

Programando Robots Reactivos



jvisca@fing.edu.uy

2013

Temario

- ¿Qué es programación?
- Programación de robots.
- Una inspiración biológica.
- Cómo programar nuestras criaturas.

¿Qué es un programa?

- Secuencia de instrucciones que una computadora puede interpretar y ejecutar.
- Escrito en un lenguaje de programación.
- Se manipulan datos y se ejecuta un algoritmo.

¿Qué es un programa?

Para ciertos datos de entrada (input) el programa aplica un algoritmo y genera una salida (output).



Los algoritmos son el objeto de estudio de la programación.

¿Qué es un programa?

(Real Academia Española. www.rae.es)

algoritmo.

1. m. Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.

Características:

- Preciso: orden en que se realizan los pasos.
- Definido: siempre se obtiene el mismo resultado sin importar el número de veces que se aplique.
- Finito: tiene fin.

¿Qué es un programa?

Ejemplo: encontrar el mayor número de una lista de números positivos, no vacía.

Voy recorriendo los números, y si alguno es mayor que el mayor que vi hasta el momento, lo recuerdo. Al terminar la recorrida, tendré el mayor de todos.

¿Qué es un programa?

Ejemplo: encontrar el mayor número de una lista de números positivos, no vacía.

Leo lista

mayor=0

Para cada valor **en** lista

si valor > mayor **entonces** mayor=valor

Devuelvo mayor

¿Qué es un programa?

Mas ejemplos:

- Encontrar las raíces de un polinomio de 2º grado.
- Ordenar una lista de números.
- Encontrar la salida de un laberinto.
- Escribir un poema.
- Traducir de un idioma a otro.
- Jugar al ajedrez.
- Ganar al ajedrez.

¿Qué es un programa?

A tener en cuenta:

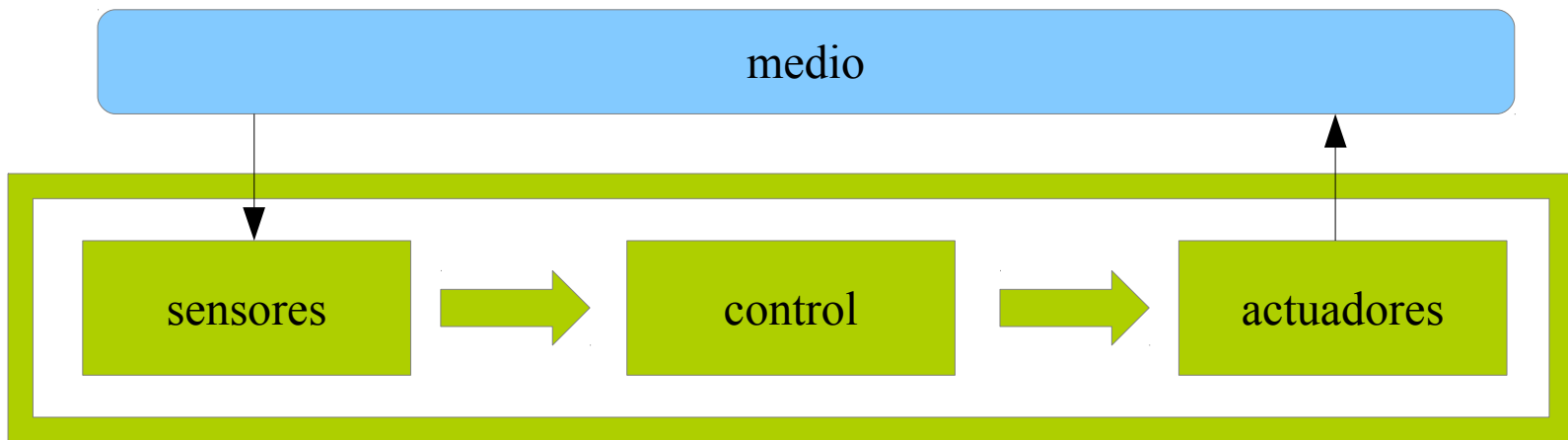
- Los datos que vamos a tener disponibles a la entrada es parte de la descripción del problema
- El algoritmo debe garantizar que termina, pero no se dice cuánto tarda.
- Problemas muy complejos llevan a aplicar Inteligencia Artificial

¿Qué es un robot?

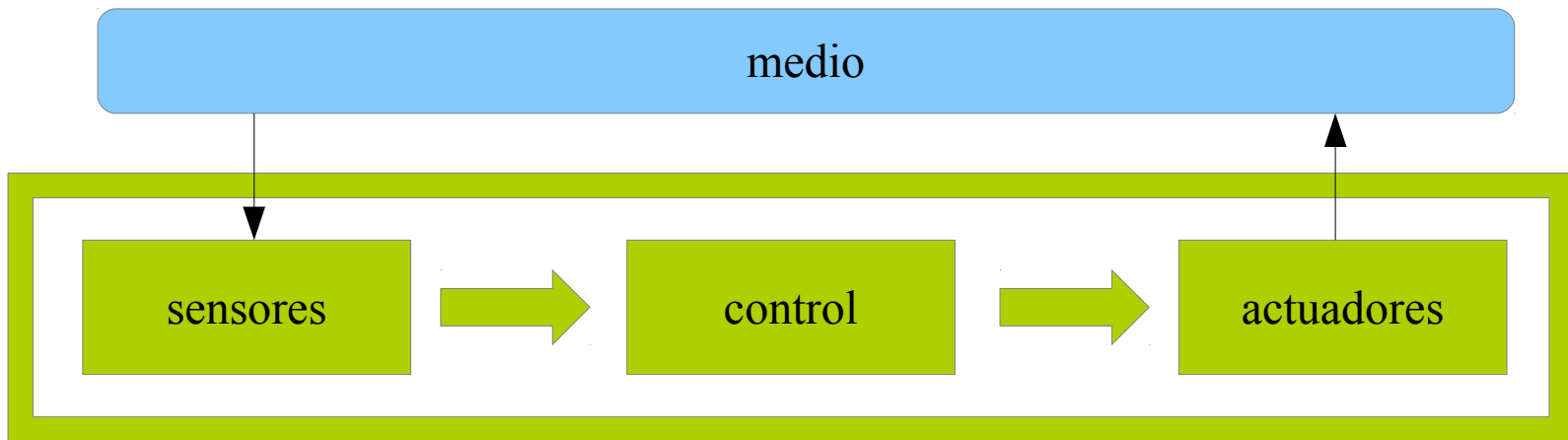
Que nos interesa (en vez de una definición):

- Debe resolver un problema.
- Debe moverse.
- Debe ser autónomo.
- Debe tomar decisiones en función de lo que perciba del mundo.
- Hacer todo esto de la mejor manera posible: robusto, eficiente, simple, etc.

¿Qué es un robot?



¿Cómo se controla un robot?



+



Paradigmas

Existen tres funciones primitivas en robótica:

- Sensar (SENSE)
- Planificar (PLAN)
- Actuar (ACT)

Paradigmas

Primitiva robótica	Entrada	Salida
Sensar (SENSE)	Datos de los sensores	Información sensada
Planificar (PLAN)	Información (sensorial o cognitiva)	Directivas
Actuar (ACT)	Información sensada o directivas	Comandos a los actuadores

Paradigmas

Existen tres paradigmas para organizar la inteligencia en un robot:

- Jerárquico (Deliberativo)
- Reactivo
- Híbrido Deliberativo/Reactivo

Paradigma jerárquico

Robot + Inteligencia Artificial = Paradigma jerárquico

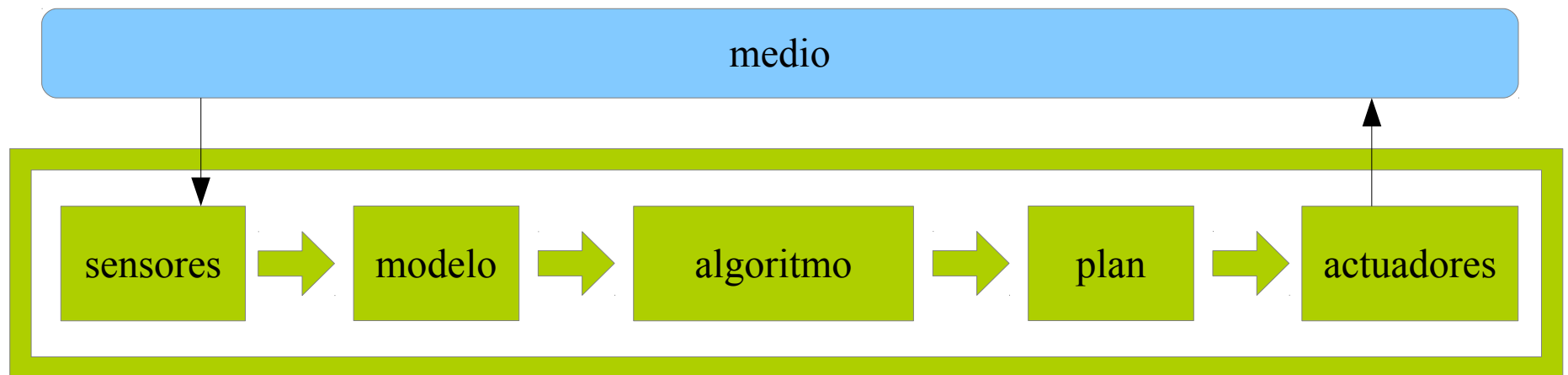
Sabemos escribir programas que juegan a ajedrez. Mover un robot no puede ser más complicado.

(1965 - 1990)

Paradigma jerárquico

Primitiva robótica	Entrada	Salida
Sensar (SENSE)	Datos de los sensores	Información sensada
Planificar (PLAN)	Información (sensorial o cognitiva)	Directivas
Actuar (ACT)	Información sensada o directivas	Comandos a los actuadores

Paradigma jerárquico



Paradigma jerárquico

Ejemplo: Shakey (1967-1971)



Paradigma jerárquico

En la práctica, el paradigma jerárquico funciona mucho peor de lo que se podría esperar de sus algoritmos de base: el mundo real no es un tablero de ajedrez.

Jugador de ajedrez - Cadete de oficina

Ajedrez vs. Realidad

¿El entorno es determinista?

Conociendo el estado actual, es posible predecir exactamente el resultado de aplicar una acción.

Jugador de ajedrez: si (“estratégico”).

Cadete de oficina: no.

Ajedrez vs. Realidad

¿El entorno es observable?

El sistema sensorial tiene acceso a toda la información necesaria para representar internamente el estado del entorno.

Jugador de ajedrez: si.

Cadete de oficina: no.

Ajedrez vs. Realidad

¿El entorno es en Tiempo Real?

En un sistema de Tiempo Real, los procesos ocurren en el tiempo físico, externo e independiente. El tiempo no se detiene ni frena, nunca.

Jugador de ajedrez: No (*).

Cadete de oficina: Si.

Ajedrez vs. Realidad

¿El entorno es estático o dinámico?

Un entorno dinámico puede cambiar mientras el robot está deliberando.

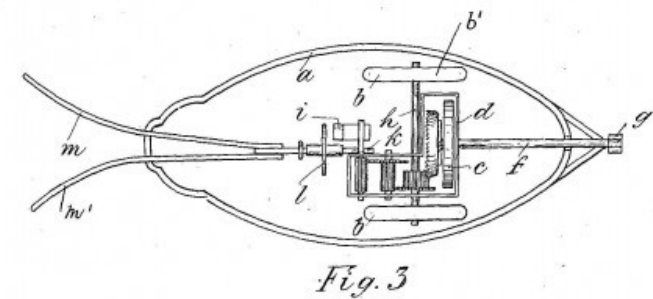
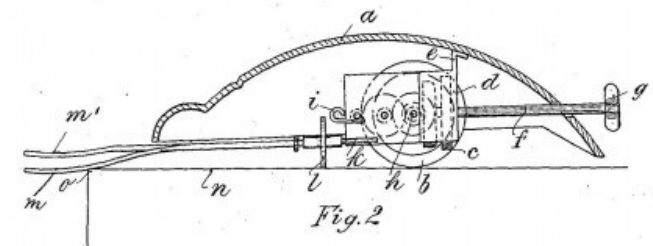
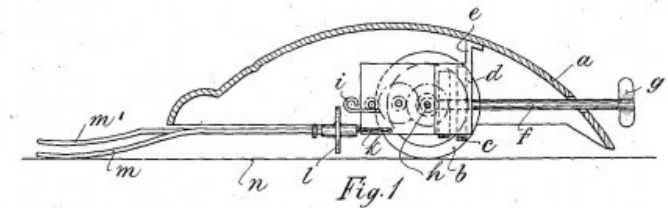
Jugador de ajedrez: estático.

Cadete de oficina: dinámico.

Robots que no piensan

¿Se hacían robots antes de la computadoras?

A. WEIGEL.
MOVING TOY.
APPLICATION FILED MAR. 20, 1911.
1,017,066. Patented Feb. 13, 1912.



Witnesses:
Max Lemmer
Paul Gschlopp

Inventor:
Adolf Weigel

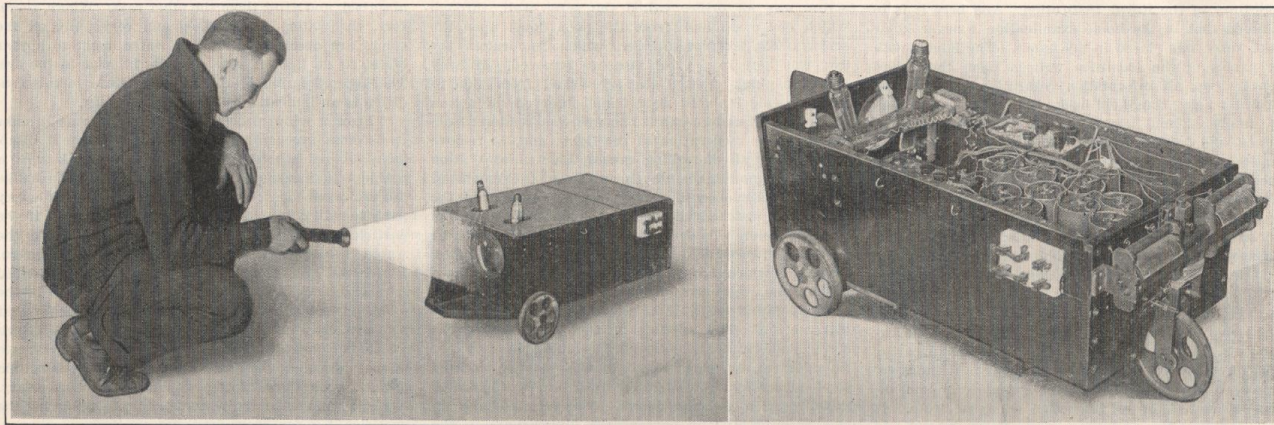
Robots que no piensan

¿Se hacían robots antes de la computadoras?

376

SCIENTIFIC AMERICAN SUPPLEMENT No. 2267

June 14, 1919



The Electric Dog following its luminous master, and a general view of its internal arrangement

The Electric Dog

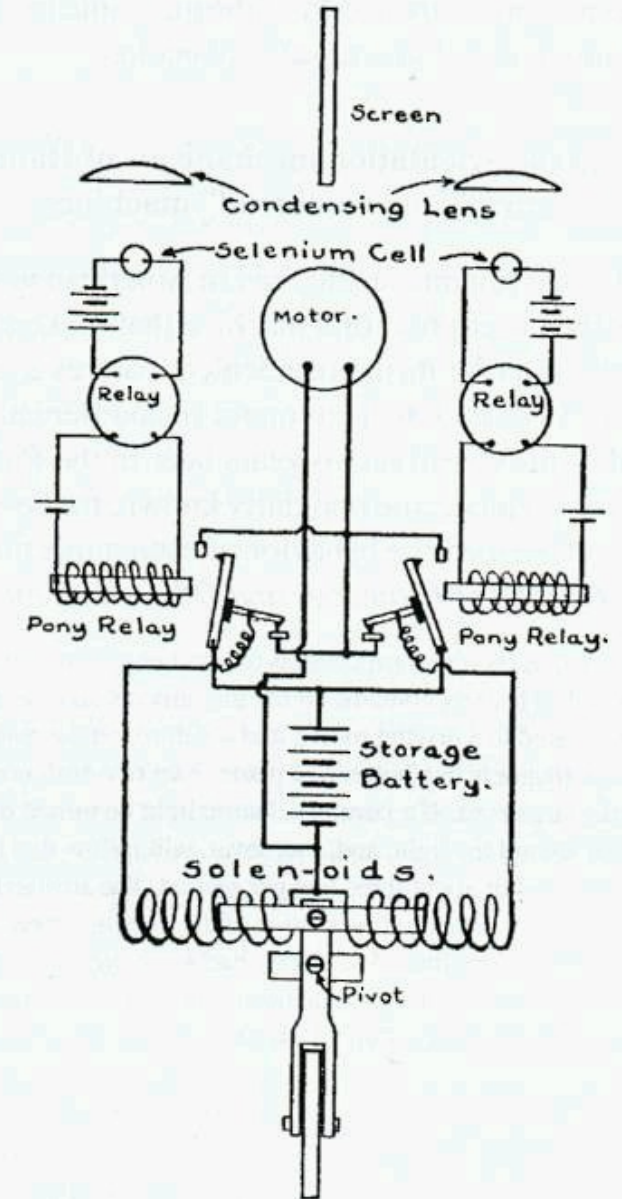
Use of the Selenium Cell to Make an Orientation Mechanism

By Benjamin Franklin Miessner

THE principle of orientation is well exemplified in nature. A large variety of plants, flowers, and other living examples of the realm of botany, are endowed with the power of turning themselves for the purpose of gaining the beneficial effect of the sun's rays. A

possessed of its own mechanical brain, my work on the electric dog represents an attempt to evolve an automaton which would conduct itself in much the same manner as do some of the lower forms of animal

fully in rendering an automaton susceptible only to the call of its master is to prevent the calls of others from reaching it. This the orientation mechanism has done for the searchlight-selenium or infra-red ray systems of



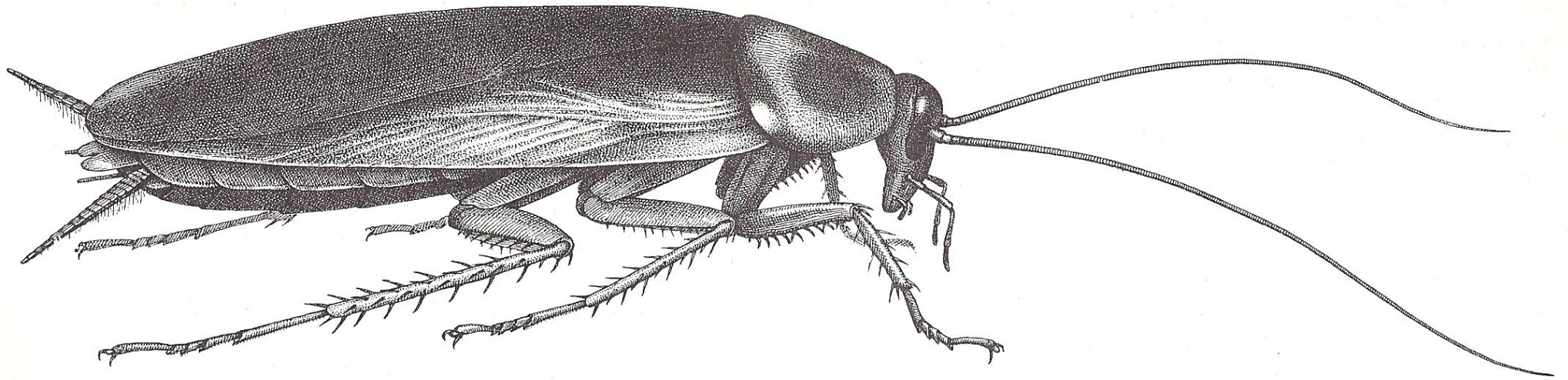
Observando la naturaleza

Inspiración: polilla



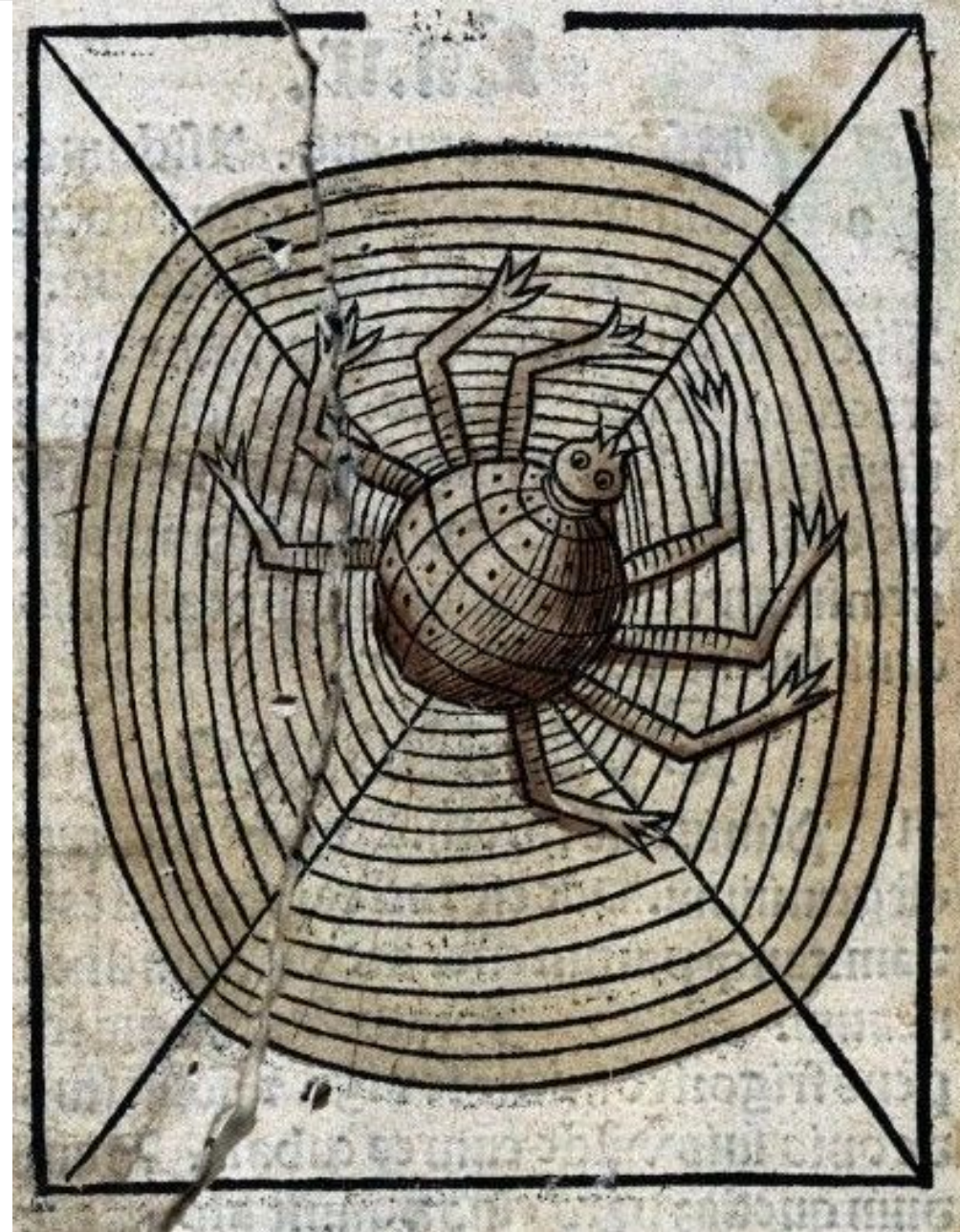
Observando la naturaleza

Inspiración: cucaracha



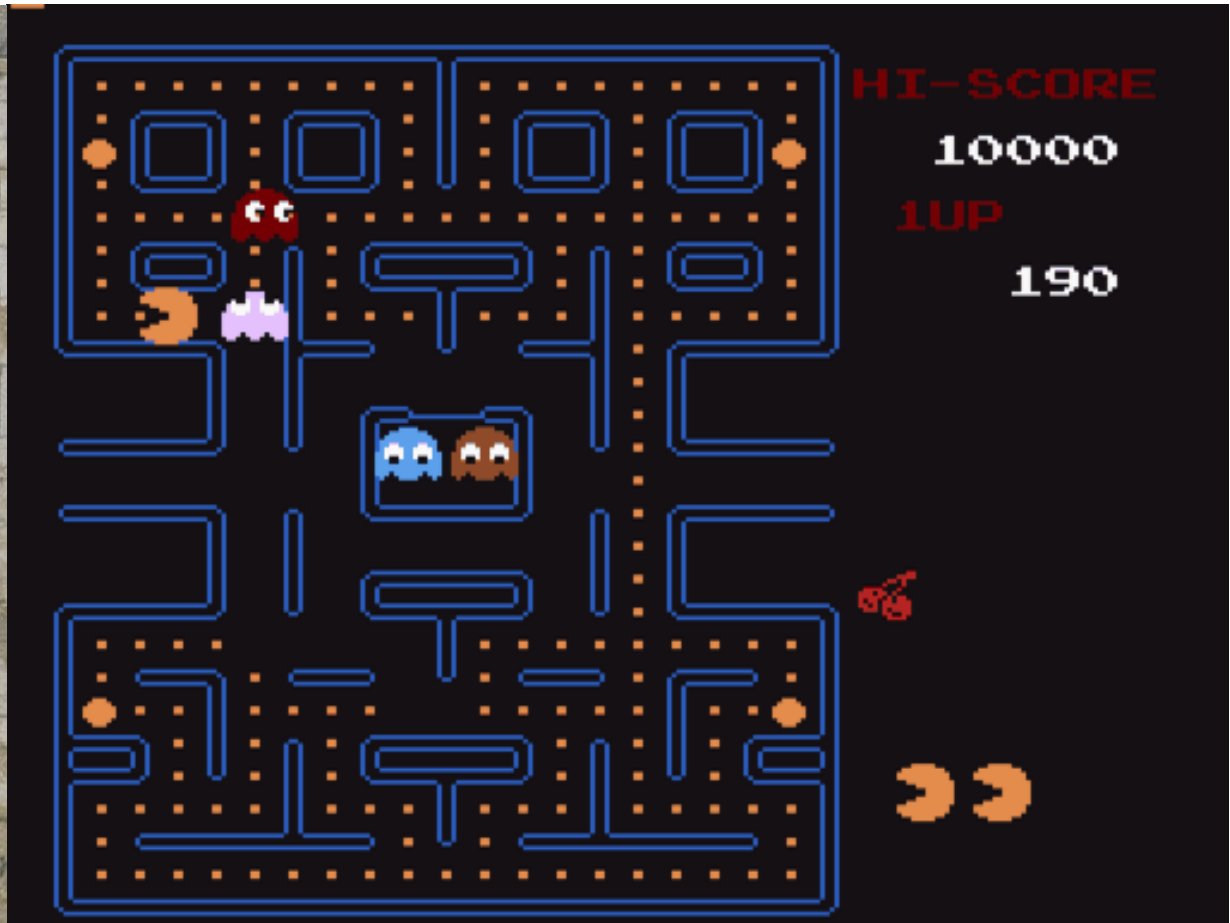
Observando la naturaleza

Inspiración: arañas



Observando la naturaleza

Inspiración: laberintos



Paradigma reactivo

- Este paradigma aparece de los estudios en biología y psicología cognitiva.
- Minimizar tiempo de respuesta, vinculando sensores y actuadores con reglas simples. No planificar, ni mantener un modelo del mundo.
- Para resolver problemas, combinar comportamientos simples.
- El comportamiento resulta de la interacción continua del programa con el medio.

¡No pienses, reacciona!

Paradigma reactivo

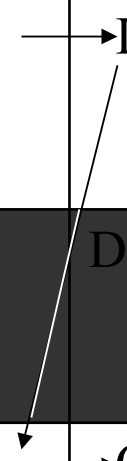
Un comportamiento es una asignación de estímulos sensoriales a un patrón de acción motora que se utiliza para realizar una tarea.



Reflejos - Memoria muscular

Paradigma reactivo

Primitiva robótica	Entrada	Salida
Sensar (SENSE)	Datos de los sensores	Información sensada
Planificar (PLAN)	Información (sensorial o cognitiva)	Directivas
Actuar (ACT)	Información sensada o directivas	Comandos a los actuadores



Paradigma reactivo

- Describir la tarea
- Describir el robot
- Describir el entorno

Especificación y Análisis

- Describir como el robot debería reaccionar al ambiente

Diseño

- Implementar cada comportamiento
- Probar cada comportamiento

Implementación

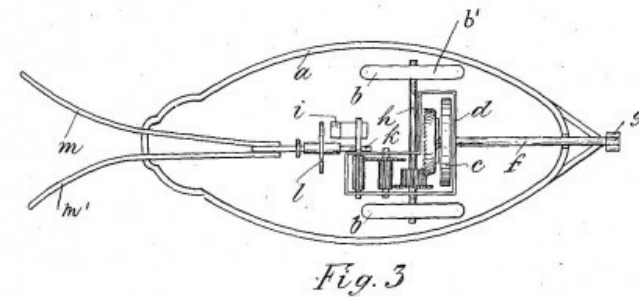
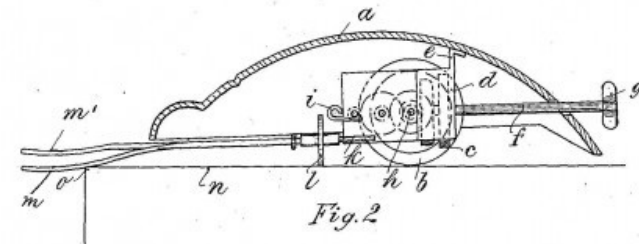
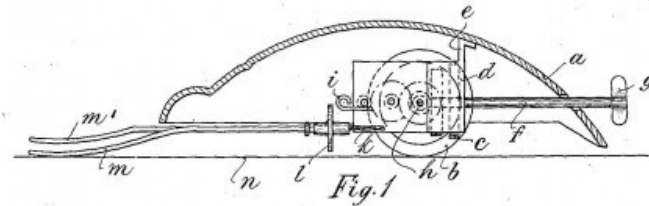
- Juntar todos los comportamientos
- Probar todos lo comportamientos

Integración

Paradigma reactivo

Por siempre
avanzar()
Si hay precipicio()
detenerse()

1,017,066. A. WEIGEL.
MOVING TOY.
APPLICATION FILED MAR. 20, 1911. Patented Feb. 13, 1912.



Witnesses:
Max Lemmer
Paul Froehner

Inventor:
Adolph Weigel

Paradigma reactivo

Por siempre

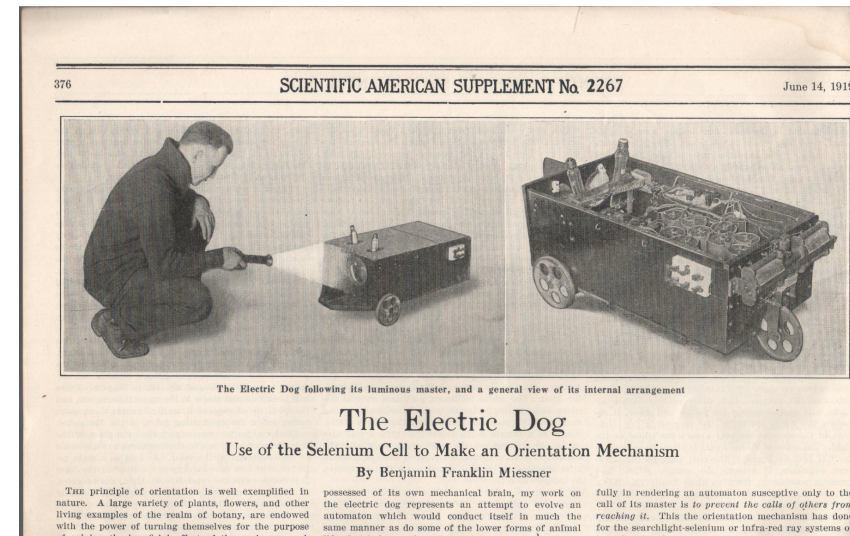
avanzar()

Si $luz_{izq}() > luz_{der}()$

giro_izq()

Sino

giro_der()

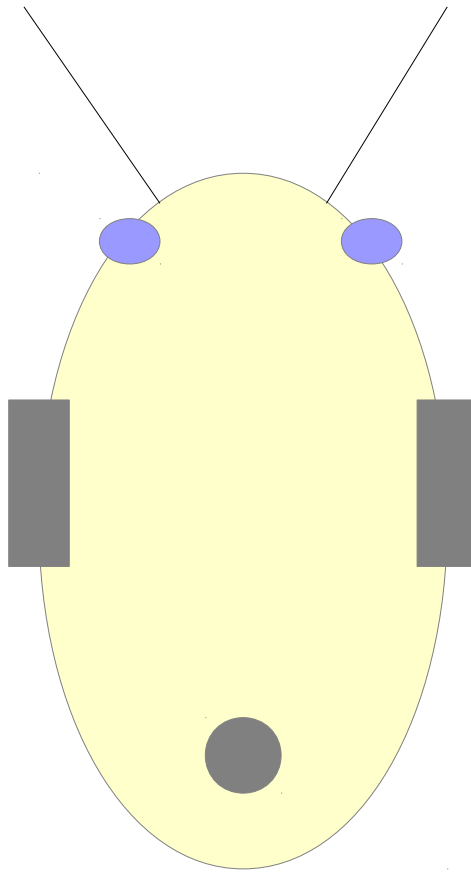


Desarrollo de un robot reactivo

Ejemplo de aplicación, basada en un insecto.

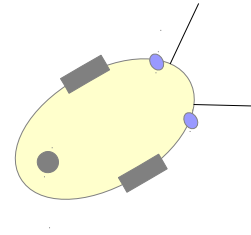
- Qué hace:
 - De noche recorre el piso recogiendo migas.
 - Huye de la luz, pegándose a las paredes.
 - Cuando huye, al llegar a un rincón espera la oscuridad y que se le pase el susto.
- Descripción del robot y del ambiente:
 - Se mueve en una habitación.
 - control diferencial (2 ruedas)
 - 2 sensores de luz (ojos) y 2 de contacto (antenas)

Desarrollo de un robot reactivo



Desarrollo de un robot reactivo

De noche recorre el piso recogiendo migas.



Por siempre

avanzar()

Si paso_tiempo() o antena_izq() o antena_der()

cambio_dirección()

Desarrollo de un robot reactivo

Huye de la luz, pegándose a las paredes.



Por siempre

avanzar()

Si luz_izq() > luz_der()

giro_der()

Sino

giro_izq()

Por siempre

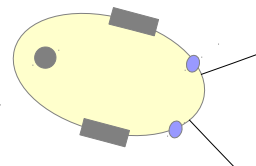
avanzar()

Si antena_izq()

giro_der()

Si antena_der()

giro_izq()

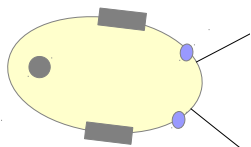


Desarrollo de un robot reactivo

Cuando huye, al llegar a un rincón espera la oscuridad y que se le pase el susto.

```
Si antena_izq() y antena_der()  
    detener()
```

```
mientras luz_izq() o luz_der() o paso_tiempo()  
    espero()
```



Desarrollo de un robot reactivo

Cuando huye, al llegar a un rincón espera la oscuridad y que se le pase el susto.

```
Si antena_izq() y antena_der()  
  detener()
```

```
mientras luz_izq() o luz_der() o paso_tiempo()  
  espere()
```

Combinando comportamientos

- Los comportamientos corren en paralelo, y pueden producir salida en simultáneo
 - Si las salidas son compatibles, no hay problema.
 - Si deciden cosas distintas, hay que elegir: prioridad.
- Un comportamiento puede inhibir a otro:
Si estoy *escapando*, no intento *comer*.
- Un comportamiento puede activar a otro:
Si *encontré* un escondite, *espero*.

Combinando comportamientos

Innate Release Mechanism (von Frisch, Lorenz & Tinbergen, 1973): **cada comportamiento tiene un liberador.**

- El liberador actúa como una señal de control para activar un comportamiento.
- Si un comportamiento no se libera, no responde a los estímulos sensoriales y no produce salida motora.



Combinando comportamientos

```
Por siempre
  avanzar()
  Si antena_izq()
    giro_der()
  Si antena_der()
    giro_izq()
```

```
Por siempre
  avanzar()
  Si luz_izq()>luz_der()
    giro_der()
  Sino
    giro_izq()
```

Huye de la luz, pegándose a las paredes.

```
Por siempre
  avanzar()
  Si antena_izq()
    giro_der()
  Sino si antena_der()
    giro_izq()
  Sino si luz_izq()>luz_der()
    giro_der()
  Sino
    giro_izq()
```

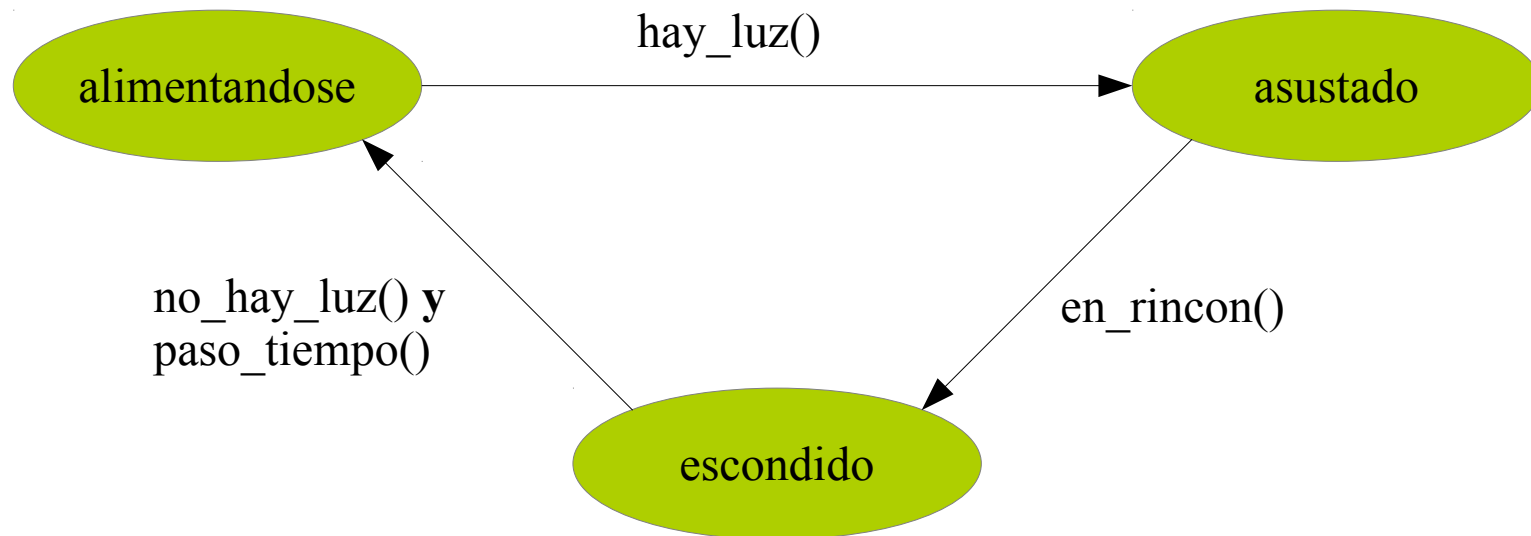
¡El orden importa!

Combinando comportamientos

- Los comportamientos solo viven en el presente: solo sentido y acción, no tienen memoria.
- Hay veces que mantener estado es inevitable: una percepción en el pasado puede condicionar un comportamiento actual. En el ejemplo:
 - Cuando se prende la luz empieza a huir y esconderse, sin importar si la luz se apaga: está *asustado*.
- El estado determina la relación entrada – salida:
En estado diferentes respondemos de manera diferente a las mismas entradas.
- Cuanto menos estados, mejor.

Combinando comportamientos

- Estados: {alimentandose, asustado, escondido}



Combinando comportamientos

- Cambiar a a un estado puede verse como un tipo de acción.
- Decidir en función de un estado puede verse como leer un sensor. A veces es posible reemplazar un estado con sensado:

```
Si estado=='alimentandose'  
y encuentre_miga()  
    levantar_miga()  
    estado = 'cargando'  
...  
Si estado=='cargando'  
    //volver al nido
```

```
Si estado=='alimentandose'  
y encuentre_miga()  
    levantar_miga()  
...  
Si estado=='alimentandose'  
y siento_carga()  
    //volver al nido
```


Desarrollando un robot reactivo

Cuando huye, al llegar a un rincón espera la oscuridad y que se le pase el susto.

```
Si estado=='asustado'  
  Si antena_izq() y antena_der()  
    estado='escondido'  
  Sino si antena_izq()  
    giro_der()  
  Sino si antena_der()  
    giro_izq()  
  Sino si luz_izq()>luz_der()  
    giro_der()  
  Sino  
    giro_izq()
```

```
Si estado=='comiendo'  
  Si hay_luz()  
    estado='asustado'  
  Si paso_tiempo()  
  o antena_izq() o antena_der()  
    cambio_dirección()
```

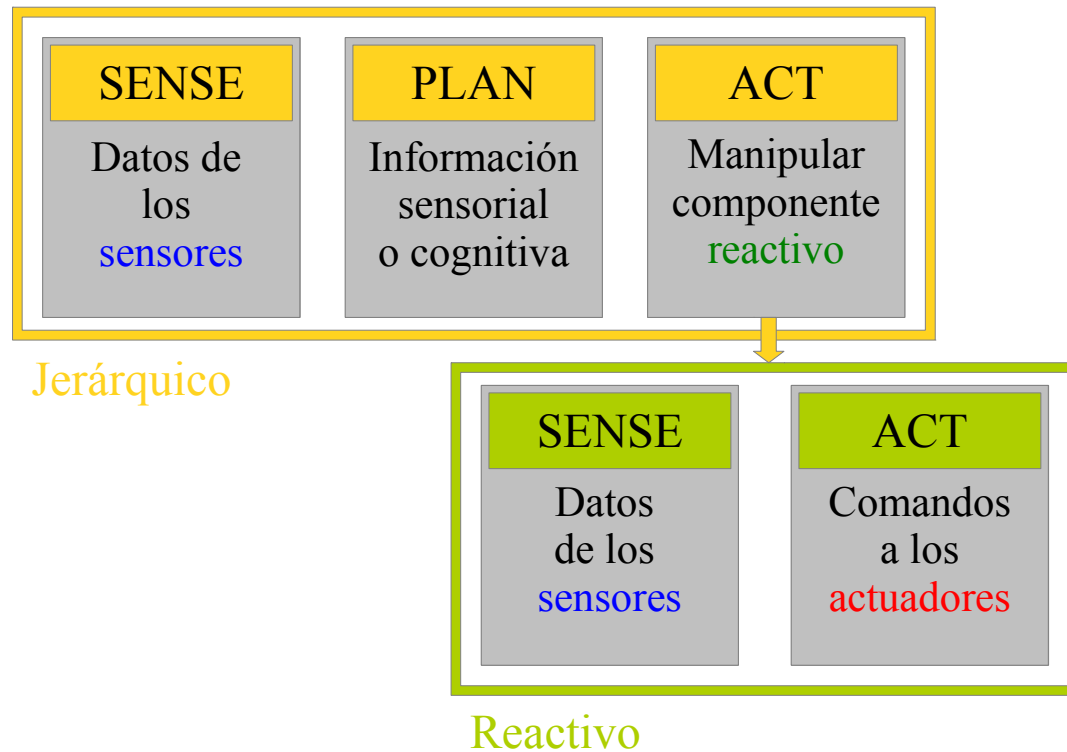
```
Si estado=='escondido'  
  Si paso_tiempo()  
  y no hay_luz()  
    estado='comiendo'
```

Recomendaciones

- Debe haber un único bucle `Por siempre`
- Antes de ponerse a programar, dominar las acciones que se pueden realizar (API), y cómo se comporta el sensado.
- Cada acción debe ser instantánea y devolver el control enseguida.
- No debe haber bucles `for`, `repeat`, etc.
- No debe haber `sleep`.
- Si va a usar una máquina de estados, dibújela primero.
- Intente reemplazar estados internos con sensado.
- Pruebe los comportamientos por separado.

Paradigma híbrido

- A veces es ventajoso planificar.
- A veces es inevitable planificar.



Referencias

- Cybernetic Zoo
<http://cyberneticzoo.com/>
- Fundamentos de programación y robótica
<https://eva.fing.edu.uy/course/view.php?id=215>
- Behavior-Based Robotics, Arkin, MIT Press, 1998.

Fin

Gracias