Inteligencia Artificial

Proyecto práctico 1 PARTE 1

Objetivo

El objetivo de la práctica es implementar distintos algoritmos de búsqueda dentro de un entorno de Unity3D. La meta final del trabajo debe ser desarrollar en un conjunto de clases de C# los algoritmos que permitan resolver los distintos problemas que plantearemos dentro del escenario a desarrollar.

Entrega

El proyecto debe ser:

- ✓ **Original**. Se pueden utilizar librerías externas para apoyar el programa general, pero la implementación del algoritmo debe ser creada desde cero.
- ✓ **Documentado**. Se debe presentar un informe donde se recoja el contexto del juego, el problema que se plantea a la IA, el algoritmo implementado, qué *controles* tiene el algoritmo, qué valor se les ha dado y cómo se maneja el programa.
  Además se debe incluir el código fuente comentado.
- ✓ **Ejecutable.** Se debe poder ejecutar sin errores dentro de Unity3D.

Para entregar esta tarea, debemos incluir en el entregable una memoria explicativa de todos los elementos desarrollados, teniendo en cuenta el formato y la información que en dicha memoria se introduzca.

La memoria deberá constar de, por lo menos:

1. Portada (**con los nombres de los integrantes del grupo**)
2. Índice
3. Descripción del algoritmo empleado para solucionar el problema
4. Características de diseño e implementación
5. **Discusión sobre los resultados obtenidos**.

Todas las decisiones de diseño que se hayan tomado durante el desarrollo de la práctica deberán ser explicadas detalladamente dentro de la memoria.

Además de la memoria explicativa, será necesario entregar todas las fuentes con los scripts creados por el grupo y el documento PDF con la memoria. Dicho paquete irá nombrado de la siguiente manera.

NombreGrupo-practica1.zip

Grupos y Modo de evaluación

Cada proyecto se debe presentar en grupos de 1, 2 o 3 personas.

Cuándo

Cada proyecto se debe subir a la tarea habilitada en el Campus Virtual antes del día que se designe a través de la misma.

Cuánto

El proyecto se evalúa de 0 a 10.

La nota depende del informe generado, la implementación y la ejecución. Asimismo, la documentación de los resultados obtenidos en las distintas pruebas y cualquier comentario aclaratorio sobre las mismas.

La calificación se atendrá a los siguientes criterios de calificación.

- No ejecución de cualquier apartado, o memoria altamente deficiente: Nota máxima 4
- Apartado 1 Offline
    - Desarrollo de un solo algoritmo para su resolución: 0 – 1 puntos
    - Desarrollo de dos algoritmos para su resolución: 0 – 2 puntos
    - Análisis de los resultados obtenidos: 0 - 1 punto extra
    - Documentación correcta: 0 – 1 punto extra
- Apartado 2 Online
    - Desarrollo de un solo algoritmo para su resolución: 0 – 1 puntos
    - Desarrollo de dos algoritmos para su resolución: 0 – 2 puntos
    - Análisis de los resultados obtenidos: 0 - 1 punto extra
    - Documentación correcta: 0 – 1 punto extra

Paquete de definición del escenario

1. Dentro del campus virtual se dispondrá de un recurso que incluye un fichero de tipo UNITYPACKAGE en el cual se encuentra todo el esqueleto de definición de las escenas y recursos necesarios para crear la práctica
2. En el directorio Scripts del proyecto que se incluye hay un ejemplo de controlador Aleatorio <RandomMind> que puede valer de punto de partida para la creación de los distintos apartados del proyecto.
3. Si se modifica cualquier fichero de los proporcionados dentro de dicho paquete deberá documentarse convenientemente dentro de la memoria en el apartado de Decisiones de Diseño.

Descripción detallada del entorno

Dispondremos de los siguientes elementos diferenciados dentro del escenario.

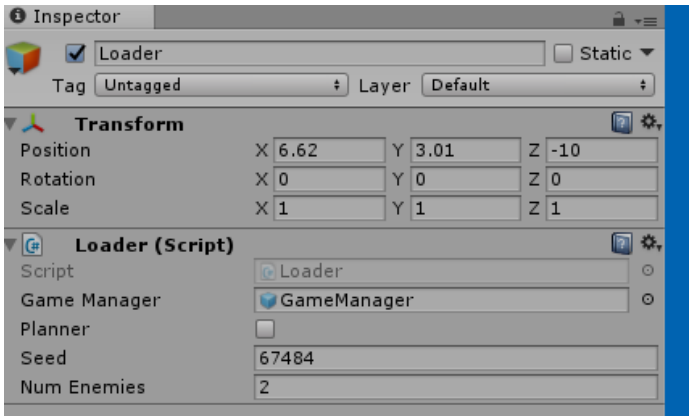
4. El personaje controlado por la IA. Será representado por un avatar .
5. Escenario compuesto por celdas cuadradas. Las baldosas que formarán el suelo serán de tamaño uniforme.
    a. Las baldosas tendrán un coste = 1 de movimiento, salvo que se especifique lo contrario, en cuyo caso, dicho coste se simbolizará visualmente en la cuadrículas y será almacenado como una propiedad de las mismas.
6. Obstáculos: Elementos que ocuparán una celda y que impedirán que sea transitable para el personaje.
7. Elementos Clave: Son elementos etiquetados con un conjunto de propiedades que permitirán al personaje completar los problemas.
    a. Los elementos contendrán una etiqueta que los identifica
8. Enemigos, los enemigos se desplazarán por la escena de modo aleatorio.

Reglas generales del escenario

En el escenario, en cualquiera de los problemas siempre dispondremos de la siguiente información y reglas fijas:

1. El tamaño del escenario (número de celdas de largo y ancho) será siempre conocido en tiempo de ejecución. Al iniciar la escena.
2. Los obstáculos son fijos dentro del escenario y el mapa no se varía en ningún momento de la ejecución.
3. El agente no puede estar en una celda ocupada por un muro.
4. Se permite el movimiento en las cuatro direcciones cardinales N-S-E-O.
5. Para activar un objeto colocarse en la ceda donde se encuentra el objeto.

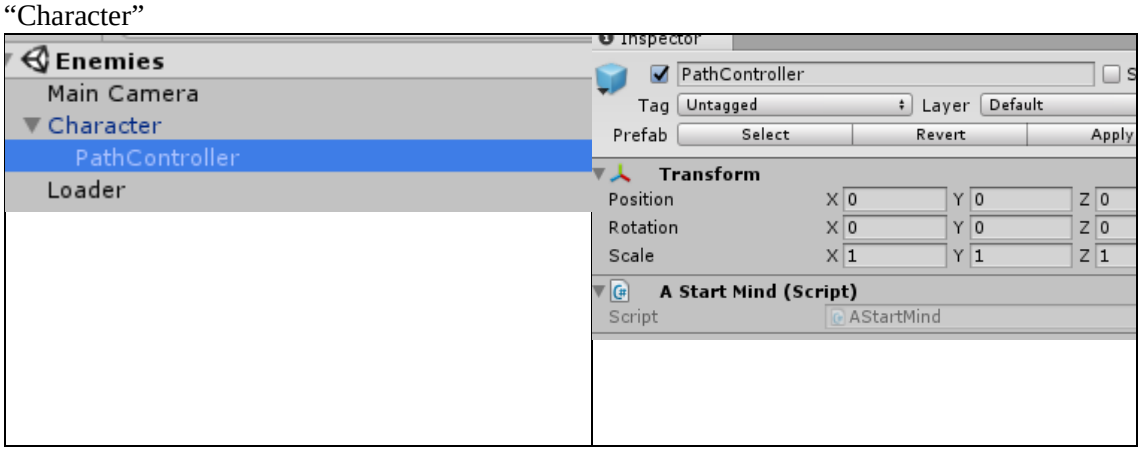
Los objetos del escenario se generan de modo aleatorio ya sea en número o en posición a través de la semilla que se especifique en el objeto "**Loader**".



Asignación de un Controlador

Para asignar el controlador que gestionará al personaje debemos crear un Script derivado de la Clase *AbstractPathMind* que implemente los métodos necesarios para proveer al sistema de Locomoción de los movimientos deseados. Dicha clase se deberá implementar acompañada de todas las clases auxiliares que necesite.

Este script se asignará como componente al GameObject "PathController" del GameObject "Character"



Las clases auxiliares del personaje permiten acceder a la información del tablero en un determinado momento o a los elementos de cada celda.

Los objetos del escenario están todos registrados en la clase *BoardInfo.*

Objetivos específicos de la práctica 1

En esta práctica se pide desarrollar dos algoritmos para el controlador en los escenarios de PathFinding y Enemies.

Apartado 1: Búsqueda de camino de salida

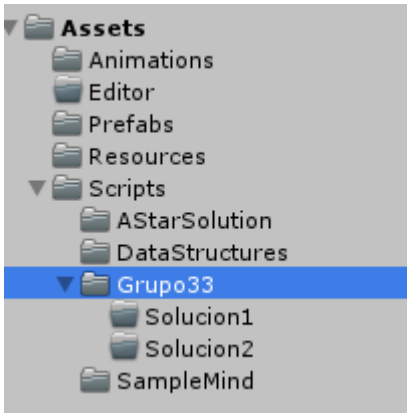
En el primero deberemos crear un algoritmo de búsqueda caminos eficiente OffLine que nos encuentre la meta (Objeto etiquetado de tipo GOAL). En dicho algoritmo se emplearán las heurísticas que se consideren oportunas y se discutirán aquellos escenarios que se seleccionen (3 Semillas asignadas por grupo libremente).

Apartado2: Atrapar a los enemigos.

El segundo de los algoritmos a crear consistirá en una búsqueda OnLine que garantice que se matan a todos los enemigos dentro de la escena antes de buscar la salida. Para ellos se recomienda emplear el algoritmo anterior como base y crear un mecanismo de encontrar el enemigo más cercano para buscar el objetivo. Para matar a un enemigo basta con entrar en la celda donde se encuentra.

Formato de la entrega dentro del proyecto

En esta primera práctica se deberán crear los controladores necesarios dentro del proyecto de Unity en un directorio que descienda de **Scripts** y que tenga el nombre del grupo (por ejemplo, para el “Grupo 33” (Nombre del Grupo o Apellidos del Alumno) debería está ubicadas todas las clases en el directorio: *Assets>>Scripts>>Grupo33* dentro de este directorio se podrá crear la estructura que se considere necesaria.



Para realizar la entrega deberá crearse un fichero ZIP que incluirá todo el proyecto de Uitny3D (Fuentes sin ejecutable) y el fichero PDF con la memoria en el formato especificado.