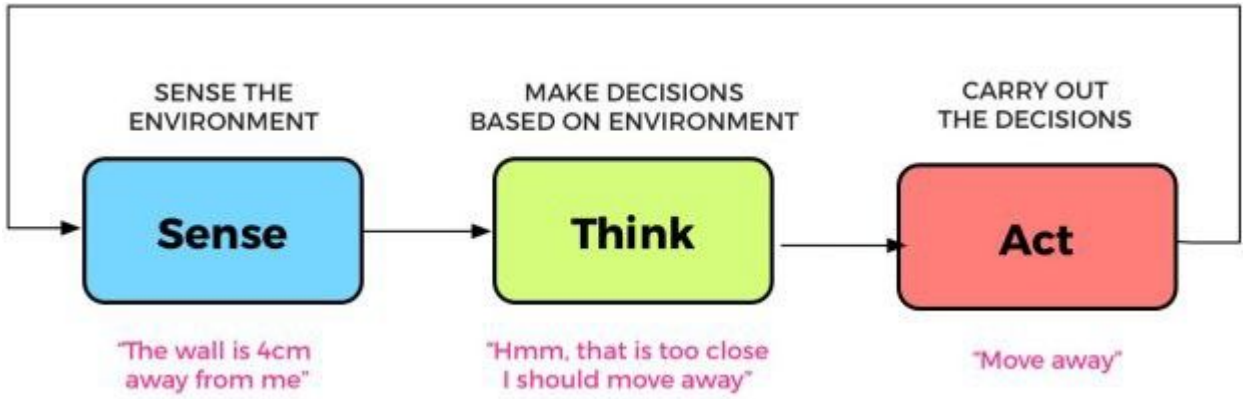
Objetivo

Revisar los principales sensores usados en vehículos autónomos y su funcionamiento

Active class

Navegación Autónoma

Ciclo Sense-Think-Act

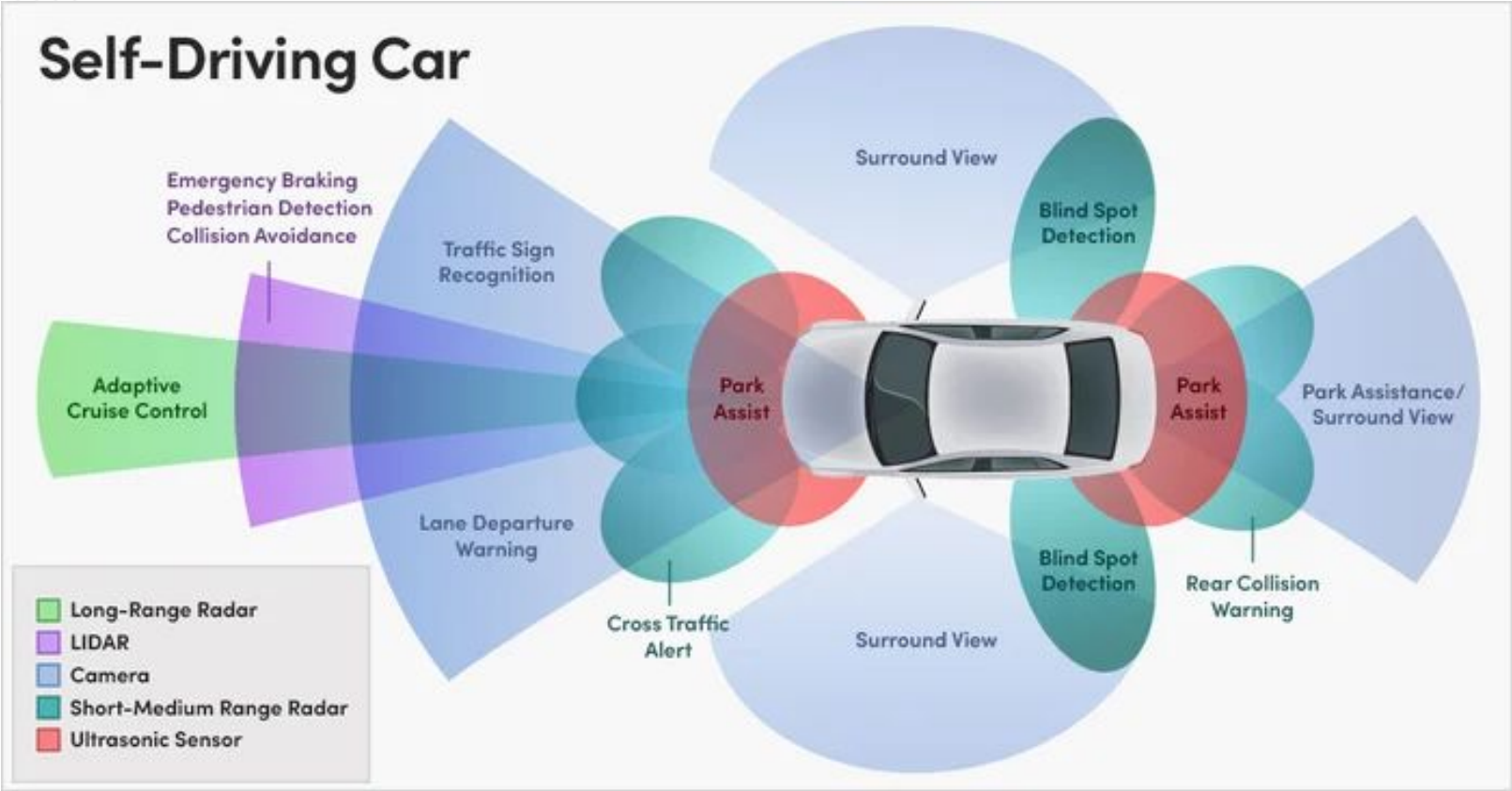


https://youngzine.org/article-expert/technology/machines-sense-think-and-act

Robótica implica diseñar máquinas que pueden detectar, pensar y actuar por sí solas, sin intervención humana y en un ciclo continuo. Si una máquina no cumple con este principio, entonces no se considera un robot.

En un vehículo autónomo, los sensores son los sentidos del vehículo, una o más computadoras proporcionan la inteligencia central del vehículo y, en función de los datos recabados del medio ambiente, el vehículo conduce independientemente a través del tráfico.

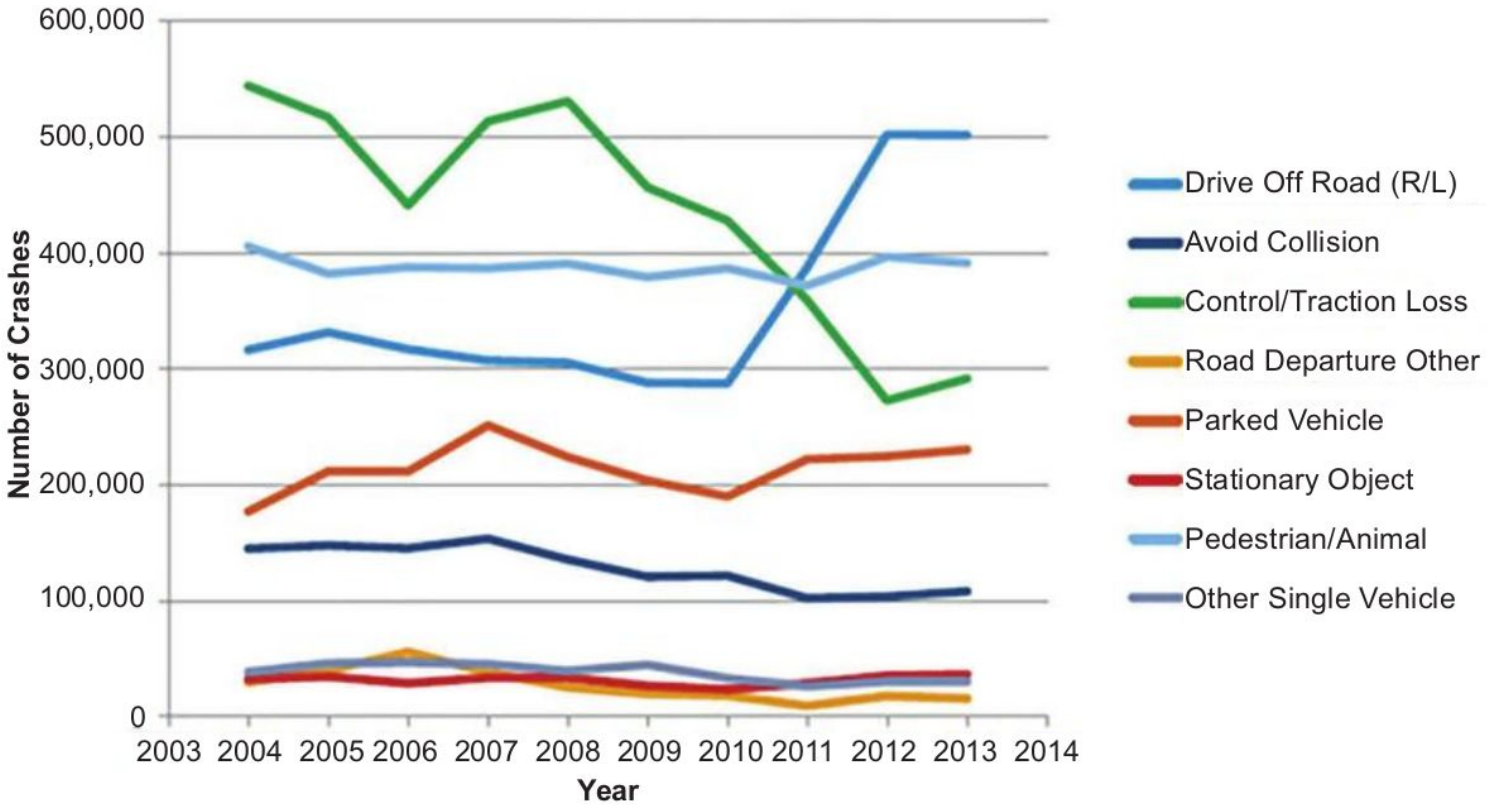
Vehículo Autónomo (Auto Robótico)



Un auto completamente autónomo (también llamado auto robótico) es un vehículo capaz de satisfacer las principales capacidades de transportación, de detectar/sensar su medio ambiente y navegar sin intervención humana.

https://blog.cloudfactory.com/autonomous-vehicle-training-conundrum

Navegación Autónoma



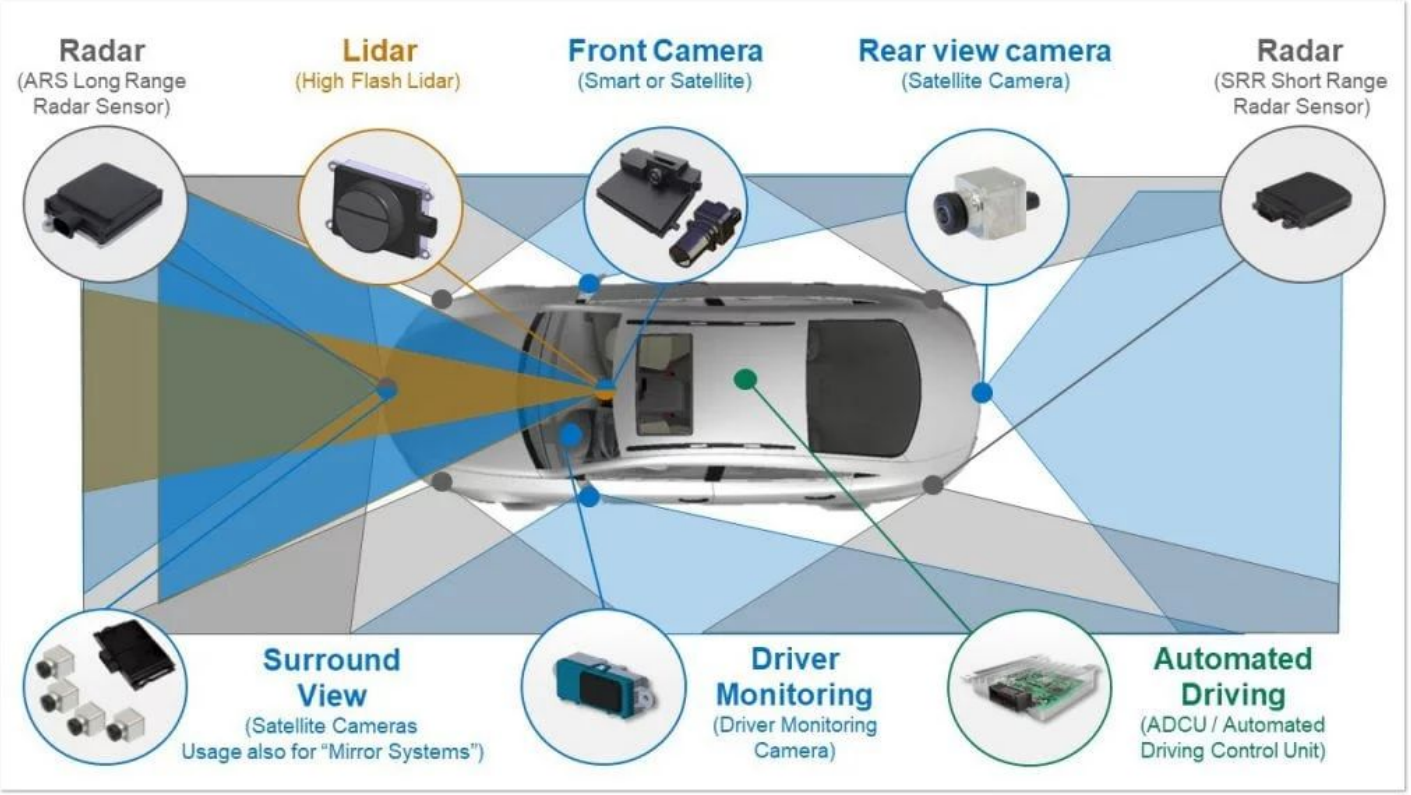
Se cree que la navegación autónoma tiene el potencial de reducir accidentes fatales y lesiones graves en un 90%.

Automated Driving and Driver Assistance Systems, Tom Denton

Active class

Advanced Driver Assistance System
ADAS

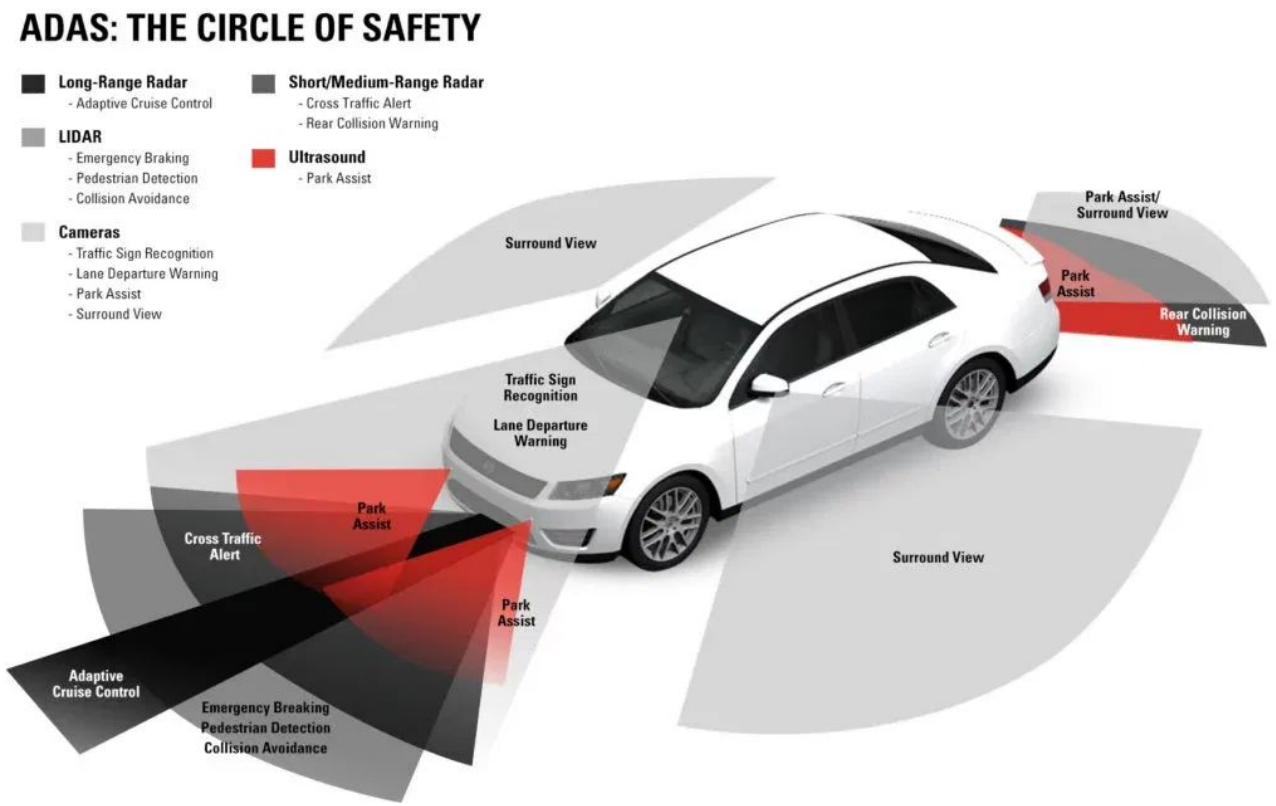
Concepto de ADAS



https://www.spinny.com/blog/index.php/adas-full-form/

ADAS basa su operación en información de varias fuentes, tales como lidar, radar, cámaras y datos del vehículo. Las medidas de seguridad se diseñan para alertar al conductor de problemas potenciales o evitar colisiones. Algunas veces, esto implica tomar control del vehículo.

Tipos de ADAS



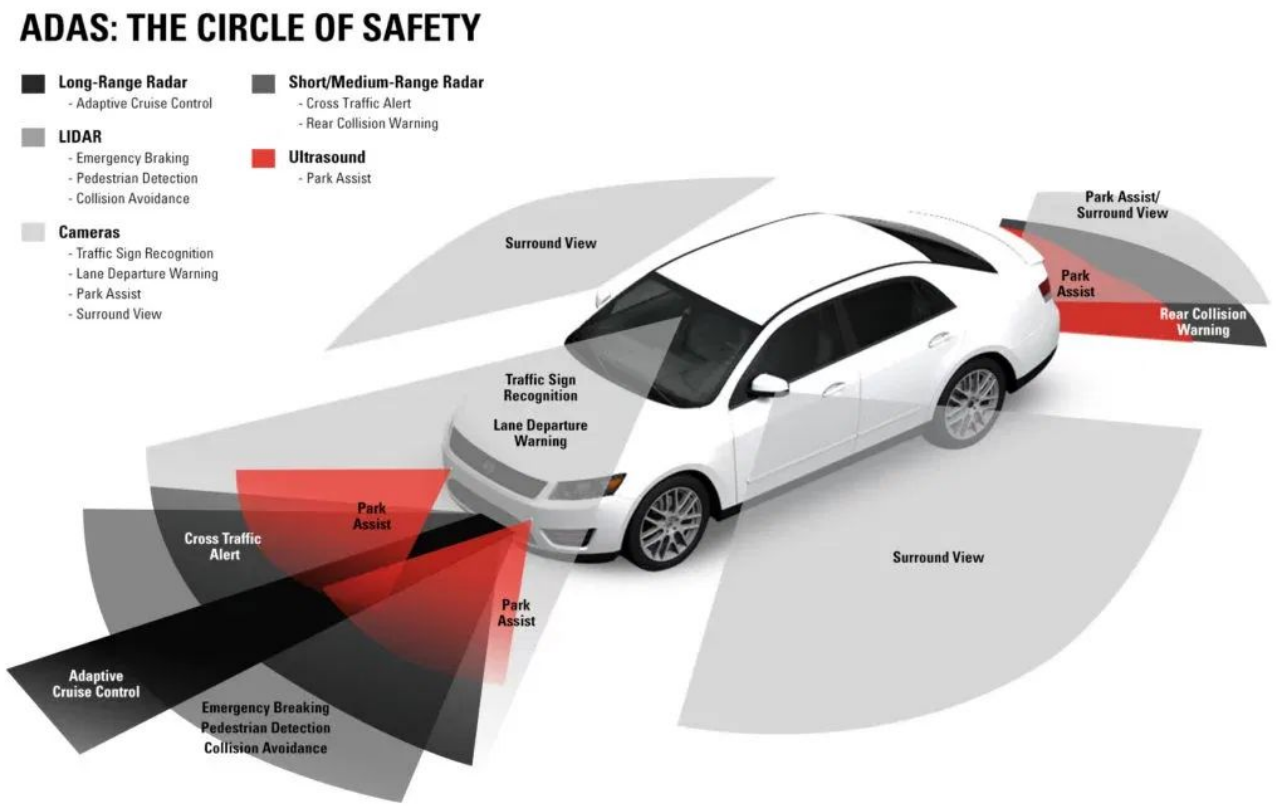
https://www.spinny.com/blog/index.php/adas-full-form/

Hay dos tipos de sistemas ADAS: pasivos y activos

Los sistemas pasivos sólo informan al conductor de peligros potenciales en tiempo para evitar un accidente:

- ABS (Anti-lock Braking System)
- ESC (Electronic Stability Control)
- TSC (Traction Control System)
- Backup Camera
- LDW (Lane Departure Warning)
- Forward Collision Warning
- Blind Spot Detection
- Parking Assistance

Tipos de ADAS



https://www.spinny.com/blog/index.php/adas-full-form/

En los sistemas activos, el auto toma control de la conducción si la computadora a bordo lo considera necesario:

- Automatic Emergency Braking
- Emergency Steering
- Adaptive Cruise Control
- Traffic Jam Assist
- Self Parking

Active class

Ubicación de Sensores en los Vehículos de Webots

Concepto de ADAS

Models of cars created using the Car PROTO

Webots ofrece siete vehículos simples para ser usados como PROTO nodes en los mundos.

- BMW X5
- Citroen CZero
- Lincoln MKZ
- Mercedes Benz Sprinter
- Range Rover Sport SVR
- Tesla Modelo 3
- Toyota Prius

https://cyberbotics.com/doc/automobile/proto-nodes

Slots para Sensores

En los vehículos de Webots, los sensores se agregan a los slots disponibles de cada vehículo. El slot central tiene la ubicación (translation) 0 0 0, es decir, dentro del vehículo. La ubicación de los slots para cada auto se muestran en la tabla.

Model	Front slot translation	Top slot translation	Rear slot translation
BmwX5	3.85 0 0.45	1 0 1.45	-1 0 0.3
LincolnMKZ	3.944 0 0.142	1.11 0 1.16	-1.06 0 0.33
RangeRoverSportSVR	3.5 0 0.5	1.4 0 1.3	-1.06 0 0.33
CitroenCZero	3.075 0 0.05	1.075 0 1.35	-0.425 0 0.3
ToyotaPrius	3.635 0 0.4	1.1 0 1.3	-0.85 0 0.3
TeslaModel3	3.79 0 0.142	1.11 0 1.16	-1.05 0 0.2
MercedesBenzSprinter	5.31 0 0.47	2.99 0 2.3	0 0 1 3.1415

Positions of the car sensor slots

https://cyberbotics.com/doc/automobile/proto-nodes

Slots para BMW X5

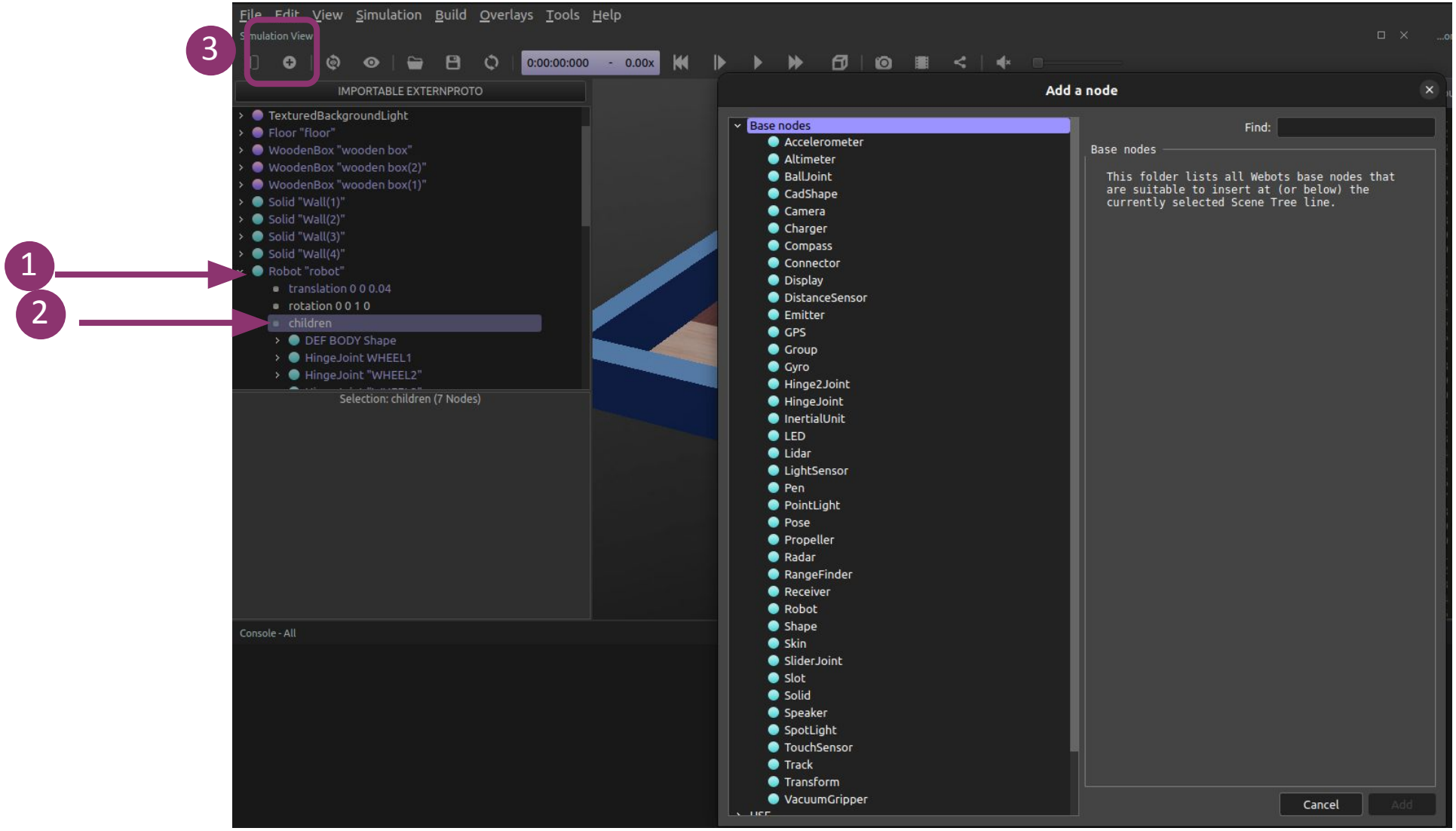
- BmwX5 "vehicle"
  - translation -45 45.9 0.4
  - rotation 0 0 1 3.14
  - color 0.43 0.11 0.1
  - engineSound "sounds/engine.wav"
  - name "vehicle"
  - controller "autonomous_vehicle"
  - controllerArgs
  - supervisor FALSE
  - synchronization TRUE
  - windshieldWipers TRUE
  - frontSpotLights FALSE
  - rearMirror FALSE
  - leftWingMirror FALSE
  - rightWingMirror FALSE
  - mirrorFar 200
  - sensorsSlotFront
  - sensorsSlotRear
  - sensorsSlotTop
  - sensorsSlotCenter

- sensorsSlotFront
  - SickLms291 "Sick LMS 291"
    - translation 0.06 0 0
    - rotation 0 0 1 0
    - name "Sick LMS 291"
    - noise 0
    - resolution 180
- sensorsSlotRear

- sensorsSlotCenter
  - GPS "gps"
    - translation 0 0 0
    - rotation 0 0 1 0
    - children
    - name "gps"
    - model ""
    - description ""
    - contactMaterial "default"
    - immersionProperties
    - boundingObject NULL
    - physics NULL
    - locked FALSE
    - translationStep 0.01
    - rotationStep 0.262
    - radarCrossSection 0
    - recognitionColors
    - type "satellite"
    - accuracy 0
    - noiseCorrelation 0
    - resolution -1
    - speedNoise 0
    - speedResolution -1
  - Gyro "gyro"
  - Display "display"

- sensorsSlotTop
  - Camera "camera"
    - translation 0.72 0 -0.05
    - rotation 0 0 1 0
    - children
    - name "camera"
    - model ""
    - description ""
    - contactMaterial "default"
    - immersionProperties
    - boundingObject NULL
    - physics NULL
    - locked FALSE
    - translationStep 0.01
    - rotationStep 0.262
    - radarCrossSection 0
    - recognitionColors
    - fieldOfView 1
    - width 128
    - height 64
    - projection "planar"
    - near 0.01
    - far 0
    - exposure 1
    - antiAliasing FALSE
    - ambientOcclusionRadius 0
    - bloomThreshold -1
    - motionBlur 0
    - noise 0
    - noiseMaskUrl ""
    - lens NULL
    - focus NULL
    - zoom NULL
    - recognition NULL
    - lensFlare NULL
- sensorsSlotCenter

Sensores genéricos en Webots



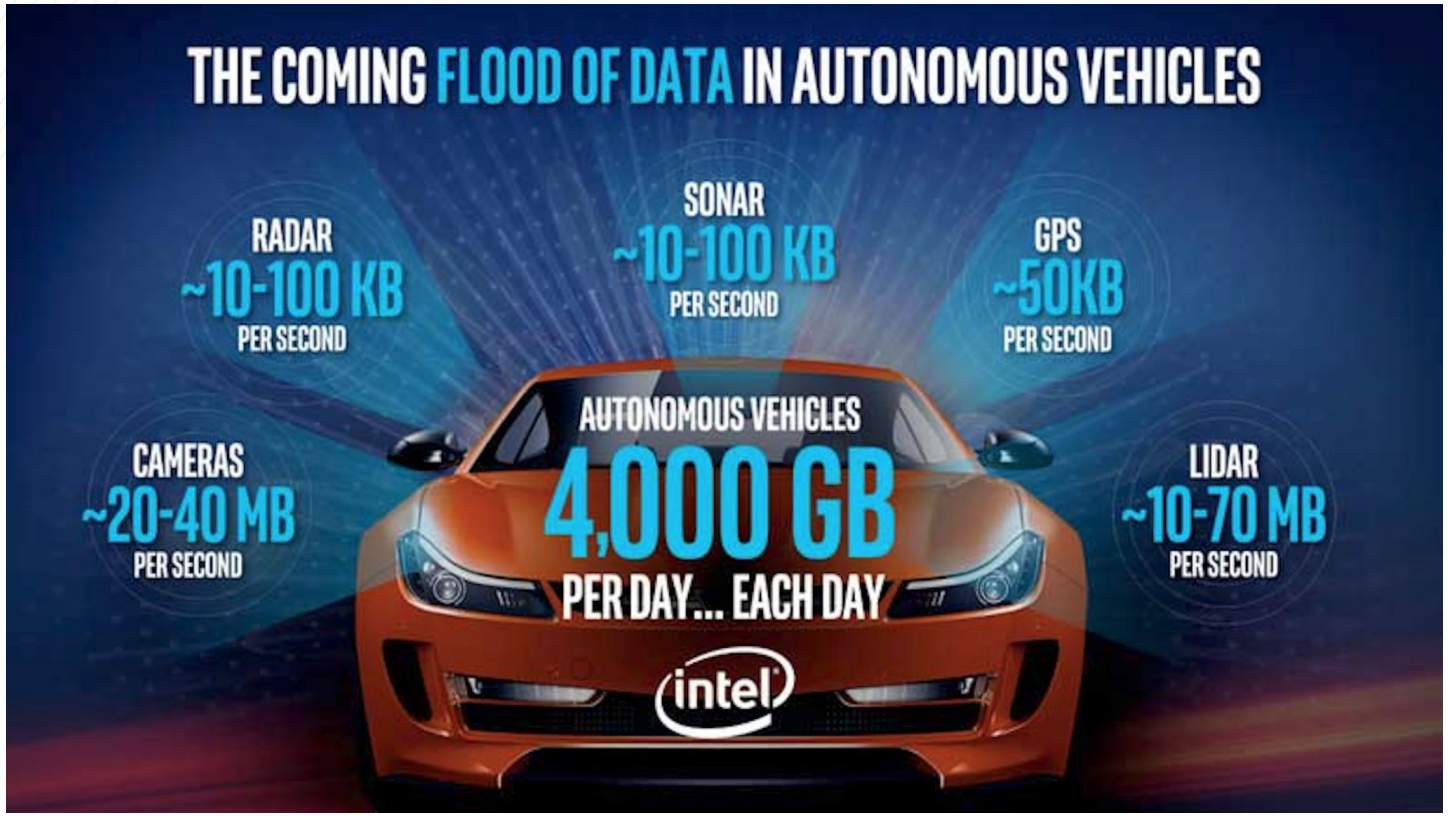
Active class

Demostración: Agregando Sensores en Webots

Active class

Sensores en ADAS

Sensores en ADAS



https://dewesoft.com/blog/types-of-adas-sensors

El vehículo percibe su entorno a través de un número de sensores, tales como

- Cámaras
- Radar
- Lidar
- Ultrasónicos
- IMU
- GNSS

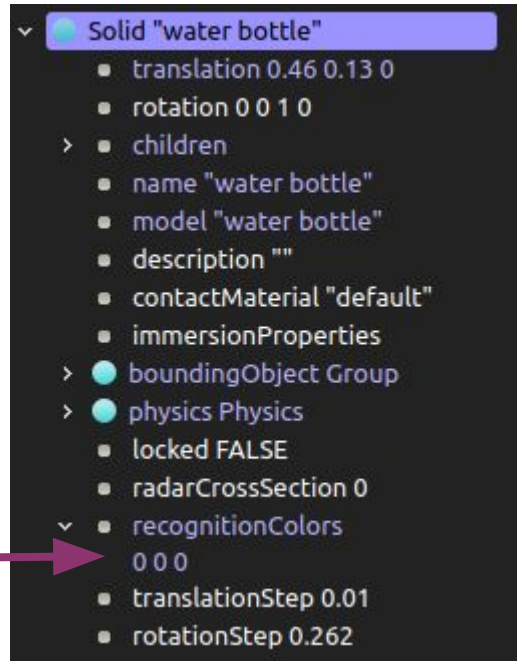
Cámaras

Los sistemas ADAS hacen uso de cámaras para detectar y reconocer, de manera rápida y precisa, otros vehículos, peatones, obstáculos, señales de tránsito, líneas de carriles, etc. Las cámaras se posicionan afuera del vehículo, al frente, en la parte posterior y a ambos lados.
Son adecuadas para detectar objetos estacionarios pequeños, mientras que el radar y el lidar son mejores para detectar objetos en movimiento.

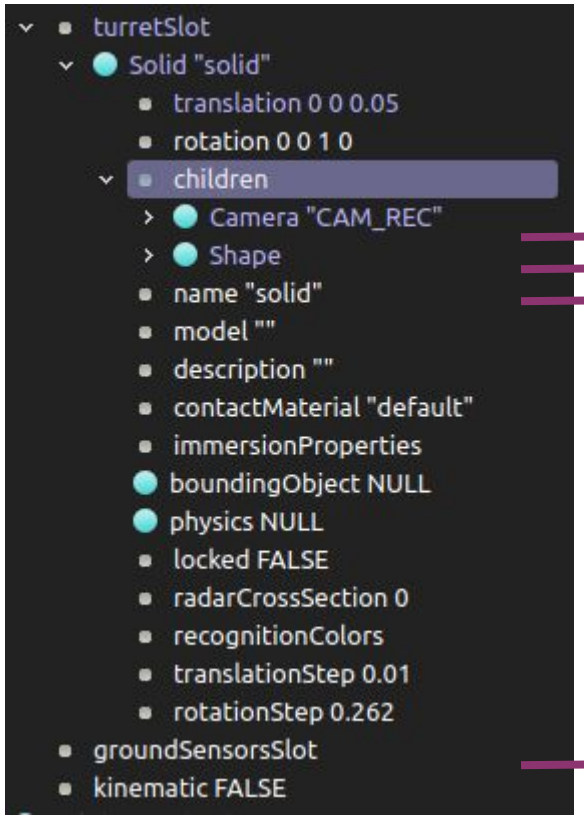
	Pros	Cons
Camera	– High resolution – Closest to human vision, detecting colors, and 2D shapes – Relatively low-cost sensor	– Sunlight can blind them – Poor vision during low visibility weather – Contrast is an issue – Depth information not available
Radar	– Detects in bad weather and in the dark	– Doesn't have high resolution at range (trouble telling bicycles from motorcycles)
Lidar	– High resolution at range – Very accurate 3d modeling and object detection	– Lesser 2D resolution than camera – Can't see through bad weather as well as radar – High cost – Requires significant computing power

https://caradas.com/adas-cameras/

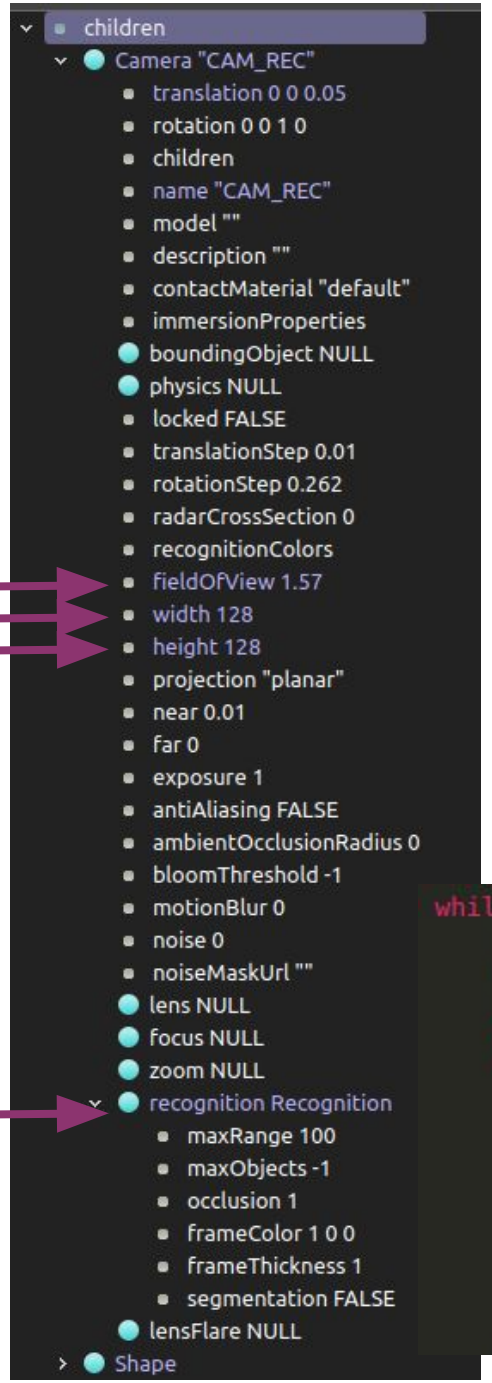
Demostración



Los objetos deben tener habilitado su recognitionColors y, para esto, deben ser convertidos a Base Node



Se agrega un sólido en forma de cilindro y, sobre este, se monta la cámara

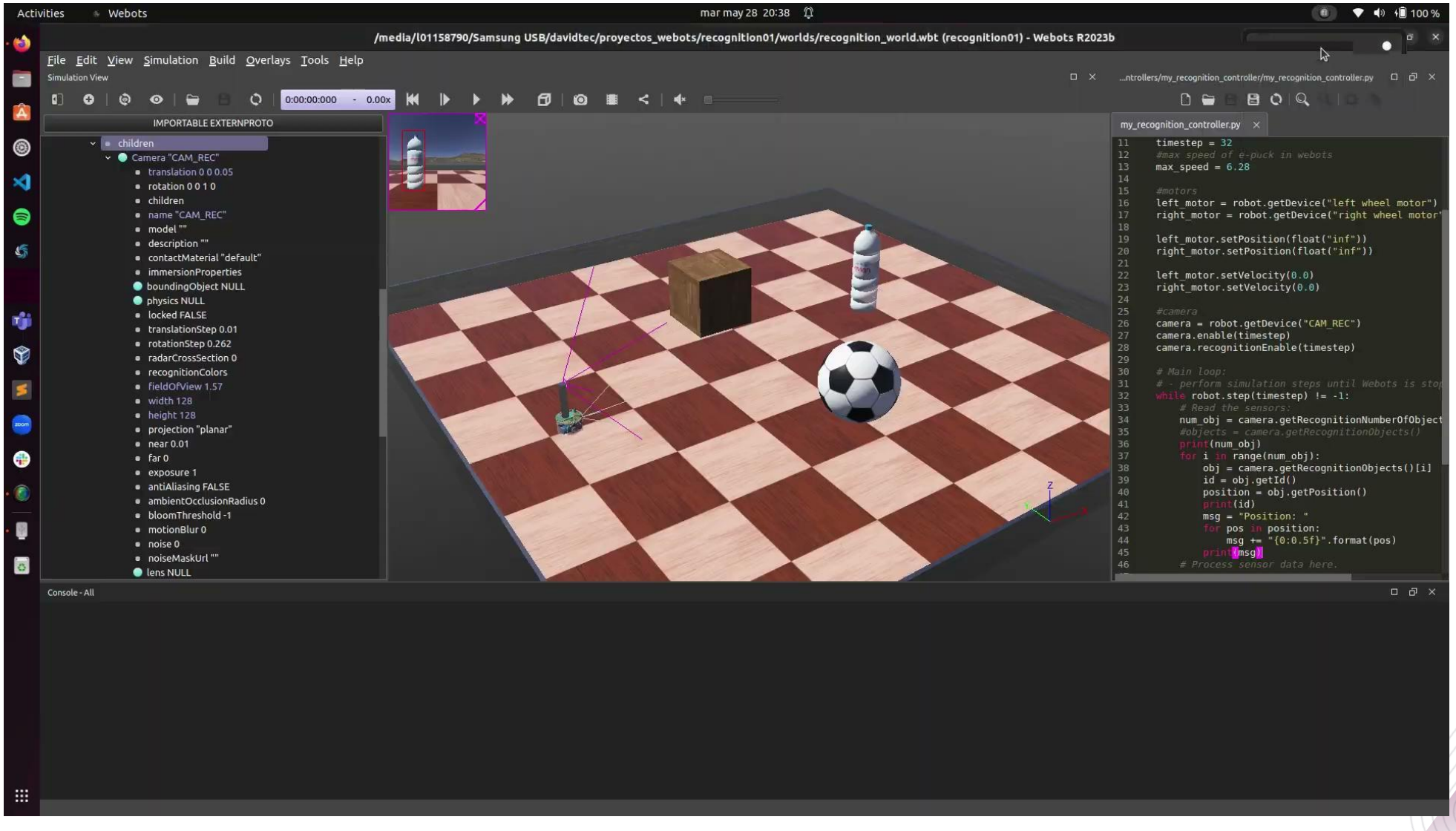


Se agrega una cámara genérica y se habilita la opción de recognition con sus valores por omisión

```
#camera
camera = robot.getDevice("CAM_REC")
camera.enable(timestep)
camera.recognitionEnable(timestep)
```

```
while robot.step(timestep) != -1:
    # Read the sensors:
    num_obj = camera.getRecognitionNumberOfObjects()
    #objects = camera.getRecognitionObjects()
    print(num_obj)
    for i in range(num_obj):
        obj = camera.getRecognitionObjects()[i]
        id = obj.getId()
        position = obj.getPosition()
        print(id)
        msg = "Position: "
        for pos in position:
            msg += "{0:0.5f}".format(pos)
        print(msg)
```

Demostración



Radar

Los sistemas de radar (Radio Detection and Ranging) emiten ondas de radio; estas señales viajan a la velocidad de la luz, se reflejan en los objetos en la trayectoria del radar y son procesados por el receptor del radar. Estos pulsos reflejados permiten al radar determinar la distancia y posición de los objetos, relativas a la posición del radar. Los sistemas de radar permiten detectar vehículos (dirección y velocidad) y otras obstrucciones en el entorno.

	Range (in meters)	General Use	Placement	ADAS Systems
Short Range	0.5–20	For detecting objects in close proximity	Rear vehicle corners	– Blind Spot Detection – Rear Collision Warning – Rear Cross Traffic Alert
Medium Range	1–60	Typically utilized by vehicles as they navigate around town because it's more effective at detecting objects at medium distances.	Front/Varies	– Front Cross Traffic Assist – Lane Change Assist
Long Range	10–250	Typically used on highways and in specific areas with high traffic density like interchanges and junctions.	Forward facing, near the front bumper or behind the grille	– Automatic Emergency Braking – Adaptive Cruise Control (& Traffic Jam Assist) – Forward Collision Warning

https://caradas.com/adas-cameras/

Demostración

Solid

Derived from Pose.

```
Solid {
  SFString name                "solid"       # any string
  SFString model               ""            # any string
  SFString description         ""            # any string
  SFString contactMaterial     "default"     # any string
  MFNode   immersionProperties [ ]           # {ImmersionProperties, PROTO}
  SFNode   boundingObject      NULL          # {node, PROTO}
  SFNode   physics             NULL          # {Physics, PROTO}
  SFBool   locked              FALSE         # {TRUE, FALSE}
  SFFloat  radarCrossSection   0.0           # [0, 1]
  MFColor  recognitionColors   []            # any color
  # hidden fields
  SFVec3f  linearVelocity      0 0 0         # any vector
  SFVec3f  angularVelocity     0 0 0         # any vector
}
```

- `radarCrossSection` : if greater than 0 this Solid node is a potential target for any Radar device. Radar cross section (RCS) is the measure of a target's ability to reflect radar signals in the direction of the radar receiver, i.e. it is a measure of the ratio of backscatter density in the direction of the radar to the power density that is intercepted by the target. Typical values are 0.01 for a bird, 1 for a human, 100 for a car and 200 for a truck.

https://cyberbotics.com/doc/reference/solid?tab-language=python

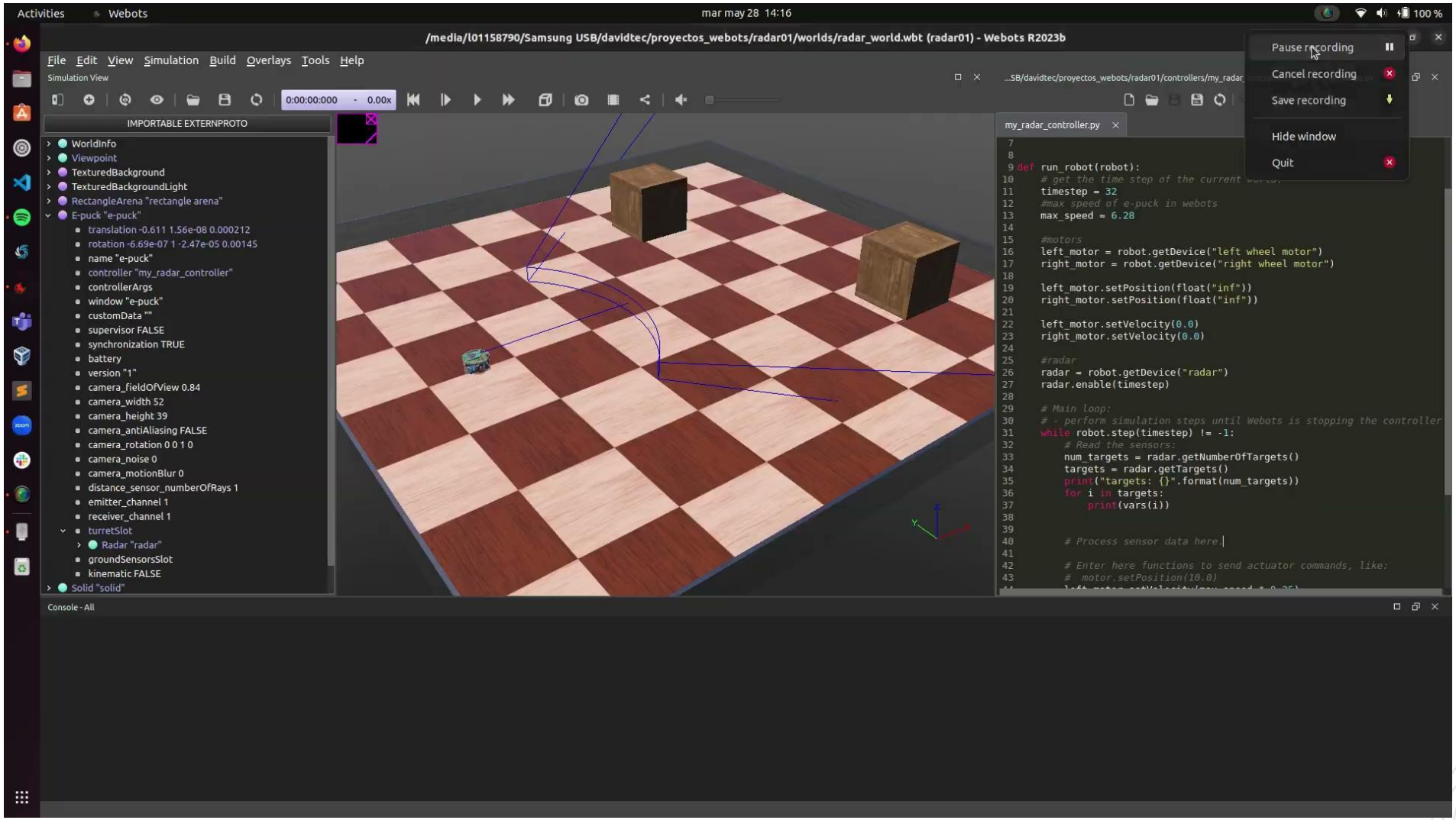
Demostración

- E-puck "e-puck"
  - translation -0.611 1.56e-08 0.000212
  - rotation -6.69e-07 1 -2.47e-05 0.00145
  - name "e-puck"
  - controller "my_radar_controller"
  - controllerArgs
  - window "e-puck"
  - customData ""
  - supervisor FALSE
  - synchronization TRUE
  - battery
  - version "1"
  - camera_fieldOfView 0.84
  - camera_width 52
  - camera_height 39
  - camera_antiAliasing FALSE
  - camera_rotation 0 0 1 0
  - camera_noise 0
  - camera_motionBlur 0
  - distance_sensor_numberOfRays 1
  - emitter_channel 1
  - receiver_channel 1
  - turretSlot
    - Radar "radar"
      - translation 0 0 0
      - rotation 0 0 1 0
      - children
      - name "radar"
      - model ""
      - description ""
      - contactMaterial "default"
      - immersionProperties
      - boundingObject NULL
      - physics NULL
      - locked FALSE
      - translationStep 0.01
      - rotationStep 0.262
      - radarCrossSection 0
      - recognitionColors
      - minRange 0.5
      - maxRange 10
      - horizontalFieldOfView 1.57
      - verticalFieldOfView 0.1
      - minAbsoluteRadialSpeed 0
      - minRadialSpeed 0
      - maxRadialSpeed 0
      - cellDistance 0
      - cellSpeed 0
      - rangeNoise 0
      - speedNoise 0
      - angularNoise 0
      - antennaGain 20
      - frequency 24
      - transmittedPower 20
      - minDetectableSignal -100
      - occlusion FALSE
  - groundSensorsSlot
  - kinematic FALSE

- Solid "solid"
  - translation 0 0.07 0.01
  - rotation 0 0 1 0
  - children
    - WoodenBox "wooden box"
      - translation 0.32 0.44 0.1
      - rotation 0 0 1 0
      - name "wooden box"
      - size 0.2 0.2 0.2
      - mass 0
      - immersionProperties
      - locked FALSE
  - name "solid"
  - model ""
  - description ""
  - contactMaterial "default"
  - immersionProperties
  - boundingObject NULL
  - physics NULL
  - locked FALSE
  - radarCrossSection 1
  - recognitionColors
  - translationStep 0.01
  - rotationStep 0.262
- Solid "solid(1)"

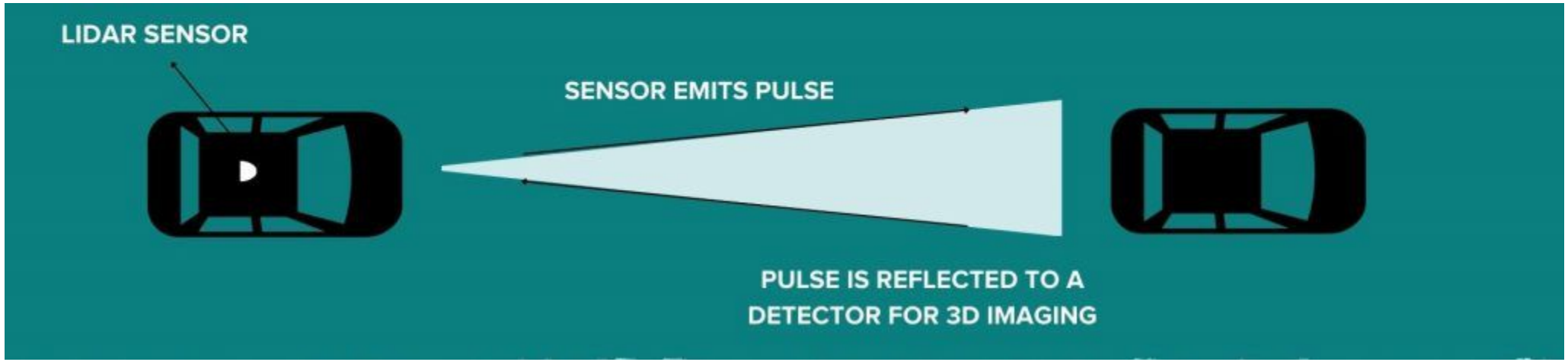
https://caradas.com/adas-cameras/

Demostración



Lidar

Un lidar (light detection and ranging) mide varios factores a lo largo de su trayectoria usando un láser pulsado, ayudando en la creación de mapas del entorno del vehículo. El pulso es emitido sobre una superficie y es reflejado. El pulso reflejado es procesado para localizar objetos en la trayectoria del vehículo.
Un lidar puede ver situaciones de tráfico complejas aún en la oscuridad y es capaz de grabar los 360 grados del entorno del vehículo.



https://loconav.com/blog/lidar/

Lidar

Una nube de puntos (point cloud) es un conjunto de puntos de datos en el espacio generados por un lidar, los cuales miden un número masivo de puntos sobre las superficies de los objetos en el entorno.

https://velodynelidar.com/blog/velodyne-lidars-alpha-prime-ready-for-prime-time/

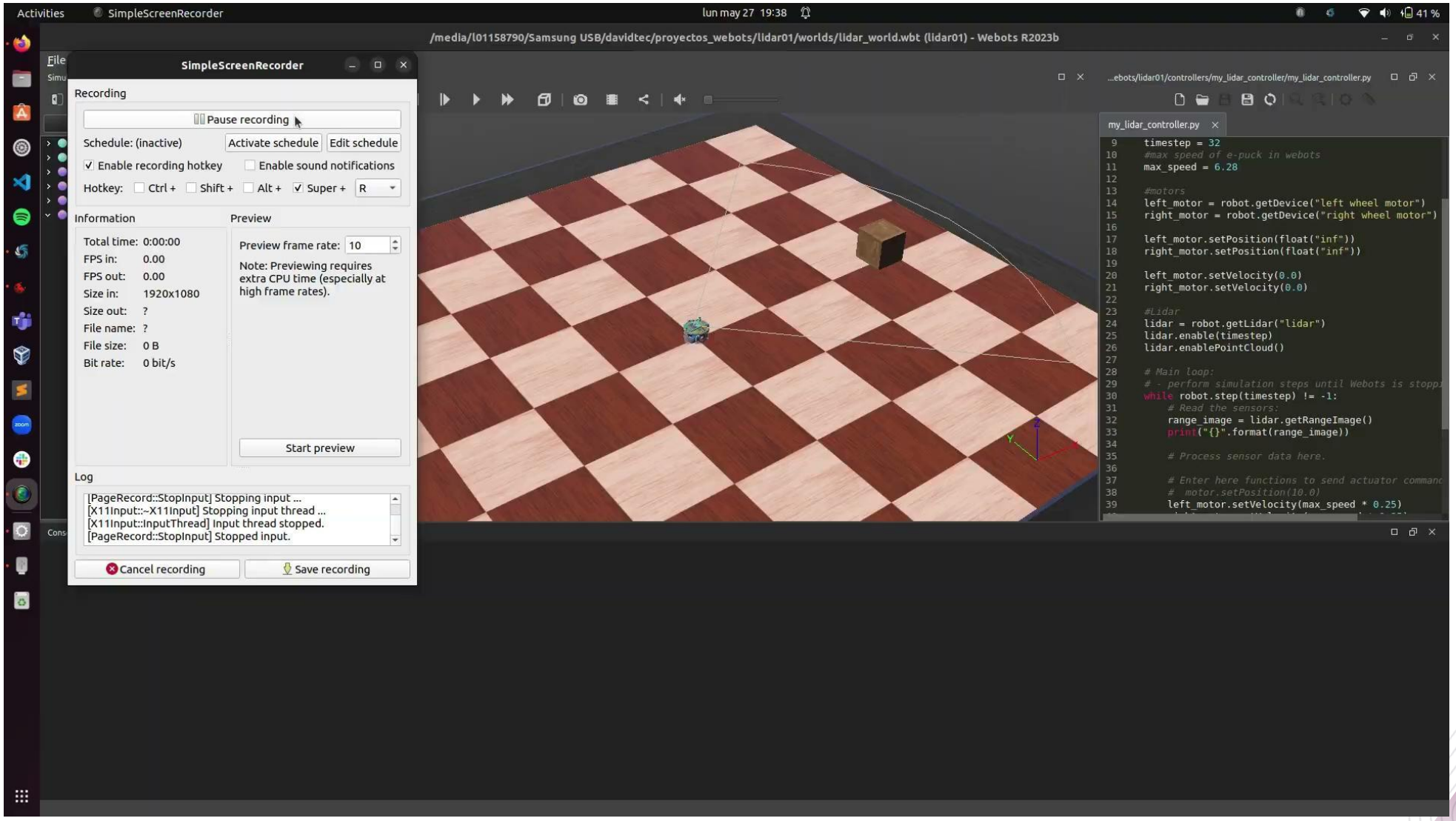
Demostración

- E-puck "e-puck"
  - translation 0 0 0
  - rotation 0 0 1 0
  - name "e-puck"
  - controller "my_lidar_controller"
  - controllerArgs
  - window "e-puck"
  - customData ""
  - supervisor FALSE
  - synchronization TRUE
  - battery
  - version "1"
  - camera_fieldOfView 0.84
  - camera_width 52
  - camera_height 39
  - camera_antiAliasing FALSE
  - camera_rotation 0 0 1 0
  - camera_noise 0
  - camera_motionBlur 0
  - distance_sensor_numberOfRays 1
  - emitter_channel 1
  - receiver_channel 1
  - turretSlot
    - Lidar "lidar"

```python
#Lidar
lidar = robot.getLidar("lidar")
lidar.enable(timestep)
lidar.enablePointCloud()

# Main loop:
# - perform simulation steps until Webots is stopping the controller
while robot.step(timestep) != -1:
    # Read the sensors:
    range_image = lidar.getRangeImage()
    print("{}".format(range_image))

    # Process sensor data here.

    # Enter here functions to send actuator commands, like:
    #  motor.setPosition(10.0)
    left_motor.setVelocity(max_speed * 0.25)
    right_motor.setVelocity(max_speed * 0.25)
```

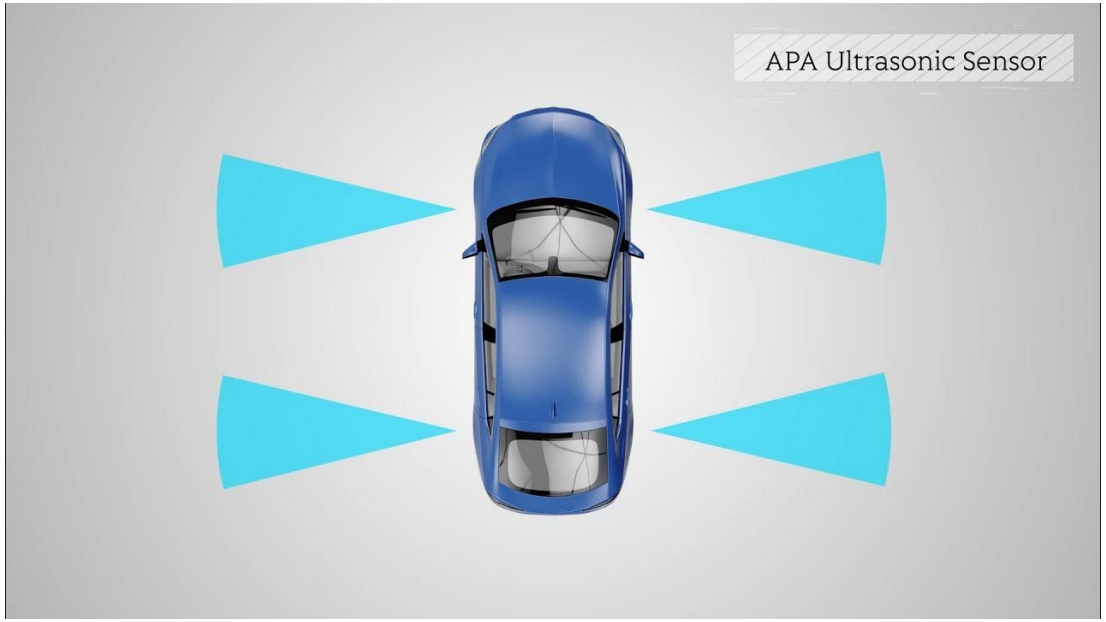
- turretSlot
  - Lidar "lidar"
    - translation 0 0 0
    - rotation 0 0 1 0
    - children
    - name "lidar"
    - model ""
    - description ""
    - contactMaterial "default"
    - immersionProperties
    - boundingObject NULL
    - physics NULL
    - locked FALSE
    - translationStep 0.01
    - rotationStep 0.262
    - radarCrossSection 0
    - recognitionColors
    - tiltAngle 0
    - horizontalResolution 512
    - fieldOfView 1.57
    - verticalFieldOfView 0.2
    - numberOfLayers 1
    - near 0.01
    - minRange 0.01
    - maxRange 1
    - type "fixed"
    - projection "cylindrical"
    - noise 0
    - resolution -1
    - defaultFrequency 10
    - minFrequency 1
    - maxFrequency 25
    - rotatingHead NULL

Demostración



Sensores Ultrasónicos

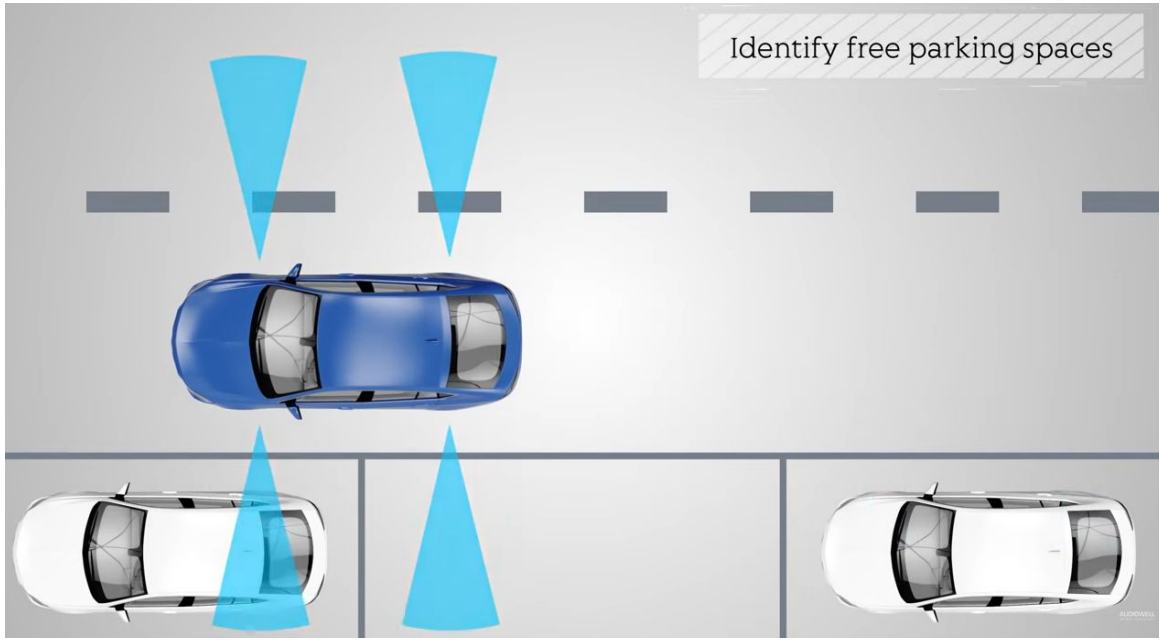
Estos sensores emiten ondas de sonido de alta frecuencia. Cuando estas ondas encuentran un objeto, son reflejadas de regreso. El tiempo que la onda se toma en regresar es usado para calcular la distancia entre el sensor y el objeto. Estos sensores se colocan en el perímetro del vehículo, por fuera, y pueden detectar objetos en movimiento y estacionarios.



https://www.youtube.com/watch?app=desktop&v=pODfyPpXr38

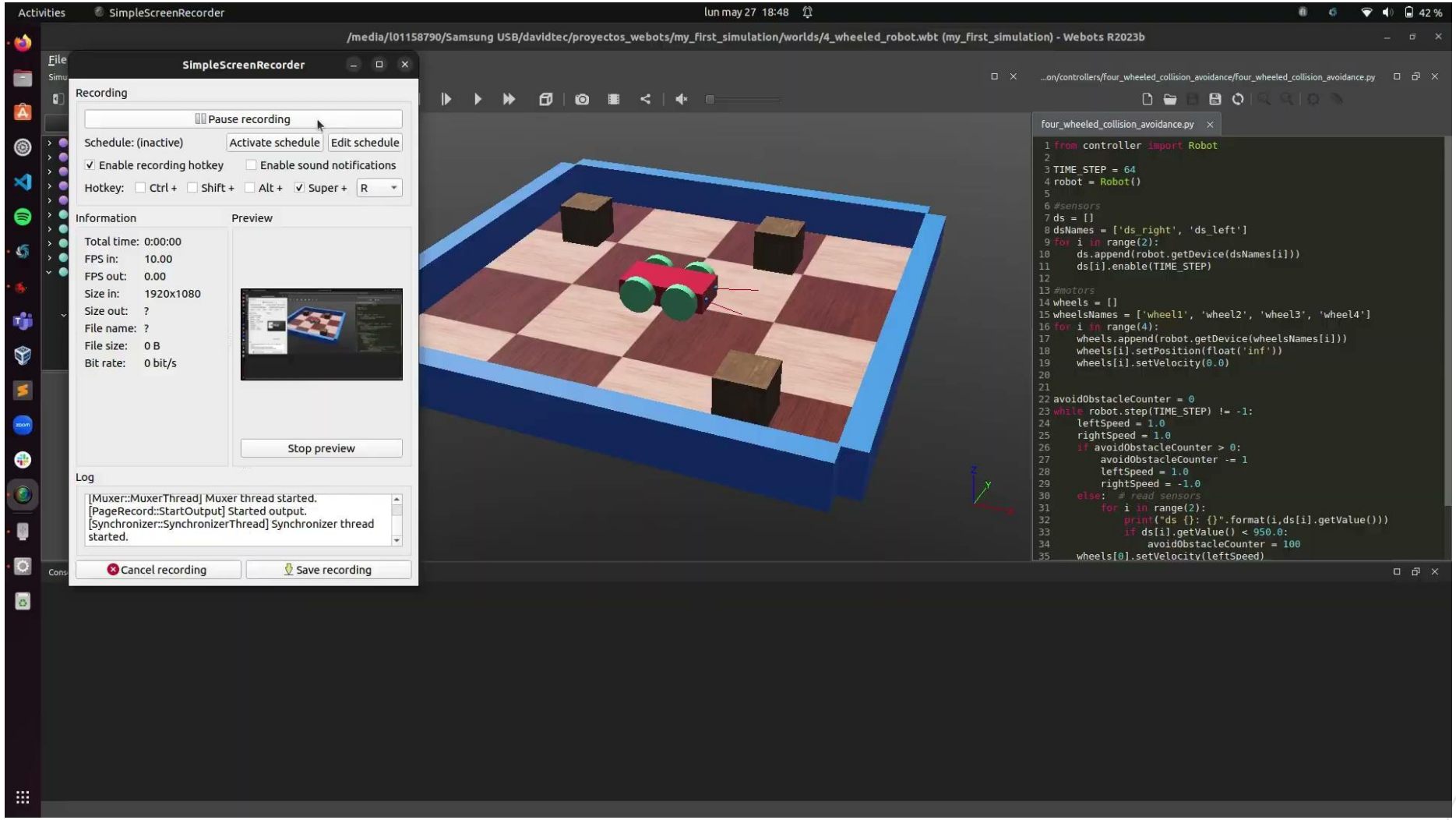
Sensores Ultrasónicos

Entre sus limitaciones, se encuentra que son susceptibles de interferencia del ruido externo, que tienen un alcance limitado y no son capaces de detectar objetos que se encuentren muy lejanos (2 a 5 m), situaciones en las cuales se sustituye su uso por un radar.



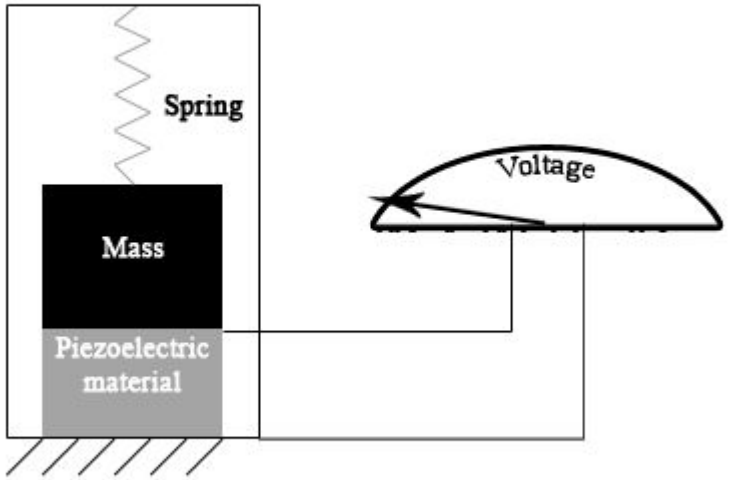
https://www.youtube.com/watch?app=desktop&v=pODfyPpXr38

Demostración



IMU

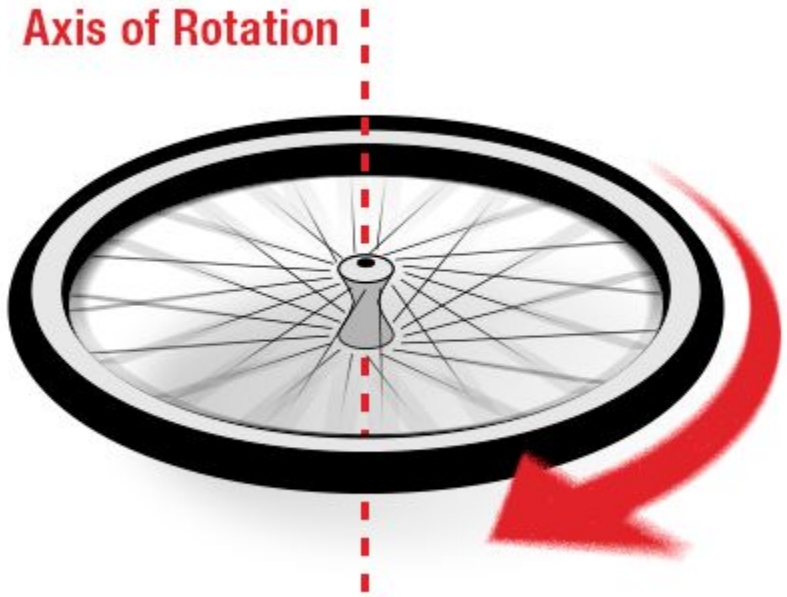
Un IMU (Inertial Measurement Unit) es un dispositivo electrónico que usualmente incluye tres sensores: acelerómetro, giroscopio y magnetómetro.
Un acelerómetro detecta movimiento lineal sin reportar la velocidad, únicamente la aceleración y la dirección (en función de la gravedad). Pueden detectar fuerzas estáticas, como la gravedad, o vibraciones y movimiento, en uno o los tres ejes.

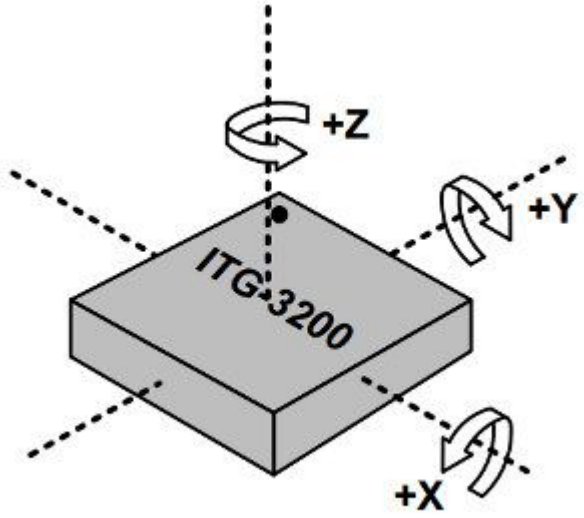


https://learn.sparkfun.com/tutorials/accelerometer-basics/all

IMU

Un giroscopio detecta el movimiento rotacional, sin reportar el ángulo actual, sino únicamente la tasa a la cual el dispositivo está girando (velocidad angular), en grados por segundo (°/s) o revoluciones por segundo (rps).
Si la rueda da una vuelta por segundo, la medición será de 360°/s. Esto puede ocurrir en sentido horario o antihorario.

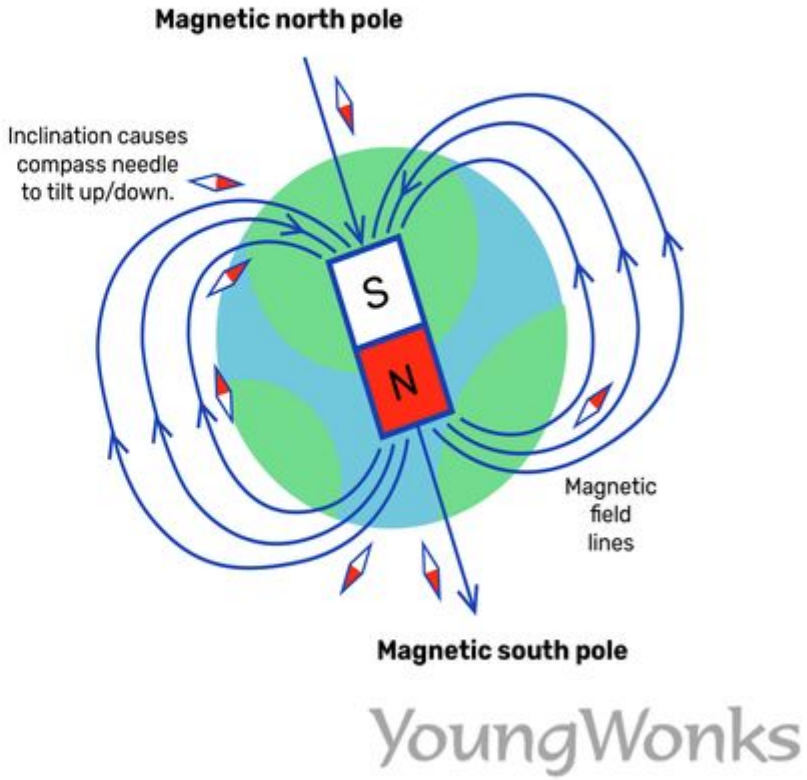




https://learn.sparkfun.com/tutorials/gyroscope?_ga=2.195309738.765125581.1713898726-2034140019.1713898724

IMU

Un magnetómetro mide el campo magnético en función de la intensidad y orientación. Un ejemplo de cómo opera el magnetómetro es la brújula. La brújula mide la orientación del campo magnético de la Tierra.



https://www.youngwonks.com/blog/What-is-a-Magnetometer-and-How-Does-It-Work

Demostración

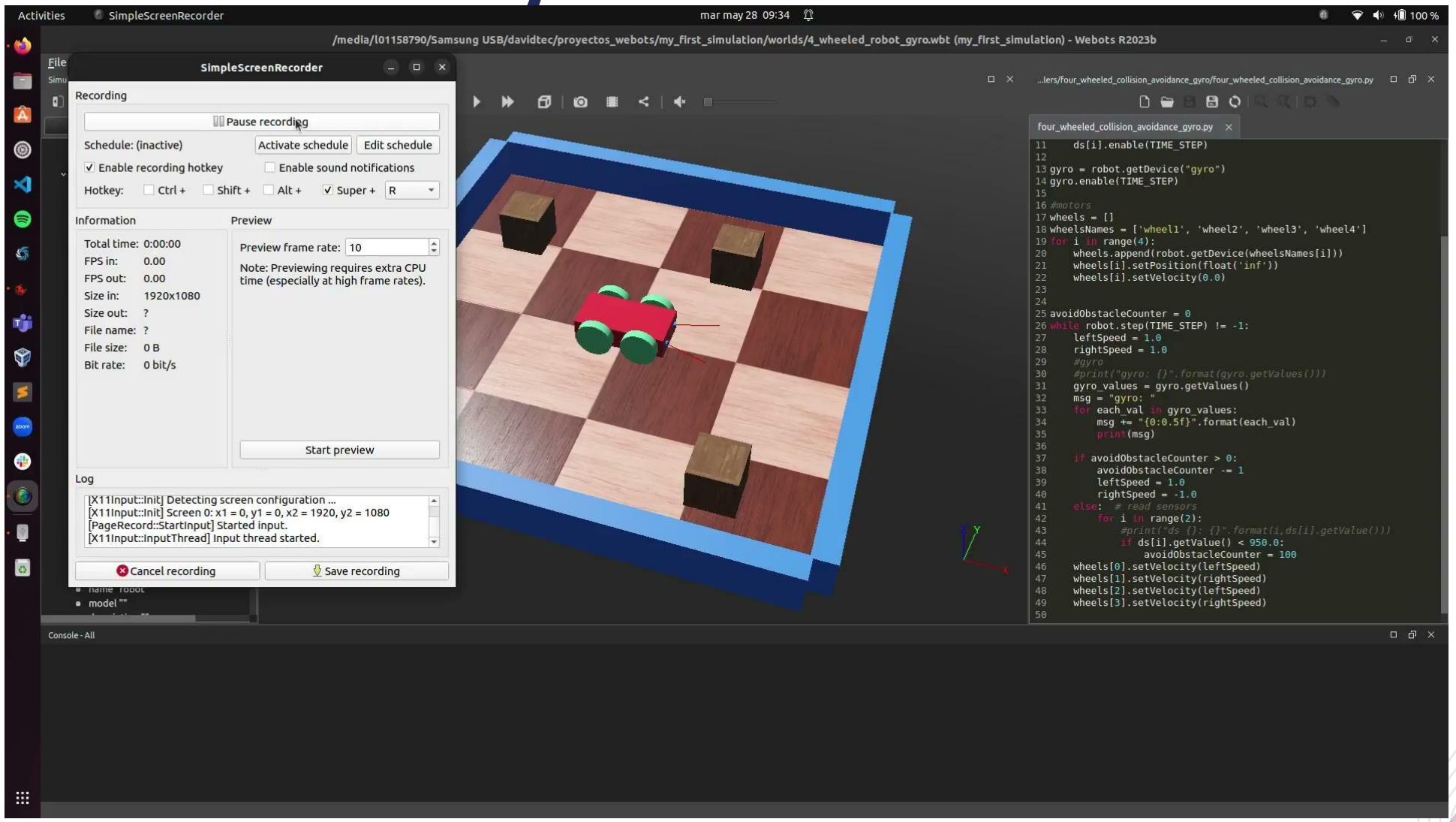
```
Robot "robot"
    translation 0 0 0.04
    rotation 0 0 1 0
    children
        InertialUnit "imu"
        Gyro "gyro"
        DEF BODY Shape
        HingeJoint WHEEL1
        HingeJoint "WHEEL2"
        HingeJoint "WHEEL3"
        HingeJoint "WHEEL4"
        DistanceSensor "ds_right"
        DistanceSensor "ds_left"
    name "robot"
    model ""
    description ""
    contactMaterial "default"
    immersionProperties
    boundingObject USE BODY
    physics Physics
    locked FALSE
    translationStep 0.01
    rotationStep 0.262
    radarCrossSection 0
    recognitionColors
    controller "four_wheeled_collision_avoidance_gyro"
```

```
children
    InertialUnit "imu"
        translation 0 0 0
        rotation 0 0 1 0
        children
        name "imu"
        model ""
        description ""
        contactMaterial "default"
        immersionProperties
        boundingObject NULL
        physics NULL
        locked FALSE
        translationStep 0.01
        rotationStep 0.262
        radarCrossSection 0
        recognitionColors
        noise 0
        xAxis TRUE
        yAxis TRUE
        zAxis TRUE
        resolution -1
    Gyro "gyro"
        translation 0 0 0
        rotation 0 0 1 0
        children
        name "gyro"
        model ""
        description ""
        contactMaterial "default"
        immersionProperties
        boundingObject NULL
        physics NULL
        locked FALSE
        translationStep 0.01
        rotationStep 0.262
        radarCrossSection 0
        recognitionColors
        lookupTable
        xAxis TRUE
        yAxis TRUE
        zAxis TRUE
        resolution -1
```

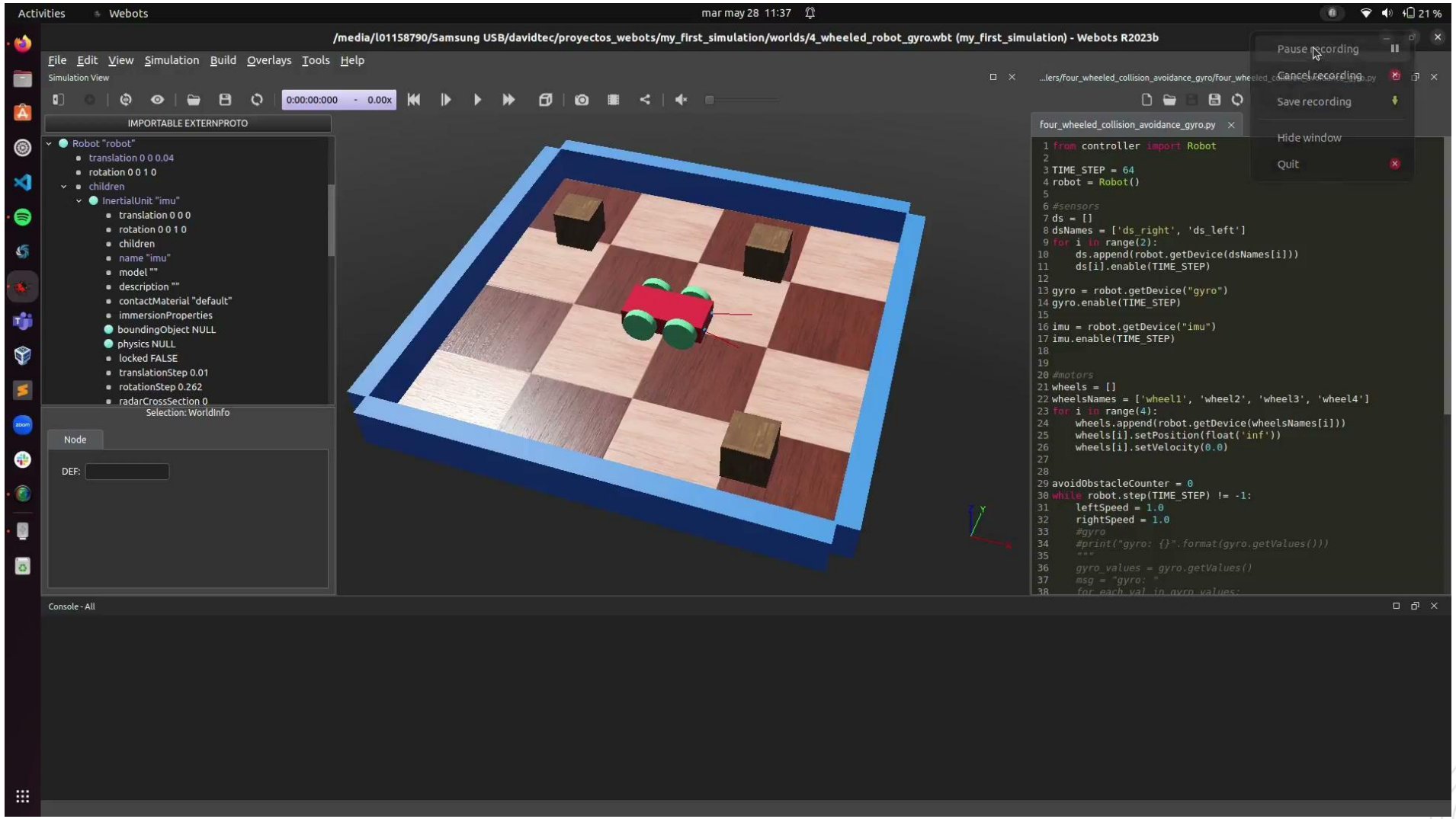
```python
gyro = robot.getDevice("gyro")
gyro.enable(TIME_STEP)

imu = robot.getDevice("imu")
imu.enable(TIME_STEP)
```

```python
    gyro_values = gyro.getValues()
    msg = "gyro: "
    for each_val in gyro_values:
        msg += "{0:0.5f}".format(each_val)
    print(msg)

    imu_values = imu.getRollPitchYaw()
    msg = "imu: "
    for each_val in imu_values:
        msg += "{0:0.5f}".format(each_val)
    print(msg)
```

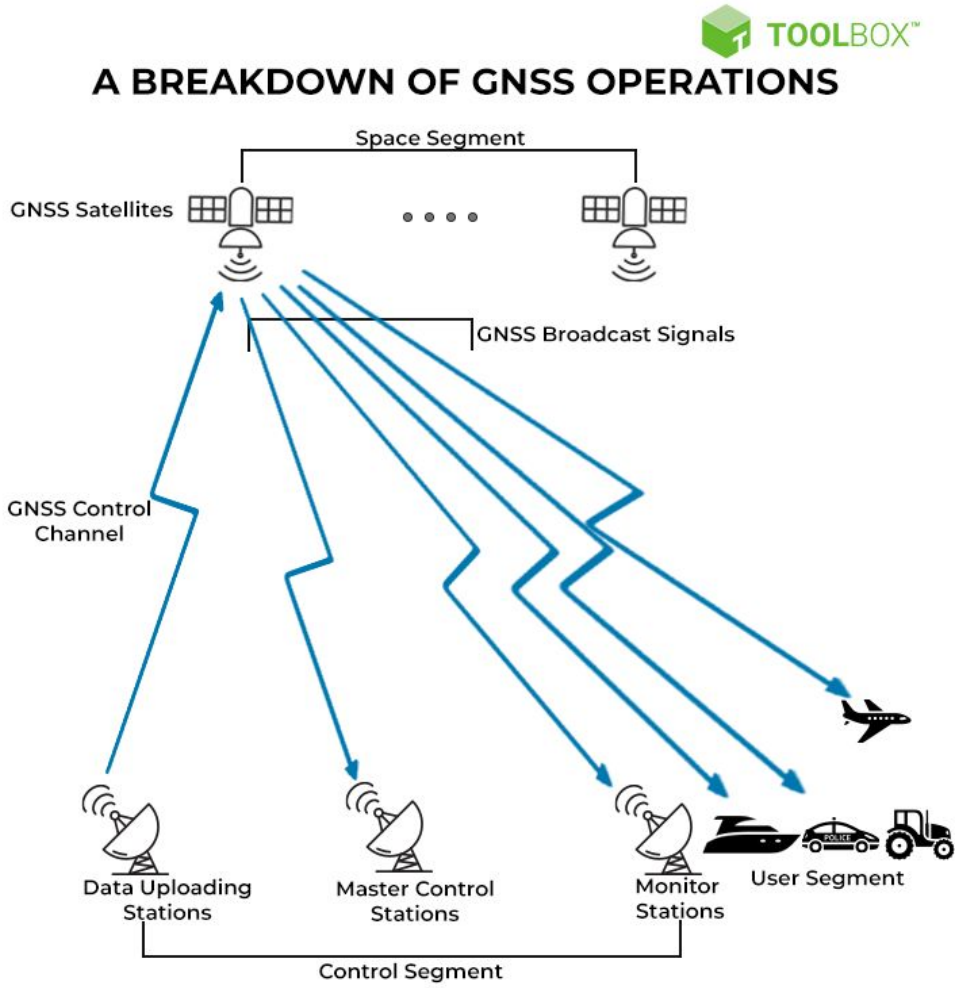
Demostración: Gyro



Demostración: IMU



GNSS



GNSS (Global Navigation Satellite System) en conjunto con IMUs proporcionan información acerca de la ubicación global, velocidad y orientación del vehículo. GPS tiene una exactitud de aprox. 5 m, mientras que en GNSS es de centímetros. GNSS incluye satélites de Rusia, China, Europa y Japón.

https://www.spiceworks.com/tech/iot/articles/what-is-gnss/

GNSS

Para fines de conducción automatizada, GNSS puede operar con un IMU para complementar la percepción del vehículo.

	Technology	Strength	Weakness
Perception	Radar	Good ranging accuracy	Cannot detect road markings
	Lidar	Highly accurate ranging	Higher cost Less effective in featureless areas (country roads)
	Camera	Good object detection Good in tunnels/urban canyons	Less effective on featureless roads Less effective in low visibility (snow, rain, darkness, fog) Roads without lane markings, construction areas
Absolute Localization	HD map + Perception	Good accuracy	High acquisition and maintenance costs Requires continuous updates Difficulty in changing surroundings (highways) May not be available everywhere
	GNSS+IMU	Good accuracy Common global reference between vehicles Good in low visibility, featureless roads and changing surroundings (highways)	Less reliable in high urban canyons Not usable in long tunnels (a few km) due to high drift rate of IMU

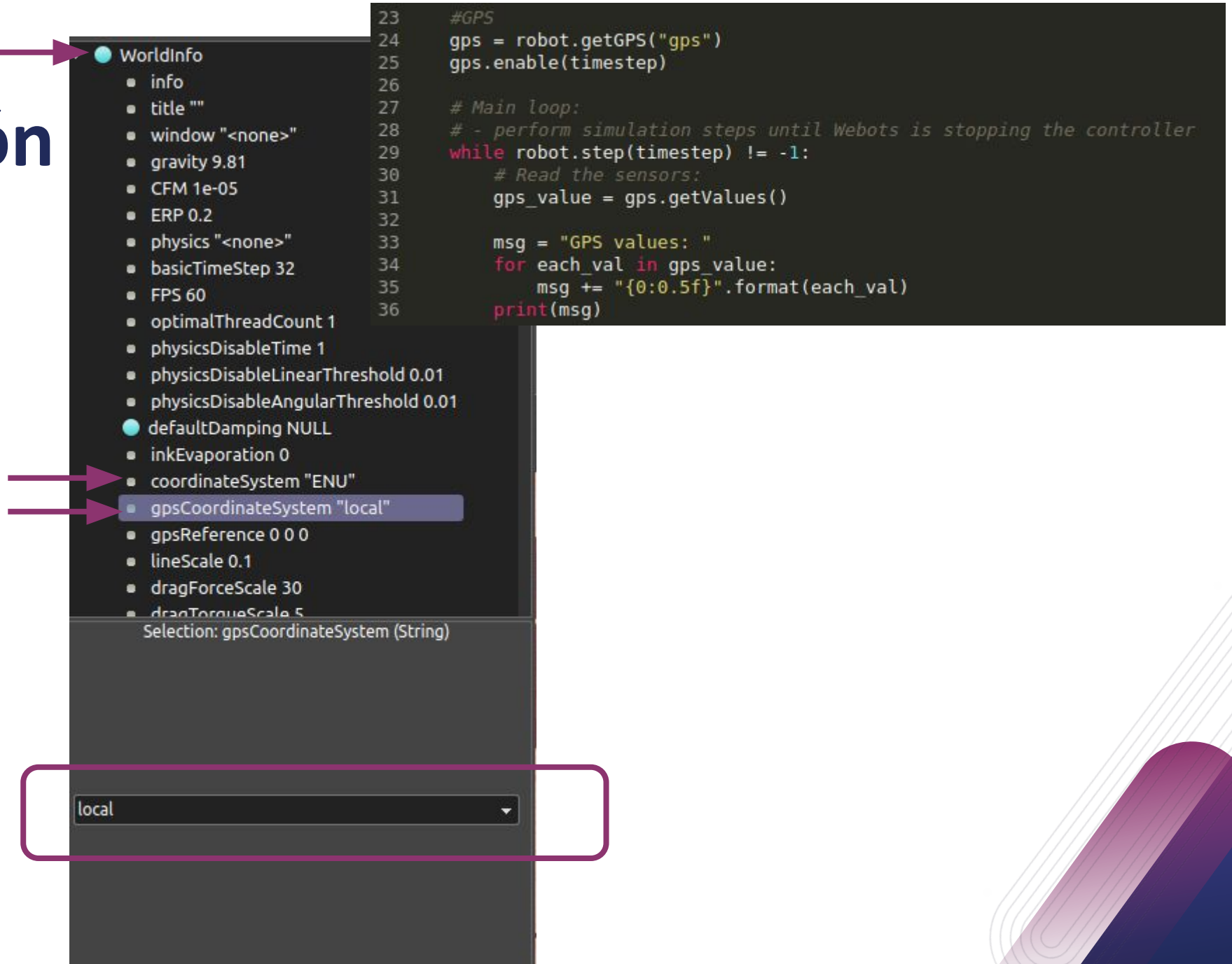
https://www.septentrio.com/en/learn-more/insights/role-gnss-localization-safe-assisted-driving

Demostración

- The `coordinateSystem` field indicates the axis convention of the global coordinate system, defining the cartesian and gravity directions. Currently it supports "ENU" (default), "NUE" and "EUN". "ENU" means East along the X-positive axis, North along the Y-positive axis and Up along the Z-positive axis. It is the most widely used axis convention in robotics, including Webots and ROS. "NUE" means North along the X-positive axis, Up along the Y-positive axis and East along the Z-positive axis. "EUN" means East along the X-positive axis, Up along the Y-positive axis and North along the Z-positive axis. It is similar to "NUE" but with the North and East inverted. Changing the coordinate system will affect the return values of the Accelerometer, Altimeter, Compass, InertialUnit and GPS devices.
- The `gpsCoordinateSystem` field is used to indicate in which coordinate system are expressed the coordinates returned by the GPS devices. If the value is `WGS84` the World Geodetic System `WGS84` and an Universal Transverse Mercatoris projection are used to compute the latitude, altitude and longitude from the cartesian coordinates of the device, otherwise, if the value is `local` the cartesian coordinates of the device are directly returned as is.

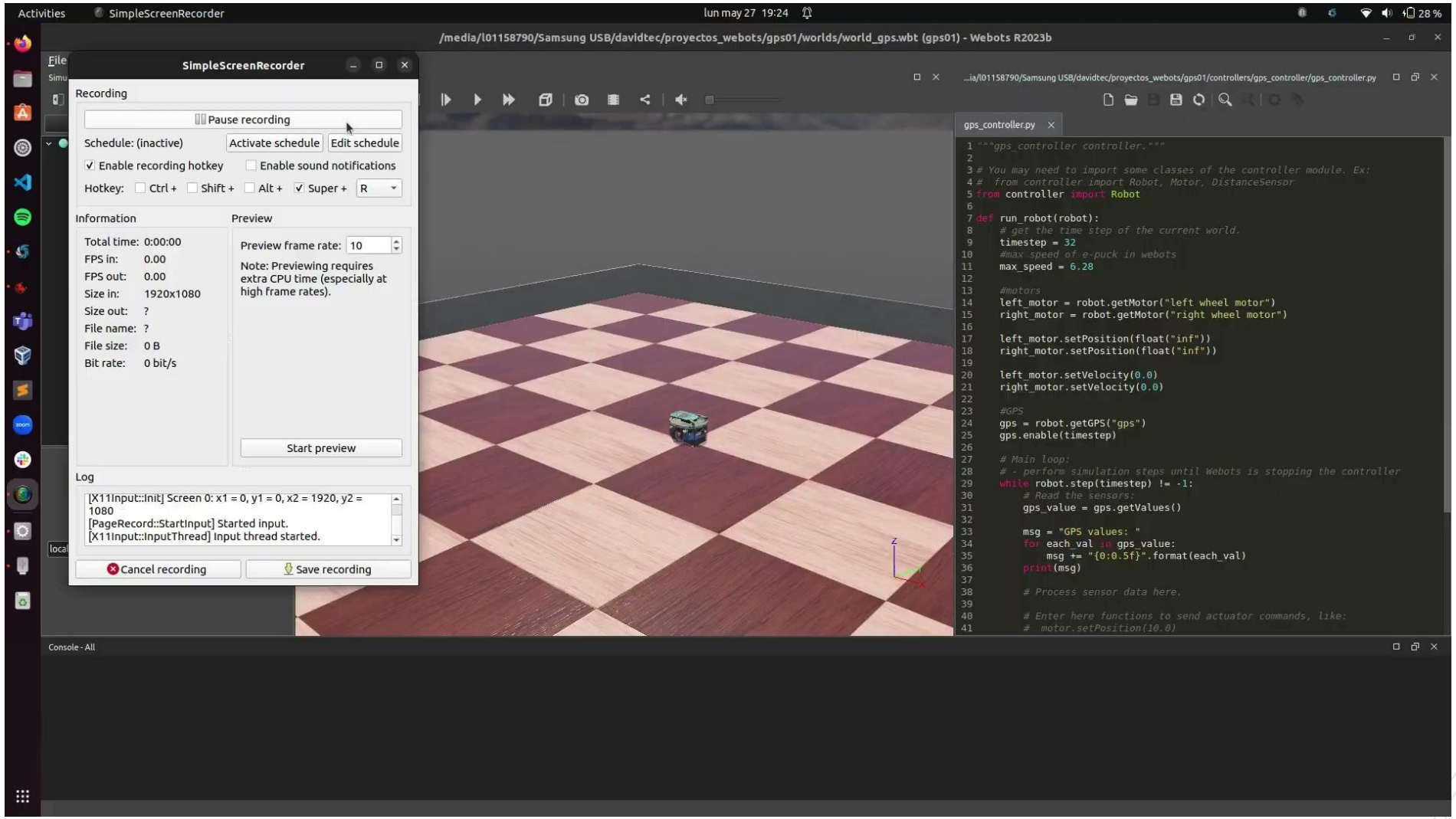
https://cyberbotics.com/doc/reference/worldinfo

Demostración



https://cyberbotics.com/doc/reference/worldinfo

Demostración



Active class

Demostración
Sumo Interface

Active class

Uso de Sensores en el Proyecto Final

Uso de Cámara

- Vamos usar un mundo en Webots sin tráfico ni peatones
- Vamos a grabar imágenes mientras conducimos el vehículo autónomo por el mundo usando el teclado
- Cada imagen guardada deberá tener un nombre particular, a la cual se le va a asociar el ángulo de conducción
- Un archivo CSV mostrará un listado de las imágenes guardadas y el ángulo de conducción correspondiente
- Esta información será usada para entrenar una CNN
- El modelo resultante será usado para lograr una conducción autónoma

Uso de Otros Sensores

- Vamos a usar un mundo en Webots con tráfico y con peatones
- Los peatones (al menos 3) los vamos a agregar manualmente con sus respectivos controladores en Python
- El tráfico va a ser manejado por SUMO de manera automática
- Vamos a agregar sensores al vehículo autónomo para que detecte peatones y vehículos
- Si se detecta peatón, el vehículo autónomo se detiene hasta que concluya la detección
- Si se detecta un vehículo, el vehículo autónomo continúa conduciendo a una cierta distancia (definida en código) evitando colisión; si la distancia se vuelve menor a la definida, el vehículo autónomo se debe detener

Active class

Referencias

Referencias

- https://youngzine.org/article-expert/technology/machines-sense-think-and-act
- https://www.spinny.com/blog/index.php/adas-full-form/
- https://cyberbotics.com/doc/automobile/proto-nodes
- https://dewesoft.com/blog/types-of-adas-sensors
- https://caradas.com/adas-cameras/
- https://loconav.com/blog/lidar/
- https://www.youtube.com/watch?app=desktop&v=pODfyPpXr38
- https://learn.sparkfun.com/tutorials/accelerometer-basics/all
- https://learn.sparkfun.com/tutorials/gyroscope?_ga=2.195309738.765125581.1713898726-2034140019.1713898724
- https://www.youngwonks.com/blog/What-is-a-Magnetometer-and-How-Does-It-Work
- https://www.spiceworks.com/tech/iot/articles/what-is-gnss/
- https://cyberbotics.com/doc/reference/worldinfo
- https://sumo.dlr.de/docs/index.html

Tecnológico de Monterrey

D.R.© Tecnológico de Monterrey, México, 2023.
Prohibida la reproducción total o parcial
de esta obra sin expresa autorización del
Tecnológico de Monterrey.