

## Proyecto Final Sistemas Inteligentes 2021-I

El objetivo es realizar el análisis, diseño y desarrollo de un programa que muestra como un león aprende a cazar un impala bajo ciertas condiciones.

### Planteamiento del problema.

Existe una manada de impalas que todas las mañanas va a beber agua a un abrevadero. Un joven león, cuya madre murió a manos de cazadores furtivos, debe aprender a cazar solo o de lo contrario morirá de hambre. Para ello se debe desarrollar un sistema en el que se simule cómo un león **aprende** a cazar a un impala. De esta manera, el joven león podrá salvar su vida.

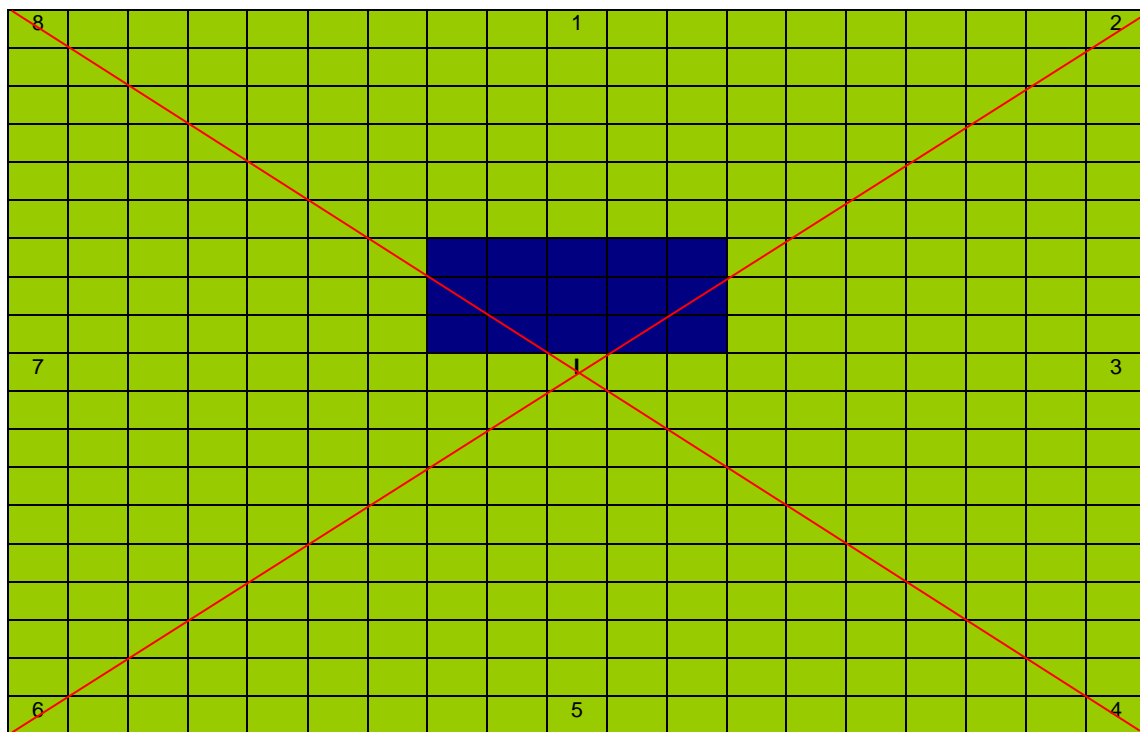


Figura 1. Mapa del abrevadero.

Por uno de esos misterios de la naturaleza, en el abrevadero siempre hay un solo impala (que será el objetivo del león), parado en la posición que se indica en el mapa con el símbolo I, y con su mirada siempre hacia el norte.

El león puede estar situado en cualquiera de las 8 posiciones marcadas en el mapa. Su objetivo es acercarse poco a poco al impala, hasta lograr una posición en la que pueda iniciar un ataque. Si el impala ve acercarse al león, o si éste comienza su ataque demasiado lejos o demasiado cerca de su víctima, el impala iniciará su huida y no podrá ser alcanzado por el depredador (ver detalles del comportamiento de los animales más adelante). Entonces, el león debe aprender cómo acercarse al impala (cuándo avanzar y cuándo esconderse), y cuándo comenzar su ataque.

## Especificaciones.

El impala y el león muestran su comportamiento en el programa por medio de acciones, cuyas consecuencias afectan el mundo virtual creado por el sistema. Todas las acciones se realizan en una unidad de tiempo  $T$ . Siempre, primero el impala actúa y el león reacciona (“en paralelo”) a dicha actuación. Una vez realizadas las respectivas acciones en una unidad de tiempo  $T$ , el sistema verifica cómo se modificó el mundo-virtual.



Entonces, una incursión de cacería del león se puede representa de la siguiente manera:

$T_1$  (inicio de la incursión),  $T_2$ ,  $T_3$  ...  $T_n$  (fin de la incursión).

El impala puede realizar las siguientes acciones:

1. Ver a la izquierda (el ángulo de visión del impala se muestra con líneas rojas en el mapa de la figura 1).
2. Ver a la derecha.
3. Ver al frente.
4. Beber agua (cuando está bebiendo sólo puede ver su reflejo en el agua).
5. Huir (a partir de que el impala comienza a huir en  $T_n$ , para  $T_n$  y  $T_{n+1}$  el impala recorre 1 cuadro/ $T$ ; en  $T_{n+2}$  recorre 2 cuadros/ $T$ ; en  $T_{n+3}$  recorre 3 cuadros/ $T$ , y así sucesivamente. El impala siempre huirá en línea recta hacia el este o el oeste).

El león puede realizar las siguientes acciones:

1. Avanzar (El león siempre avanza 1 cuadro en línea recta hacia el impala desde su posición actual).
2. Escondarse entre la maleza (el impala no puede ver al león cuando éste está escondido).
3. Atacar (El león ataca a una velocidad constante de 2 cuadros/ $T$ ).
4. Situarse en cualquiera de las 8 posiciones permitidas (ver figura 1).

El impala siempre huirá en los siguientes casos:

- Cuando ve al león (el impala verá al león cuando éste se encuentre dentro de su ángulo de visión. El impala NO podrá ver al león, aunque esté dentro de su ángulo de visión, si el felino se encuentra escondido entre la maleza).
- Cuando el león comienza un ataque (independientemente de la posición del león y hacia dónde esté viendo el impala).
- Cuando existe una distancia menor a tres cuadros entre el león y el impala.

Cada unidad de tiempo T el sistema debe verificar si existen las condiciones necesarias para que el impala huya. Una vez que el león inicia un ataque no podrá realizar otra acción. Una vez que el impala comienza a huir no podrá realizar otra acción. La incursión de cacería termina cuando el sistema detecta que:

- El león no podrá alcanzar al impala.
- El león alcanza al impala.

### Ejemplos.

Supongamos que el león está en la posición 1 (ver mapa).

T=1:

Impala = Ver a la izquierda.

León = Avanzar.

Puesto que el león está fuera del ángulo de visión del impala, puede avanzar un cuadro hacia su víctima.

T=2:

Impala = ver al frente.

León = Escondarse entre la maleza.

Puesto que el león está escondido, el impala no puede verlo. El león mantiene su misma posición.

T=3:

Impala = ver al frente.

León = Avanza.

Puesto que el impala ve al león, huye del abrevadero y se frustra el ataque del león.

Termina la incursión de cacería con un fracaso.

El programa debe tener dos modos de operación:

- Fase de entrenamiento.
- Cacería paso a paso.

### Fase de entrenamiento.

El león debe aprender a cazar con base en su experiencia. Para ello, se debe realizar un entrenamiento el cual consiste en simular incursiones de cacería. El león debe recolectar el conocimiento que surge de estas experiencias y usarlo para guiar sus acciones en cada nueva incursión. Cuando el león no sepa qué hacer por falta de suficiente conocimiento en el sistema, podrá escoger al azar la siguiente acción a realizar. **Bajo ninguna circunstancia el diseñador podrá indicar en forma explícita las acciones que debe ejecutar el león.** El objetivo del proyecto es que éste aprenda por si solo.

Ciclo de entrenamiento.- El sistema debe estar diseñado de tal forma que se pueda ciclar para realizar MUCHAS incursiones de cacería en forma automática. Debe ser posible para el usuario definir el número de incursiones a realizar en un ciclo de entrenamiento.

Un elemento muy importante del entrenamiento es la posición inicial del león en cada incursión de cacería. Debe existir un parámetro que permita definir las posibles posiciones iniciales para un ciclo de entrenamiento.

En caso de que exista más de una, el sistema escoge al azar una de ellas para cada incursión. Por ejemplo, supongamos que realizo un ciclo de entrenamiento de 1,000 incursiones donde sólo se permite tener como posición inicial el punto 3:

Ciclo de entrenamiento = 1,000 incursiones.

Posición Inicial válida = [3]

El león debe convertirse en un experto cuando comienza su incursión de ataque en la posición 3, pero muy malo cuando comienza en cualquier otra posición.

También sería muy interesante entrenar al león desde cualquier punto inicial de ataque con excepción del 5, y ver si descubre cómo atacar desde dicho punto 5.

Ciclo de entrenamiento = 10,000 incursiones.

Posición Inicial válida = [1,2,3,4,6,7,8]

Durante la fase de entrenamiento, el impala tiene dos modos de comportamiento:

- aleatorio: el sistema escoge al azar las acciones que ejecuta el impala durante toda la incursión de cacería.
- programado: el usuario selecciona una secuencia de acciones las cuales se repiten cíclicamente.

### **Manejo del conocimiento.**

El adecuado manejo del conocimiento constituye una parte muy importante del sistema. Como parte del proyecto debes decidir todo lo referente a la representación del conocimiento. La única restricción es que, para hacer más eficiente la toma de decisiones del león, el sistema debe ser capaz de abstraer (generalizar) su conocimiento. Por ejemplo, si el sistema tiene representada en su base de conocimientos el hecho de que si el león está en la posición 1 y el impala está volteando a la izquierda el león puede avanzar, y en otra estructura tiene representado que si el león está en la posición 1 y el impala está volteando a la derecha el león puede avanzar, debe generalizar este conocimiento y crear una sola estructura para representarlo. Es decir, en forma automática debe crear una estructura en la que quede establecido que , si el león está en la posición 1 y el impala está volteando a la derecha o la izquierda, el león puede avanzar.

El diseño de la base de conocimientos requiere de un análisis cuidadoso y de varias pruebas de escritorio antes de comenzar a escribir su código. Entre otras cosas, debe quedar claro:

- ¿Cómo se va a representar el conocimiento?
- ¿Cómo se va a guardar ese conocimiento?
- ¿Cómo se va a acceder a ese conocimiento?
- ¿Cómo se va a actualizar el conocimiento?
- ¿Cómo se va a abstraer?

Es importante que se incluya en el programa una opción para:

- Detener un ciclo de entrenamiento sin perder el conocimiento ya generado.
- Salvar en archivos el conocimiento producido por el sistema.

- Cargar en memoria conocimiento producido anteriormente.

**Cacería paso a paso.**

Durante este modo de operación, el sistema realiza una incursión de cacería la cual puede ser seguida paso a paso por el usuario del sistema. Debe ser posible ver por unidad de tiempo qué acción realiza el impala y cómo reacciona el león.

Debe existir la opción de que el usuario le pregunte al sistema el porqué del comportamiento del león.

Debe ser posible visualizar, en un formato entendible, toda la base de conocimientos del programa (por ejemplo, se puede imprimir en un archivo de texto).