

E



# ALGORITMOS BÚSQUEDA

JESÚS FERMÍN VILLAR RAMÍREZ

INGENIERÍA DEL CONOCIMIENTO: INTELIGENCIA ARTIFICIAL



# TABLA CONTENIDOS

- 03      Introducción**
- 04      Descripción del algoritmo**
- 05      Diseño e implementación**
- 06      Resultados**

**TÍTULO:** ALGORITMOS DE BÚSQUEDA  
**AUTOR:** JESÚS FERMÍN VILLAR RAMÍREZ  
**ASIGNATURA:** INGENIERÍA DEL CONOCIMIENTO: INTELIGENCIA ARTIFICIAL  
**DOCENTE:** LUIS PEÑA SÁNCHEZ  
**GRADO:** DISEÑO Y DESARROLLO DE VIDEOJUEGOS, ESNE  
**FECHA:** FEBRERO 2019  
**FOTOGRAFÍA DE PORTADA:** Fuente: <https://www.pexels.com/photo/man-standing-infront-of-forest-768267/>

# INTRODUC- CIÓN

---

Esta memoria corresponde a la primera práctica “Práctica 1. Búsqueda Offline” de la asignatura “Ingeniería del Conocimiento: Inteligencia Artificial” del grado oficial de Diseño y Desarrollo de Videojuegos, impartida por Luis Peña Sánchez en ESNE. En dicha práctica se buscaba implementar cuatro algoritmos de búsqueda para la resolución de cualquier laberinto; dos de ellos offline y otros dos online.

Este documento constituye la memoria de dicha entrega, mostrando la descripción de los algoritmos utilizados, el diseño e implementación de los mismos así como un análisis de los resultados obtenidos.

En Madrid, febrero de 2019

**Jesús Fermín Villar Ramírez  
Diseño y Desarrollo de Videojuegos**

# DESCRIP- CIÓN DEL ALGORITMO

---

Los algoritmos que se han utilizado para el método Offline son la búsqueda en amplitud y la búsqueda A estrella; mientras que los algoritmos Online elegidos son la Búsqueda por Ascenso de Colinas y la Búsqueda por Horizontes.

El algoritmo de búsqueda en amplitud recorre todos los nodos posibles hasta llegar al estado meta; expandiendo uno a uno cada nodo de la lista al sacarlo de la misma. Esto nos genera, al llegar a la meta, una lista de acciones que debe realizar el personaje hasta llegar a dicha meta “viajando” a través de la estructura generada por los enlaces a los padres de cada nodo.

El algoritmo de búsqueda A estrella es algo similar aunque incluye una heurística débil; un cálculo aproximado de la distancia a la meta por medio de la distancia Euclídea. Gracias a esto, en cada expansión podemos elegir aquel que tiene una menor distancia aproximada y suponer que nos acercamos a la solución.

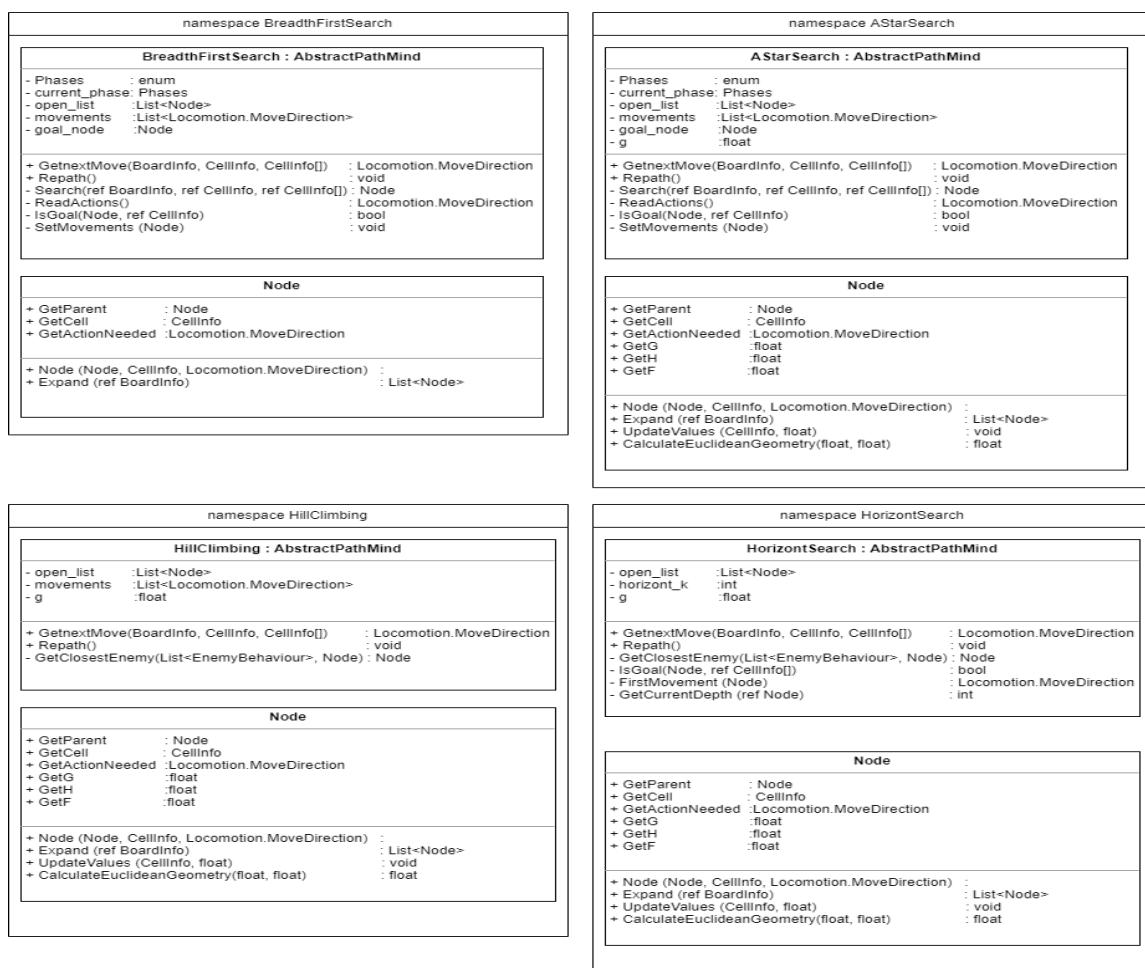
El algoritmo de ascenso de colinas es un poco distinto. Este, al ser un método de búsqueda online, en vez de buscar el camino a seguir al comienzo de la aplicación, decide escoger el paso más prometedor mediante una heurística. De este modo, cada vez que realiza una expansión escoge el nodo con menor distancia euclídea a la meta y realiza ese movimiento, no volviendo a expandir hasta encontrarse en este mismo nodo.

Por último, el algoritmo de búsqueda por horizonte también es un método online que se ejecuta cada vez que el personaje da un paso en su ruta. En este caso, al dar un paso, el algoritmo expande (por un mecanismo como la búsqueda en amplitud) hasta llegar a una profundidad dada. En este nivel escogerá el nodo más prometedor (igual que el algoritmo anterior) y realizará el primer paso que tiene que hacer para llegar hasta ese mismo nodo. Una vez dado el primer paso se vuelve a realizar la búsqueda con el mismo grado de profundidad.

# DISEÑO E IMPLEMEN- TACIÓN

La combinación de los cuatro algoritmos anteriores da un escenario variado. Todos ellos necesitan de un modelo de datos que contenga nodos; pero como estos son distintos dependiendo del algoritmo he optado por usar namespaces distintos para cada algoritmo.

Así, el código desarrollado se estructuraría de la siguiente forma:



# RESULTADOS

---

A nivel de resultados no he sabido muy bien exactamente cómo medir la performance desde Unity pero sí puedo comentar los siguientes aspectos:

## MEMORIA USADA

Sorprendentemente, y contra toda expectativa, las búsquedas online gastan más memoria RAM que las búsquedas offline tal y como se muestra en la gráfica inferior. La seed usada para guardar estos datos es la seed "30".

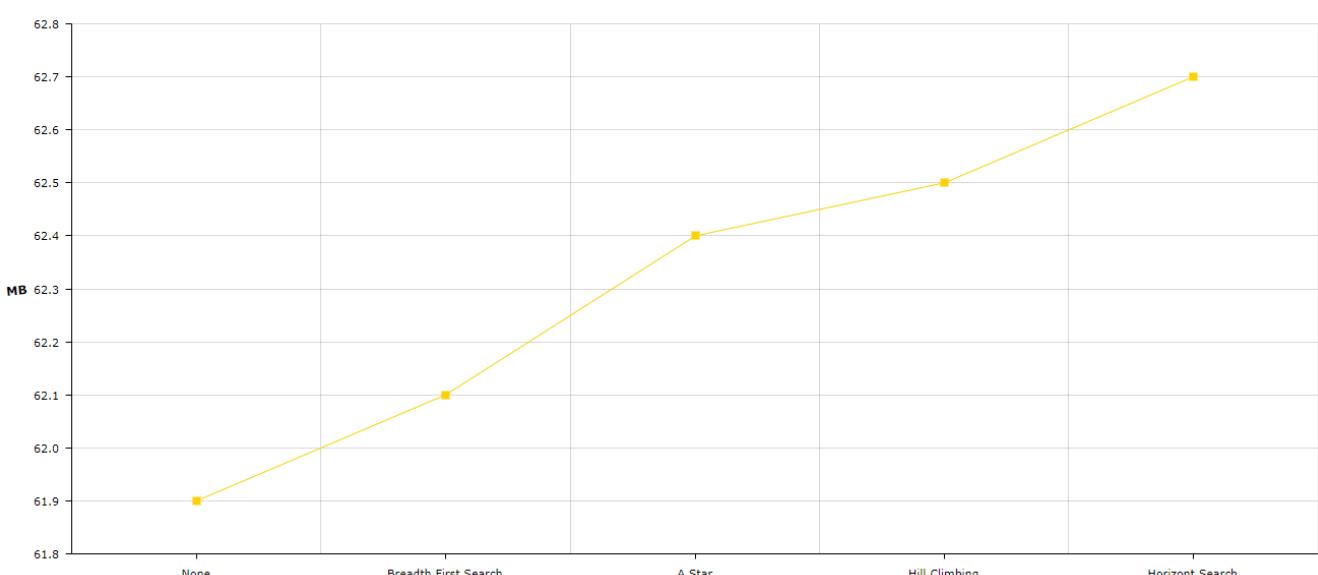
A la vista de estos resultados se entiende que o bien los algoritmos están mal implementados, o bien no se libera la memoria en los online en cada ciclo y, por tanto, se necesita más memoria para almacenar todos los nodos de todas las comprobaciones.

## BÚSQUEDAS ONLINE

La búsqueda en horizonte, pese a tener incorporada la funcionalidad de perseguir a los enemigos como el algoritmo de ascenso de colinas, va directamente a la salida pese a que debugueando su "goal" indique que es un enemigo.

## ESCENARIO

Se ha observado que hay algún tipo de error en la generación de mundo ya que hay algunos tiles no walkables que poseen la flag de walkable por lo que el algoritmo los reconoce como hábiles cuando no lo son y se queda "atascado" chocando contra esas paredes.





E