1. 專案目標  
   此專案程式目標為處理Multi Agent Path Finding Problem，此內容涉及Task Assignment與Path Finding兩大方向，於此專案我們著重於Path Finding部分。
2. 專案架構

專案內有兩個Java Project，MPP\_Collision與MPP\_Collision\_Demo，建議使用前者進行修改，後者為Demo用之專案。

以下列出MPP\_Collision中所有Java檔及說明

1. Main\_Dynamic.java : 專案主程式，執行專案需執行此程式。
2. NewCollision.java : 偵測與處理碰撞。
3. MapBuilder.java : 產生 map (grid) 的主程式，定義map所有點、邊、距離、方法。
4. Robot.java : 定義Robot 物件與其屬性、方法。
5. Graph.java : 定義map上的資料結構與方法，處理map之資料結構。
6. Node.java : 定義 Node的資料結構與方法。
7. DijkstraAlgorithm.java : 定義最短路徑演算法。
8. AddRobot.java : 處理「指派靜止Robot新任務」。
9. draw.java : 視覺化處理。
10. 專案使用說明

輸入：機器人數量、機器人工作相對優先權、機器人起終點

輸出：每台機器人路徑

以下為程式執行之流程  
1. 執行Main\_Dynamic.java。  
2. Console Mode 會出現「**Please input the number of robot**」並輸入預先設定之Robot數量。  
3. Console Mode 會出現「**Please input the (int)weight,(String)Source,(String)Destination of robot**」 則輸入每個Robot的任務優先權、起點、終點。  
4. 輸入完成後會先出現「**cal Collision-Free Path:**」即計算出Collision Free Path 所使用的時間(單位為ns)。

5. 接著會顯示完整的路徑（橫軸為每個Robot R0~Rn、縱軸為每個Time step），對應到的是此機器人在此Timestep抵達的Node。

6. 每3秒機器人會走到Path的下一個Node（時間可以自行修改，但建議不要低於1秒），當Console Mode顯示「**Robot1 offer to Stop Queue**」，可以在任何時間點在Console Mode按下鍵盤「i」，即可以使完成工作的Robot從結束的點到新的位置。