



## Inteligencia Artificial para Videojuegos

Grado en Desarrollo de Videojuegos

Prácticas del curso

Departamento de Ingeniería del Software e Inteligencia Artificial

Facultad de Informática

Universidad Complutense de Madrid



## Práctica 2: El hilo de Ariadna

Fecha de entrega: **9 de abril de 2021**

**Importante:** Haz la entrega en tiempo y forma, subiendo al campus virtual un fichero *IAVP2GXX.txt* donde *XX* sea el número del grupo al que representas, con dos dígitos. Dentro de ese fichero incluye los datos del grupo, una breve explicación de la práctica y los enlaces al repositorio donde está el fichero *README.md* con toda la documentación técnica enlazada, la carpeta *IAVGXX* con todo el proyecto de la asignatura (código fuente, recursos y plugins), el ejecutable generado para Windows de 64bits *IAVP2GXX.exe* (con sus carpetas y ficheros acompañantes) y el video con vuestros comentarios *IAVP2GXX.mp4*.

### 1. Introducción

*“El segundo canto de El Hilo de Ariadna narra las aventuras de Teseo y su llegada a la isla de Creta, en busca de la cabeza del Minotauro. Ariadna, decidida a salvar al famoso héroe, le entregó un ovillo de hilo para que el laberinto perdiera por completo su secreto, y así Teseo pudiera escapar vivo de la isla. La princesa cretense empezaba a enamorarse perdidamente de él, y él le ofreció todo tipo de promesas que un hombre puede hacerle a una mujer.*

*El destino de Teseo estaba escrito, mató al Minotauro y se llevó consigo a Ariadna hasta la isla de Naxos, en la cual se celebró hasta el amanecer la gloria del héroe. A la mañana siguiente, Teseo abandonó a su prometida, mientras ella dormía en la playa. Por eso el canto termina con la maldición que Ariadna realiza sobre su amado...”*

El conocido mito griego (Figura 1) nos sirve como excusa para profundizar ahora en el problema de la navegación de entornos virtuales, tan habitual en el mundo de los videojuegos. En la historia tenemos un protagonista, Teseo, en principio heroico. Hay también un villano, el temido Minotauro. Tenemos también el propio entorno donde se encuentran ambos personajes, el laberinto del Minotauro, y por último, el hilo mágico que Ariadna entregó a Teseo. Este hilo tiene la propiedad de burlar el embrujo del laberinto y ayudarnos a salir de él. Aunque en el mito se habla de la victoria de Teseo sobre el Minotauro, nosotros vamos a aplicarle la “maldición” que le lanza Ariadna de forma anticipada, y haremos que sea imposible que pueda acabar con la dichosa criatura. Lo único que hará el héroe cuando se encuentre próximo al monstruo es entablar combate, lo que en la práctica supone ralentizar el movimiento de ambos.



**Figura 1.** Óleo sobre lienzo de Niccolò Bambini donde la princesa Ariadna entrega su famoso ovillo de hilo al héroe ateniense Teseo, bajo la tutela de la diosa Atenea.

En esta ocasión, crearemos un prototipo en el que el jugador controla el movimiento de Teseo por los pasillos del laberinto. La bestia mitológica por su parte será un agente inteligente que se dedica a merodear por allí, siguiendo a Teseo en caso de que lo vea. El hilo de Ariadna funcionará, cuando lo activemos, dibujando con una línea blanca el camino de menor coste que puede conducir a Teseo hasta la *baldosa de salida*. El hilo, al ser mágico, tiene en cuenta todos los costes, ya que la baldosa donde se encuentra el Minotauro es intransitable, y moverse a las 9 baldosas vecinas tienen 5 veces mayor coste que el movimiento a través de una baldosa normal.

Este prototipo servirá para probar el algoritmo A\* como técnica de navegación (búsqueda de caminos con información heurística), de los más usados en videojuegos, en combinación con comportamientos de dirección como el seguimiento y técnicas como el suavizado del camino.

## 2. Planteamiento del proyecto

Desarrolla un prototipo de IA para Videojuegos, dentro de un entorno virtual laberíntico, con un agente inteligente que merodea por allí y un avatar, en principio controlado por el jugador. Al activar el hilo de Ariadna Teseo pasará a ser controlado por la máquina y procederá a salir del laberinto de manera tranquila pero automática, hasta que lo desactivemos. El prototipo será fácilmente usable y podrá utilizar cualquier tipo de estructura (ya sean habitaciones o pasillos) para representar el entorno, el avatar y el enemigo.



**Figura 2.** Ejemplo de representación del laberinto del Minotauro.

La entrega será realizada en tiempo y forma [1 pto.], el proyecto estará bien diseñado, organizado y comentado [1 pto.], y la documentación explicará con claridad cuáles fueron la implementación utilizada [1 pto.], las pruebas realizadas [1 pto.] y los resultados obtenidos.

El prototipo ejecutable será usable y funcional, permitiendo:

- Mostrar el entorno virtual laberíntico (el laberinto del Minotauro), con un esquema de división de *grafo de baldosas* que incluirá una baldosa de salida, donde se ubica inicialmente el avatar (Teseo). Debe haber varios caminos alternativos para llegar a la salida, algunos serán anchos y otros estrechos. Este estará controlado, *a priori*, por el jugador mediante los *cursores*. En el centro del laberinto estará el agente inteligente que representa al enemigo (el Minotauro) en el centro del laberinto, que realizará un merodeo monótono y lento, pasando a seguimiento algo más rápido si en su línea de visión detecta a Teseo [1 pto.].
- Representar el hilo de Ariadna (camino más corto) pintado con línea blanca, y realizando una navegación automática desde Teseo hasta la baldosa de salida, mientras está pulsada la *barra espaciadora* [1 pto.]. Se mostrará algo de información (*métricas*) sobre la ejecución del algoritmo, como el número de nodos explorados [1 pto.] o el tiempo tardado en calcular el camino mínimo [1 pto.].
- Elegir (activando o desactivando la funcionalidad con la tecla *S*) si *suavizar* o no el camino generado por el algoritmo anterior, reduciendo un poco sus nodos [1 pto.].
- Desarrollar el movimiento completo de Teseo, que mientras tenemos pulsada la *barra espaciadora*, va moviéndose automáticamente hacia la baldosa de salida. Esto hace desaparecer la parte del hilo que ya no se utiliza, al completo en cuanto se suelta la barra espaciadora y se vuelve al movimiento manual, el más rápido de todos [1 pto.].

### 3. Restricciones y consejos

A la hora de desarrollar este proyecto es obligatorio:

- No utilizar herramientas o plugins de terceros, ni reutilizar código ajeno al del profesor.
- Documentar claramente los algoritmos y heurísticas utilizados.
- Diseñar y programar de la manera más limpia y elegante posible, separando la parte visual e interactiva del juego, del modelo y las técnicas de IA implementados.
- Evitar, en la medida de lo posible, el uso de recursos audiovisuales pesados.

Se pueden organizar todas las prácticas en un único proyecto, siempre que se estructuren en distintas carpetas, espacios de nombres, escenas, recursos, etc. Pensando tanto en las pruebas como en la revisión del profesor, y también con ánimo de reutilizar el esfuerzo de desarrollo entre ellas, conviene crear menú y HUD cómodos con el título de la práctica correspondiente, vuestros datos, instrucciones de uso, etc. Su manejo debe ser ágil e intuitivo para poder repetir rápidamente todas las pruebas que sean necesarias con las variaciones que hagan falta.

### 4. Referencias y ampliaciones

Como punto de partida para la investigación, además de la bibliografía de la asignatura, puedes utilizar las siguientes referencias. En ningún caso debes replicar el código que encuentres por ahí sin entenderlo y asegurarte primero de que funciona *exactamente* como pide este enunciado.

- Unity 2018 Artificial Intelligence Cookbook, Second Edition (Repositorio)  
<https://github.com/PacktPublishing/Unity-2018-Artificial-Intelligence-Cookbook-Second-Edition>
- Unity Artificial Intelligence Programming, Fourth Edition (Repositorio)  
<https://github.com/PacktPublishing/Unity-Artificial-Intelligence-Programming-Fourth-Edition>

Para ir más allá en tu aprendizaje, puedes considerar estas posibles ampliaciones:

- Genera procedimentalmente el laberinto, utilizando algoritmos específicos para ello.
- Modifica al Minotauro para que patrulle siguiendo un camino con un patrón concreto.
- Añade zonas de baldosas con distinto coste al laberinto, como agua, barro, pendientes...
- Da la opción de poder cambiar la heurística utilizada en el algoritmo A\*.
- Permite añadir más salidas al laberinto y modifica a Teseo para que, si hay varias salidas, salga por la más cercana, utilizando para ello el algoritmo de Dijkstra.