

## Técnicas de los Sistemas Inteligentes

Curso 2018-19

### Práctica 1: Desarrollo de un agente basado en búsqueda heurística para el entorno GVG-AI

La Práctica 1 consiste en desarrollar un controlador, basado en alguna variante de A\*, dentro del entorno GVG-AI que guíe a un avatar a resolver un juego en distintos niveles. El juego escogido es el juego con índice 11 en los tipos de juego “singleplayer”, que se pueden encontrar en el fichero "examples/all\_games\_sp.csv" de la distribución de GVG\_AI, denominado Boulder Dash.

#### 1. Descripción breve del juego

En este juego (ver Figura más abajo), el avatar debe cavar en una cueva con una pala, para encontrar al menos 10 diamantes, de entre un conjunto mayor de ellos inicialmente distribuidos en diferentes posiciones, antes de salir por una puerta. Algunas rocas pesadas pueden caer mientras cava, matando al jugador si es golpeado desde arriba. Hay enemigos en la cueva que pueden matar al jugador, pero si dos enemigos diferentes chocan, se genera un nuevo diamante.



**Puntuación:** se otorgan 2 puntos por cada diamante recolectado. 1 punto cada vez que se genera un nuevo diamante. Se otorga -1 para un nivel si el avatar es asesinado por una roca o un enemigo.

## Departamento de Ciencias de la Computación e Inteligencia Artificial




**Acciones:** se pueden ejecutar 4 acciones de Movimiento (IZQUIERDA, DERECHA, ARRIBA, ABAJO) y una acción de USO de la pala

Consultar las transparencias de la presentación de la práctica y el documento Tutorial 1 de GVG-AI para ver más información sobre cómo instalar el entorno con IntelliJ y cómo desarrollar un controlador básico para el juego. En las transparencias del seminario sobre videojuegos hay instrucciones sobre cómo instalarlo en Eclipse.

El juego 11, boulderdash es “casi” determinista. Hay enemigos confinados en “habitáculos” y si éstos no se abren, no suponen amenaza. No obstante, si se abren los habitáculos el juego es no-determinista porque, en cada *tick* del juego (e.d. en cada ciclo de ejecución de una acción del avatar), el enemigo se mueve aleatoriamente a una posición contigua, aunque no necesariamente persigue al avatar.

En los 5 niveles visualizados más abajo en este documento, se puede observar que el avatar puede conseguir al menos 10 diamantes sin necesidad de liberar enemigos. En todo caso, es importante tener en cuenta que el controlador desarrollado por el alumno debe considerar la posibilidad de que haya enemigos fuera de habitáculos que puedan hacer peligrar la misión, porque la corrección de la práctica incluirá niveles distintos a los mostrados y que permitirán que los enemigos deambulen por todo el entorno.

Nivel 0	
Nivel 1	

Nivel 2	
Nivel 3	
Nivel 4	

## 2. Descripción de la tarea a realizar

El juego tiene varias características que lo convierten en un desafío desde el punto de vista de la búsqueda heurística. Un agente que resuelva este juego tiene que tener la capacidad de plantearse varios objetivos (determinar cuáles son las 10 gemas a recoger) que deben ser priorizados (recoger las gemas en el menor tiempo posible). Además, la consecución de cada objetivo es un problema de búsqueda heurística en el que no solo hay que considerar la distancia al objetivo, otros aspectos del estado influyen también en la evaluación de un nodo y en la determinación de cómo de prometedor es un nodo (estimación de la distancia a la que un estado se encuentra del objetivo). Por ejemplo, el número de enemigos o distancia del avatar a éstos, la presencia de rocas, la amenaza de caída de una roca, etc.



Por otro lado, el agente implementado debe integrar comportamientos reactivo y deliberativo. Reactivo porque hay que considerar el estado actual del mundo y ejecutar en cada ciclo (en cada *tick* del juego) la acción más adecuada para conseguir un objetivo inmediato (que puede variar desde ejecutar directamente una acción del un plan ya elaborado, hasta ejecutar una acción para evitar una situación de peligro). Deliberativo porque hay que considerar el estado actual del mundo para elaborar un plan con el fin de alcanzar objetivos a medio/largo plazo. También es interesante observar que el plan elaborado puede fallar porque ha surgido una situación inesperada y puede ser necesario replanificar. Incluso la nueva situación no prevista puede obligar a replantearse el conjunto de gemas a recoger.

**Restricciones en la ejecución.** Es importante tener en cuenta que cada ciclo de decisión está limitado de tal forma que la decisión de qué acción ejecutar en **cada *tick* debe tomarse en un tiempo no superior a 40ms**. Además, **cada nivel debe superarse en menos de 2000 ticks**.

**Restricciones en la implementación.** Se pide que la estrategia de búsqueda de la parte deliberativa esté basada en una versión o extensión del algoritmo A\*.

Pasos a seguir:

- Descargar e instalar el entorno GVG-AI (seguir indicaciones del seminario de videojuegos).
- Seguir las indicaciones de las últimas transparencias del seminario sobre videojuegos para probar varios juegos y niveles
- Consultar la documentación sobre el código, en el seminario de videojuegos. La estructura del código y documentación básica está en <https://github.com/EssexUniversityMCTS/gvgai/wiki/Code-Structure>
- Consultar los tutoriales proporcionados como material de la práctica:
  - Tutorial 1 de GVG-AI, en el que se describe cómo instalar el entorno con IntelliJ, cuáles son las funciones de la API de GVG-AI más relevantes para la implementación del controlador basado en A\* y cómo desarrollar un controlador básico para el juego.
  - Tutorial 2 de GVG-AI, en el que se indica cómo usar un wrapper que simplifica el acceso a la API del entorno GVG-AI y proporciona una API específica para el juego 11 que puede usarse para implementar un agente de forma cómoda.



### 3. Material a entregar

El material a entregar será un fichero zip con el siguiente contenido:

- Una carpeta denominada “fuentes” que incluya el código fuente en java cumpliendo las siguientes restricciones:
  - a. Debe ser un paquete java que se denomine “practica\_busqueda”
  - b. Debe contener al menos un fichero “Agent.java” en el que se defina la clase que implementa el controlador, tal y como se describe en los tutoriales que se entregan como material de la práctica o en los tutoriales del entorno. Podrán entregarse otros ficheros fuente adicionales si así lo considera el alumno.
- Un fichero en pdf con la memoria de la práctica. En la siguiente sección se describe cómo debe ser el contenido de la memoria.

### 4. Criterios de evaluación

En la evaluación se considerarán como criterios el número de puntos obtenido en total en los 5 niveles, el número de niveles que ha superado y el tiempo medio empleado en superar los 5 niveles (si no supera un nivel).

Además, se considerará también la adecuación de la estrategia de búsqueda y heurística empleadas para resolver el problema, lo que se debe documentar en la memoria.

Finalmente, la elaboración de la memoria se considerará también como criterio en la evaluación. La memoria consistirá en un documento en pdf, con una extensión máxima de 7 páginas, la primera de ellas con el nombre y apellidos y grupo de los autores, y las 6 restantes describiendo claramente las siguientes secciones:

**Descripción general de la solución.** Dos páginas como máximo describiendo en pseudocódigo cómo se han integrado el comportamiento reactivo/deliberativo, además de una descripción general y breve de qué estrategia de búsqueda y heurística se ha empleado y qué heurística. Explicar claramente cuál es el aspecto más relevante y destacar su valor en el trabajo realizado en la práctica

**Comportamiento reactivo.** Dos páginas como máximo describiendo en pseudocódigo cómo se ha planteado el comportamiento reactivo, explicando los aspectos más relevantes del trabajo realizado en esta parte de la práctica.

**Comportamiento deliberativo.** Dos páginas como máximo describiendo en pseudocódigo cómo se ha planteado la búsqueda y explicando en detalle en qué consiste la heurística empleada.