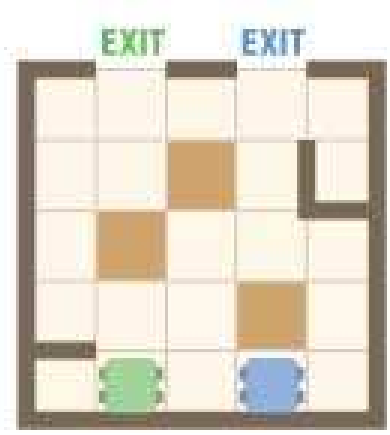
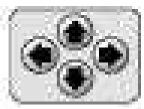
Práctica 6 - Introducción a la Inteligencia Artificial

RESOLUCIÓN DE PROBLEMAS

Profesor responsable: Patricio García Báez
Semana de corrección: 3 al 5 de junio del 2003
Objetivo: Habilidad para elegir los componentes adecuados para un sistema de producción





El juego **Bumper Cars** consta de un laberinto con una serie de obstáculos y un grupo de coches que se mueve en él teniendo que alcanzar los puntos destinos (véase link en http://www.discover.com/nov_01/bogglers.html). El problema consiste por tanto en obtener los movimientos mínimos necesarios para ello y se dificulta por el hecho de que los coches se mueven simultáneamente bajo la misma instrucción, siempre que puedan.

Esta práctica se basará en **diseñar un agente inteligente** que haciendo uso de **un sistema de producción** trate de **obtener los pasos mínimos necesarios** para resolver problemas de este tipo. Además habrá de implementarse dos modalidades (la 1 y 2 de http://www.discover.com/nov_01/bogglers.html), relacionadas con el hecho de si al llegar un coche a un destino puede seguir moviéndose o se queda parado en él (ver el parámetro *:sim* de la función *inic-bump* de *bumper.lsp).* Por otro lado, simplificaremos el problema, eliminando los obstáculos tipo muro y haciendo que los puntos destinos sean una casilla dentro del laberinto.

Para la realización de la práctica tendremos que utilizar **obligatoriamente** para generar las reglas y expandir los estados las funciones implementadas el fichero ***shell.lsp*** y las funciones implementadas en ***bumper.lsp*** y ***check-bump.o*** para generar los problemas y testear el funcionamiento del agente. Ambos ficheros son códigos lisp, están disponibles en el ftp anónimo del centro (*ftp://ftp.csi.ull.es/pub/asignas/INTART/prac6_03/*) y suficientemente documentado su modo de uso.

Adicionalmente existen dos ficheros en los que se puede apoyar en la realización de su práctica: ***agente-humano.lsp*** contiene el código de un agente trivial que juega solicitando los movimientos por teclado. Por otro lado ***q_search.lsp*** consiste en una librería que implementa un marco genérico de como codificar las distintas estrategias de búsqueda en el espacio de estados.

La práctica constará por tanto de los siguientes apartados:

1. Indicar las **características** que nos guían en la elección de un sistema de producción que tiene este **problema** y sus implicaciones.
2. Diseñar, en pseudocódigo, la **representación de estados,** funciones de **estado inicial y final** y **reglas** que crea mas adecuadas.
3. Indicar que **características** posee el **sistema de producción** que resultará del apartado anterior y sus implicaciones.
4. Indicar las **propiedades** que se crean oportunas para las **estrategias de búsqueda**, a partir de los estudios anteriores.
5. **Codificar** el agente que haga uso de las estrategias de control de **búsqueda en anchura y búsqueda A*** bajo **2 heurísticas** diferentes, mediante el uso de *shell.lsp, bumper.lsp* y *check-bump.o.*
6. **Testear el agente** con distintos tipos de laberintos y con distinto tipo de soluciones, ver tabla ***test.ps***.

En la corrección de la práctica **se valorará** especialmente:

- La **originalidad** en el planteamiento y justificación del sistema de producción.
- El lograr **soluciones** de laberintos de distintos **niveles de dificultad**.
- El lograr **soluciones** empleando **poco tiempo de cómputo** para ello.