



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



Desarrollo de un prototipo de Inteligencia Artificial para un posible videojuego de tema libre

Revisión del enunciado y del trabajo: **5 de mayo de 2022**

Revisión de los resultados: **24 de mayo de 2022**

Importante: Aunque hayas coordinado tu proyecto con los de otros alumnos para producir algo más ambicioso, las revisiones se realizarán *a título individual* en tiempo y forma, incluyendo en la documentación el reparto realizado de tareas y esfuerzo. El profesor tendrá tus datos y el enlace a tu repositorio en GitHub “IAV22-*Apellidos*” donde estará el código fuente del proyecto, los recursos, *plugins* y toda la documentación del prototipo en el fichero raíz *README.md*. La documentación contendrá tus datos, un resumen del enunciado (tema libre pero aprobado por el profesor), la descripción del punto de partida, el diseño del prototipo de IA (con diagramas y pseudocódigo) y el enlace a un video oculto de YouTube “IAV22 Proyecto final - *Apellidos, Nombre*” con la grabación de las pruebas realizadas (exitosas o no), personalizado, comentado, y de menos de 3 *minutos* de duración. La versión ejecutable para Windows de 64bits “IAV22-*Apellidos.zip*” será publicada como *lanzamiento* descargable en el repositorio.

1. Introducción

“La ventaja de ser inteligente es que así resulta más fácil pasar por tonto. Lo contrario es mucho más difícil.” - Kurt Tucholsky.

Las técnicas de Inteligencia Artificial (IA) con las que se desarrollan los videojuegos son muy variadas y prácticamente cada problema específico merece ser estudiado con detenimiento y resuelto con algún algoritmo, heurística o truco, hasta dar con los mejores resultados.

Como proyecto final de la asignatura los alumnos participarán en el *desarrollo de un prototipo* de **Inteligencia Artificial** para un **posible videojuego** (aún sin desarrollar) de tema libre, aunque aprobado previamente por el profesor.

Aunque la autoría de los proyectos es **individual**, varios alumnos podrían coordinarse para sumar esfuerzos y combinar sus contribuciones hacia un resultado final más ambicioso.

2. Propuesta

Desarrolla un **prototipo de IA** para videojuegos con Unity Pro, Visual Studio Community, C# y los *plugins* vistos en clase donde pongas en práctica aquello que hayas aprendido pero no hayas podido implementar en las prácticas. El tema es **completamente libre**, para que seas creativo y puedas afrontar un desafío que te interese personalmente. No hay límites en cuanto al **alcance** del proyecto, pero debes confirmar previamente con el profesor que se trata de un proyecto

factible -que se puede realizar con el tiempo y los recursos disponibles- y suficientemente **interesante** -de dificultad razonable teniendo en cuenta lo estudiado y lo ya disponible en la Web-.

Puedes usar todo el **material de terceros** que necesites, pero intenta mantener a raya el tamaño del proyecto y el ejecutable, evita usar recursos de pago y deja bien claro cuáles son **tus contribuciones**, separándolas nítidamente de todo aquello que no sea de tu autoría.

Tras la correspondiente investigación y búsqueda de referentes, toda la **documentación del prototipo** se redactará preferiblemente en español. La extensión del documento se ajustará a unas **2500 palabras** como máximo y se incluirá el **nombre completo de los alumnos** (aclarando sus contribuciones individuales, si son varios que han cooperado), el **título** y la **descripción del problema** (equivalente al resumen del enunciado que se suele hacer para las prácticas), la **descripción del punto de partida** (sea un proyecto base, una práctica... poniendo capturas y dejando claro dónde quedan esos recursos en las rutas internas del proyecto), los **controles** que se utilizan, el **diseño del prototipo de IA** (explicado con diagramas y pseudocódigo) y el **enlace al video con todas las pruebas realizadas**, sean exitosas o no.

El **proyecto completo de desarrollo** quedará en su repositorio de GitHub, con todos sus ficheros fuente (cuya publicación bajo licencia abierta podría considerarse). Deberán nombrarse adecuadamente los recursos (ya sea en inglés o en español, pero con coherencia), organizarse debidamente en carpetas (algunas propias y otras correspondientes a los paquetes de terceros), al igual que se organizarán debidamente los objetos de las distintas escenas del juego. Entre los recursos a organizar y comentar debidamente se incluyen las **clases de código C#**, con sus interfaces, funciones, variables, etc. El **ejecutable resultante** se distribuirá en forma de aplicación ejecutable para Windows 64bits, jugable con teclado y ratón. Se añadirá al repositorio como “lanzamiento” para poder ser descargado directamente desde allí.

3. Revisiones y defensas

El **trabajo de diseño** sobre el prototipo deberá ser el adecuado (investigación de los referentes, resumen del enunciado -factible e interesante-, punto de partida elegido...) y quedar debidamente reflejado en la documentación [**1 pto.**], especialmente todo el diseño de la solución (usando diagramas y pseudocódigo, como trabajamos en clase) [**2 ptos.**]. Esta documentación será revisada y defendida ante el profesor.

El **prototipo** será usable y funcional, cubriendo las tareas propuestas de percepción, movimiento, navegación, decisión, evaluación, coordinación u otras distintas, con la calidad mínima esperable en la jugabilidad y los contenidos del juego [**4 ptos.**]. El banco de pruebas realizadas, y documentadas en el video, dejarán en evidencia la funcionalidad y las limitaciones del prototipo, mostrando aquellas métricas que permitan valorar los resultados y la eficiencia de la implementación [**2 ptos.**]. Esta ejecución será revisada y defendida ante el profesor.

El **código y las buenas prácticas de desarrollo software** (commits frecuentes en la rama principal, organización de objetos en la escena, escritura sistemática de los comentarios, etc.) serán revisadas al cierre del proyecto directamente en el repositorio [**1 pto.**].

La exigencia aplicada a la hora de evaluar estos apartados, al ser un proyecto de tema libre, dependerá de la **ambición manifestada por el alumno** a la hora de plantearlo. Los defectos o carencias encontrados tendrán *mayor penalización* si se trata de un prototipo simple, donde es fácil hacer una revisión exhaustiva para corregirlos, que si se trata de un producto complejo. La *calificación máxima* que es posible obtener con un proyecto, depende de este alcance.

El criterio de evaluación queda reflejado en esta tabla:

Ambición	Penalización	Calificación máxima
Mínima	Cualquier fallo funcional o de rendimiento	5 sobre 10
Moderada	Cualquier fallo funcional o fallo <i>serio</i> de rendimiento	6,75 sobre 10
Adecuada	Fallos <i>serios</i> funcionales o de rendimiento	8,75 sobre 10
Excelente	Problemas evidentes de funcionalidad o rendimiento	10 sobre 10

4. Restricciones y consejos

A la hora de desarrollar este proyecto es obligatorio:

- Utilizar únicamente las herramientas de Unity y opcionalmente los *plugins* de terceros acordados con el profesor, sin reutilizar código ajeno a estos.
- 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 implementadas.
- Evitar, en la medida de lo posible, el uso de recursos audiovisuales pesados o ajenos.

Pensando tanto en la realización de pruebas como en las revisiones del profesor, y con ánimo también de reaprovechar esfuerzo de desarrollo, conviene crear herramientas visuales cómodas para mostrar escenarios de ejemplo interesantes, instrucciones de uso, etc. Por ejemplo, habilitando una “consola de trucos” (teclas rápidas) que permitan al desarrollador reiniciar la ejecución, cambiar la cámara, establecer situaciones específicas para hacer pruebas, volverte invencible, etc. con una sola tecla. El manejo debe ser ágil e intuitivo para poder repetir rápidamente todas las pruebas necesarias con aquellas variaciones que hagan falta.

5. Referencias

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 necesitamos que lo haga.

- AI Game Programming Wisdom - Serie de libros
- Artificial Intelligence for Games - Repositorio antiguo en C++
<https://github.com/idmillington/aicore>

- Aversa, D., Kyaw, A.S., Peters, C.: Unity Artificial Intelligence Programming, Fourth Edition (Repositorio)
<https://github.com/PacktPublishing/Unity-Artificial-Intelligence-Programming-Fourth-Edition>
- Bolt, Visual Scripting
<https://unity.com/es/products/unity-visual-scripting>
- de Byl, P.: Artificial Intelligence for Beginners (Learn Unity)
<https://learn.unity.com/course/artificial-intelligence-for-beginners>
- Game AI Pro - Serie de libros
- Game AI Pro 360 - Serie de libros
- Game AI Summit
<https://gdconf.com/summit/game-ai>
- Opsive, Behavior Designer
<https://opsive.com/assets/behavior-designer/>
- Smith, M.: Unity 2018 Cookbook
<https://github.com/PacktPublishing/Unity-2018-Cookbook-Third-Edition>
- 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>