



Inteligencia Artificial para Videojuegos

Grado en Desarrollo de Videojuegos

Proyecto final

Departamento de Ingeniería del Software e Inteligencia Artificial

Facultad de Informática

Universidad Complutense de Madrid

Proyecto: LUABerinto

Fecha del enunciado: **25 de mayo de 2021**

Fecha de defensa: **15 de junio de 2021**

Fecha de entrega: **16 de junio de 2021**

Importante: Haz la entrega en tiempo y forma, subiendo al campus virtual un fichero *IAVFinal-Apellidos.txt* sobre el proyecto final. Dentro de él incluye los datos y contribuciones de los alumnos implicados, una breve explicación del proyecto, los problemas existentes y el enlace al repositorio donde están: el fichero *README.md* con el índice de la documentación técnica, la carpeta *IAVFinal-Apellidos* con el proyecto (plugins, recursos y el código fuente), la versión ejecutable para Windows de 64bits *IAVFinal-Apellidos.exe* (con sus carpetas y ficheros acompañantes) y el video comentado con las pruebas *IAVFinal-Apellidos.mp4*.

1. Introducción

Tras un largo día dedicado a Proyectos 3 y a “debuguear” scripts de LUA, los integrantes de este grupo decidieron tomarse un descanso y charlar sobre los trabajos y tareas que les quedaban por entregar. Esto culminó en una sucesión interminable de memes, a cada cual más cuestionable que el anterior, como: LUAprobamos, LUHAcemos y LUAcabamos. Al final, entre memes y risas, los integrantes de este grupo llegaron a la conclusión de que hacer un “LUABerinto” como trabajo final para la asignatura de Inteligencia Artificial para Videojuegos era una idea insuperable. Y así fue como, en base a un comentario random, se prendió la mecha que iniciaría todo el proceso para la creación de este prototipo.

¿En qué consiste este LUABerinto, te preguntarás?. Podríamos definirlo como una serie de muchos “oye, ¿y si le metemos <x> al proyecto del de IA?. Seguro que queda tri chulo”, gracias a la cual llegamos a la conclusión de crear de forma procedural un laberinto que nos servirá como escenario para probar la inteligencia artificial de un agente que contará con sistemas de percepción y memoria para recorrer y encontrar objetos por dicho escenario.

En esta ocasión, esta pequeña anécdota nos sirve como pretexto para presentar el proyecto final para la asignatura de IAV en el que trabajaremos Amaro Blest Polo (ablest@ucm.es), Aurora García Jover (augar01@ucm.es) y Leire Osés Sánchez (leoses@ucm.es).

El prototipo que vamos a desarrollar se centra en la creación de un laberinto de manera procedural, empleando el algoritmo clásico de generación de laberintos y crear un agente con inteligencia artificial que irá explorando él sólo dicho laberinto. Por tanto, abordaremos 2 de los temas que más nos han llamado la atención en relación a la asignatura; la generación procedural de contenido y el sistema de percepción y memoria de un agente inteligente.

Durante su exploración, el agente podrá encontrarse con diversos NPCs que le encomendarán misiones. Contará con un sistema de percepción, el cual le servirá para recorrer dicho laberinto y encontrar e interactuar con los diversos objetos y personajes que se encuentren cerca de su posición o a la vista. También contará con un sistema de memoria para solventar las misiones que le especifiquen ciertos NPCs, las cuales consistirán en conseguir objetos que se encuentran dispersos por el laberinto y según si dichos objetos han sido registrados en su sistema de memoria o no deberá seguir explorando o ir a la posición en la que recuerda que se encontraban.

El objetivo del agente es encontrar a todos los NPCs, completar todas sus misiones y salir de nuevo del laberinto.

2. Planteamiento del proyecto

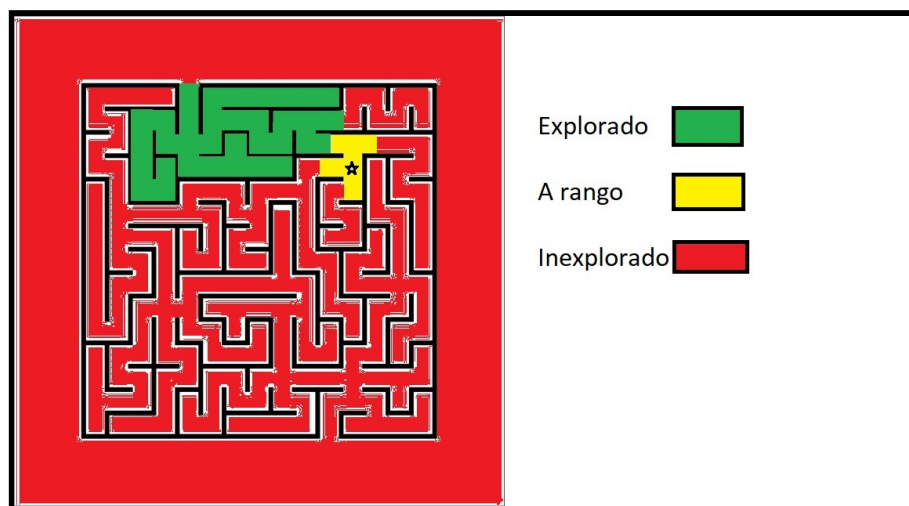


Figura 1. Representación básica del estado de juego en un momento concreto.

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 las técnicas implementadas **[2 ptos.]**, las pruebas realizadas y los resultados obtenidos **[de 1 pto.]**.

El prototipo ejecutable será usable y funcional, permitiendo:

- Crear el **laberinto de forma procedural**, empleando el algoritmo de Prim de creación de laberintos. [1.5 ptos.]
- Que el personaje sea capaz de realizar la **exploración del laberinto** de forma **autónoma**. [1 pto.]
- Mostrar el entorno virtual de forma que se **diferencie** a simple vista **las zonas** que no han sido exploradas de las que sí. [1 pto.]
- Que el personaje sea capaz de **realizar las misiones** de forma correcta y autónoma, **aplicando el conocimiento** que ha **obtenido** durante la exploración a la hora de resolver dichas misiones. [1.5 ptos.]

3. Restricciones y consejos

A la hora de desarrollar este proyecto es obligatorio:

- Utilizar únicamente las herramientas de Unity y opcionalmente los plugins de terceros *Bolt* o *Behavior Designer*, sin reutilizar código ajeno al que proporciona el 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.

Pensando tanto en las pruebas como en la revisión del profesor, y también con ánimo de reutilizar el esfuerzo de desarrollo, conviene crear herramientas visuales cómodas para mostrar escenarios de ejemplo interesantes y con instrucciones de uso, etc. El 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; asegúrate de entenderlo y verifica que funciona *exactamente* como pide este enunciado.

- Bolt, Visual Scripting
<https://unity.com/es/products/unity-visual-scripting>
- Opsive, Behavior Designer
<https://opsive.com/assets/behavior-designer/>
- Unity, Navegación y Búsqueda de caminos
<https://docs.unity3d.com/es/2019.3/Manual/Navigation.html>
- 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>