



Introducción a robótica y a Duckietown Gym

Nociones básicas de Robótica Móvil

25 de agosto 2021

Informaciones generales

Informaciones generales

- Credenciales

Informaciones generales

- Credenciales
- Comunicaciones
 - Discord -> capacitaciones_2021
 - Respetar los horarios de consultas

Informaciones generales

- Credenciales
- Comunicaciones
 - Discord -> capacitaciones_2021
 - Respetar los horarios de consultas
- Horario de consultas
 - **Oficial:** Viernes a las 16:15

Informaciones generales

- Credenciales
- Comunicaciones
 - Discord
 - Respetar los horarios de consultas
- Horario de consultas
 - **Oficial:** Viernes a las 16:15
 - **Extra oficial:** Lunes a las 16:15

Informaciones generales

- Credenciales
- Comunicaciones
 - Discord
 - Respetar los horarios de consultas
- Horario de consultas
 - **Oficial:** Viernes a las 16:00
 - **Extra oficial:** Lunes a las 16:00*

Bitácoras



Nicolás Fredes

Training Manager

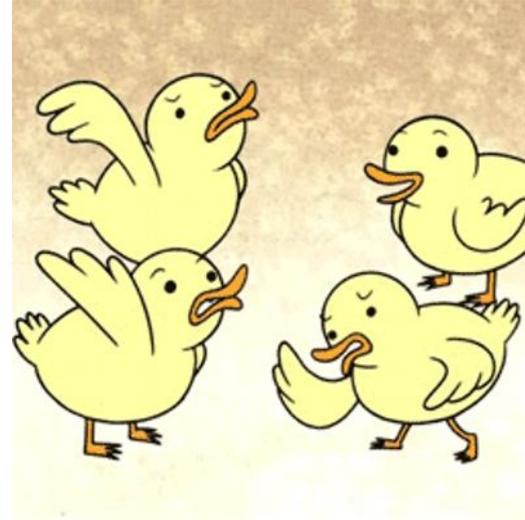
nicolas@ccuac.cl

¿Para qué una bitácora?

- Seguimiento del trabajo desarrollado
- Aprendizajes y dificultades
- Desafíos semanales

¿Cuál es el foco?

- Según tus progresos y dificultades, podremos ir trabajando junto al ED en avanzar de manera integral en todos los aspectos del curso y ver tus logros y autogestión a lo largo del semestre



Bitácora 1

- Entrega por U-cursos (Día anterior a la próxima capacitación)
 - Martes 23:59 hrs
- Preguntas guías en Discord
- Documento libre (Word, Pdf)
- Individual



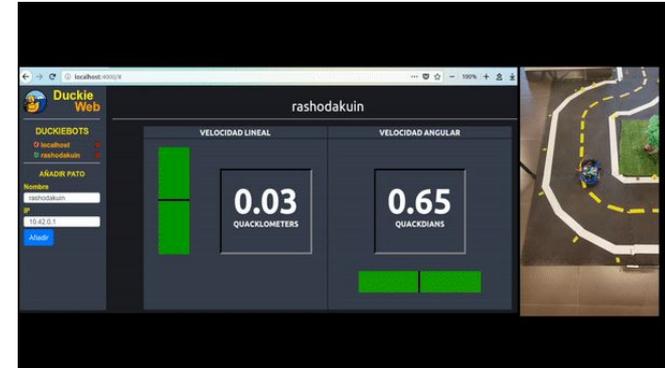
Próximas Bitácoras

- Semanales e individual
- Después de capacitación “*Herramientas de Software*” entrega en wiki-Git
- Bitácoras por grupo cuando comienzan los proyectos
- Nuevo formato cuando comienzan los proyectos

Proyectos realizados

Monitoreo web de Duckiegyim

- **Tutor/a:** Miguel Campusano *
- **Número de personas:** 3 (1 equipo)
- **Requerimientos:**
 - Aprender a desarrollar páginas web.
 - Uso de Python, servidores web y websockets.
 - Adaptar monitor de Duckietown para monitorear Duckiegyim.
 - Analizar necesidades actuales y futuras de la plataforma



mcampus@dcc.uchile.cl

localhost:4000

Duckie Web

rashodakuin

DUCKIEBOTS

- localhost
- rashodakuin

AÑADIR PATO

Nombre
rashodakuin

IP
10.42.0.1

Añadir

VELOCIDAD LINEAL

0.03
QUACKLOMETERS

VELOCIDAD ANGULAR

0.65
QUACKDIANS

0.03
0.65

Duckiegym User Interface (DUI)

- **Tutor/a:** Miguel Campusano, Rodrigo Delgado, Lukas Pavez, Jose Astorga*
- **Número de personas:** 3 (1 equipo)
- **Requerimientos:**
 - Aprender a desarrollar interfaz de usuario en Python.
 - Tomar requisitos de usuario.
 - Uso de Python y alguna herramienta para interfaz (e.g., PyQt).
 - Conectar herramientas de visión y cámaras con interfaz.
 - Conectar herramientas de control remoto con interfaz.
 - Conectar información de Duckiegym con interfaz.

mcampusa@dcc.uchile.cl





Introducción a robótica y a Duckietown Gym

Nociones básicas de Robótica Móvil

25 de agosto 2021



Lukas Pavez

Chief Human Resources Officer

lukas@ccuac.cl



Mattias Prieto

Duckielearning Manager

mattias@duckietown.cl

Capacitación de hoy

- Introducción a robótica
- Presentación Duckiebot
- Simulador duckietown-gym
- Tutorial entorno del simulador
- Misión Semanal 1

Introducción: Robótica

¿Qué es la Robótica Móvil?

Rama de la Ingeniería que estudia los robots capaces de **moverse** en su entorno (que no están fijados a un punto en el espacio)





¿Qué es un Robot?



¿Qué es un Robot?

Máquina **autónoma** que mediante el uso de **sensores y actuadores**, es capaz de interactuar con su entorno

¿AUTONOMÍA?

“Los seres vivos son autónomos, en los que su autonomía se da en su autorreferencia y son sistemas cerrados en su dinámica de constitución como sistemas en continua producción de sí mismos.”

F. Varela, H. Maturana

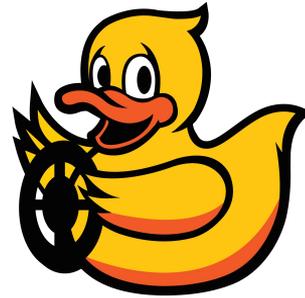




Autonomía

“Capacidad de una entidad a reaccionar a un estímulo externo según sus propias reglas y componentes”

Nosotros



¿Qué es un Robot?

Máquina **autónoma** que mediante el uso de **sensores y actuadores**, es capaz de interactuar con su entorno

Sensor



Procesador

Actuador



Algunos robots reales

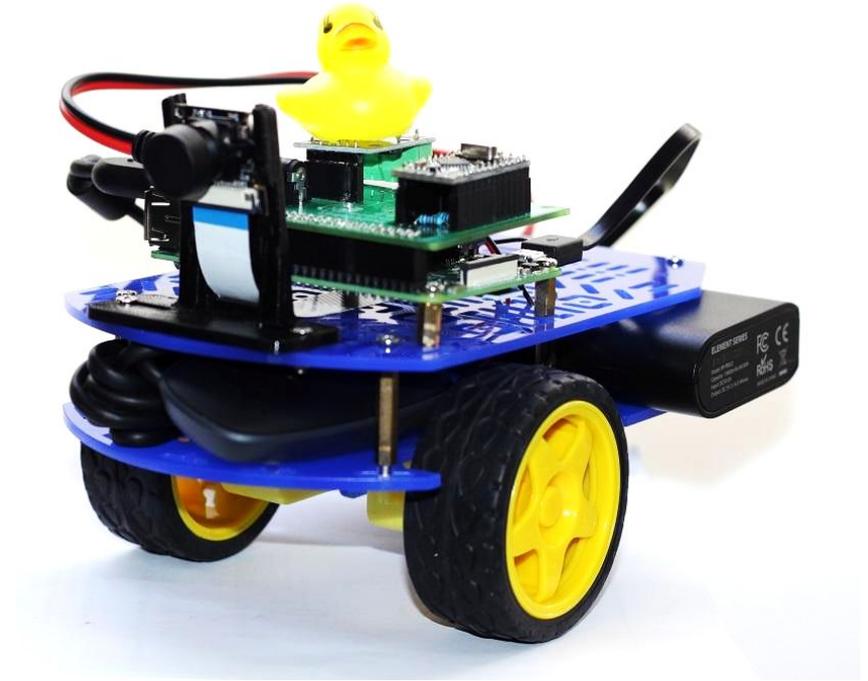
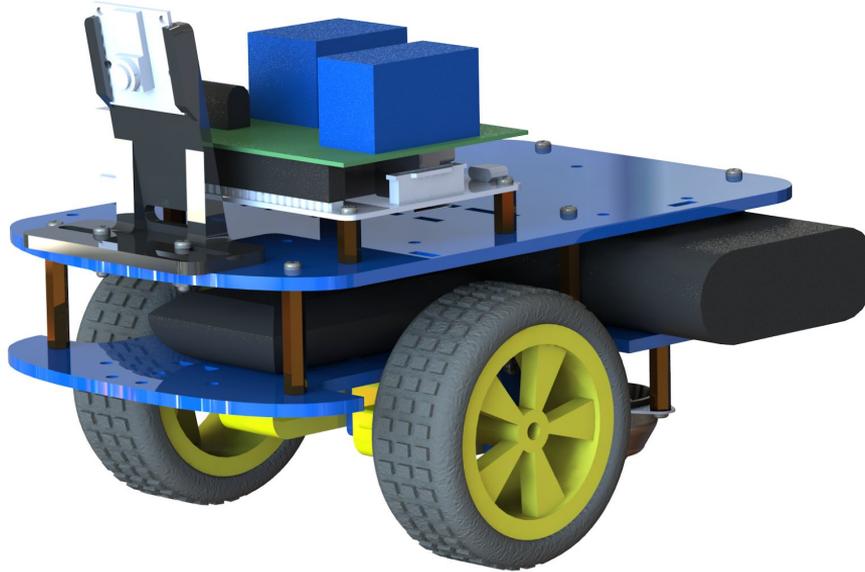




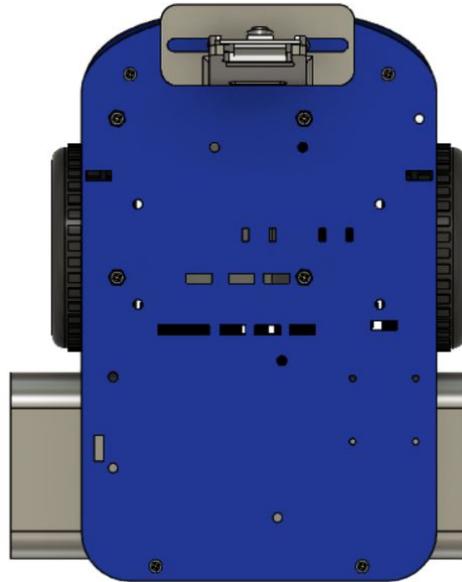
David Szymanski

WHERE'S WALL·E?

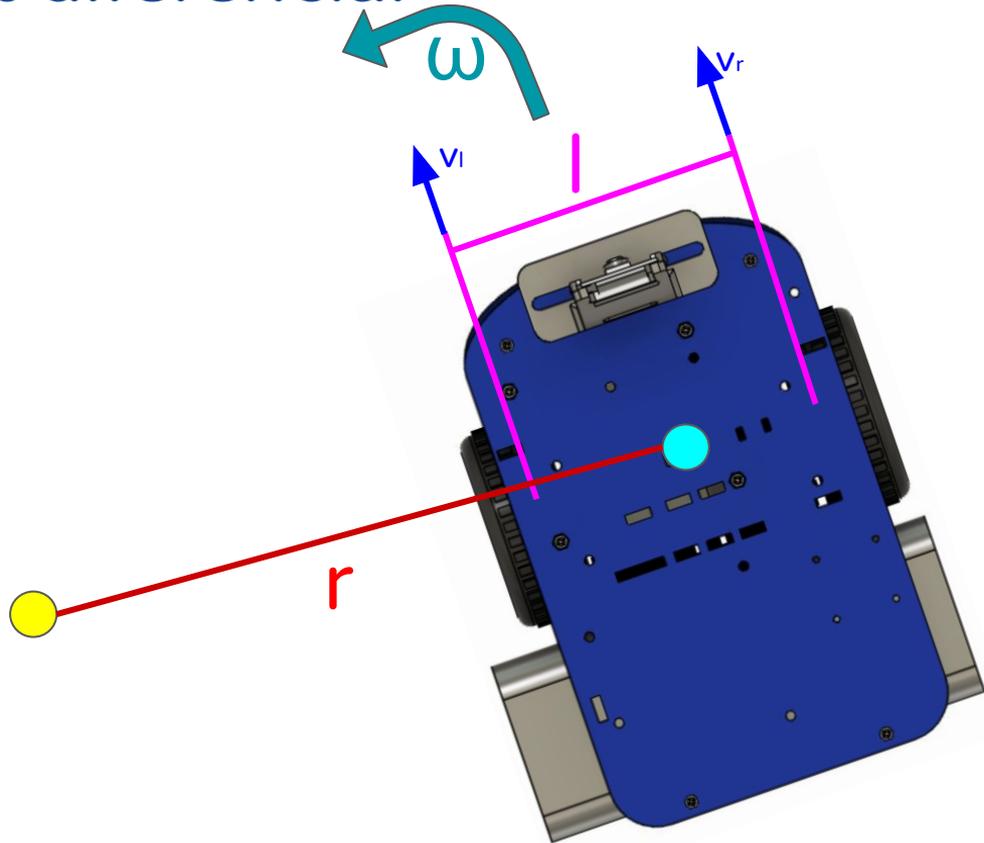
Duckiebot Mark 3



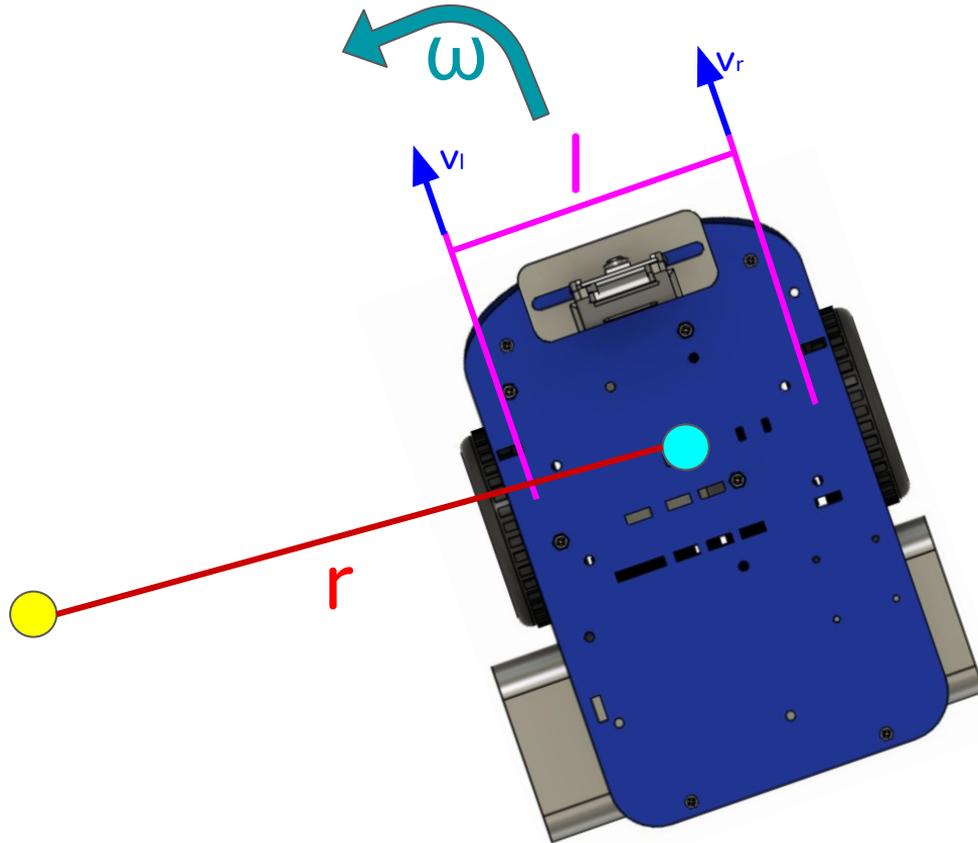
Robot diferencial



Robot diferencial



Robot diferencial

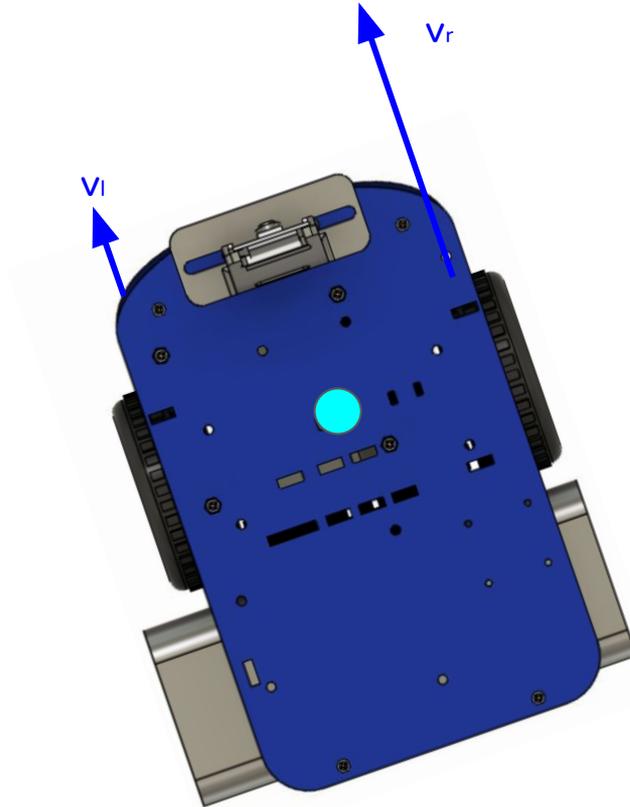


$$\omega = \frac{v_r - v_l}{l}$$

$$r = \frac{l(v_r + v_l)}{2(v_r - v_l)}$$

Robot diferencial

¿Qué pasa si una rueda gira más rápido que la otra?



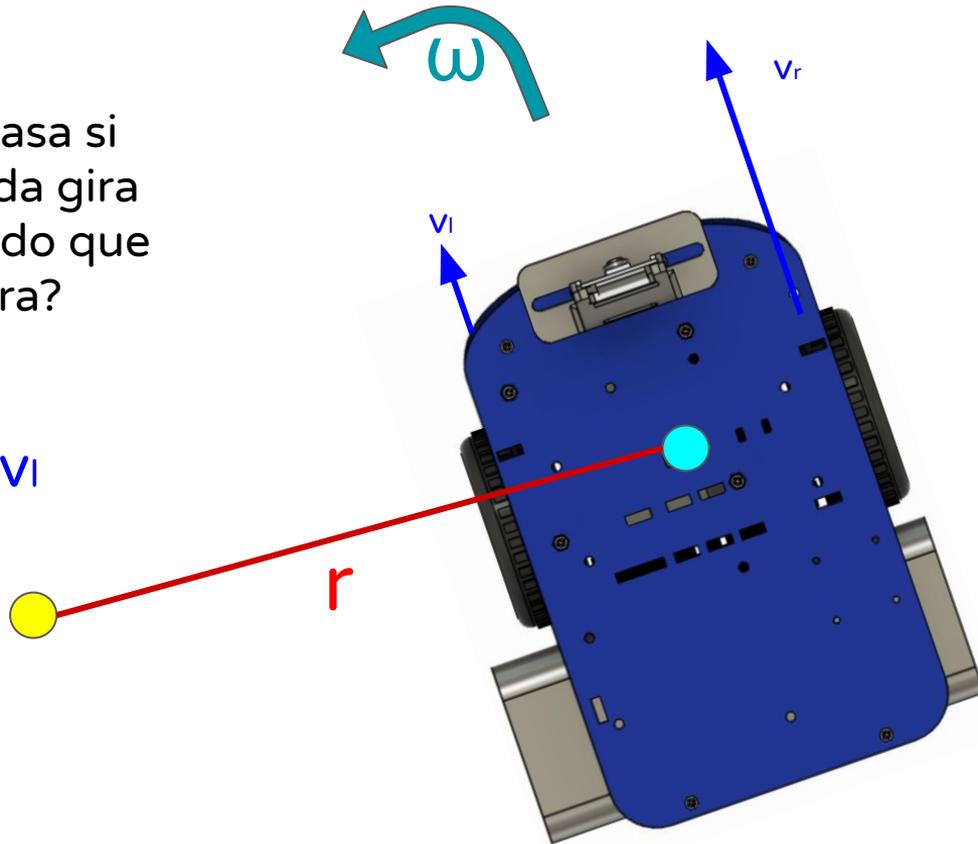
$$\omega = \frac{v_r - v_l}{l}$$

$$r = \frac{l(v_r + v_l)}{2(v_r - v_l)}$$

Robot diferencial

¿Qué pasa si una rueda gira más rápido que la otra?

$$v_r > v_l$$

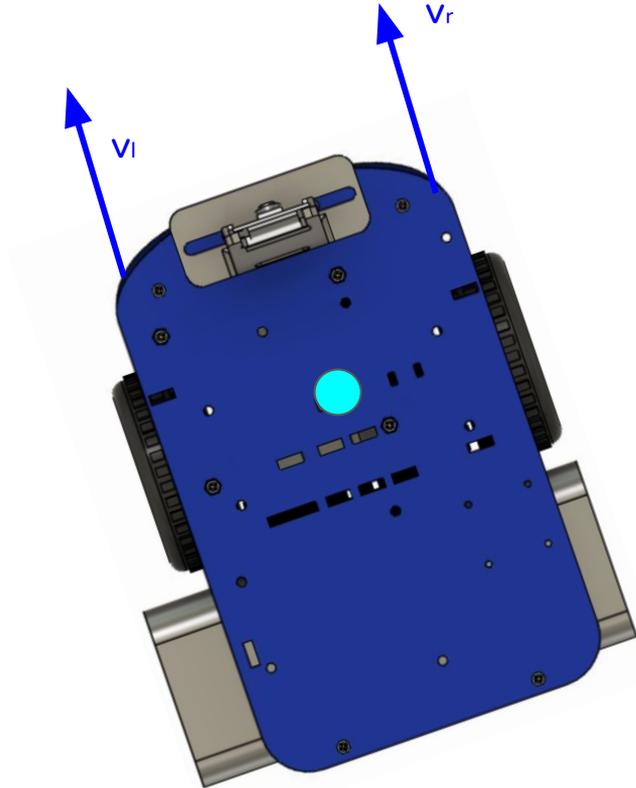


$$\omega = \frac{v_r - v_l}{l}$$

$$r = \frac{l(v_r + v_l)}{2(v_r - v_l)}$$

Robot diferencial

¿Qué pasa si
ambas ruedas
giran a la
misma
velocidad?



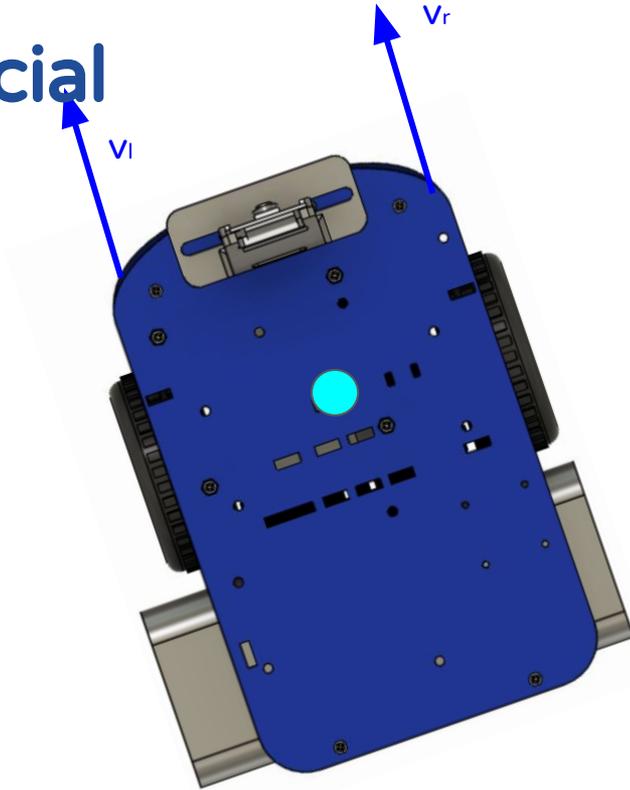
$$\omega = \frac{v_r - v_l}{l}$$

$$r = \frac{l(v_r + v_l)}{2(v_r - v_l)}$$

Robot diferencial

¿Qué pasa si
ambas ruedas
giran a la
misma
velocidad?

$$v_r = v_l$$

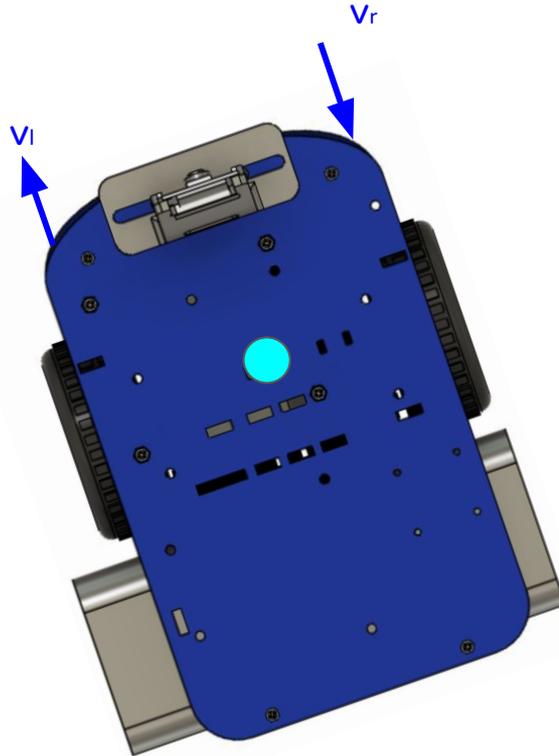


$$\omega = \frac{v_r - v_l}{l} = 0$$

$$r = \frac{l(v_r + v_l)}{2(v_r - v_l)}$$

Robot diferencial

¿Qué pasa si giran a la misma velocidad en sentido contrario?



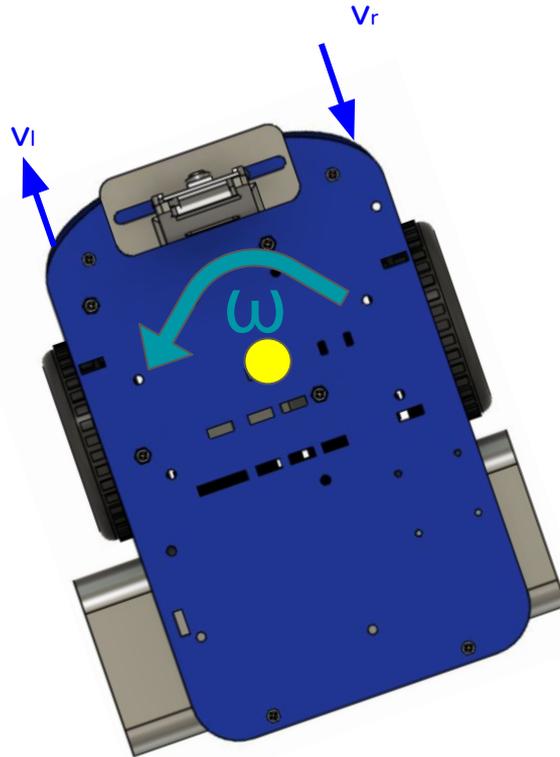
$$\omega = \frac{v_r - v_l}{l}$$

$$r = \frac{l(v_r + v_l)}{2(v_r - v_l)}$$

Robot diferencial

¿Qué pasa si giran a la misma velocidad en sentido contrario?

$$v_r = -v_l$$



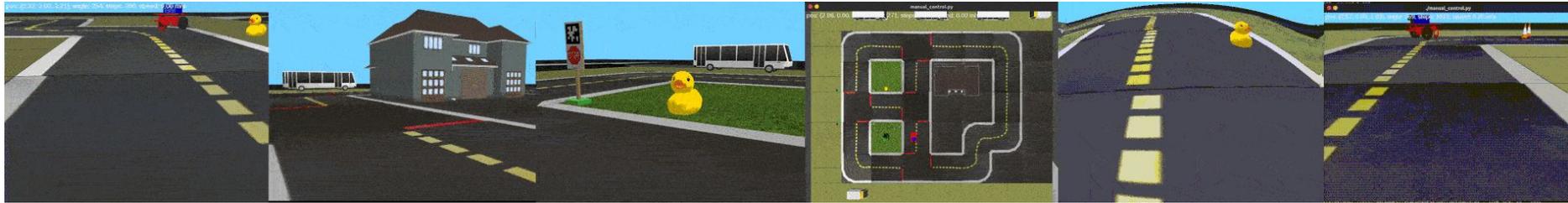
$$\omega = \frac{v_r - v_l}{l}$$

$$r = \frac{l(v_r + v_l)}{2(v_r - v_l)} = 0$$

Simulador

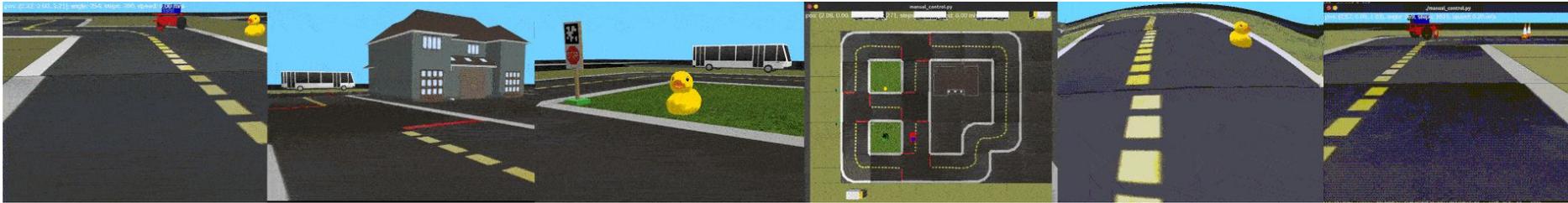
Duckietown Gym

- Simulador para el universo Duckietown



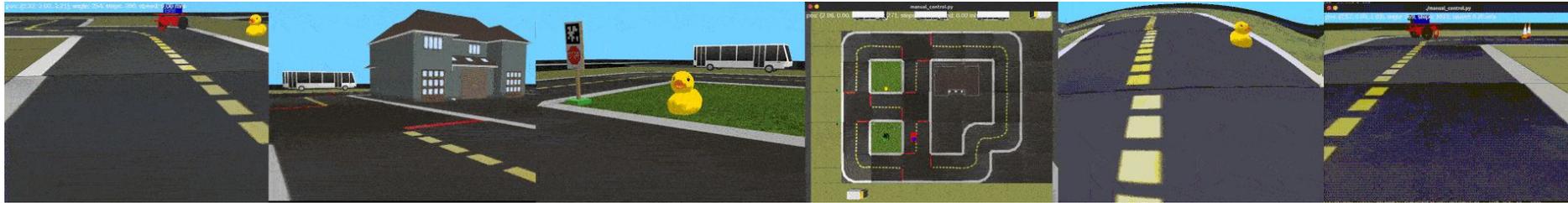
Duckietown Gym

- Simulador para el universo Duckietown
- Calles, casas, patos, semáforos, árboles, duckiebots, intersecciones, etc.



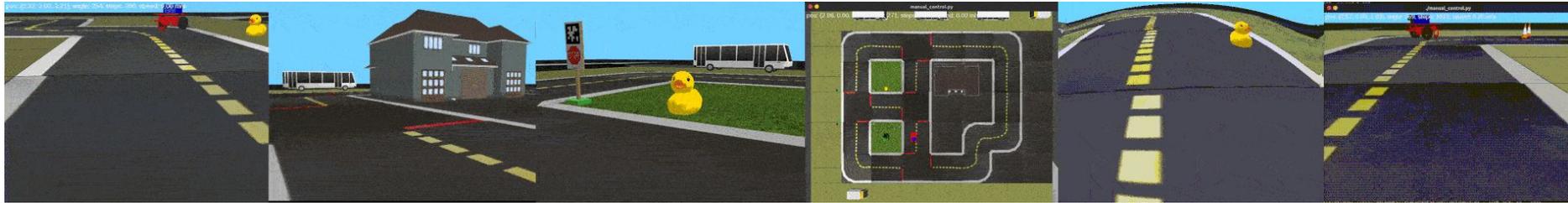
Duckietown Gym

- Simulador para el universo Duckietown
- Calles, casas, patos, semáforos, árboles, duckiebots, intersecciones, etc.
- Rápido, Open Source y personalizable



Duckietown Gym

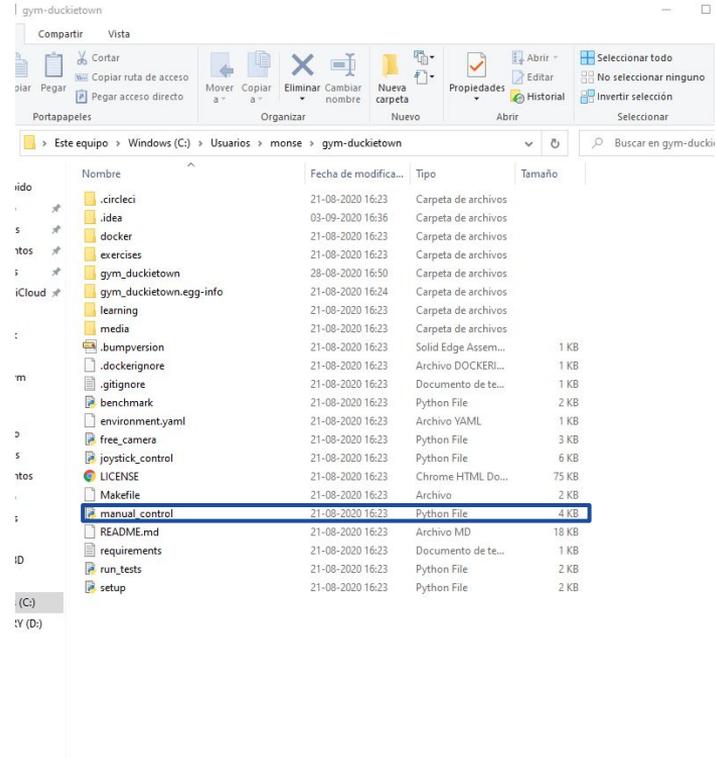
- Simulador para el universo Duckietown
- Calles, casas, patos, semáforos, árboles, duckiebots, intersecciones, etc.
- Rápido, Open Source y personalizable
- Permite realizar las mismas acciones que el Duckiebot



**¿Cómo modificar un mapa del
simulador?**

Archivos importantes*

- `manual_control.py`: corre el simulador.



Carpetas importantes*

Nombre	Fecha de modifica...	Tipo	Tamaño
.idea	28-08-2020 17:07	Carpeta de archivos	
__pycache__	21-08-2020 16:51	Carpeta de archivos	
envs	21-08-2020 16:51	Carpeta de archivos	
maps	21-08-2020 16:23	Carpeta de archivos	
meshes	21-08-2020 16:23	Carpeta de archivos	
randomization	21-08-2020 16:51	Carpeta de archivos	
textures	21-08-2020 16:23	Carpeta de archivos	
__init__	21-08-2020 16:23	Python File	2 KB
collision	21-08-2020 16:23	Python File	8 KB
config	21-08-2020 16:23	Python File	1 KB
distortion	21-08-2020 16:23	Python File	6 KB
graphics	21-08-2020 16:23	Python File	8 KB
launcher	21-08-2020 16:23	Python File	3 KB
objects	21-08-2020 16:23	Python File	14 KB
objmesh	21-08-2020 16:23	Python File	10 KB
requirements	21-08-2020 16:23	Documento de te...	0 KB
simulator	21-08-2020 16:23	Python File	59 KB
utils	21-08-2020 16:23	Python File	2 KB
wrappers	21-08-2020 16:23	Python File	7 KB

- maps: contiene los mapas.
- meshes: contiene los objetos que aparecen en el mapa.
- textures: contiene las baldosas (suelo) del mapa.

maps

2 partes:

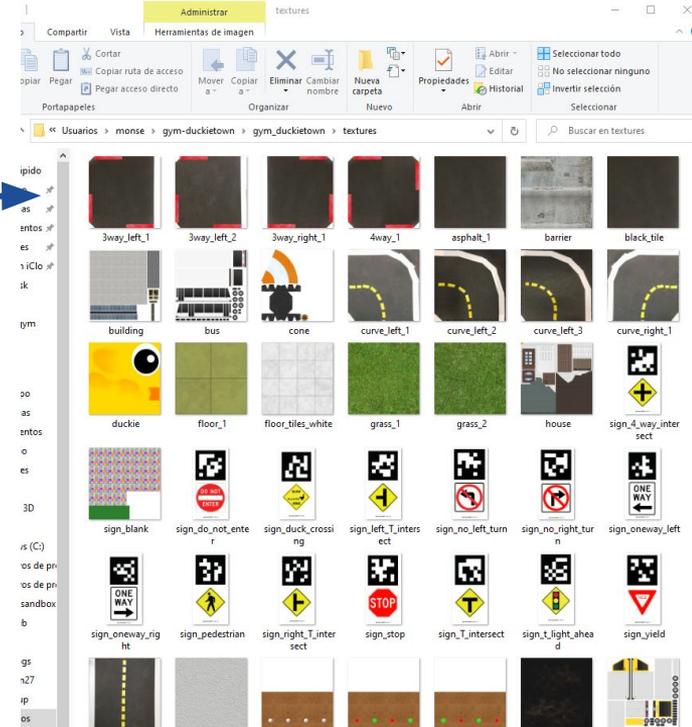
- tiles
- objects

```
manual_control.py × objects.py × udem1.yaml ×
1 | # The following section defines the grid layout.
2 | ●
3 | # Each row is an array of tiles descriptions.
4 | # The top-left tile in this file maps to coordinates (0, 0) in the world
5 | # Each tile contains a string with the kind of tile and its orientation.
6 | # The orientations are North, South, East, West. This is the direction in
7 | # which the agent is expected to drive when going onto this tile.
8 | tiles:
9 | - [floor, floor, floor, floor, floor, floor, floor, floor]
10 | - [floor, curve_left/W, straight/W, 3way_left/W, straight/W, straight/W, curve_left/N, asphalt]
11 | - [floor, straight/S, grass, straight/N, asphalt, asphalt, straight/N, asphalt]
12 | - [floor, 3way_left/S, straight/W, 3way_left/N, asphalt, asphalt, straight/N, asphalt]
13 | - [floor, straight/S, grass, straight/N, asphalt, curve_right/N, curve_left/E, asphalt]
14 | - [floor, curve_left/S, straight/E, 3way_left/E, straight/E, curve_left/E, asphalt, asphalt]
15 | - [floor, floor, floor, floor, floor, floor, floor, floor]
16 |
17 | # The following is a listing of 3D objects placed on the map
18 | #
19 | # pos: object positions are scaled according to the tile size, so
20 | # that (1.5, 2.5) would be exactly in the middle of tile (1, 2)
21 | # height: Objects can be scaled according to their real-world height in meters
22 | # Duckietown tiles are 61cm, or 0.61 meter
23 | # rotate: the rotation angle determine the object's orientation
24 | # a positive rotation angle corresponds to a counter-clockwise
25 | # (leftward) rotation. The angles are in degrees.
26 | # Models should, at angle 0, face towards the positive X axis.
27 | objects:
28 |
29 | - kind: tree
30 |   pos: [2.5, 4.5]
31 |   rotate: 180
32 |   height: 0.25
33 |   optional: true
34 |
35 | - kind: duckie
36 |   pos: [2.5, 2.9]
37 |   rotate: -90
38 |   height: 0.08
39 |   optional: true
40 |
41 | - kind: sign_stop
42 |   pos: [2.08, 4.05]
43 |   rotate: 90
44 |   height: 0.18
45 | - kind: sign_left_T_intersect
46 |   pos: [0.94, 3.96]
47 |   rotate: 90
48 |   height: 0.18
49 | - kind: sign_stop
```



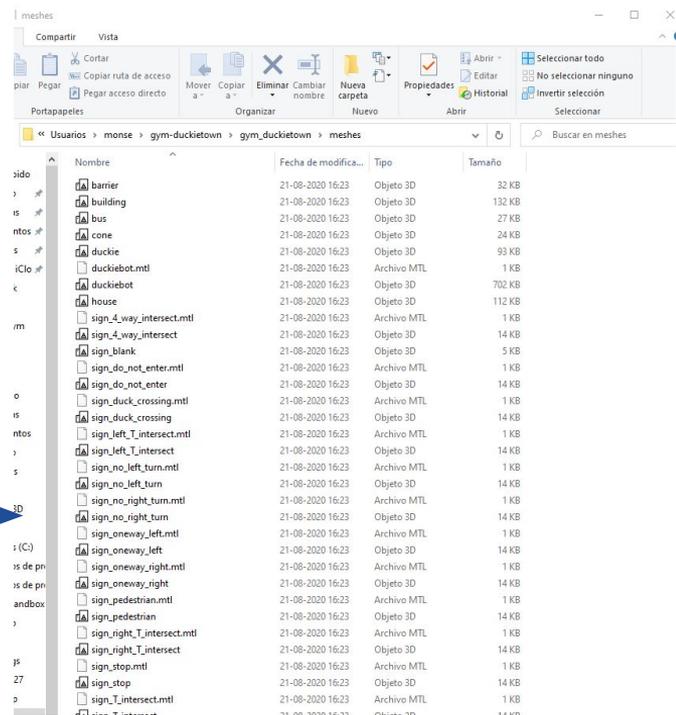
maps

```
manual_control.py | objects.py | udem1.yaml
1  # The following section defines the grid layout.
2
3  # Each row is an array of tiles descriptions.
4  # The top-left tile in this file maps to coordinates (0, 0) in the world
5  # Each tile contains a string with the kind of tile and its orientation.
6  # The orientations are North, South, East, West. This is the direction in
7  # which the agent is expected to drive when going onto this tile.
8
9  tiles:
10 - [floor, floor, floor, floor, floor, floor, floor, floor, floor]
11 - [floor, curve_left/W, straight/W, 3way_left/W, straight/W, straight/W, curve_left/N, asphalt]
12 - [floor, straight/S, grass, straight/N, asphalt, asphalt, straight/W, asphalt]
13 - [floor, 3way_left/S, straight/W, 3way_left/N, asphalt, asphalt, straight/W, asphalt]
14 - [floor, straight/S, grass, straight/N, asphalt, curve_right/N, curve_left/E, asphalt]
15 - [floor, curve_left/S, straight/E, 3way_left/E, straight/E, curve_left/E, asphalt, asphalt]
16 - [floor, floor, floor, floor, floor, floor, floor, floor, floor]
17
18 # The following is a listing of 3D objects placed on the map
19
20 # pos: object positions are scaled according to the tile size, so
21 # that (1.0, 0.5) would be exactly in the middle of tile (1, 2)
22 # height: Objects can be scaled according to their real-world height in meters
23 # Duckietown tiles are 61cm, or 0.61 meter
24 # rotate: the rotation angle determine the object's orientation
25 # a positive rotation angle corresponds to a counter-clockwise
26 # (leftward) rotation. The angles are in degrees.
27 # Models should, at angle 0, face towards the positive X axis.
28
29 objects:
30
31 - kind: tree
32   pos: [2.5, 4.5]
33   rotate: 180
34   height: 0.25
35   optional: true
36
37 - kind: duckie
38   pos: [2.5, 2.9]
39   rotate: -90
40   height: 0.08
41   optional: true
42
43 - kind: sign_stop
44   pos: [2.08, 4.05]
45   rotate: 90
46   height: 0.18
47
48 - kind: sign_left_T_intersect
49   pos: [0.94, 3.96]
50   rotate: 90
51   height: 0.18
52
53 - kind: sign_stop
```



maps

```
1 # The following section defines the grid layout.
2
3 # Each row is an array of tiles descriptions.
4 # The top-left tile in this file maps to coordinates (0, 0) in the world
5 # Each tile contains a string with the kind of tile and its orientation.
6 # The orientations are North, South, East, West. This is the direction in
7 # which the agent is expected to drive when going onto this tile.
8
9 tiles:
10 - [floor, floor, floor, floor, floor, floor, floor, floor]
11 - [floor, curve_left/W, straight/W, 3way_left/W, straight/W, straight/W, curve_left/N, asphalt]
12 - [floor, straight/S, grass, straight/N, asphalt, asphalt, straight/N, asphalt]
13 - [floor, 3way_left/S, straight/W, 3way_left/N, asphalt, asphalt, straight/N, asphalt]
14 - [floor, straight/S, grass, straight/N, asphalt, curve_right/N, curve_left/E, asphalt]
15 - [floor, curve_left/S, straight/E, 3way_left/E, straight/E, curve_left/E, asphalt]
16 - [floor, floor, floor, floor, floor, floor, floor, floor]
17
18 # The following is a listing of 3D objects placed on the map
19 #
20 # pos: object positions are scaled according to the tile size, so
21 # that (1.0, 0.5) would be exactly in the middle of tile (1, 2)
22 # height: Objects can be scaled according to their real-world height in meters
23 # Duckietown tiles are 61cm, or 0.61 meter
24 # rotate: the rotation angle determine the object's orientation
25 # a positive rotation angle corresponds to a counter-clockwise
26 # (leftward) rotation. The angles are in degrees.
27 # Models should, at angle 0, face towards the positive X axis.
28
29 objects:
30 - kind: tree
31   pos: [2.5, 4.5]
32   rotate: 180
33   height: 0.25
34   optional: true
35
36 - kind: duckie
37   pos: [2.5, 2.9]
38   rotate: -90
39   height: 0.08
40   optional: true
41
42 - kind: sign_stop
43   pos: [2.08, 4.05]
44   rotate: 90
45   height: 0.18
46 - kind: sign_left_T_intersect
47   pos: [0.94, 3.96]
48   rotate: 90
49   height: 0.18
50 - kind: sign_stop
```



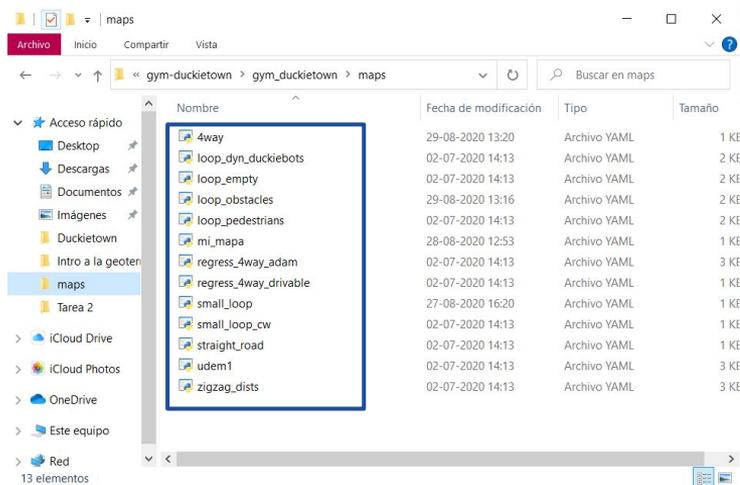
¿Cómo elijo en qué mapa estar?

Configurar mapa

```
manual_control.py  objects.py  udem1.yaml
1 #!/usr/bin/env python
2 # manual
3
4 """
5 This script allows you to manually control the simulator or Duckiebot
6 using the keyboard arrows.
7 """
8
9 import ...
10
11 # from experiments.utils import save_img
12
13
14 parser = argparse.ArgumentParser()
15 parser.add_argument('--map-name', default=None)
16 parser.add_argument('--map-name', default='udem1')
17 parser.add_argument('--distortion', default=False, action='store_true')
18 parser.add_argument('--draw-curve', action='store_true', help='draw the lane following curve')
19 parser.add_argument('--draw-bbox', action='store_true', help='draw collision detection bounding boxes')
20 parser.add_argument('--domain-rand', action='store_true', help='enable domain randomization')
21 parser.add_argument('--frame-skip', default=1, type=int, help='number of frames to skip')
22 parser.add_argument('--seed', default=1, type=int, help='seed')
23 args = parser.parse_args()
24
25 if args.env_name and args.env_name.find('Duckietown') != -1:
26     env = DuckietownEnv(
27         seed = args.seed,
28         map_name = args.map_name,
29         draw_curve = args.draw_curve,
30         draw_bbox = args.draw_bbox,
31         domain_rand = args.domain_rand,
32         frame_skip = args.frame_skip,
33         distortion = args.distortion,
34     )
35 else:
36     env = gym.make(args.env_name)
37
38 env.reset()
39 env.render()
40
41 @env.unwrapped.window.event
42 def on_key_press(symbol, modifiers):
43     """
44     This handler processes keyboard commands that
45     control the simulation
46     """
47     if symbol == key.BACKSPACE or symbol == key.SLASH:
```

En manual_control.py:

Cambiar 'udem1' por uno de los mapas



(demo en vivo)

Misión semanal 1

¡Crea tu propio mapa!

Requisitos mínimos

- 5x3 baldosas
- 1 baldosa de pasto
- 8 patos en diferentes posiciones
- 1 objeto a elección de adorno en el pasto



Referencias

- <https://github.com/duckietown/gym-duckietown>

Para rotar objetos y reducir su tamaño

- <http://www.meshmixer.com/>
- <https://www.onshape.com/>

Para encontrar modelos 3D

- <https://thingiverse.com>



Introducción a robótica y a Duckietown Gym

Nociones básicas de Robótica Móvil

25 de agosto 2021