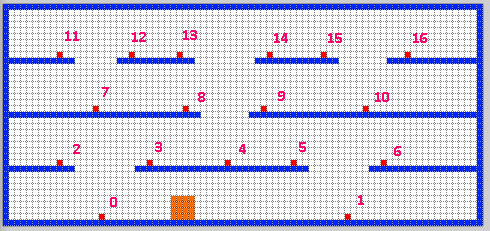
在2D平台使用的find path

* 使用 find path 工具匯出 Path array (如C++ array)
* 敵人的行動模式「巡邏」及「路徑」
* 巡邏 --- 在指定區域往來行走
* 路徑 --- 主動決定目的位置，並制定及遵循路徑走到位置
* 範例示範在2D平台路徑行走，不展示巡邏



**路徑、段落、位置點、行走方法**

路徑：例如由0到16的

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 3 | 4 | 9 | 10 | 16 |

段落：路徑段落及位置點(共5組)

|  |  |
| --- | --- |
| 由 | 到 |
| 0 | 3 |
| 3 | 4 |
| 4 | 9 |
| 9 | 10 |
| 10 | 16 |

段落位置之間的行走方法

|  |  |  |
| --- | --- | --- |
| 由 | 到 | 行走方法 |
| 0 | 3 | 向目標跨跳 |
| 3 | 4 | 直接行走 |
| 4 | 9 | 向目標跨跳 |
| 9 | 10 | 直接行走 |
| 10 | 16 | 向目標跨跳 |

重新思考行動

路徑

段落

已到達

核心使用state machine方法，只需3個主要狀態，初步AI便可使用：

**思考行動**

指示要巡邏或按路徑走到目的位置

**執行路徑**

確認路徑下一個段落，即把路段分成多個段落，順序執行

如走完路徑或走錯路，需重新進入思考行動；

**執行段落**

每個段落分為由A到B位置

按A, B位置，使用特定的2D平台方法 (如：直走，跨跳，墮下)