

## (一) 何謂 A\* Algorithm

常用於路徑查詢和圖形遍歷等算法，有較好的效能和準確度。而有較好的效能，則因採用 Heuristic Estimate 評估公式，此公式避免一些不必要的路徑

## (二) Heuristic Estimate 公式

$$F(n) = G(n) + H(n) \quad n: \text{目前節點}$$

目前節點的  
評估分數  
總和

從起點到目前  
節點實際移動  
距離

目前節點到終  
點的估計值

### (2.1) $H(n)$ 主導 A\* 表現方式

(2.1.1)

目前節點到終點的距離  $\begin{cases} > \\ = \\ < \end{cases} H(n) \Rightarrow \begin{cases} \Rightarrow \text{保證找到最短路徑, } H(n) \downarrow \Rightarrow \text{搜索越深} \\ \Rightarrow \text{只找到最佳路徑, 且能快速找到結果} \\ \Rightarrow \text{不能保證找到最短路徑, 但計算較快} \end{cases}$

(2.1.2)

$H(n) = 0$ : A\* Algorithm = Dijkstra Algorithm, 且保證找到最短路徑。

(2.1.3)

當  $H(n)$  與  $G(n)$  高度相關: A\* Algorithm 成為 BFS (Best-First Search)

## (三) 應用場景

常用於遊戲中 NPC 移動計算、網路遊戲 BOT 計算