David Berga 146716
David Madrid 146722

Computación inteligente y lenguaje natural
Práctica 3 - Una aplicación de planning

Propuesta

Nuestra aplicación consiste en un dado de 12 caras. Simplemente cada cara tiene un identificador (puede ser un numero, un nombre...) y el problema consiste en que a partir de estar en una cara, girar en diferentes direcciones hasta llegar a la ultima cara.


Este tipo de problema es bastante simple. La gracia está en que según los valores que se pongan a las caras, se puedan hacer diferentes juegos de azar, o bien de rol, o de twister u otros.

Las secuencias consisten en ir de la cara que esta hacia arriba (la que vemos) ir hacia qualquier cara en el menor numero de giros posibles. Con un dado de 12 caras, por cada cara hay 5 lados donde ir, lo cual seran 5 posibles acciones de giro.

Aplicación

Hemos programado una aplicación en C++ ya que nos parecía el lenguaje que con más facilidad trabajaba con archivos de texto trabajando facilmente con poca memoria en este caso.

El programa tiene gestión de errores en cuanto a encuentro de parámetros y archivos. Lo cual si no se encuentra un archivo y/o el parámetro se ha puesto mal, el programa se sale y reporta al usuario lo que se ha hecho mal y el usage que debe efectuar.

El programa efectua los procesos por el siguiente orden:
1.Crea el domain
2.Genera un problema aleatorio (init y goal) usando como objects las caras de la config
3.Ejecuta el planner (probe) y muestra su información de tiempo y demás
4.Muestra las secuencias de salida del planner

La configuración de entrada (config-pddl/config.txt) sigue la siguiente nomenclatura:
cara = num1
cara = num2
cara = num3
cara = num4
cara = num5
cara = num6
cara = num7
cara = num8
cara = num9
cara = num10
cara = num11
cara = num12
-Por tanto los valores que se pueden cambiar son los “numX” por ejemplo por “mickey”

La secuencia de salida (probe-files/result.txt.1) sigue la siguiente nomenclatura:
(SPINLR NUM8 NUM7)
(SPINLR NUM7 NUM11)
(SPINLL NUM11 NUM5)
-Al ser un dado de 12 caras, el numero maximo de secuencias usadas son 3, ya que con 3 puede llegar a la cara opuesta mas lejana.

Ejecución/Compilación

Simplemente, para ejecutar el programa hay que poner en terminal:
./app "config-pddl/config.txt"

Para compilarlo de nuevo así:
g++ -o app application.cpp

Veamos qué nos devuelve al ejecutar el programa:

```
dber@dber-K52JU:~/app12$ ./app "config-pddl/config.txt"


---------------CREATING DOMAIN---------------n
Done!


---------------GENERATING PROBLEM with config data---------------n
Done!


---------------RUNNING PROBE PLAN---------------n
 --- OK.
Fluents=14
Operators=63

================================================
;; PROBE searching...
================================================
[0] [1] [2]
================================================
;; SOLUTION 1
;;       Plan cost: 3.000000, steps: 3
;;       Time: <1 msec
================================================
Done!


---------------RESULTS---------------n
(SPINLR NUM4 NUM11)
(SPINUR NUM11 NUM7)
(SPINLL NUM7 NUM9)


Done!
dber@dber-K52JU:~/app12$
```