



Universidad
Rey Juan Carlos



Escuela Técnica Superior de
Ingeniería de Telecomunicación

Mejoras en entorno de robótica educativa para niños

Trabajo de fin de grado

Rubén Álvarez Martín

Índice

1. Introducción

2. Objetivos

3. Herramientas

4. Mejoras a WebSim

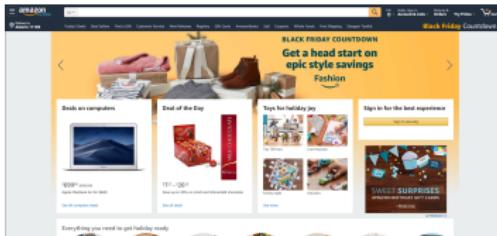
- ▶ Soporte a drones en WebSim
- ▶ Teleoperadores en WebSim
- ▶ Ejercicios individuales
- ▶ Ejercicios competitivos

5. Conclusiones

Introducción

Tecnologías web

- HTTP
- Tecnologías en cliente: HTML, CSS y JavaScript.
- Tecnologías en servidor: Node.js, Django y Spring.



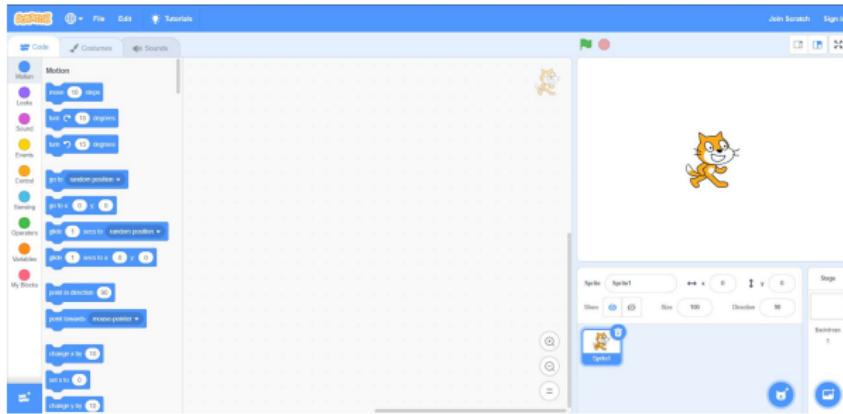
Robótica



- ▶ *robot = hardware + software*
- ▶ *hardware = sensores + actuadores + controladores*

Robótica educativa

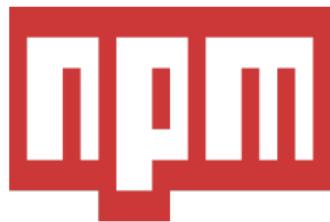
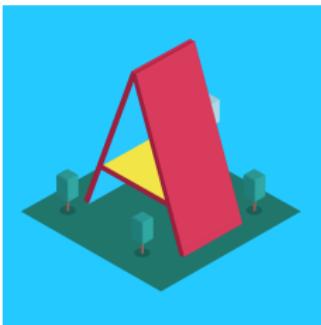
- Lenguajes de programación visual:
Scratch!, LEGO, Kodu, Snap! o Blockly.



- Plataformas hardware: *LEGO mindstorm*, *Makeblock* o *Arduino*.

Herramientas

Herramientas



WebSim



- *WebSim* -> simulador robótico para enseñar conceptos básicos de tecnología, robótica y programación.

Objetivos

Objetivos

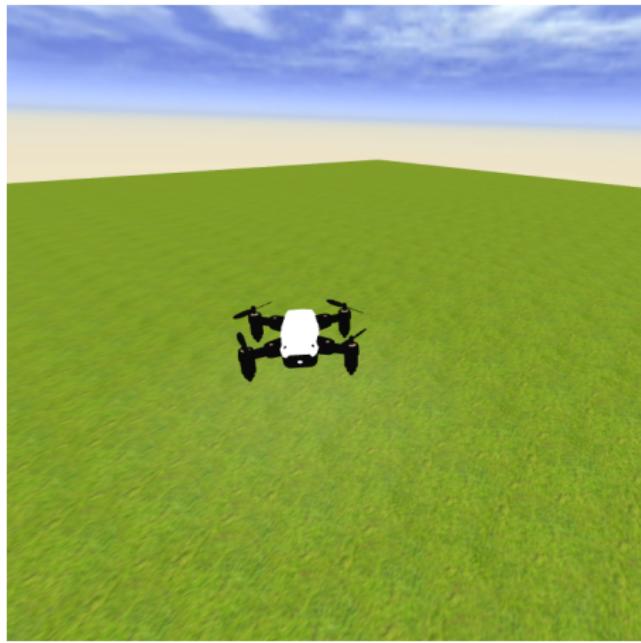
- ▶ Mejorar *WebSim*:
 1. Soporte a *drones*.
 2. Teleoperadores y ficheros de configuración.
 3. Ejercicios individuales.
 4. Ejercicios competitivos y evaluadores automáticos.

Mejoras a WebSim

Soporte a drones: Drivers

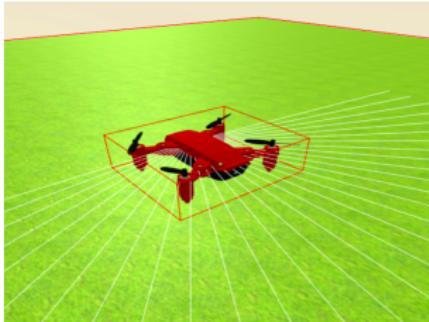
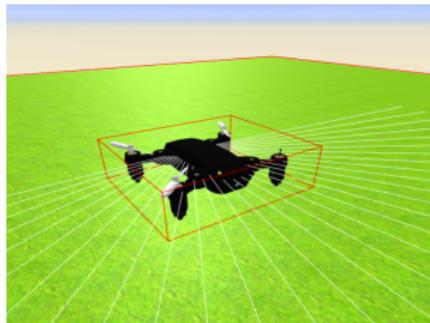
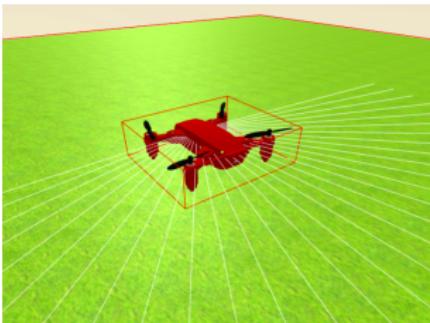
Método	Descripción
.setL(integer)	Comanda velocidad ascendente
.getL()	Devuelve velocidad ascendente
.despegar()	Despega <i>drone</i>
.aterrizar()	Aterriza <i>drone</i>
.move(integer, integer, integer)	Comanda velocidades

Soporte a drones: Modelo 3D



- Formato *A-Frame*
- Modelo *low-poly*
- Animación hélices

Soporte a drones: Modelo 3D



Soporte a drones: Bloques Scratch

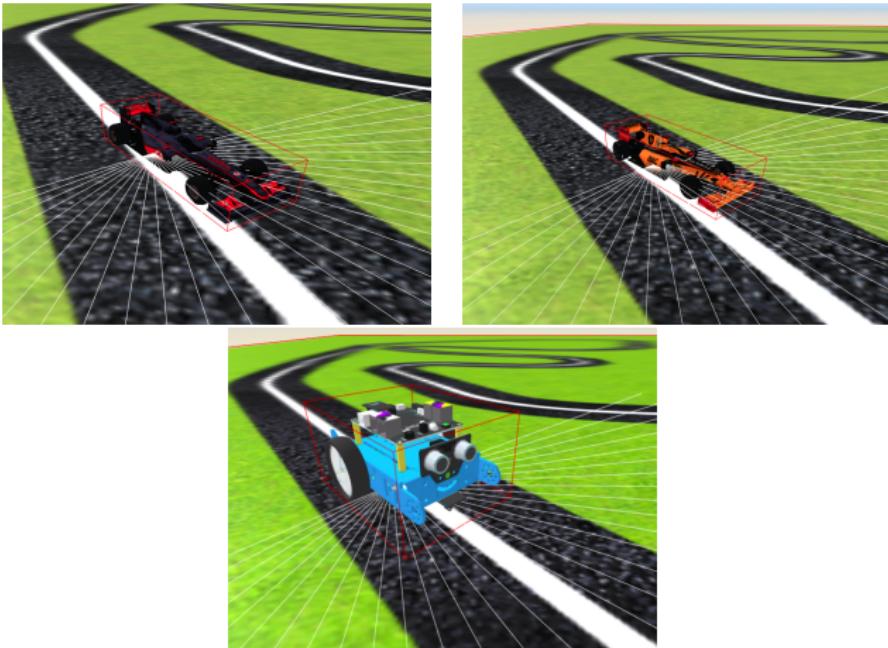
Eleva **myRobot** a velocidad

Obtener la velocidad de elevación de **myRobot**

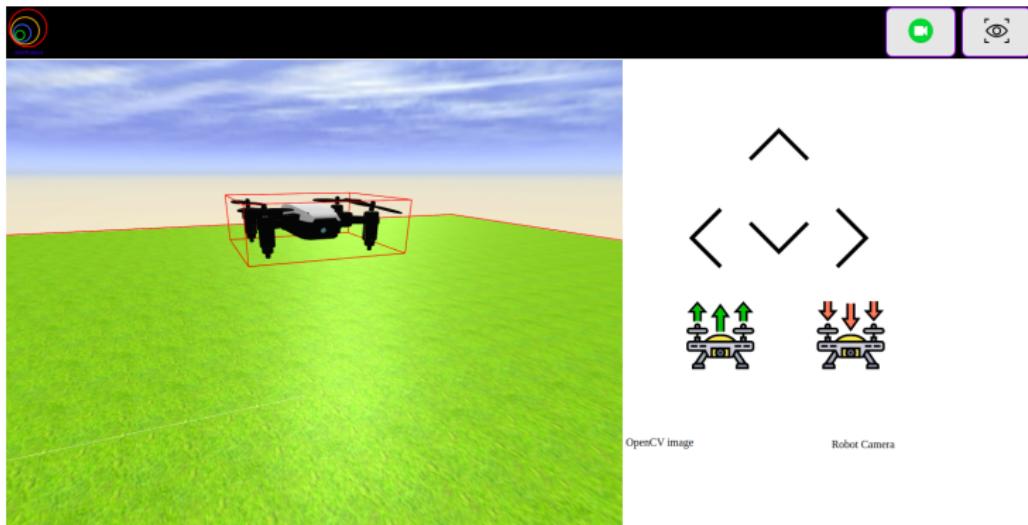
Aterrizar **myRobot**

Despegar **myRobot**

Otros modelos



Teleoperadores



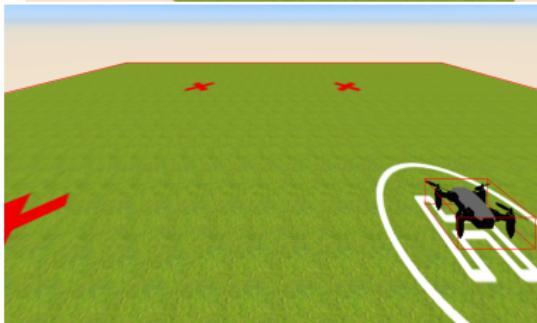
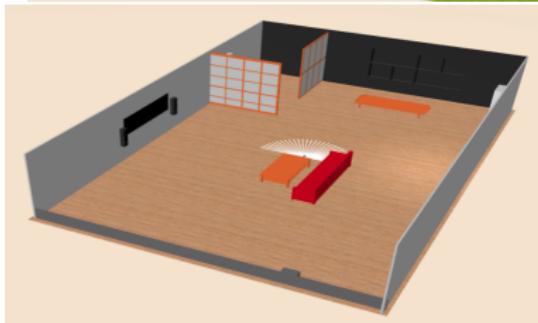
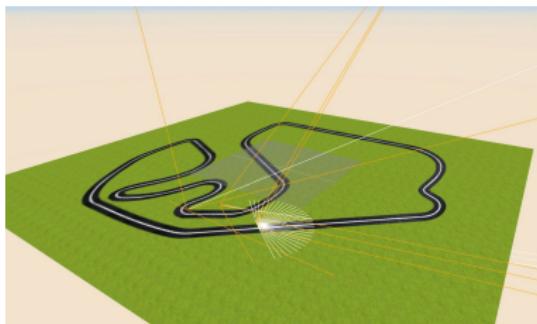
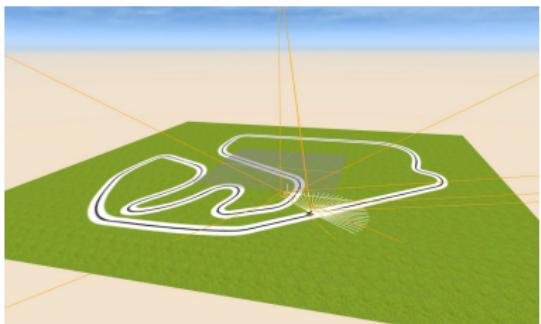
- Posibilidad de controlar los robots sin necesidad de programarlos

Teleoperadores: configuración

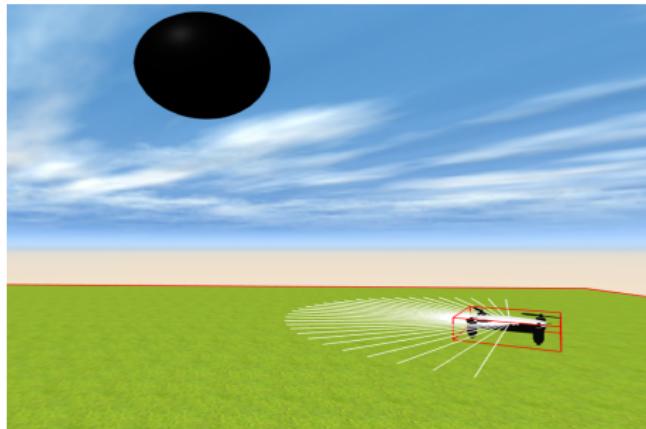
- Ficheros de configuración -> crear escenario sin tener elementos en el código fuente.
- Formato JSON -> escenario, robot, gravedad, elementos, etc.

Ejercicios individuales

Ejercicios individuales

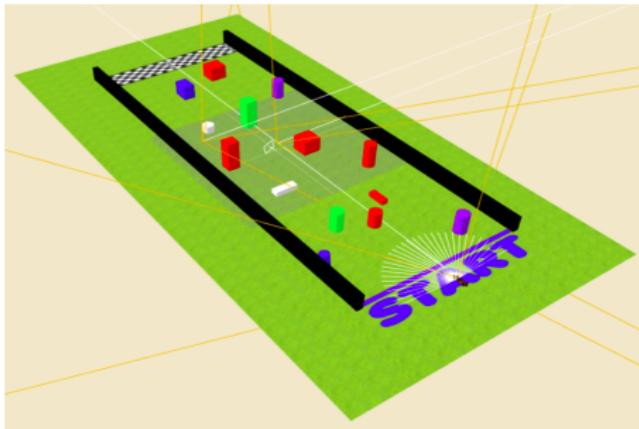


Sigue pelota



```
Bucle infinito
  establecer centroX a De la cámara de [myRobot] obtener "centerX" del objeto de color [black]
  establecer centroY a De la cámara de [myRobot] obtener "centerY" del objeto de color [black]
  establecer area a De la cámara de [myRobot] obtener "area" del objeto de color [black]
  establecer errorX a 65 - centroX * 0.0005
  establecer errorY a centroY - 33 * 0.0005
  si errorX < 0
    hacer Girar [myRobot] a la izquierda a velocidad [-errorX * 0.5]
  sino si errorY < 0
    hacer Girar [myRobot] a la derecha a velocidad [-errorY * 0.5]
  sino
    hacer Girar [myRobot] a la izquierda a velocidad [-0.2]
  si errorX > 0
    hacer Elevar [myRobot] a velocidad [0.2]
  sino si errorY > 0
    hacer Elevar [myRobot] a velocidad [0.5]
  hacer Parar [myRobot]
  si area < 100
    hacer Avanzar [myRobot] a velocidad [1]
  sino si area > 100
    hacer Retroceder [myRobot] a velocidad [1]
```

Atraviesa bosque



```
Bucle infinito establecer distancias [a] Para myRobot [ ] obtener las distancias en arco 180 grados
Avanzar [myRobot] a velocidad [0.25]
Girar [myRobot] a la izquierda a velocidad [0]
establecer menor [a] a [15]
establecer id [a] a [15]
contar con [id] desde [10] hasta [20] de a [1]
hacer
  si [en la lista [distancias] contiene [a] <= [15] & menor [a] >= [15]]
    hacer establecer menor [a] a [en la lista [distancias] contiene [a] <= [15] & menor [a] >= [15]]
    establecer id [a] a [15]
  si menor [a] <= [15]
    hacer Avanzar [myRobot] a velocidad [0.75]
    Girar [myRobot] a la derecha a velocidad [0.1]
  sino si [id] <= [15]
    hacer Avanzar [myRobot] a velocidad [0.1]
    Girar [myRobot] a la izquierda a velocidad [0.05]
  sino si [id] >= [15]
    hacer Avanzar [myRobot] a velocidad [0.1]
    Girar [myRobot] a la derecha a velocidad [0.05]
```

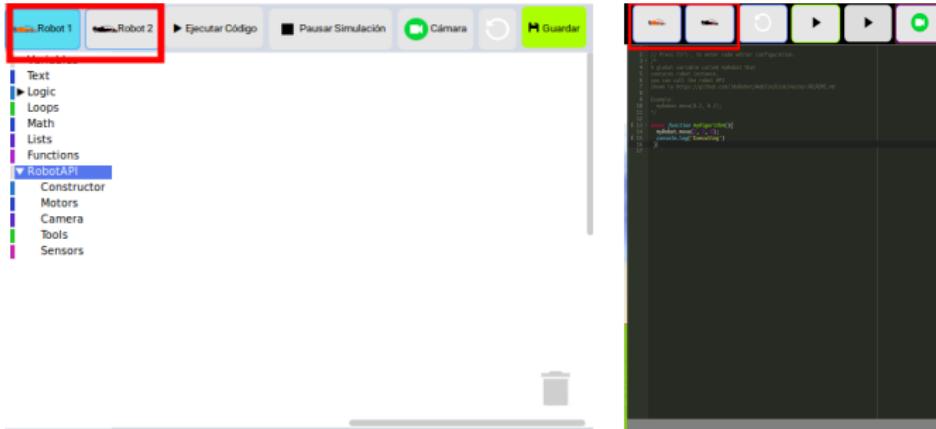
Ejercicios competitivos

Arquitectura de cómputo

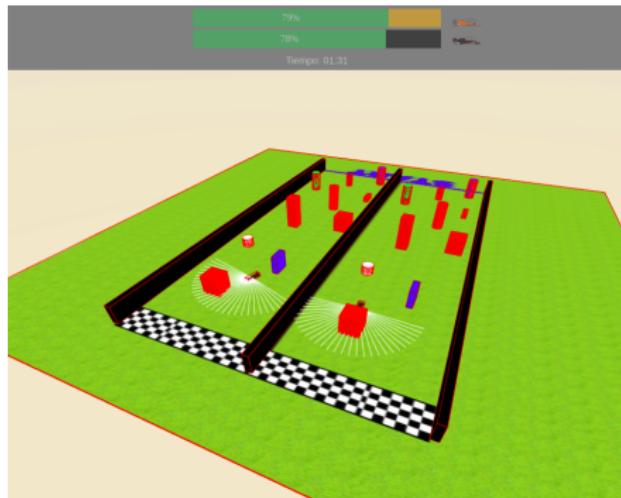
- Módulo ***brains*** ampliado.
- Módulo ***evaluators***:
`runEvaluator(arrayRobots,evaluator)`
- Módulo ***agents***:
`runAgent(idRobot, path)`

Arquitectura de cómputo

► Nuevos editores:



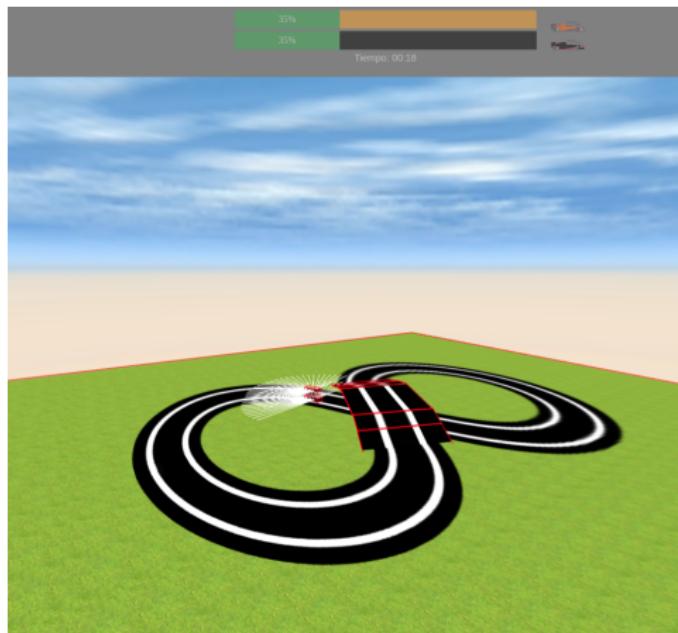
Atraviesa bosque competitivo



- % distancia recorrida
- Mismo recorrido

Sigue líneas visión competitivo

- ▶ % distancia recorrida
- ▶ Puente primitivas
A-Frame



Gato-ratón



- Distancia *drones* en gráfica
- Módulo *agents*

Conclusiones

Conclusiones

- Soporte a *drones*.
- Ficheros de configuración.
- Ejercicios individuales con funcionalidad existente.
- Ejercicios competitivos con nueva funcionalidad.

Líneas futuras

- Nuevos modelos de *robots*.
- Nuevos ejercicios.
- Gravedad.
- *WebWorkers*.
- Control en posición.