

GRADO EN DISEÑO Y DESARROLLO DE VIDEOJUEGOS

DESARROLLO DE JUEGOS CON INTELIGENCIA ARTIFICIAL

Práctica 2: Machine learning

Objetivos

El objetivo de la práctica es implementar un agente inteligente cuyo comportamiento sea aprendido por técnicas de machine learning. En concreto, se emplearán técnicas de Q-Learning para entrenar el cerebro del agente. La práctica se desarrollará utilizando el motor de videojuegos **Unity™** y su lenguaje de scripting C#. Esta práctica requerirá desarrollar las clases que permitan aprender y ejecutar el comportamiento que resuelva el objetivo que vamos a plantear.

Descripción del entorno

Desde el aula virtual puede descargarse el proyecto **Unity™** que servirá como base para la práctica. Al abrirse el proyecto, dentro de la carpeta **Scenes\QLearning** pueden encontrarse dos escenas, **TestPlayGround** y **TrainPlayGround**.

TrainPlayGround

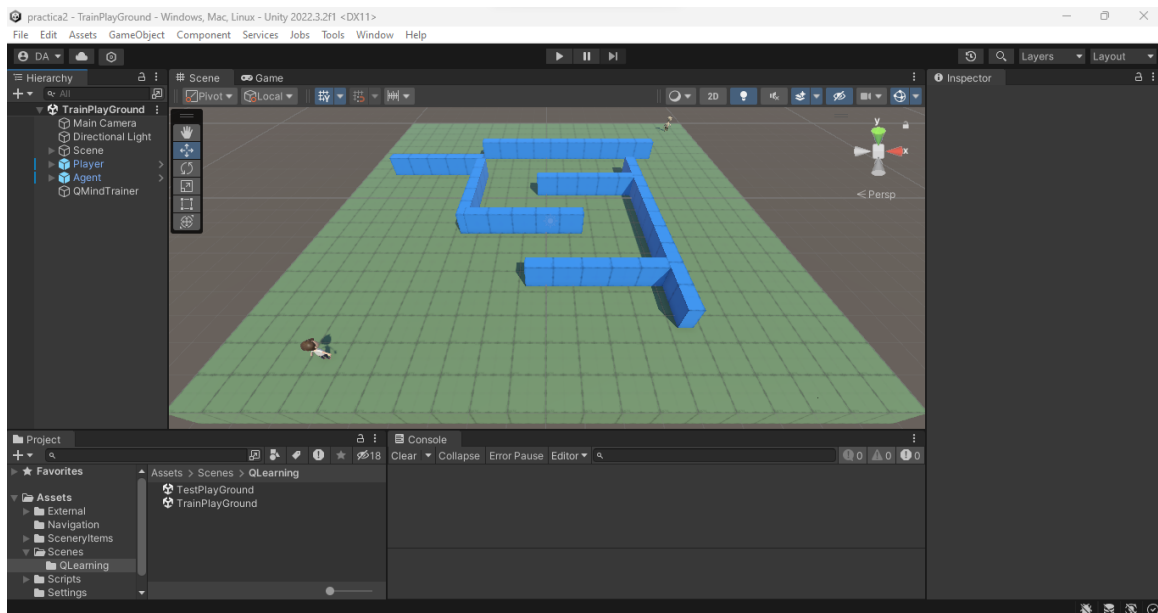


Figura 1: Escena de entrenamiento del agente inteligente

En la escena **TrainPlayGround** es donde se realizará el entrenamiento de nuestro agente. Dentro de este mundo podemos encontrar los siguientes elementos de juego:

1. El agente inteligente, representado por el game object **Agent**. Es el personaje controlado por la IA y objeto de la práctica.

2. El rival de nuestro agente inteligente, representado por el game object **Player**. Este rival se desplazará hacia la posición del agente inteligente usando un algoritmo A*.
3. El game object **QMindTrainer** contiene el script homónimo que entrena a nuestro agente. Este script utiliza una implementación de la interfaz **IQMindTrainer** que puede seleccionarse en el inspector. En el proyecto base, la implementación seleccionada es **QMindTrainerDummy**, una implementación básica de ejemplo.
4. El escenario de juego, representado por el game object **Scenery**. Los objetos del mundo son game objects hijos de este. La escena incluye:
 - a) **El suelo/tablero**, representado por el game object **Floor** en color verde. Está construido en forma de bloque rectangular y tiene un tamaño de 20x20 unidades.
 - b) **Obstáculos**, representados por los game objects **Wall (*)** en color azul.

Objetivos específicos de la práctica

El objetivo de la práctica es implementar un algoritmo de aprendizaje por refuerzo basado en Q-Learning que enseñe al agente inteligente *Agent* a escapar de su oponente *Player*. Para ello deberá programarse una implementación de la interfaz **IQMindTrainer**. Posteriormente se probará el agente inteligente ya entrenado en una escena de test.

Programar el sistema de entrenamiento del agente (10 puntos)

El agente (*Agent*) deberá evitar ser atrapado por el rival (*Player*). Para ello, deberá ser entrenado mediante técnicas de QLearning. El entrenamiento se realiza de forma automática a partir de la implementación de la interfaz **QMind\Interfaces\IQMindTrainer** que se implementará como parte de esta práctica. El resultado de este entrenamiento debe ser una tabla Q en formato **CSV**.

También debe programarse una implementación de la interfaz **QMind\Interfaces\IQMind**. Esta implementación usará la tabla Q anteriormente mencionada y se empleará para testear de forma automática el aprendizaje efectuado. El testing se realizará en una escena con obstáculos aleatorios y posiciones aleatorias de salida, en el proyecto de los profesores.

Se realizarán 10 ejecuciones de test durante las cuales se contará el número de pasos efectuados por el agente inteligente sin ser atrapado por el oponente. El número de pasos promedio entre las 10 ejecuciones se utilizará para obtener la calificación final, según la siguiente tabla:

Promedio de pasos	Calificación
250-1000	6
1000-5000	8
>5000	10

Para que la práctica se califique, debe poder ser ejecutada en el entorno de test. Además el algoritmo de aprendizaje debe poder ser ejecutado y replicar el entrenamiento. La tabla Q debería poder ser regenerada desde el algoritmo de entrenamiento, y revisada empleando Excel.

Le falta de cumplimiento de cualquiera estos requisitos supondrá una nota de suspenso en la práctica.

Grupos y modo de evaluación

El proyecto se debe realizar en grupos de 2 personas, en principio los mismos grupos de la práctica 1. De forma excepcional, se podrá realizar de forma individual acordando este punto con los profesores de la asignatura.

La práctica se evaluará de 0 a 10, según los objetivos especificados en el apartado anterior.

La fecha límite de entrega se ha fijado el **domingo 21 de Enero de 2024, a las 23:59 horas**.

Si se produce alguna modificación al enunciado de la práctica, se avisará con la suficiente antelación.

Entrega

El proyecto a entregar debe ser:

- **Original:** Se pueden utilizar librerías externas para apoyar el programa general, pero la implementación del agente y los algoritmos de búsqueda deben ser creadas desde cero.
- **Documentado:** Se debe presentar un informe/memoria donde se recoja el contexto del juego, el problema que se plantea a la IA, los algoritmos implementados, que controles tienen los algoritmos, que valores se les ha dado y como se maneja el programa. Además se debe incluir el código fuente debidamente comentado.
- **Funcional:** Los algoritmos deben poder ejecutarse sin errores dentro de un proyecto **Unity™**

Para implementar el proyecto, se creará una carpeta dentro de **Scripts** con el nombre del grupo sin espacios en blanco, por ejemplo para el grupo A: **Scripts\GrupoA**. Dentro de esta carpeta se incluirán **todos los archivos de código fuente de la práctica y la tabla Q además de la memoria en formato PDF**. La estructura de carpetas en el interior de esta queda a la elección del alumno.

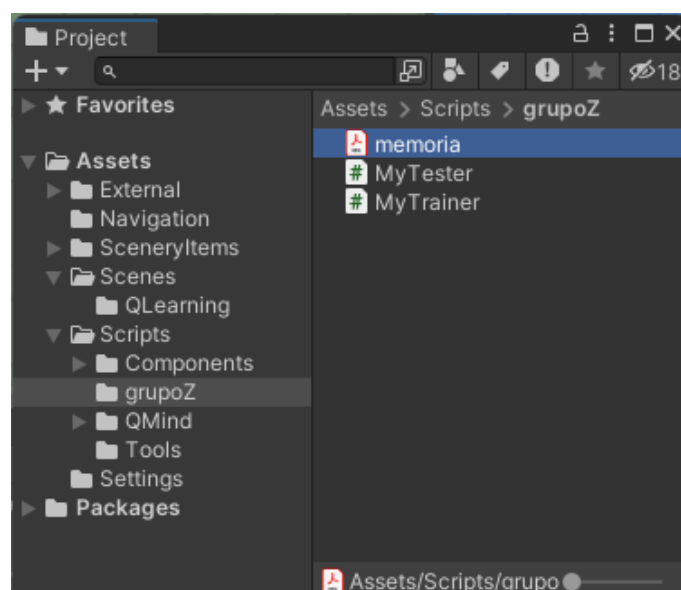


Figura 2: Carpeta para el desarrollo de la práctica

La entrega incluirá:

- Un package de **Unity™** (ver figura 3) que incluirá **únicamente** esta carpeta. Este package **Unity™** se subirá a la tareá que se creará en el Aula Virtual. El package **Unity™** se nombrará *<nombre de grupo>.unitypackage*. Incluirá:
 - Los scripts C# programados para la práctica, completamente funcionales y listos para ser ejecutados.
 - La tabla Q con la que se evaluará el rendimiento del agente inteligente.
 - La memoria especificada anteriormente.
- Un **archivo de texto** con todo el código fuente desarrollado para la práctica, todas las clases de forma consecutiva. El nombre de este fichero será *<nombre de grupo>.txt*

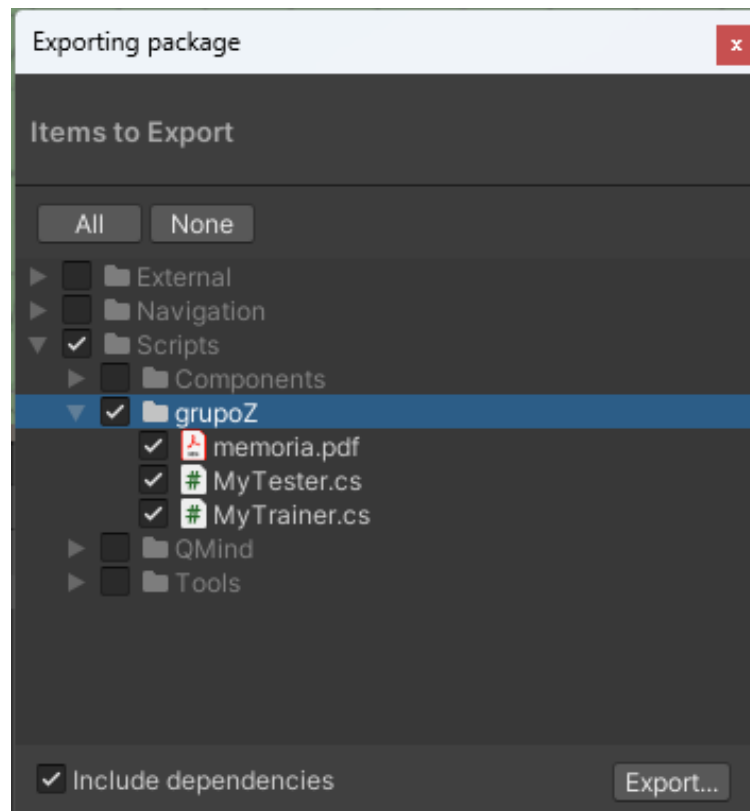


Figura 3: Export de la entrega de la práctica