

NOMBRE: José Arcos Aneas
D.N.I: 74740565-H

PRACTICA 1 INTELIGENCIA ARTIFICIAL AGENTES REACTIVOS

Las posibilidades para crear el agente son varias. Expondré algunas en las que empecé a implementar.

Teniendo en cuenta las suposiciones del tamaño de la habitación la posición en que partimos y controlando la orientación podríamos crear una matriz en la que almacenaremos una prioridad después de recorrer la matriz por completa registrando las paredes.

Los resultados son bastante buenos en relación a otras implementaciones porque se conoce las paredes y objetos pero además el tamaño. El grado de suciedad llega a ser muy bajo de unos 64000.

Mi implementación entregada tiene carácter aleatorio. Aunque no tiene nada que ver con el ejemplo entregado para prueba. En primer lugar se comprobaba si la superficie inicial esta sucia entonces la limpia.

Después comprueba cuales fueron las ultimas acciones. Siempre que sea un giro avanzara para probar si choca (o sea si se activa bump_).

Si choca comprueba los movimientos guardado en 4 posiciones antes de now ya que después de un giro o un hacia delante siempre limpiara. Por ejemplo si giro a derecha después avanzo y limpio. Luego voy hacia delante si chocara, entonces compruebo que hice en las posiciones anteriores y actuó conforme a ello.

Si en las posición anterior limpie esta no limpio y si las posiciones anteriores en 4 a now ya que la anterior sera FORWARD , y pongo 4 porque después de cada giro o avance se limpia y eso queda guardado en el array ACTUAL como una acción mas..

Esta forma de implementación es no determinista incluyo aleatorio ya que si se fueran incrementado el numero de muros o objetos a esquivar por el agente que memorice un recorrido no es factible y menos aun si hay objetos móviles.

El código es sencillo y esta explicado y comentado en la medida de lo posible. He elegido entregar esta porque al ser un agente reactivo veo importante que disponga de cierta aleatoriedad. Aunque sea mucho menos efectiva. ya que sus movimientos son aleatorios y no almacena ninguna matriz en memoria que guarde donde están los obtaculos. De la otra manera el agente memorizaba el recorrido y cuando llegaba a una situación en la que conocía el terreno solo utiliza el sensor de suciedad.

Esta implementación es totalmente reactiva y no determinista, a pesar de eso sus resultados no son tan malos como esperaba en un principio.

Básicamente el funcionamiento de esta practica seria el siguiente:

En primer lugar comprobamos si la superficie esta sucia y si la ultima accion no fue limpiar.

También comprobaba si el movimiento anterior es un giro ya que entonces tendría que avanzar una posición en el sentido del giro que realizo anteriormente.

Otra comprobación que realiza este agente es la de si se encuentra un obtaculo. en ese caso miraría cual era el movimiento que realizo dos veces antes(ya que el anterior sera siempre avanzar) y genera otro posible movimiento excluyendo el movimiento que realizo. Por ejemplo si el agente giro a la derecha y luego avanzo prestaríamos atención al giro a la derecha y generaríamos un posible movimiento que fuera girar a la izquierda(o a derecha aquí esta la aleatoriedad) si chocara después de avanzar haríamos que girara otra vez con aleatoriedad, esto podría mejorarse incluyendo otra comprobación mas pero por motivos de tiempo no ha sido posible.

