

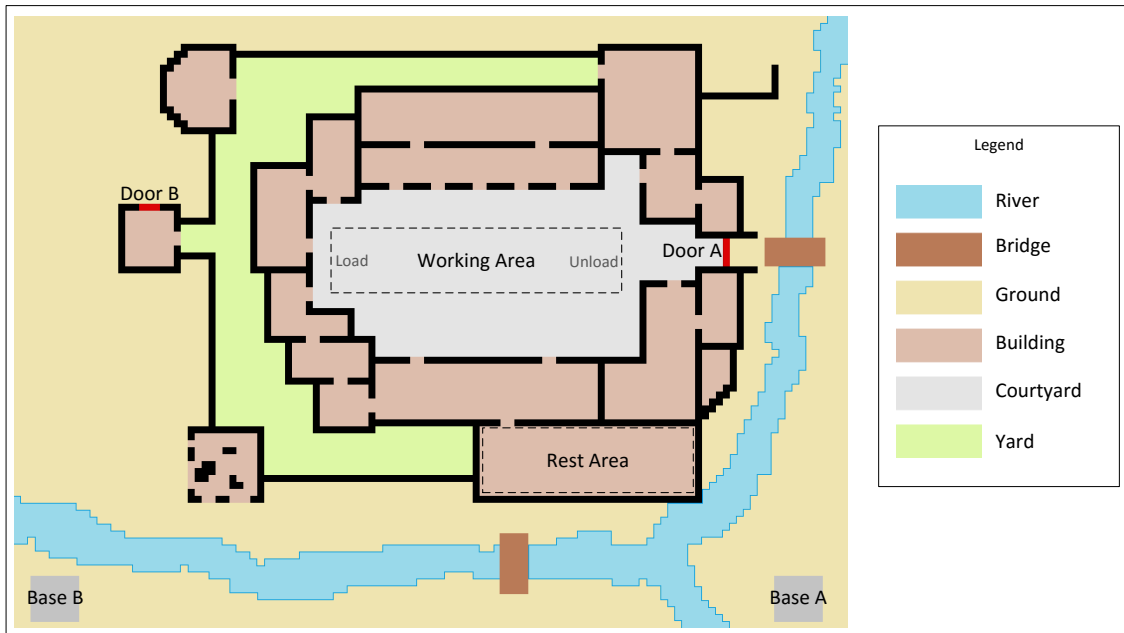
HND in Videogames Programming	3INF	Profesor titular / Asesor	Antonio Barella
ARTIFICIAL INTELLIGENCE FOR VIDEOGAMES		Fecha asignación: / Starting date:	19 / 05 / 2016
ARTIFICIAL INTELLIGENCE FOR VIDEOGAMES		Fecha entrega: / End date:	13 / 06 / 2016
<b>Práctica Final / Final Assignment</b>			

ALUMNO / STUDENT :	
--------------------	--

PROFESOR / ASSESOR :	Antonio Barella
----------------------	-----------------

ENUNCIADO DE LA PRACTICA / ASSIGNMENT BRIEF	CHECKING		EVALUACIÓN / MARK	R	P	M	D
<p><b>Directrices de investigación y desarrollo / Research directions:</b></p> <p><u>Práctica</u></p> <p>La práctica consiste en desarrollar un juego aplicando las técnicas de Inteligencia Artificial vistas en clase. El objetivo del juego es liberar a los prisioneros de una fortaleza. Existe un grupo de guardias cuyo objetivo es vigilarlos y evitar que escapen. Los guardias, para mantener ocupados a los prisioneros, les obligan a realizar tareas pesadas. Concretamente, deben transportar cajas de un lado a otro de la fortaleza. La mitad de los prisioneros están desempeñando estas tareas, mientras que la otra mitad está en un patio cercano, paseando y esperando su turno para relevar al otro grupo de prisioneros. Un grupo de soldados, aliados de los prisioneros, debe rodear la fortaleza y abrir las puertas para que los prisioneros escapen. En ningún momento del juego se produce combate, es decir, no hay armas, ni combate de ningún tipo (ni siquiera cuerpo a cuerpo).</p> <p><u>Mapa</u></p> <p>Para la realización del ejercicio se facilita dos mapas, uno cuya finalidad es visual y de tamaño 960 x 704, y otro que es de costes, que debe usarse para el cálculo de caminos, y de tamaño 120 x 88. Las posiciones iniciales de los soldados al comienzo del juego serán alrededor de la base B situada en la esquina inferior izquierda. Sólo es posible entrar o salir de la fortaleza por medio de las puertas (en rojo en el mapa visual), y no están incluidas en el mapa de costes. La única puerta de entrada es la puerta A. Es posible salir de la fortaleza a través de las puertas A y B. Una puerta puede ser abierta (la puerta B sólo desde el interior de la fortaleza) por un soldado cuando pasa a una distancia menor que <math>D_{DOOR}</math>. Un guardia la puede cerrar si está a una distancia menor que <math>D_{DOOR}</math> y no hay ningún soldado dentro de ese radio. Ningún agente puede quedarse quieto a una distancia menor que <math>D_{DOOR}</math> de una puerta.</p> <p><b>Interior de la fortaleza</b></p> <ul style="list-style-type: none"> <li>Zona de descanso: es la zona donde la mitad de los prisioneros descansan. Sólo se entra en ella para descansar y sólo se sale de ella para trabajar o huir.</li> <li>Zona de trabajo: la zona en la que los prisioneros trabajan. Transportan cajas desde un extremo de la zona hasta el extremo opuesto. Los prisioneros no deben salir de ella excepto para descansar o huir. Los guardias no deben entrar en ella, excepto si se está en ALERTA.</li> </ul> <p><b>Exterior de la fortaleza</b></p> <ul style="list-style-type: none"> <li>Río: zona intransitable (los agentes no pueden pisarla). La única forma de cruzar los ríos es a través de los puentes.</li> <li>Paredes de la fortaleza: zona intransitable. Los agentes sólo pueden entrar y salir de la fortaleza a través de las puertas (en rojo en el mapa visual).</li> <li>Bases: La base B es la base de la que salen los soldados para liberar a los prisioneros. Los prisioneros y los soldados</li> </ul>							

deben alcanzar cualquiera de las dos bases durante el proceso de huida.



## Tipos de Agentes

- Guardias:
  - pasean por los alrededores de la zona de trabajo y también pueden hacerlo por las estancias de la fortaleza, pero no hacen un recorrido cíclico. Si ven a un soldado, lo persiguen. Si ven una puerta abierta, se acercan a cerrarla.
- Prisioneros:
  - si están trabajando (zona de trabajo), cargan mercancía en la zona de carga y la transportan a la zona de descarga. A continuación, vuelven a la zona de carga y repiten el proceso, de manera cíclica. Cuando van cargados, y debido al peso, su velocidad se reduce a la mitad. Cuando se da la situación de ALARMA, dejan las cajas que transportan y comienzan a buscar una puerta abierta para huir. Si la encuentran, intentan salir por ella y ya no vuelven a entrar a la fortaleza, huyendo lejos de la fortaleza. Si no están trabajando, esperan paseando en la zona de descanso que hay en el edificio. En situación de ALARMA, los prisioneros que descansan también intentan escapar.
- Soldados:
  - situados fuera de la fortaleza, deben de acercarse a ella y abrir sus puertas para que escapen los prisioneros. Los soldados sólo pueden huir si se ha dado la alarma.

## Estado de alarma

En situaciones normales, los agentes están desarrollando sus tareas, y no están alarmados. En caso de que los guardias observen un soldado, pasan a SOSPECHA y se dedican a perseguirlo. Si perciben una puerta abierta, los guardias pasan a ALERTA, van a cerrar la puerta y dan la alarma. En este estado, los guardias persiguen a aquellos prisioneros y/o soldados que tengan más cerca, y los persiguen constantemente. Un guardia puede pasar de NORMAL a ALERTA sin pasar por SOSPECHA. Una vez un agente pasa a ALARMA, todos los agentes han de tener conocimiento de que se está en ALARMA. Si los guardias no ven soldados o puertas abiertas durante un periodo mayor que  $T_{ALARM}$ , se pasa automáticamente a una situación normal.

## **Requisitos y Entregable del Alumno**

- El número mínimo de agentes de cada tipo en el juego es de diez.
- En distancias cortas, han de utilizarse los algoritmos de movimiento básico vistos en clase, combinando diversos tipos de movimientos por un mismo agente en función de su posición y del estado de la partida.

- El algoritmo que se debe implementar para calcular el recorrido de largas distancias es A\*. Se facilita un mapa de costes para implementar el algoritmo.
- La técnica básica de razonamiento ha de ser un conjunto de máquinas de estado. El movimiento debe ser independiente del razonamiento, como se ha estudiado en el primer semestre de la asignatura.
- El juego ha de funcionar al menos a 25 imágenes por segundo, es decir, con fluidez y sin parones en su ejecución.
- El trabajo ha de ser implementado en C++, usando el esqueleto de proyecto que se facilitó a principio de curso.
- El trabajo debe ser realizado individualmente. La presentación podrá ser realizada en inglés o en castellano.
- Debe entregarse el proyecto (código fuente) y las transparencias de la presentación.

## Criterios de Mejora

- Utilización de A\* de varios niveles (diferente granularidad). Para la realización de esta mejora se facilita un mapa de costes adicional de tamaño 120 x 88 y otro de 60 x 44.
- Uso de grafos de adyacencia para que los guardias recorran todas las estancias de la fortaleza en su labor de vigilancia.
- Utilización de un agente *pathfinder* común a todos: los agentes no calculan el camino, sino que lo piden al agente *pathfinder*. Este agente usa A\* para el cálculo de caminos. Debe optimizar su gestión almacenando caminos (o parciales de caminos). Puede apoyarse en grafos de adyacencia.
- Uso de comunicación entre agentes mediante mensajes.
- Uso de técnicas de *flocking*.
- Realizar una gestión de objetivos mediante grafos.

**Anotaciones de corrección / Feedback to the student:**

Fecha / Date	Elementos evaluados / Checked elements	Comentarios / Observations

## Criterios de evaluación / Assessment criteria

### REQUISITO MINIMO PARA EVALUAR / MINIMAL REQUIREMENTS FOR EVALUATION

#### PASS

La asistencia obligatoria a todas aquellas clases marcadas como clases de desarrollo de la práctica. / Presence in all the class hours dedicated to this assignment.
La entrega de la práctica dentro de los límites de plazo de entrega. / Hand out the solved assignment within the designated time frame.
La correcta realización de los objetivos básicos de la práctica. / Correct development of the core objectives of the assignment

#### MERIT

Se han utilizado otros métodos y técnicas para recopilar, analizar y procesar la información
Se han usado técnicas y teorías relevantes en el análisis del conocimiento
Se ha presentado coherentemente el trabajo utilizando un lenguaje técnico apropiado

#### DISTINCTION

Se ha validado/contrastado la información recopilada
Se han evaluado y resumido las técnicas y teorías usadas para generar las conclusiones/solución
Se ha presentado coherentemente el trabajo utilizando un lenguaje técnico de forma fluida

#### Objetivos del módulo ARTIFICIAL INTELLIGENCE FOR VIDEOGAMES / Unit Outcomes

1	Conocer los conceptos básicos tras la lógica de un videojuego	Describir la estructura del código fuente de un videojuego
X		Evaluar dónde situar el código de IA en un videojuego
		Proveer ejemplos de estructuras lógicas para diferentes tipos de juego
2	Desarrollar componentes de IA usando el paradigma de los Sistemas Multi-Agente	Diseñar un agente software para un videojuego
X		Describir las características de un agente software
		Proveer ejemplos de agentes software
		Identificar las relaciones entre los elementos de un entorno virtual y los de una plataforma de agentes software
3	Entender las técnicas modernas de IA aplicadas al videojuego	Identificar la técnica de IA adecuada para resolver un problema
X		Implementar algoritmos de búsqueda de caminos en un videojuego
		Implementar algoritmos de toma de decisiones para un videojuego
		Implementar código de IA en un lenguaje de programación seleccionado
4	Emplear middleware IA para resolver problemas de IA	Identificar la estrategia de IA a seguir en un juego por su género
X		Identificar las partes y capas que componen un motor de IA
		Demostrar el uso de un motor de IA

#### Material de apoyo

Microsoft Visual Studio 2008 / 2010

Bibliografía:

"Applying Agent Technology". N.R.Jennings and M.Wooldridge  
"AI Game Programming Wisdom (series)". Ed. by Steve Rabin.

Web:

<http://aigamedev.com> – AI Game Dev  
<http://www.gamedev.net> - GameDev.net