

YAKS - Yet Another Khepera Simulator

El simulador YAKS puede funcionar tanto en windows como en Linux. Para aquellos que deban usar Linux lo pueden ejecutar a través del emulador de windows (wine). En este caso deberán ejecutar la siguiente línea de comando: `wine yaks.exe archivo_de_configuracion` dentro del directorio donde se encuentra descomprimido el simulador. El parámetro `archivo_de_configuracion` es un archivo de extensión `opt` que necesita el YAKS para arrancar y saber como es el entorno de funcionamiento. El simulador arrancará y la forma de comunicarle las acciones que debe tomar es a través de un socket que está escuchando en el puerto 1033 por defecto (configurable en el archivo `.opt`). En este caso los que usen Linux deberán ejecutar un telnet desde la línea de comandos. Por ejemplo: `telnet localhost 1033`. También hay una versión del simulador que se compila y no es necesario usar el emulador de windows. Corriendo el script `launch.sh` que se encuentra dentro del directorio `bin` se puede compilar una versión del YAKS. Para los que usen Windows pueden hacer un archivo `.bat` para incluir el parámetro de `simple.opt` en la llamada a `yaks.exe`. Para comunicarse con el simulador lo único que deben hacer es iniciar un Hyperterminal en localhost en el puerto 1033 o en su defecto un Putty en modo RAW en el localhost puerto 1033

Archivo de configuración

El archivo de configuración es un archivo de texto de extensión `.opt`. Cada línea contiene el nombre de un parámetro y su valor correspondiente, separados por uno o más espacios. Las líneas que comienzan con el símbolo numeral (`#`) son consideradas comentarios. En la siguiente tabla se describen todos los parámetros aceptados. Al final de esta página puede ver un archivo de ejemplo.

Parámetro	Significado	Valores posibles
A_#ROBOTS	Cantidad de robots	1..n
WORLD_PATH	Directorio donde se encuentra la definición del entorno	.\worlds
WORLD_FILE	Nombre de archivo que contiene la definición del entorno	nuclear.world
SERVER_PORT	Puerto TCP/IP para utilizar en la comunicación con el controlador	1033
TEST_SAME_START_POSITION	Hace que los robot comiencen siempre en la misma posición de inicio	1: sí. 0: no.
UPDATE_DELAY	Tiempo de espera entre actualizaciones de la visualización, en milisegundos	100
VERBOSE_LEVEL	Nivel de información de debug que se imprime en la consola	0: desactivado. 3: completo
SCALE	Factor de escala de la visualización	0.5
SENSOR_NOISE	Agrega ruido a los sensores	1: sí. 0: no.
NOISE_PERCENTAGE	Cantidad máxima de ruido (porcentaje)	5
MOVE_OBSTACLES	Mueve los obstáculos aleatoriamente entre corridas	1: sí. 0: no.

Ejemplo

```
#####  
# This files includes the parameters that are adjustable in the simulator,#  
# use # in the beginnig of lines to comment them out. #  
# The file is not case sensitive. #  
#####  
  
# Number of robots  
A_#ROBOTS 1  
  
# define path and file for the world definition file  
WORLD_PATH .\worlds  
WORLD_FILE nuclear.world  
  
# This option makes the robot start always at the same position  
TEST_SAME_START_POSITION 1  
# milliseconds to wait between updates  
UPDATE_DELAY 100  
VERBOSE_LEVEL 0  
  
# Graphical options  
SCALE 0.5  
  
# Server TCP/IP port for controller communication  
SERVER_PORT 1033  
  
SENSOR_NOISE 0 # If you want noise added to your input, 1 - yes, 0 - no  
NOISE_PERCENTAGE 5 # Amount of noise to add 0 - 100  
MOVE_OBSTACLES 1  
  
#####  
# This file was created by the author of the simulator #  
# Johan Carlsson, johanc@ida.his.se, 1999 #  
#####
```

Archivo de definición del mundo o entorno

El archivo de definición del mundo o entorno es un archivo de texto. Cada línea contiene las coordenadas para definir un único elemento. Las líneas inválidas son descartadas. En la siguiente tabla se describen todos los parámetros aceptados.

Nombre	Significado	Parámetros
wall	Pared	x1, y1, x2, y2
zone	Zona	x, y, r
light	Fuente de luz	x, y
roundobst	Obstáculo grande	x, y
sroundobst	Obstáculo pequeño	x, y
start	Posición de inicio	x, y, ang
radius	Radio de todos los obstáculos	r
sradius	Radio de todos los obstáculos pequeños	r

NOTAS

Todo archivo de definición de ambiente debe contener una única definición para el radio de los obstáculos (radius y sradius) y al menos tantas posiciones de inicio (start) como robots se desee utilizar (parámetro A_#ROBOTS).

Si se utiliza el parámetro MOVE_OBSTACLES (su valor es 1), la posición de los obstáculos no es tomada en cuenta.

Ejemplo

```
#Paredes del exterior
wall 0 0 1000 0
wall 0 0 0 1000
wall 1000 0 1000 1000
wall 0 1000 1000 1000

#Paredes dentro del entorno
wall 300 0 300 700
wall 500 400 500 1000
wall 500 700 800 700
wall 700 500 1000 500

#Zonas objetivos
zone 600 900 70
zone 300 300 50

#Radio asignado a todos los obstáculos
radius 15

#Obstáculos grandes
roundobst 300 700
roundobst 500 400
roundobst 800 700
roundobst 700 500
```

```
#Fuente de luz
light 600 600
```

```
#Posiciones de inicio para dos robots
start 50 50 90
start 150 50 90
```

Comandos de control

Cada comando está formado por una letra que identifica el comando (que puede estar precedida por un modificador), una lista de parámetros que dependen de cada comando y un caracter de fin de línea. El simulador envía siempre una respuesta por cada comando recibido exitosamente. Si el comando es inválido, el simulador devuelve el mensaje ***Command not found***.

La siguiente tabla contiene la lista de comandos aceptados por el simulador y la respuesta asociada.

Comando	Significado	Respuesta
N	Leer los sensores de proximidad	n,<s0>,<s1>,<s2>,<s3>,<s4>,<s5>,<s6>,<s7> 0 <= sn <= 1024
O	Leer los sensores de luminosidad	o,<s0>,<s1>,<s2>,<s3>,<s4>,<s5>,<s6>,<s7> 0 <= sn <= 1024
D,<left>,<right> -10 <= left, right <= 10	Asignar velocidades a las ruedas	d
E	Leer las velocidades de las ruedas	e,<left>,<right> -10 <= left,right <= 10
G,<left>,<right>	Asignas valores a los encoders de las ruedas	g
*<num>	Cambia el robot activo	*<num>
#E	Lee el sensor de energía	#e,<energy> 0 <= energy <= 1
#G	Verifica si el robot se encuentra dentro de una zona	#g,<ground> ground = 0 o 1
#C	Lee el compas	#c,<compass> 0 <= compass <= 1

Con el comando #C lee la orientación, esto es, cuando apunta a derecha mide 1, arriba mide 0,75 a la izquierda mide 0,5 y abajo mide 0,25. Es decir va de 1 a 0 en sentido antihorario.