



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 1: El flautista de Hamelín

Fecha de entrega: **12 de marzo de 2021**

Importante: Haz la entrega en tiempo y forma, subiendo al campus virtual un fichero *IAVPIGXX.txt* donde *XX* sea tu número de grupo 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, el video comentado *IAVPIG03.mp4*, la carpeta *IAVGXX* con todo el proyecto de la asignatura (código fuente, recursos y plugins), y el ejecutable generado para Windows de 64bits *IAVPIGXX.exe* (con sus carpetas y ficheros acompañantes).

1. Introducción

“Había una vez una pequeña ciudad al norte de Alemania, llamada Hamelín. Su paisaje era placentero y su belleza era exaltada por las riberas de un río ancho y profundo que surcaba por allí, y sus habitantes se enorgullecían de vivir en un lugar tan apacible y pintoresco.

Pero un día, la ciudad se vio atacada por una terrible plaga: ¡Hamelín estaba lleno de ratas! Había tantas y tantas que se atrevían a desafiar a los perros, perseguían a los gatos, sus enemigos de toda la vida; se subían a las cunas para morder a los niños allí dormidos y hasta robaban enteros los quesos de las despensas para luego comérselos, sin dejar ni una miguita.”

La famosa leyenda de El flautista de Hamelín (Figura 1) recogida por los hermanos Grimm nos sirve de inspiración para explorar el movimiento de múltiples agentes en un entorno virtual. La historia nos habla de una manada de ratas que iban tras el misterioso flautista, a la par que molestaban a los demás animales y habitantes del pueblo.

Lo que haremos nosotros será un prototipo en el que el jugador controla el movimiento del flautista y los demás personajes son controlados mediante agentes inteligentes. Nuestro fiel compañero será un perro, que nos seguirá a todas partes, y también estarán por allí las ratas, merodeando por el pueblo. Tendremos la posibilidad de tocar o no la flauta, y mientras que lo hacemos, las ratas comenzarán a seguirnos y el perro huirá de nosotros, asustado al saber que estamos invocando tan desagradable compañía.

Este prototipo servirá para probar algoritmos de movimiento que, lejos de ser anticuados, en realidad siguen usándose de forma habitual en la industria, para animar toda clase de criaturas y personajes que se mueven tanto en solitario, como en “bandada”.



Figura 1. Representación pictórica de El Flautista de Hamelín, original de la iglesia de San Nicolás en Hamelín, Alemania (siglo XVI).

2. Planteamiento del proyecto

Desarrolla un prototipo de IA para Videojuegos, dentro de un entorno virtual con obstáculos y con un avatar controlado por el jugador, que representa el comportamiento cinemático de un perro y de una manada de ratas. El perro, si el flautista no está tocando su flauta, le seguirá, y si la está tocando, huirá de él. Las ratas de la manada, si el flautista no está tocando su flauta, merodearán por el escenario, y si la está tocando, se dirigirán en formación hacia él, controlando la llegada, y se moverán hipnotizadas a su alrededor. El prototipo será fácilmente usable y podrá utilizar formas geométricas simples para representar el entorno y a los distintos agentes.

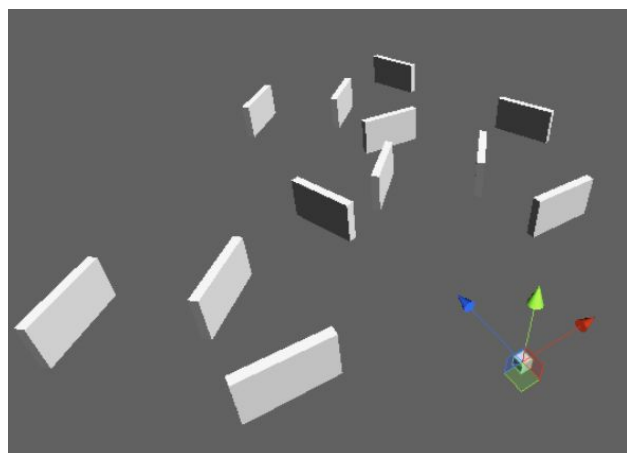


Figura 2. Ejemplo de representación del entorno y los agentes que lo habitan.

La entrega será realizada en tiempo y forma **[1 pto.]**, la documentación explicará con claridad cuál fue la implementación utilizada y cuáles las pruebas realizadas y los resultados obtenidos **[2 ptos.]**, y el proyecto estará bien diseñado, organizado y comentado **[2 ptos.]**.

El prototipo ejecutable será usable y funcional, permitiendo:

- Mostrar el entorno virtual con obstáculos (pueblo de Hamelín) donde se ubican tanto el avatar (flautista) controlado por el jugador mediante los *cursores*, como el agente inteligente que lo acompaña (su perro) compañero y la bandada de agentes que lo siguen (manada de ratas) [1 pto.].
- Representar el acompañamiento del perro (su seguimiento) cuando el flautista no esté tocando su flauta (es decir, no esté pulsando la *barra espaciadora*) [1 pto.].
- Representar la huida del perro cuando el flautista esté tocando su flauta [1 pto.].
- Representar el merodeo individual (movimiento errático, desordenado) de la manada de ratas cuando el flautista no esté tocando su flauta [1 pto.].
- Representar el desplazamiento en bandada (hipnosis) de las ratas, con movimiento en formación y control de llegada hasta el flautista cuando esté tocando su flauta [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, heurísticas o cualquier “truco” utilizado.
- 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 o ajenos.

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:

- Coloca los obstáculos pseudoaleatoriamente o utilizando una secuencia de Halton.
- Genera procedimentalmente el terreno mediante el algoritmo de ruido de Perlin.
- Añade vista y predicción al perro para que se evada de las ratas y no del avatar.