



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

*DSIC*  
DEPARTAMENTO DE SISTEMAS  
INFORMÁTICOS Y COMPUTACIÓN

Escuela Técnica Superior de Ingeniería Informática



Dpto. Sistemas Informáticos y Computación  
Escuela Técnica Superior de Ingeniería Informática  
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

## **SISTEMAS INTELIGENTES**

### **PRÁCTICA 1**

## **DISEÑO, IMPLEMENTACIÓN Y EVALUACIÓN DE UN SISTEMA BASADO EN REGLAS**

**- Problema a resolver -**

**Recogida de paquetes en un edificio**

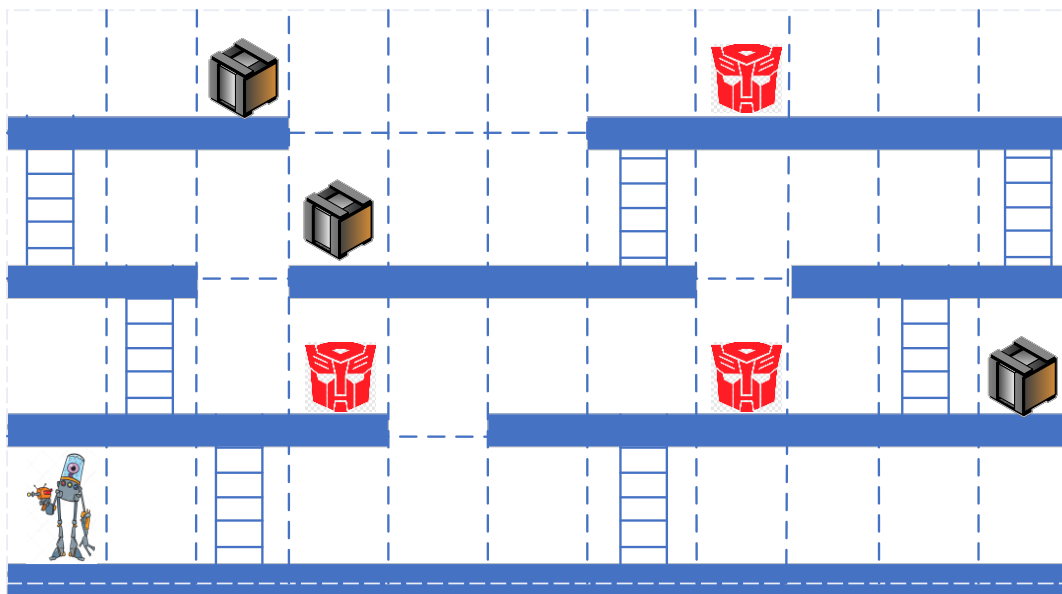
**Septiembre 2020**



## Recogida de paquetes en un edificio

El problema a resolver es un juego que se desarrolla en un entorno como el que se muestra en la figura. Un robot equipado con un arma tiene como misión recoger las cajas que están situadas en las plantas de un edificio. Las plantas se encuentran a diferente altura y cada planta se divide en un conjunto de casillas. Hay unas escaleras distribuidas por el edificio que conectan una casilla de una planta con una casilla de la planta inmediatamente superior o inferior. Además, puede haber “huecos” en una planta, lo que significa que no existe suelo en algunas de sus casillas y por tanto el robot no puede atravesar dichos huecos. Por otro lado, el robot sí puede atravesar una casilla que contenga una escalera.

En este entorno existe también una serie de enemigos (iconos de color rojo) que el robot debe destruir con su arma para poder desplazarse a lo largo de una planta. Concretamente, cuando robot y enemigo se encuentran en la misma planta, y la casilla donde se encuentra el robot es adyacente a la del enemigo, el robot debe eliminarlo con su arma para poder situarse en la posición del enemigo. El número de disparos que el robot tiene disponibles es limitado y puede ser inferior al número de enemigos existentes. El objetivo del programa es que el robot consiga recoger todas las cajas de la escena.



Por comodidad y sencillez en el diseño siempre supondremos que los diferentes elementos del entorno se distribuyen en una cuadrícula (usando coordenadas x,y) como la que se muestra en la imagen. En esta figura en particular tenemos:

- que la dimensión del edificio es 11x4 (4 plantas, 11 casillas por planta)
- que el robot se encuentra en la casilla (1,1)
- que hay un enemigo en la casilla (4,2), (8,2), etc.
- que hay una caja en la casilla (4,3), (11,2), etc.
- que hay una escalera en la casilla (3,1), (7,1), (2,2), etc.
- que el número disponible de disparos del robot son 2

El sistema debe estar preparado para trabajar con cualquier número y disposición de plantas, escaleras, cajas y enemigos. Una caja y un enemigo nunca pueden estar en una misma casilla, ni tampoco una caja o un enemigo pueden estar en una casilla de donde parta o llegue una escalera.

Se pide resolver este problema utilizando un diseño basado en estados mediante un SBR implementado en CLIPS. Como estrategias de búsqueda para resolver el SBR se empleará ANCHURA y PROFUNDIDAD.

1. El programa deberá solicitar al usuario el nivel de profundidad máximo del árbol a desarrollar (ver ejemplo del programa del 8-puzzle).
2. Para cada ejecución del SBR con una de las dos estrategias de búsqueda, ANCHURA o PROFUNDIDAD, el programa debe devolver la profundidad o nivel del árbol donde la estrategia encuentra la solución y el número de nodos generados (no es necesario devolver el camino solución)
3. La representación debe ser generalista y permitir realizar cambios fácilmente. Se debe poder trabajar con un número variable de plantas, número de casillas en cada planta, escaleras, cajas y enemigos sin necesidad de modificar las reglas. Para todas las instancias de este dominio se considerará que solo existe un robot.
4. Toda la información inicial se puede representar directamente en el comando `def facts` que contiene los hechos iniciales.

## INDICACIONES

1. El patrón para representar la información de un estado del problema debe contener únicamente la información dinámica susceptible de cambiar entre estados; la información que puede cambiar de un estado a otro es:
  - la posición del robot
  - la información de las cajas
  - la información de los enemigos
  - número de disparos que el robot tiene disponibles
2. La información estática de una instancia de problema para este dominio (información que no cambia a lo largo de la ejecución del SBR) sería:
  - la casilla donde se encuentran las escaleras
  - la casilla donde se encuentran los huecos
  - el tamaño de la cuadrícula (número de plantas y casillas en cada planta)
3. Las acciones que pueden realizarse en este dominio son:
  1. Mover el robot una casilla a la derecha (se puede mover aunque haya una caja en dicha casilla pero no en el caso de que haya un enemigo).
  2. Mover el robot una casilla a la izquierda (se puede mover aunque haya una caja en dicha casilla pero no en el caso de que haya un enemigo).

3. Subir una planta (el robot debe estar en una casilla C donde haya una escalera y al finalizar la acción el robot terminará en la casilla situada encima de C)
4. Bajar una planta (el robot debe estar en una casilla C tal que exista una escalera en la casilla situada debajo de C y al finalizar la acción el robot terminará en la misma casilla que la escalera).
5. Recoger una caja (el robot debe estar en la casilla donde se encuentra la caja).
6. Disparar a enemigo. El robot debe estar en una casilla adyacente a la del enemigo (a derecha o izquierda) y le deben quedar disparos. Como resultado eliminará al enemigo.

## EVALUACIÓN DE LA PRÁCTICA

La evaluación de la práctica se realizará a través de un examen **INDIVIDUAL** que tendrá lugar el día señalado en el calendario de prácticas para cada uno de los grupos de SIN (ver transparencias del *Bloque 0- Presentación y normativa de la asignatura*).

El examen consistirá en una(s) pregunta(s) sobre el código realizado, evaluación del SBR o realización de alguna modificación del código para atender una nueva funcionalidad en el problema. Se generará una tarea en PoliformaT para la evaluación de la práctica.

Se deberá subir a la tarea de PoliformaT la versión del código realizada durante las sesiones de prácticas (versión antes del examen), así como la versión de código modificado acorde a las instrucciones del examen (**IMPORTANTE**: hay que indicar claramente el nombre de los dos componentes del grupo -si es el caso- en la versión del código de la práctica antes del examen).