* 操作：

1. 選地圖：輸入地圖編號（ex： 要選擇mission1.txt ，輸入 1 即可）
2. 推箱子：輸入W（上）、D（右）、S（下）、A（左），大小寫均可
3. 拉箱子：輸入P + WDSA（向WDSA方向拉箱子），大小寫均可
4. 符號：&（玩家）、2（終點）、-（路）、/（牆）、@（箱子）

* Main說明：

1. 讀檔：將輸入編號與mission和.txt組合成字串讀檔
2. 執行：根據選擇模式呼叫相應物件

* Class說明：

1. Path：讓電腦判斷玩家是否可以走到指定位置（遞迴嘗試所有路徑）
2. Private：

* \_row、\_col、\_playerX、\_playerY、\_endpointX、endpointY參數：紀錄地圖的行列大小、玩家當前位置、終點位置
* \_arrive參數：紀錄是否能到達指定位置(bool)
* Vector<vector<char>> \_copyMp：複製一份地圖

1. Public：

* 建構子：初始化參數
* robot()：電腦遞迴嘗試路徑
* getArrive()：回傳\_arrive

1. Player：
2. Private：

* \_x、\_y、\_dx、\_dy、\_row、\_col、\_count參數：紀錄玩家位置、移動量、地圖行列大小、走了幾步
* vector<vector<char>> \_mp：紀錄地圖的二維vector
* vector<pair<int,int>> \_end：紀錄每個終點的座標
* vector<bool > \_finish：紀錄每個終點是否有箱子走到過
* setXY(int,int)：設定玩家位置
* Move()：判定玩家是否可移動
* moveXY()：移動\_x, \_y
* moveMap()：根據輸入的Move設定 \_dx和\_dy，並回傳是否有要拉箱子(當模式為拉箱子時)
* printMap()：輸出整張地圖

1. Public：

* Player()：初始化變數
* melcome()：輸出歡迎語
* setMap()：將地圖檔案的內容存入二維vector內

1. Box：
2. Private：

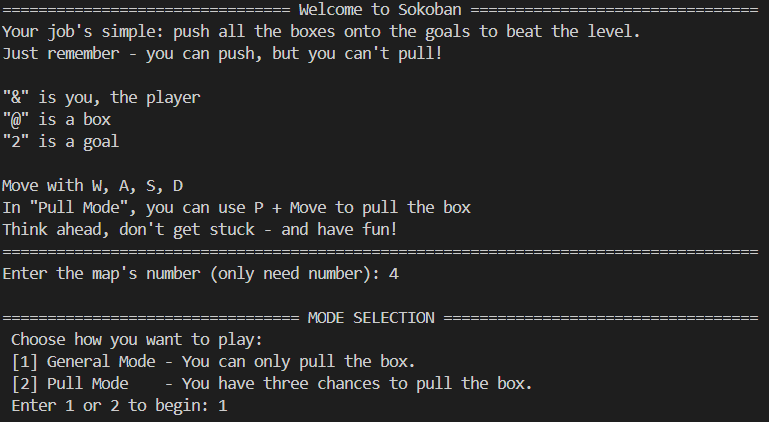
* \_temp\_box\_x、\_temp\_box\_y、\_idx：紀錄某一箱子的位置，以及箱子的索引
* vector<pair<int,int>> \_box：記錄每一個箱子的位置

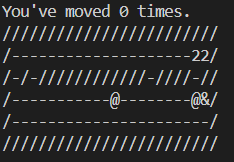
1. Public：

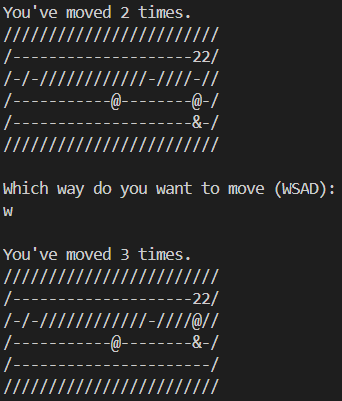
* Box()：初始化數據
* setBox()：設定\_box
* moveBox()：根據移動決定玩家及箱子的位置
* Pull()：推箱子模式
* Move\_box\_XY()：移動箱子座標
* Can\_move\_box()：判斷箱子是否可以移動
* Judge()：判斷遊戲是否結束
* 判斷結束邏輯：
  1. 判斷箱子周圍是否三面為牆，或是兩面為牆且兩面牆相鄰
  2. 若箱子兩面為牆且兩面牆相對，判斷是否存在路徑使玩家可以走到箱子旁邊
* 加分項：

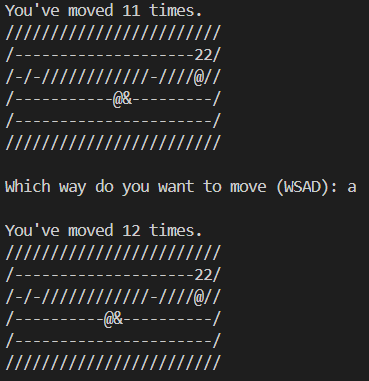
1. 可自動判定是否結束遊戲
2. 可處理多個箱子的情況
3. 拉箱子模式

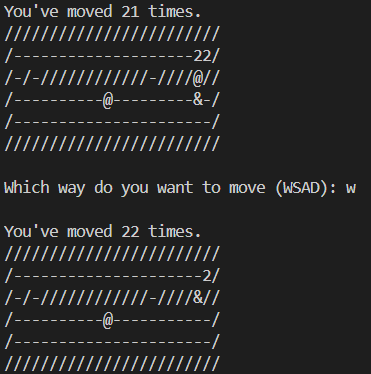
* 程式展示：

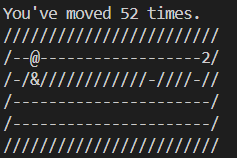
 < 開始介面

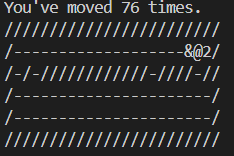
 < 可以兩個箱子同時在場

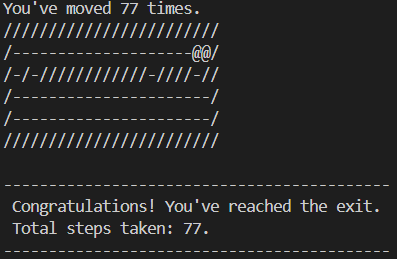
 < 推動箱子

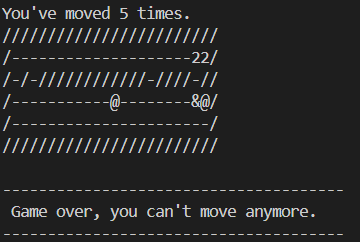
 < 推動另一個箱子

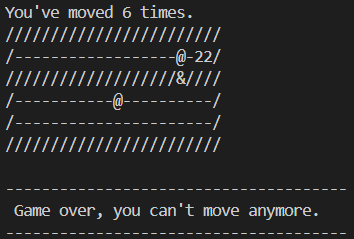
 < 一個箱子到達終點，箱子及終點一起消失，表任務完成

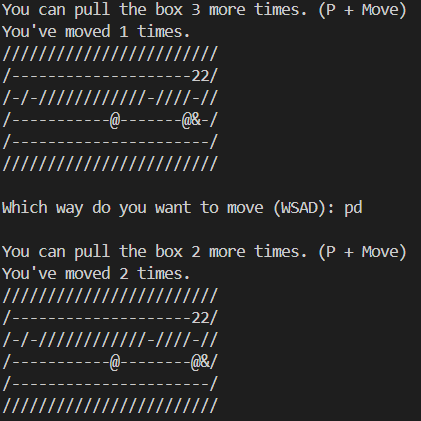
 < 仍有路徑可以到達箱子旁邊，因此遊戲不會結束

 < 不會重複判定已到達過的終點

 < 遊戲結束後，輸出所有箱子（終點）的位置

 < 推入死角，遊戲結束

 < 沒有路徑可以到箱子旁邊，遊戲結束

 < 拉箱子模式可以拉動箱子