UNIVERSIDADE DA CORUÑA

Ejercicios con 

Diseño de arquitecturas subsumidas

Arquitectura Subsumida

- Las arquitecturas reactivas o arquitecturas subsumidas son un enfoque en robótica y sistemas inteligentes que se centra en cómo un robot puede responder de forma rápida y directa a su entorno sin depender tanto de un procesamiento centralizado o de un modelo interno complejo del mundo.
- Este tipo de arquitectura es popular en robótica autónoma y es especialmente útil para robots que deben operar en entornos dinámicos.

Conceptos Básicos

En la arquitectura subsumida, el comportamiento del robot se divide en módulos o capas de comportamiento que operan de manera independiente, cada una enfocada en tareas específicas y de complejidad creciente.

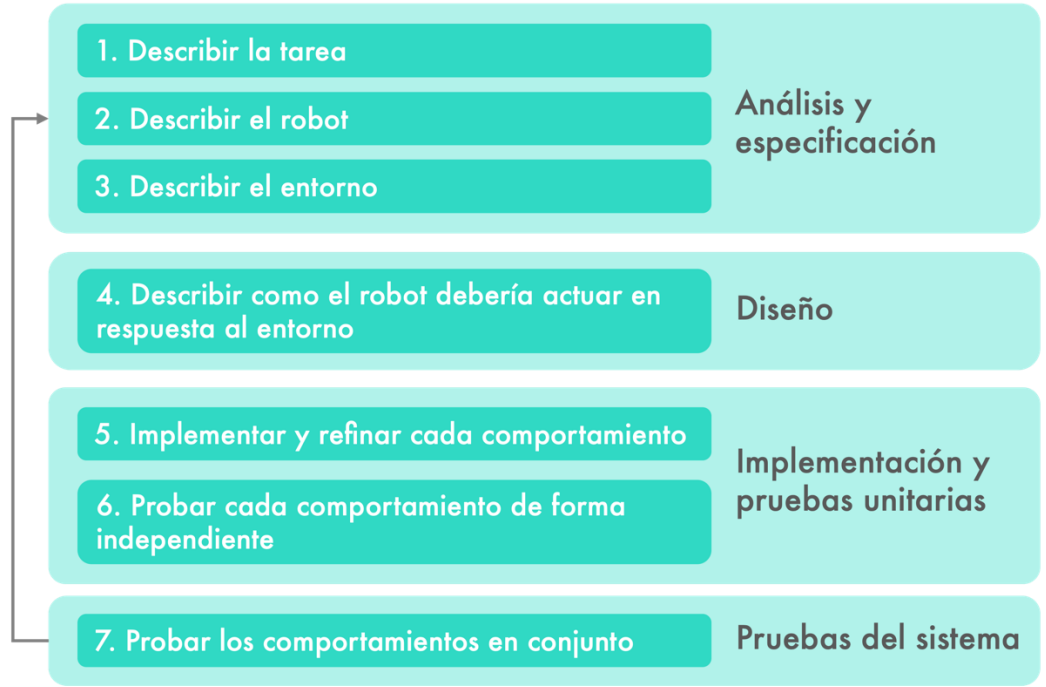
Las ideas clave son:

- Capas Jerárquicas de Comportamiento:
  - Cada capa tiene un objetivo particular y básico, como "evitar obstáculos" o "seguir una meta".
- Prioridad entre Capas:
  - Las capas más básicas y esenciales (por ejemplo, evitar obstáculos) tienen mayor prioridad que las capas de mayor nivel (como explorar un área).
- Acción Basada en Estímulos Directos:
  - Cada capa responde de manera casi inmediata a los estímulos de los sensores, en lugar de pasar por un procesamiento complejo.
- Subsumción:
  - Las capas superiores pueden anular o "subsumir" las inferiores si es necesario. Por ejemplo, una capa de "huida" puede sobreponerse a la capa de "exploración" cuando el robot detecta peligro.

Diseño de una Arquitectura Subsumida

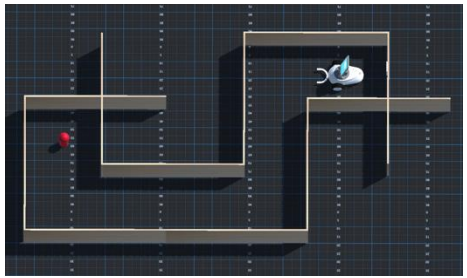
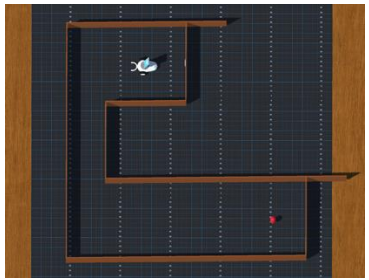
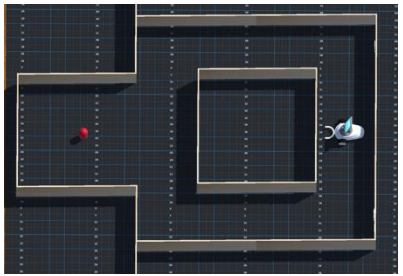
ESQUEMA

Esquema con los pasos a seguir para el diseño de una arquitectura reactiva.



Diseño de una Arquitectura Subsumida

- El ejemplo consiste en que un robot cumpla la misión de encontrar una pelota de color en un mundo formado por paredes, pero de forma totalmente desconocida y variable.
- Utilizaremos el mundo del simulador “Maze” con comportamiento aleatorio.

Diseño de una Arquitectura Subsumida

- PASO 1: DESCRIBIR LA TAREA (GLOBAL)

El robot tiene que encontrar una pelota de color, y una vez que la encuentre centrarse mirando hacia ella.

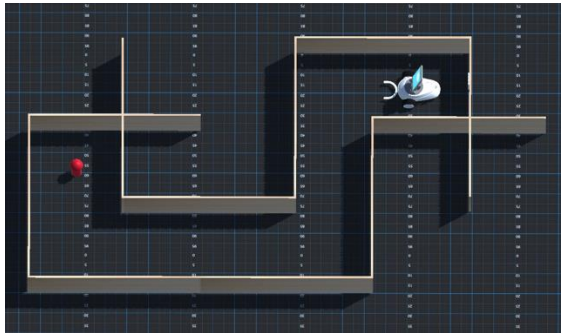
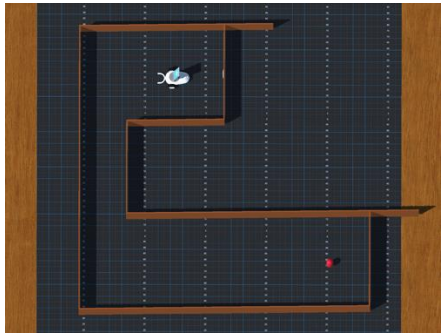
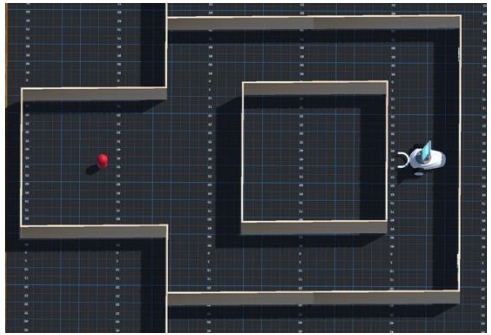
- PASO 2: DESCRIBIR EL ROBOT

- Sensores: IR, giroscopio, micrófono, encoders, cámara.
- Actuadores: dos ruedas, unidad Pan-Tilt, altavoz, LEDs, pantalla...

Diseño de una Arquitectura Subsumida

- PASO 3: DESCRIBIR EL ENTORNO

- El entorno puede variar pero manteniendo estas características:
  - Entorno vacío excepto por la presencia del robot y el objeto de color.
  - El entorno contiene paredes y es posible recorrerlo siguiendo las paredes.

Diseño de una Arquitectura Subsumida

- PASO 4: DESCRIBIR ACTUACIÓN DEL ROBOT

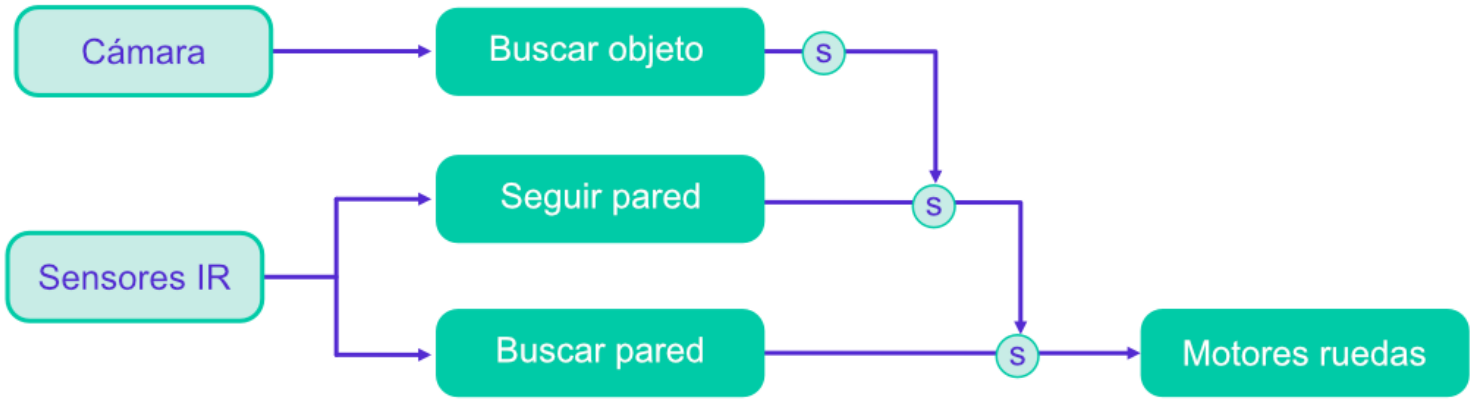
Tabla de comportamientos:

Lanzador	Suprime a	Comportamiento	Acción actuadores	Percepción	Procesado percepción
Color size > 0	Seguir pared Buscar pared	Buscar color	Girar para orientarse hacia el color	Cámara: detección color	Centro del blob de color
Front IR > 80	Buscar pared	Seguir pared	Avanzar hacia delante con la pared un lado	Sensores IR	No requiere procesamiento
		Buscar pared	Moverse recto hacia delante	Sensores IR	No requiere procesamiento

Diseño de una Arquitectura Subsumida

- PASO 4: DESCRIBIR ACTUACIÓN DEL ROBOT

Diagrama con los comportamientos definidos:



Diseño de una Arquitectura Subsumida

- PASO 5: REFINAR CADA COMPORTAMIENTO

Definimos de forma detallada cada comportamiento:

- Buscar pared

  Descripción detallada: el robot avanzará recto y se detendrá cuando los sensores IR detecten que se encuentra cerca de la pared. Usaremos un control proporcional.

  Precondiciones: ninguna, es el comportamiento con el que se inicia.

Diseño de una Arquitectura Subsumida

- PASO 5: REFINAR CADA COMPORTAMIENTO

Definimos de forma detallada cada comportamiento:

- Buscar pared

Pseudocódigo:

```
inicializamos kp
mientras IRFrontal < valor_objetivo:
   error = valor_objetivo – valor IRFrontal
   salida = error * kp
   mover robot recto con velocidad = salida
```

Diseño de una Arquitectura Subsumida

- PASO 5: REFINAR CADA COMPORTAMIENTO

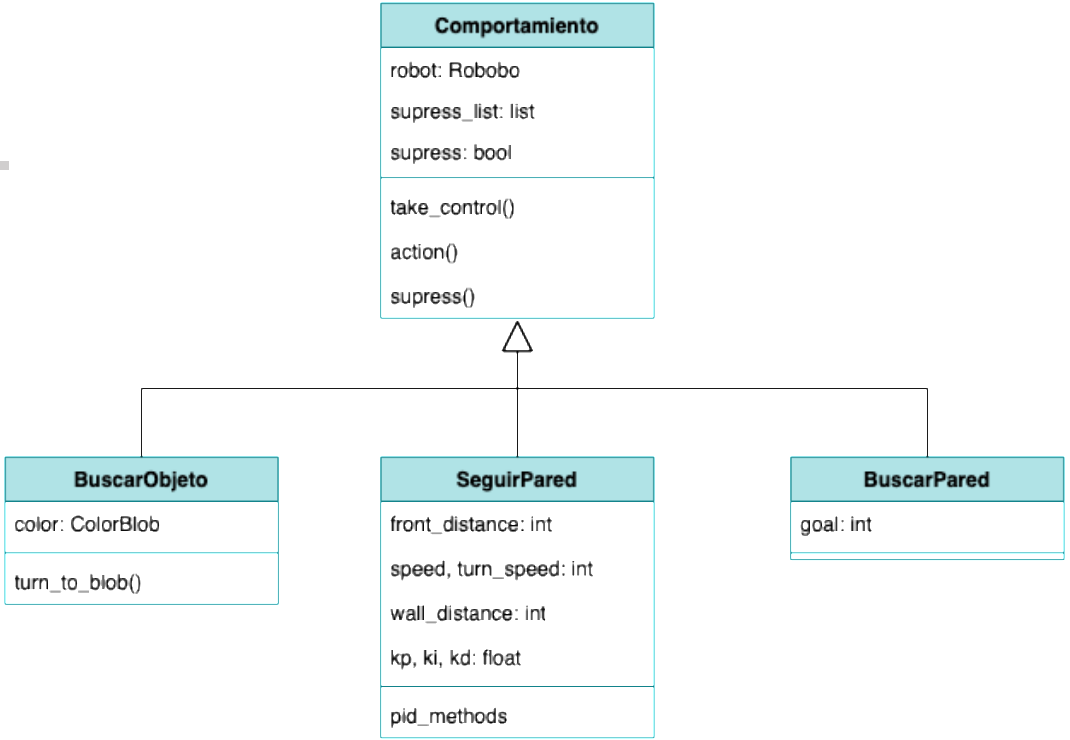
- Cada comportamiento realiza una tarea determinada pero también tienen características comunes → Usaremos programación orientada a objetos.

- Los comportamientos tienen que ejecutarse de forma simultánea → Usaremos programación paralela.

Diseño de una Arquitectura Subsumida

Clases

Diagrama de clases de los comportamientos.



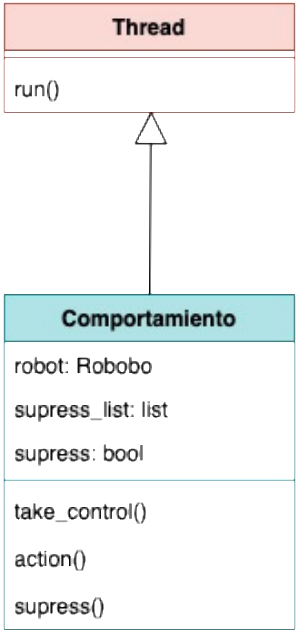
Diseño de una Arquitectura Subsumida

Paralelismo

Utilizaremos hilos.

Usaremos el Módulo de threading:

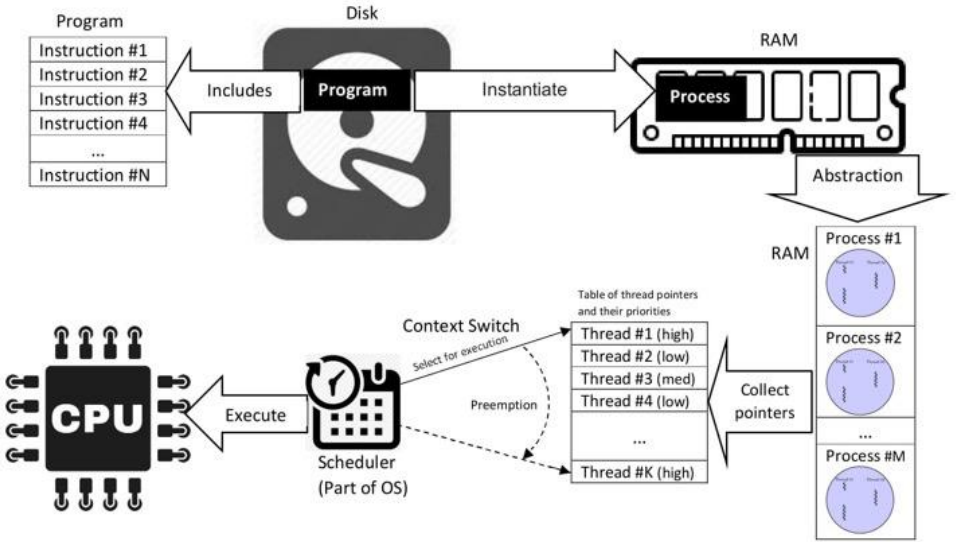
Especialización de la clase Thread.



Diseño de una Arquitectura Subsumida

Paralelismo

- Un proceso es una instancia en ejecución de un programa.
- Puede ejecutar bloques de código simultáneamente: hilos de ejecución (threads).



https://en.wikipedia.org/wiki/Process_(computing)

Diseño de una Arquitectura Subsumida

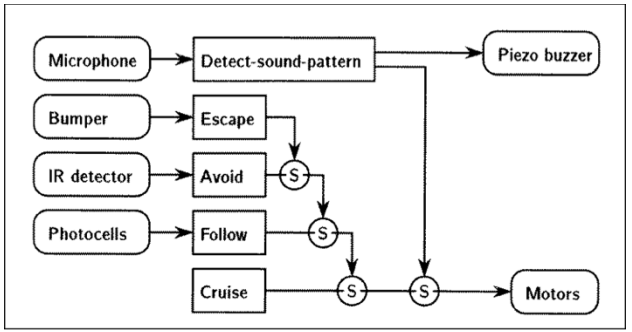
SIGUIENTES PASOS

- PASO 6. PROBAR CADA COMPORTAMIENTO DE FORMA INDEPENDIENTE
- PASO 7. PROBAR LOS COMPORTAMIENTOS EN CONJUNTO
- Los pasos 6 y 7 los realizaremos con el simulador y el entorno de desarrollo.

Arquitecturas Subsumidas. Ejemplos

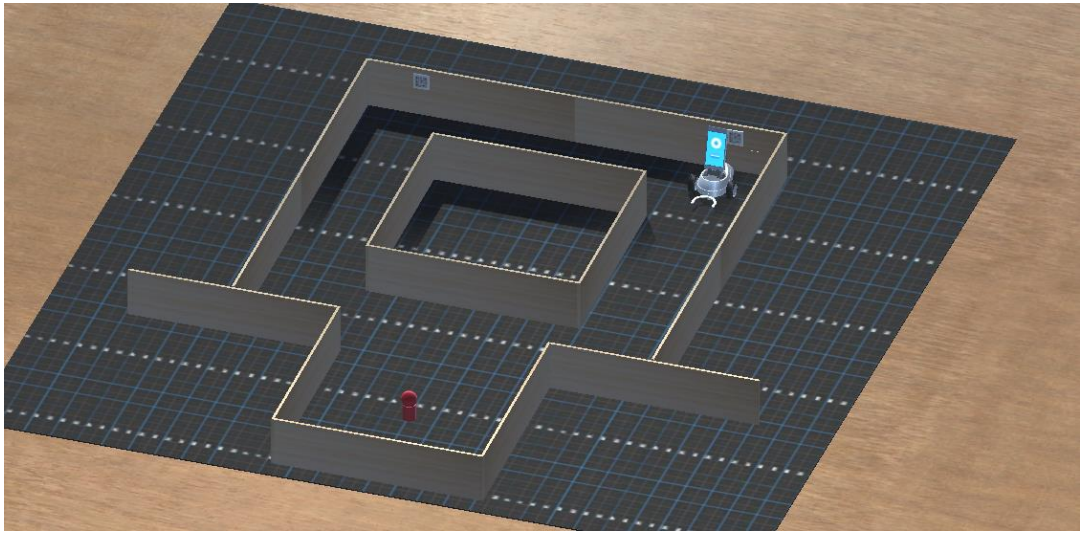
Ejemplo de comportamiento que busca y se dirige hacia fuentes luminosas al tiempo que está a la escucha de una posible orden de detención.

- Follow: actúa cuando la luz percibida por los fotosensores supera un determinado umbral, girando el robot hacia la dirección del más brillante.
- Avoid: evita obstáculos: si el obstáculo está a la derecha gira robot hacia la izquierda, si está a la izquierda gira a la derecha, y si está enfrente gira también hacia la derecha.
- Escape: actúa cuando los infrarrojos fallan, caso de algunos objetos. Mueve el robot alejándolo del obstáculo detectado con los bumpers al chocar y gira siguiendo un patrón similar al de avoid.
- Detect-sound-pattern: detecta secuencias específicas de aplausos y pausas. Si es así, el robot emite un tono particular y se detiene.
- Cruise: mueve el robot hacia delante.



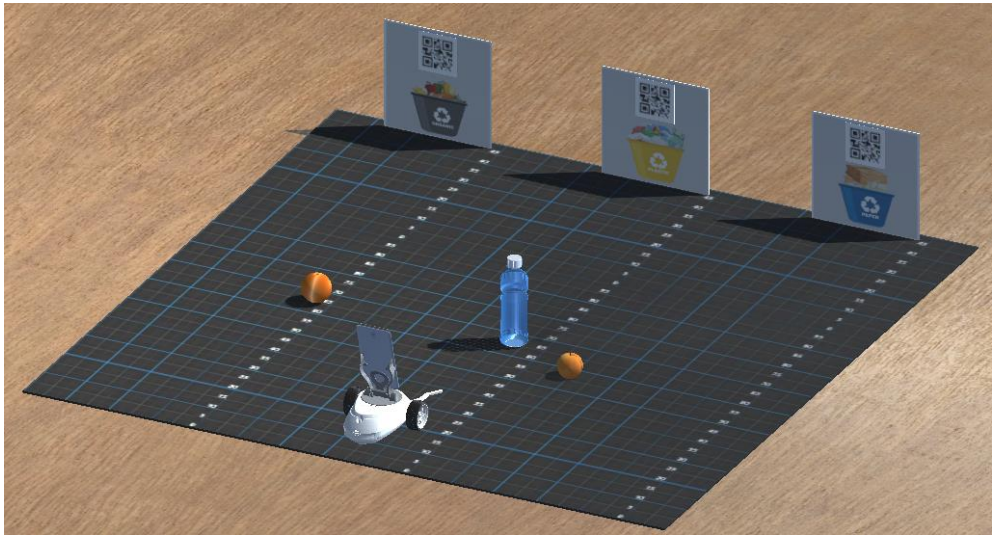
Ejemplos con RoboboSim

- Robot que busca un objeto en un entorno de forma desconocida y lo devuelve a un punto marcado con un código QR.



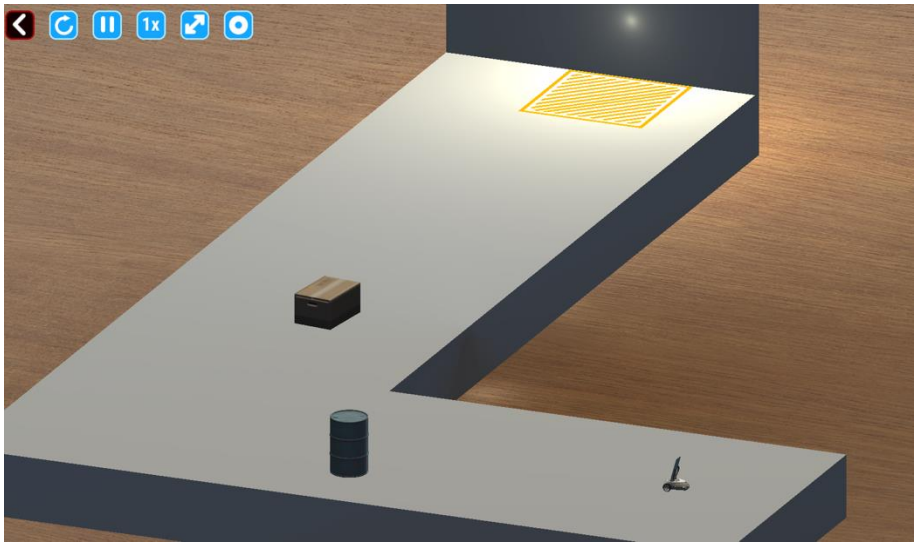
Ejemplos con RoboboSim

- Robot que detecta 3 tipos de objetos (orgánico, plástico y papel) y los deposita en un punto marcado por un código QR.



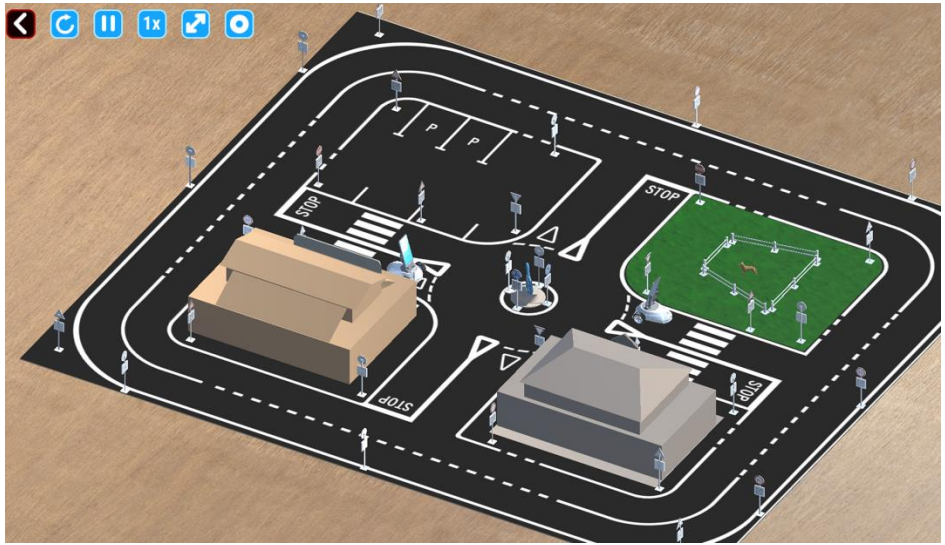
Ejemplos con RoboboSim

- Robot que va hacia una luz esquivando objetos y sin caer de una altura.



Ejemplos con RoboboSim

- Robot que navega de forma autónoma por una ciudad respetando las señales, y cuando recibe un aviso, aparca.



Ejercicio 1: Arquitectura Subsumida

OBJETIVO DEL EJERCICIO

Este ejercicio consiste en la implementación de una arquitectura subsumida con al menos cuatro comportamientos (aunque pueden ser más), que permita a Robobo actuar en un entorno y cumplir sus objetivos independientemente de su posición u orientación inicial e independientemente de que cambien las posiciones de algunos de los objetos.

Ejercicio 1: Arquitectura Subsumida

OBJETIVO DEL EJERCICIO

Cada grupo debe definir un objetivo de actuación y un conjunto de reglas de entorno que serán las que condicionen la actuación del robot. Ejemplos de estas reglas podrían ser acercarse a los objetos de color azul o evitar chocar con las paredes o el resto de los objetos.

- ¡Es importante que el robot realice una tarea útil!

Ejercicio 1: Arquitectura Subsumida

OBJETIVO DEL EJERCICIO

Una vez definidas las reglas de actuación, habrá que generar y programar los comportamientos base del robot, así como la estructura de subsunción de la arquitectura de manera que la actuación del robot resulte lo más inteligente posible para un observador externo.

Ejemplos de comportamientos:

- Dirigirse hacia un objeto de un color determinado.
- Evitar obstáculos y choques.
- Salir de situaciones de colisión.
- Seguir paredes o contornos de objetos.
- Empujar objetos.

Ejercicio 1: Arquitectura Subsumida

OBJETIVO DEL EJERCICIO

- Se aconseja implementar y probar aisladamente los diferentes comportamientos antes de integrarlos dentro de la arquitectura.

- No confundir esto con entregar los comportamientos por separado. La arquitectura debe incluir y coordinar todos los comportamientos implementados.

Entrega de los Ejercicios

DETALLES DE LA ENTREGA

- Entregar un archivo .py con el código Python del ejercicio.
- Agregar comentarios en el código para facilitar la comprensión del mismo.
- Agregar un documento .pdf explicando cómo se ha resuelto el ejercicio.
- Agregar un archivo de video comentado mostrando el funcionamiento del sistema.

IMPORTANTE

- Este trabajo se presentará en la última semana de clase (10 minutos por grupo: presentación y preguntas).
- Se valorará que el trabajo se realice con el robot real (3 puntos), así como la originalidad y la complejidad de la propuesta y de la solución.