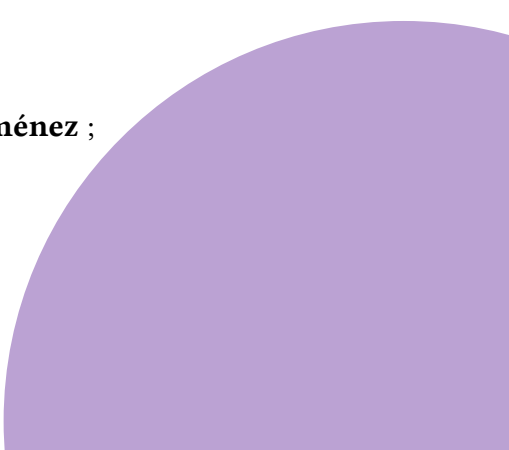


# Seminario 1

## Redes Bayesianas en Juegos

2024-02-25

**Pablo Santana González ; Pablo Hernández Jiménez ;**  
Universidad de La Laguna  
Curso 2023-2024  
*Inteligencia Artificial Avanzada*



# Índice

I. Introducción .....	3
I.1. Porcentaje de Participación .....	3
II. Red Utilizada .....	4
II.1. Descripción de la Red .....	4
II.2. Respuesta a Cuestion Planteada en el Apartado 2 .....	4
¿Por qué crees que se ha hecho así? .....	4
¿Cómo sería la red puesta de forma causal? ¿Cuál el tamaño de las tablas? .....	4
III. Personalidad del Bot .....	5
IV. Justificación de las Tablas de Probabilidad Condicional .....	5
IV.1. Estado del Bot $S_t$ .....	5
IV.2. Estado del Bot $S_{t+1}$ .....	6
IV.3. Estado del Sensor del Bot $H$ .....	7
IV.4. Estado del Sensor del Bot $W$ .....	7
IV.5. Estado del Sensor del Bot $OW$ .....	7
IV.6. Estado del Sensor del Bot $HN$ .....	7
IV.7. Estado del Sensor del Bot $NE$ .....	8
IV.8. Estado del Sensor del Bot $PW$ .....	8
IV.9. Estado del Sensor del Bot $PH$ .....	8
V. Ejemplos de Calculo con el <b>GeNIe</b> .....	9

## I. Introducción

En este seminario se va a realizar el diseño y análisis de una red bayesiana que modela el comportamiento de un bot en un videojuego.

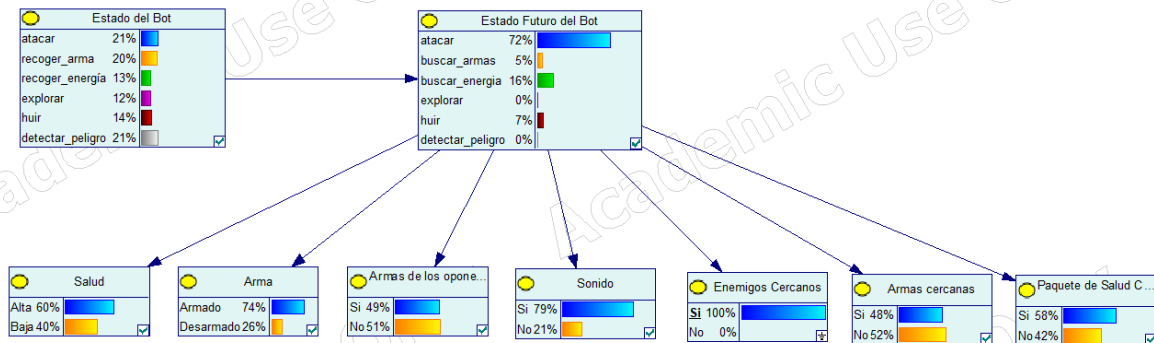
La implementación de la red bayesiana se ha realizado con el software **GeNIe** y se ha exportado a un archivo **.xds1**. Y está basada en la publicación [« Teaching Bayesian Behaviours to Video Game Characters »](#).

### I.1. Porcentaje de Participación

Identificador ALU	Apellidos	Nombres	Participación
alu0101*****	Santana González	Pablo	50%
alu0101495934	Hernández Jiménez	Pablo	50%

## II. Red Utilizada

Fig. 1 — « Red Bayesiana »



### II.1. Descripción de la Red

$$\begin{aligned}
 &P(S_t, S_{t+1}, H, W, OW, HN, NE, PW, PH) \\
 &= P(S_t) \\
 &P(H|S_{t+1}) \\
 &P(W|S_{t+1}) \\
 &P(OW|S_{t+1}) \\
 &P(HN|S_{t+1}) \\
 &P(NE|S_{t+1}) \\
 &P(PW|S_{t+1}) \\
 &P(PH|S_{t+1})
 \end{aligned}$$

### II.2. Respuesta a Cuestion Planteada en el Apartado 2

A la cuestion « La factorización no sigue la causalidad pues variables en tiempo  $t$  dependen de variables en tiempo  $t+1$ . » Se plantean las siguientes preguntas:

#### ¿Por qué crees que se ha hecho así?

Porque si la causalidad se hubiera representado de la manera tradicional, la red sería mucho más compleja y las tablas de probabilidad condicional serían mucho más grandes, tanto que sería imposible de manejar.

#### ¿Cómo sería la red puesta de forma causal? ¿Cuál el tamaño de las tablas?

En la red causal, el nodo de  $S_{t+1}$  dependería de todos los nodos de  $S_t$ ,  $H$ ,  $W$ ,  $OW$ ,  $HN$ ,  $NE$ ,  $PW$ ,  $PH$ . Por lo que el tamaño de las tablas de probabilidad condicional sería de  $2^7 * 6 = 768$  combinaciones a rellenar.

### III. Personalidad del Bot

Hemos considerado que el bot tiene una personalidad agresiva, por lo que atacará siempre que pueda. Sin embargo, tampoco queremos que sea un bot suicida, por lo que si detecta peligro, atacará con una probabilidad casi segura si no está herido, pero si lo está buscará energía hasta que la encuentre.

### IV. Justificación de las Tablas de Probabilidad Condicional

A continuación se justificará la tabla de probabilidad condicional de cada nodo de la red.

#### IV.1. Estado del Bot $S_t$

$S_t$	%
atacar	0.1666666666666667
recoger_arma	0.1666666666666667
recoger_energía	0.1666666666666667
explorar	0.1666666666666667
huir	0.1666666666666667
detectar_peligro	0.1666666666666667

Para esta tabla se ha considerado que el bot tiene una probabilidad uniforme de realizar cada acción. Puesto que no depende de ningún otro nodo.

IV.2. Estado del Bot  $S_{t+1}$ 

$S_t$	atacar	buscar_armas	buscar energia	explorar	huir	detectar_peligro
atacar	0.9999	0.83329	1e-05	0.249995	0.33332	0.99995
buscar_armas	1e-05	0.16666	1e-05	0.249995	1e-05	1e-05
buscar energia	1e-05	1e-05	0.99995	0.249995	1e-05	1e-05
explorar	1e-05	1e-05	1e-05	0.249995	0.33332	1e-05
huir	1e-05	1e-05	1e-05	1e-05	0.33332	1e-05
detectar_peligro	1e-05	1e-05	1e-05	1e-05	1e-05	1e-05

Donde cada columna representa el estado del bot en el tiempo  $t$  y cada fila representa el estado del bot en el tiempo  $t + 1$ .

Para esta tabla se han tenido las siguientes consideraciones para cada estado:

- $S_t = \text{atacar}$ : El bot tiene una probabilidad alta de seguir atacando y una probabilidad infinitesimal de realizar cualquier otra acción.

Por su personalidad agresiva, cuando el bot ataca seguirá atacando hasta que no detecte peligro.

- $S_t = \text{buscar\_armas}$ : El bot tiene una probabilidad alta de que en  $S_{t+1}$  de atacar y una probabilidad más baja de seguir buscando armas.

Si esta buscando armas, es porque no tiene armas, por lo que tiene una probabilidad alta de atacar en el siguiente estado. Y si no encuentra armas, lo más probable es que siga buscando.

- $S_t = \text{buscar\_energia}$ : El bot tiene una probabilidad alta de seguir buscando energía y una probabilidad más baja de realizar cualquier otra acción.

Aunque su personalidad es agresiva, si no tiene energía, buscará energía hasta que la encuentre.

- $S_t = \text{explorar}$ : El bot tiene una probabilidad uniforme de realizar cualquiera de las acciones (**atacar**, **buscar\_armas**, **buscar\_energia**, **explorar**) y una probabilidad infinitesimal de realizar cualquier otra acción.

- $S_t = \text{huir}$ : El bot tiene una probabilidad uniforme de seguir huyendo, atacar o huir y una probabilidad infinitesimal de realizar cualquier otra acción.

- $S_t = \text{detectar\_peligro}$ : El bot tiene una probabilidad casi segura de atacar y una probabilidad infinitesimal de realizar cualquier otra acción.

### IV.3. Estado del Sensor del Bot $H$

Salud	atacar	buscar_armas	buscar_energia	explorar	huir	detectar_peligro
Alta	0.7	0.5	0.3	0.7	0.3	0.5
Baja	0.3	0.5	0.7	0.3	0.7	0.5

Para esta tabla se ha considerado que el bot tiene una probabilidad alta de atacar si tiene la salud alta y una probabilidad alta de buscar energía si tiene la salud baja. Y una probabilidad alta de huir si tiene la salud baja.

### IV.4. Estado del Sensor del Bot $W$

Armas	atacar	buscar_armas	buscar_energia	explorar	huir	detectar_peligro
Armado	0.9	0.1	0.5	0.7	0.1	0.5
Desarmado	0.1	0.9	0.5	0.3	0.9	0.5

Para esta tabla se ha considerado que el bot tiene una probabilidad alta de atacar si tiene armas y una probabilidad alta de buscar armas si no tiene armas. Si no tiene armas, las probabilidades de huir son extremadamente alta. También tiene una probabilidad alta de explorar si tiene armas.

### IV.5. Estado del Sensor del Bot $OW$

Armas Oponente	atacar	buscar_armas	buscar_energia	explorar	huir	detectar_peligro
Armado	0.49	0.5	0.5	0.5	0.5	0.5
Desarmado	0.51	0.5	0.5	0.5	0.5	0.5

Para esta tabla se ha considerado que el bot tiene una probabilidad casi uniforme de realizar cualquier acción si el oponente tiene armas o no. Es decir que le importa poco si el oponente tiene armas o no.

### IV.6. Estado del Sensor del Bot $HN$

Sonido	atacar	buscar_armas	buscar_energia	explorar	huir	detectar_peligro
Si	0.9	0.5	0.5	0.7	0.5	0.9
No	0.1	0.5	0.5	0.3	0.5	0.1

Para esta tabla se ha considerado que el bot tiene una probabilidad alta de atacar si detecta un sonido y una probabilidad alta de detectar peligro si detecta un sonido. También tiene una probabilidad alta de explorar si detecta un sonido.

#### IV.7. Estado del Sensor del Bot NE

Enemigos Cercanos	atacar	buscar_armas	buscar_energia	explorar	huir	detectar_peligro
Si	0.9999	0.6	0.6	1e-05	1	1e-05
No	1e-05	0.4	0.4	0.9999	0	0.9999

Para esta tabla se ha considerado que el bot tiene una probabilidad alta de atacar si tiene enemigos cercanos, una probabilidad alta de buscar armas si no tiene enemigos cercanos y sólo huirá si hay enemigos cercanos.

#### IV.8. Estado del Sensor del Bot PW

Armas Cercanas	atacar	buscar_armas	buscar_energia	explorar	huir	detectar_peligro
Si	0.5	0.1	0.5	0.5	0.5	0.5
No	0.5	0.9	0.5	0.5	0.5	0.5

Para esta tabla se ha considerado que el bot tiene una probabilidad alta de atacar si tiene armas cercanas y una probabilidad alta de buscar armas si no tiene armas cercanas.

#### IV.9. Estado del Sensor del Bot PH

Energía Cercana	atacar	buscar_armas	buscar_energia	explorar	huir	detectar_peligro
Si	0.7	0.5	0.1	0.5	0.5	0.5
No	0.3	0.5	0.9	0.5	0.5	0.5

Para esta tabla se ha considerado que el bot tiene una probabilidad alta de buscar energía si tiene energía cercana y una probabilidad más alta de atacar si tiene energía cercana.



## V. Ejemplos de Calculo con el **GeNIe**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguere possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et.