

## Inteligencia Artificial

### Trabajo práctico 1

***Objetivo:** construir un agente inteligente, para comprender como éste se relaciona con el mundo en donde se desenvuelve y comprender cómo utiliza búsqueda parra tomar las decisiones sobre las acciones que puede emprender.*

Suponga que tiene que diseñar dentro de un videojuego un agente-PAC-MAN que tenga como objetivo: la supervivencia dentro de su mundo, para lo cual necesita alimentarse y evitar ser eliminado por sus enemigos. Inicialmente, tiene una cierta cantidad de energía que le permite moverse por su mundo, compuesto por celdas o habitaciones, en busca de alimento (ver figura 1). El tamaño del mundo es 4x4. Es un mundo estático, los enemigos y el alimento están fijos, el que se mueve por el mundo es el agente.

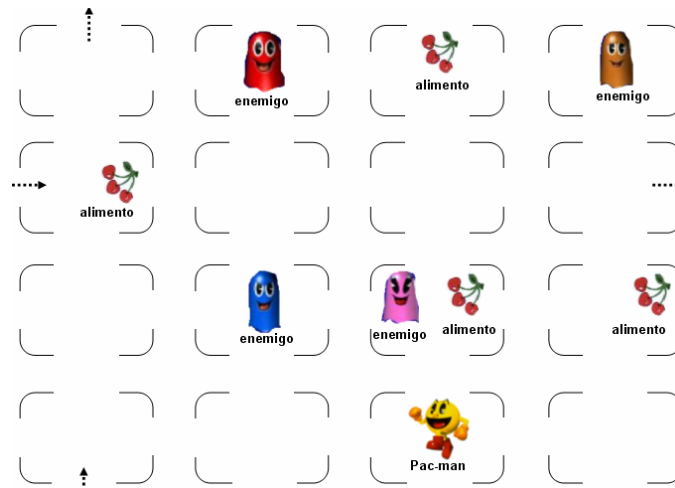


Figura 1: el agente-Pac-man y su mundo

Para poder aumentar sus energías, Pac-Man necesita alimentarse, lo que implica llegar hasta las celdas donde se encuentra la comida. Si llega a una celda donde existe un enemigo, entrarán en lucha y el enemigo siempre morirá y Pac-Man siempre perderá algo de energía. Si su energía queda en cero, significará que ha muerto.

Pac-Man no tiene conocimiento de todo su entorno, sino que lleva consigo un equipo con sensores que le provee la siguiente información:

- un **contador de energía**, que le indica su valor de energía actual. Si este valor está en cero es porque el Pac-Man ha muerto. Se conoce que, inicialmente, como máximo Pac-Man puede tener 50 puntos de energía.
- **sensores de celdas vecinas**, que, según lo que haya en las celdas adyacentes a su posición actual, emiten un 2 si hay alimento, un 1 si hay un enemigo y 0 si no hay nada.

Tenga en cuenta que el agente no tiene control sobre su energía; cuando entra en guerra o come, no conoce exactamente cuanta energía pierde o gana. Tampoco conoce la energía de sus enemigos como para decidir si es o no bueno entrar en lucha.

Las acciones que puede realizar el Pac-Man son:

- avanzar de celda en celda (solamente movimientos verticales y horizontales, no diagonales)
- pelear
- comer

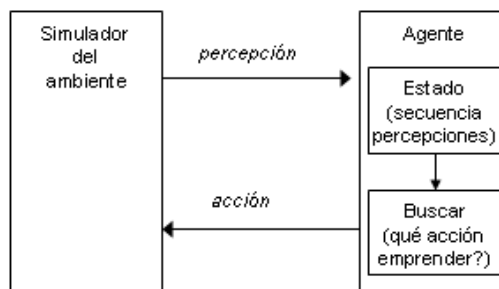
Implemente el Pac-Man como un agente inteligente basado en objetivos. Su meta es sobrevivir. Pac-Man no tiene conocimiento total del universo, sabe cuántas habitaciones hay en su mundo y su posición inicial dentro del mismo, pero no sabe dónde hay alimento o enemigos. En su estado interno mantiene la secuencia de percepciones para aprender de ellas y tomar mejores decisiones. Es decir, al no conocer el ambiente, mantendrá una imagen propia de ese ambiente (no puede acceder al ambiente real.)

El desarrollo de este trabajo práctico se hará en 3 etapas:

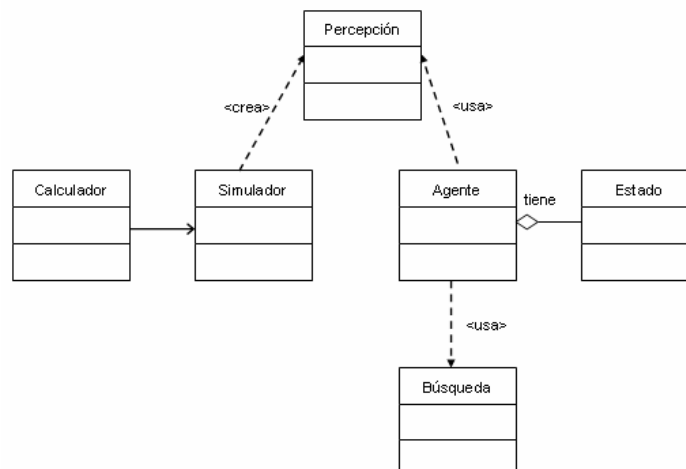
- 1) Definición del agente-Pac-Man. Para ello especifique: arquitectura del agente (diagrama de clases), percepciones, objetivo y representación de estado. Cada grupo debe entregar un informe con sus decisiones de diseño, por correo electrónico, a la cuenta de la cátedra y hacer una presentación pública de su propuesta de solución para implementar el agente. Esta actividad de puesta en común promueve la comparación de soluciones con otros grupos y la discusión en clase de las distintas propuestas. De este modo cada grupo controla su diseño de la solución antes de pasar a su implementación y así evitar errores. **Entrega: 26/03/07**
- 2) Definición de cómo el Agente usando búsqueda decide qué acción ejecutar: se deben definir los operadores de búsqueda, estado inicial y final del problema, prueba de meta y estrategia de búsqueda a utilizar. La entrega de esta etapa se hará con un informe con sus decisiones de diseño, por correo electrónico, a la cuenta de la cátedra, y una presentación en clase por grupo, del mismo modo que en la etapa 1. **Entrega: 16/04/07**
- 3) Definición del simulador del ambiente y su interacción con el agente: se debe definir el simulador del mundo y cómo interactúa con el agente. El simulador obtendrá información acerca de inicialización del mundo, posición inicial del agente dentro del mundo, cálculos de energía y performance del agente, de la clase *calculador* que será provista por la cátedra. Cada grupo hará la entrega del TP1 (etapa 1+etapa 2+etapa 3) por correo electrónico a la cuenta de la cátedra. Junto con el programa fuente se debe presentar un informe que contenga cualquier suposición o consideración que deba ser tenida en cuenta para su prueba y corrección, secuencia de acciones ejecutadas por Pac-man para lograr su objetivo y al menos un árbol de búsqueda utilizado para la toma de decisiones. **Entrega: 21/05/07**

Notas:

- ✓ Todas las entregas deben hacerse por e-mail: **ia@frsf.utn.edu.ar**
- ✓ Utilice **Java** para desarrollar el práctico.
- ✓ Ayuda para la arquitectura del agente:



- ✓ Ayuda para el diagrama de clases del agente



La clase *Calculador* es proporcionada por la cátedra. Esta clase es usada por el simulador para calcular posiciones iniciales del Pac-Man, enemigos y alimento. También calcula la energía que tiene el Pac-Man después de haber realizado alguna acción.

Cuando se crea una instancia de *Calculador*, este le asigna un valor aleatorio entre 1 y 50 a la energía del Pac-Man y de los enemigos (todos los enemigos tienen la misma energía). También asigna una posición aleatoria al Pac-Man.

Cuando el simulador crea el ambiente debe determinar la posición de los enemigos. Esto lo hará invocando un método en *Calculador* llamado *inicializarEnemigo()* el cual no tiene argumentos y retorna un array de dos elementos conteniendo los valores de la fila y columna del enemigo.

El método *calcularEnergiaPacman(String a)* debe ser invocado por el simulador cuando quiere conocer la energía que tiene el pacman después de realizar una acción. El cálculo de la energía es una fórmula que solo es conocida por la clase *Calculador* es por eso que el simulador debe invocarla para conocer la nueva energía y armar así la percepción que le dará al agente. El argumento del método es un string que representa la acción que el pacman realizó y por la cual quiere conocer su nueva energía. Los valores que puede tomar el argumento es: “comer”, “pelear” o “avanzar”.

También, el simulador tiene que calcular el desempeño del agente, para ello por cada acción que emprende el pacman se le solicitará a la clase *Calculador* que calcule la performance del agente. Para ello se debe invocar el método *calcularPerformance(String a)* donde el argumento es un string que representa la acción del agente y que puede tomar los mismos valores que fueron mencionados anteriormente. El método no retorna la performance. Sólo cuando la simulación haya finalizado entonces el simulador obtendrá el valor de desempeño del agente invocando al método *getPerformance()*, que retorna un entero indicando qué tan bien se desempeñó el agente.

Este valor será usado por la cátedra para compara el desempeño de los distintos agentes propuestos por el alumno a modo de competencia.

**para consultas y entrega del TP**  
**ia@frsf.utn.edu.ar**