

物件導向程式設計期末報告
報告題目：成大地圖導航

學生：林德祐

系級班別：機械碩一

學號：N16054530

E-mail：jj21065@gmail.com

內容

一、摘要.....	2
二、動機與目的.....	3
三、程式設計流程.....	4
四、程式類別架構.....	5
五、路徑規劃演算法.....	6
Floyd 演算法.....	6
A*演算法	7
軟體介面/操作介紹	9
結果與未來展望.....	11
參考文獻.....	12

一、摘要

物件導向的概念，在程式設計中非常實用，而本次的專題研究題目為結合 3D 模型的顯示做地圖導航，便是一個很好應用物件導向概念的一個題目，地圖中的節點、個別的地圖、3D 模型物件等等都可以是獨立的物件，方便管理，且邏輯清楚。

目前此程式以成功大學為主要地圖，建立了自強校區的建築 3D 模型和地圖節點資料庫，使用者可以以 2D 方式檢視地圖，或是在 3D 模式下對地圖做平移、旋轉和縮放等動作，能夠更清楚檢視地圖的資訊。程式也包含如 Google Map 的路徑導航功能，使用者能夠點地圖上的兩個節點作為起點終點，規劃最短路徑，除了起點終點外也能加入中途想經過的其他節點做路徑規劃。

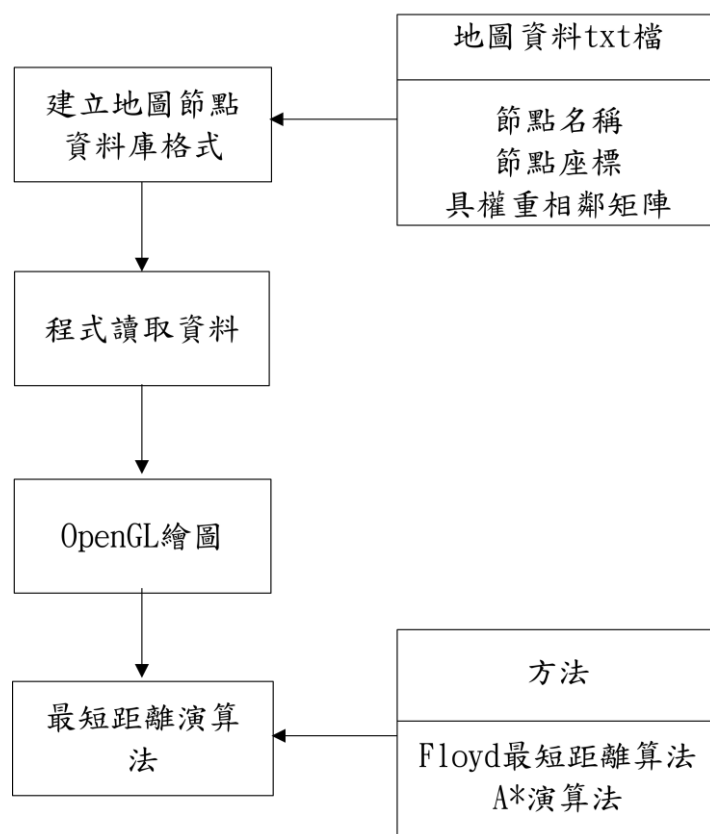
二、動機與目的

對於大學新生來說，對學校還不熟悉時，在找教室、行政單位等地點或建築時總是比較困難，且有的人對於 2D 的地圖可能較沒有方向感，從 2D 地圖上可能無法完全清楚的顯示細部的資訊，而 GoogleMap 中對於學校內部的資訊可能也不是最完整或是最新的，因此希望能實做類似 GoogleMap 功能且專屬學校的地圖導航系統。

本專題實作目的為實做校園地圖導航系統，能夠以 3D 方式顯示地圖建築、地標，且使用者能夠自訂起點、終點或中途點來規劃路徑，甚至能包含教室、辦公室位置等等細部資訊，最終目的希望能夠整合至手機、平板等等行動裝置。

三、程式設計流程

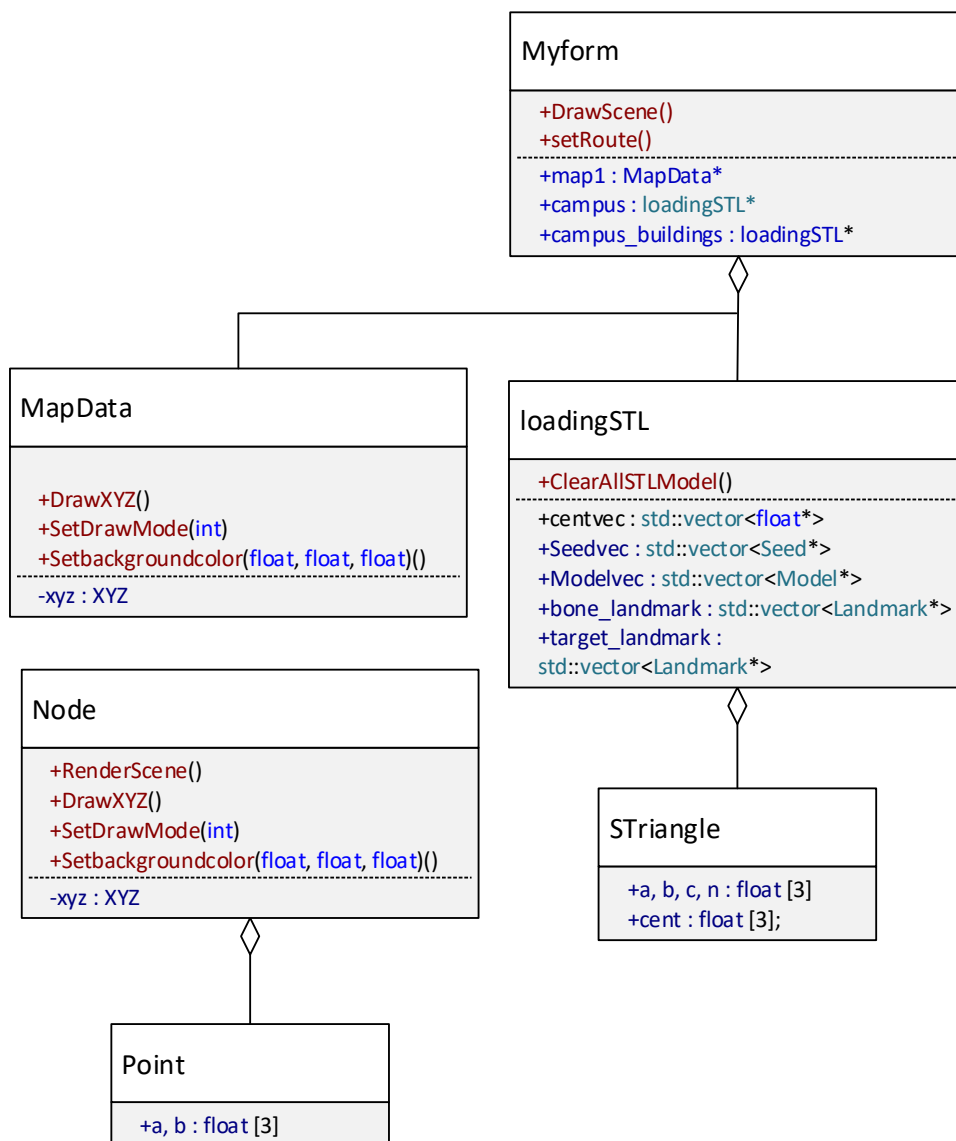
為了紀錄地圖的資訊，在 excel 檔案中記錄地圖節點的座標、名稱、相鄰矩陣等等資訊，輸出成 txt 檔後程式再做讀取。有了這些資料便可以做後續繪出節點以及節點間最短路徑的分析。最短路徑演算法這邊使用 Floyd 最短距離演算法和 A*演算法來做路徑規劃。



圖一、程式設計流程圖

四、程式類別架構

主要架構包含一 windows form 類別，負責介面的設計和 OpenGL 的繪圖顯示等等。其中包含 LoadingSTL 類別做 STL 3D 模型物件的建立與顯示。MapData 類別記錄地圖物件，管理地圖中節點、路徑等等的運算，Node 類別記錄節點資訊。



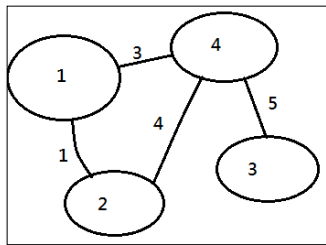
圖二、程式類別架構

五、路徑規劃演算法

Floyd 演算法

全名 Floyd-Warshall 演算法，可以用來解決有向圖中任一兩點間最短距離的演算法，其概念和使用都很簡單。演算法的時間複雜度為 $O(N^3)$ 。

使用 Floyd 演算法前，要先建立相鄰矩陣。假設有一無向圖如下圖三，可以其相鄰矩陣 D 便如圖四， $D(i,j)$ 代表第 i 節點到 j 節點的距離，因此可以看到當 $i == j$ 時距離是 0，而不相鄰的節點距離則設為 x (無限大)。



圖三、一無向圖

0	1	X	3
1	0	X	6
X	X	0	5
3	6	5	0

圖四、相鄰矩陣 D

其演算法也很簡單，概念為藉由上方相鄰矩陣的資料，對每個節點檢查到其他節點有沒有更短的距離，有的話便取代原來矩陣中的資料，結果如下圖五。

0	1	8	3
1	0	9	4
8	9	0	5
3	4	5	0

圖五、最短距離矩陣

Floyd程式實作概念：


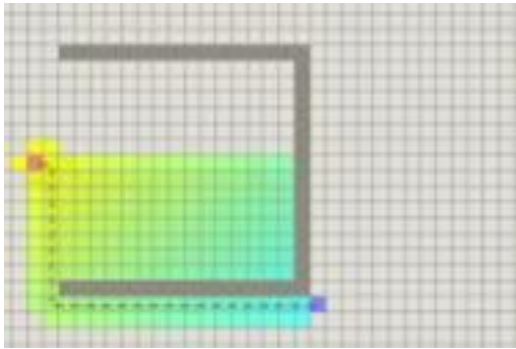
```
for (int k = 0; k < Node_num; k++)
for (int i = 0; i < Node_num; i++)
for (int j = 0; j < Node_num; j++)
    if (dist2[i*Node_num + j] > dist2[i + k*Node_num] + dist2[k*Node_num + j])
        dist2[i*Node_num + j] = dist2[i + k*Node_num] + dist2[k*Node_num + j];
```

A*演算法

A*演算法為一廣泛被使用的演算法，用於一般的 RPG 遊戲、戰略遊戲等等。其概念以一個評估公式 $F = G + H$ 來判斷下一步應該走哪個節點。G 代表走到鄰近節點的移動代價，H 代表從所選的相鄰節點到終點的移動代價，而這裡的移動代價就是節點間的最短距離，因此可以藉由 Floyd 演算法所得到的矩陣來計算。

在程式中，從起始節點出發，先尋訪週圍相鄰的節點計算出 F，選擇 F 值最小的節點做為下一個路徑點，反覆計算移動到終點為止，便能得到從起點到終點間路徑。

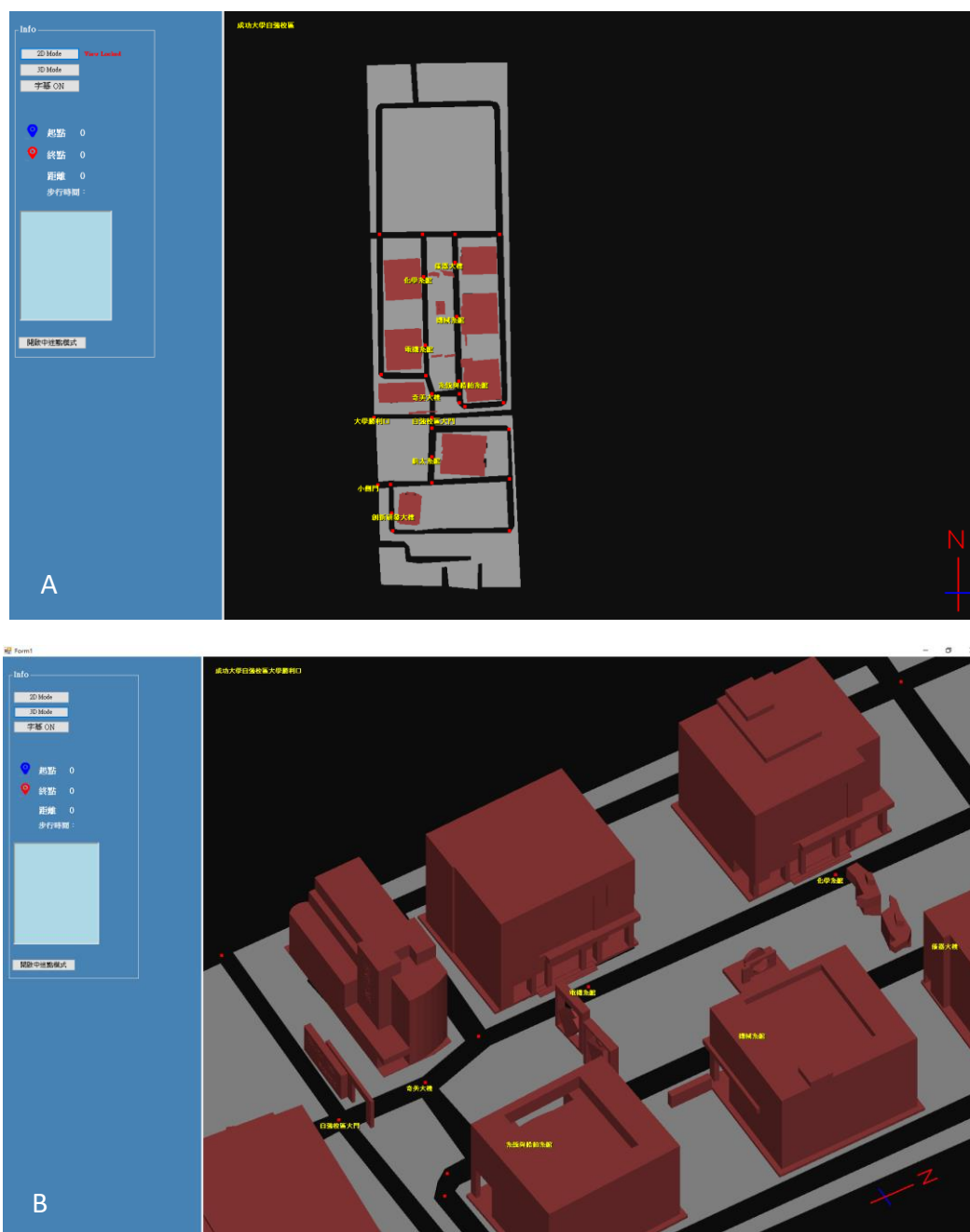
實做中以 vector 作為容器，不斷地塞入節點作為路徑資料。

	A*演算法
無障礙物路徑	
有障礙物路徑	

*張傑,以改良的 A*演算法規劃較佳導引路徑之研究,大同大學資訊工程研究所,2009

軟體介面/操作介紹

下圖六為軟體介面，右邊欄位包括選擇 2D/3D 視角的切換、切換地圖地標名稱的顯示、起點終點、距離、步行時間、經過的地點等功能。地圖中的紅點為節點，讓使用者方便點選。



圖六、A:軟體介面，2D 視角，B:3D 視角下旋轉縮放視角

在 2D 視角模式下無法做旋轉，只能平移、縮放地圖。3D 視角模式下可以自由旋轉平移與縮放。

用滑鼠點選地圖任意兩節點，第一點為起點，第二點為終點，便會產生一綠色虛線代表其規劃的最短路徑，從左方資訊欄也可以看到路徑資訊。



圖七、最短路徑規劃

除了一般的最短路徑模式，中途點模式可以先點選起點終點後，再點選其他節點作為欲經過中途點。如下圖，起點為航太系館，終點為化學系館，但若再點選機械系館和儀器大樓，便會在自動計算一新的最短路徑。



圖八、中途點模式路徑規劃

結果與未來展望

目前本程式可以讓使用者決定起點、終點或中途節點，並規劃最短路徑。建立 3D 地圖，包含對地圖的平移、旋轉、縮放。Floyd 演算法雖然須執行三個 for 迴圈，但只需在剛載入地圖時執行一次運算即可，目前最短路徑演算法效率還算高。

未來能夠繼續增加地圖資訊資料庫，地圖或是細部教室資料庫，提供使用者更多細部的地圖資訊，當功能和資料庫完整後，未來希望可以整合到行動裝置 app 上，如此導航程式的價值會更高。

參考文獻

- [1]. 張傑,以改良的 A*演算法規劃較佳導引路徑之研究,大同大學資訊工程研究所,2009
- [2]. Google Map