

亞洲大學-資訊工程學系

(109-1 學年度)

資訊數學-期末專題

轉換矩陣動畫

(報告超連結點此)

郭諺勳(109021058)

顏宏宇(109021009)

林書雋(109021062)

劉子萱(109021362)

陳昇新(109021076)

呂偉宏(109021065)

指導老師：蔡志仁老師

目錄

摘要.....	- 4 -
第一章 專題研究的題目設計的動機和目的.....	- 5 -
1-1 專題研究動機	- 5 -
1-2 專題研究目的	- 5 -
第二章 開發環境、硬體、軟體介紹.....	- 6 -
2-1 開發環境簡介	- 6 -
2-2 硬體環境介紹	- 6 -
2-3 軟體環境介紹	- 8 -
第三章 研究方法及步驟、GUI 設計介紹.....	- 9 -
3-1 繪製初期圖像	- 9 -
3-2 定位初始車廂	- 11 -
3-3 優化豐富度之前置作業.....	- 12 -
3-3 繪製摩天輪與車輛動畫.....	- 13 -
3-4 GUI 介面外觀設計.....	- 16 -
3-5 GUI 介面 call back 嫁接.....	- 18 -
第四章 實驗結果	- 20 -
第五章 小組成員工作分配表	- 22 -

第六章 進度表 (直條圖)	- 23 -
第七章 計畫配合事項及限制	- 24 -
第八章 結論與心得.....	- 24 -
第九章 未來展望	- 27 -

摘要

本次的專題主題為"摩天輪(轉換矩陣動畫)"以研究數學與動畫的相關關係，並運用特定的數學方程式以及創意設計製作出動畫製圖以及遊戲場景。

在現今科技發達跟資訊爆炸的時代，數學在生活上的應用變得更加普及也相對枯燥乏味，故在現今資訊化的數學應用的文載探討方面仍然需要注入更多的創意設計部分。

在本研究主題中，團隊遇到的困難點在於 GUI 介面在 MAC 電腦設計後於 Windows 系統開啟操作介面會與原設計位置相異。在此方面團隊仍然還在尋找問題點以及解決的辦法。

專題設計之實行方向為將開發程式搭載團隊設計之人機互動 GUI 以使使用者能操作並輸入部分參數來與城市互動並調整視覺化結果。

團隊希冀在未來此研究在這個資訊爆炸的時代裡是一個有效的運用，諸如在網頁頁面上用於美化以及分類的用途使頁面更加的生動活潑，應用於遊戲中提高畫面表現與優化使用者體驗，亦可結合人工智慧方面簡化在操作上之繁複程序致使連孩童都能更直覺的與人工智慧產物共同互動。

第一章 專題研究的題目設計的動機和目的

1-1 專題研究動機

於 1980 年代，桌上型個人電腦日益普及，乃至現今手持電子裝置人手不離，團隊可以預見在未來物聯網甚至車聯網的時代必然將至。

在各式電子裝置滿足人們生活基本生活需求的現代，娛樂性電子亦是開發重點之一。團隊藉研究初步的矩陣轉換來了解數據化動畫的製程，並搭載簡易之 GUI 介面使使用者能以之互動。

1-2 專題研究目的

本團隊藉由了解數據化動畫的基本生成原理與製作方式作為未來於電子設備上娛樂性軟體亦或網頁介面等美化與操作之學習基石。

然而僅有設計優良之程式軟體並無法令使用者與之互動。故本團隊亦藉由研究 GUI 介面之設計，並製作初步之 GUI 介面使使用者能輕易地操作其內部數據，使其能依使用者自由意識改變視覺化結果。

綜合以上兩者之研究結果能提升團隊相關開發經驗，並致力於開發出更人性化的直覺操作介面，以達到所有成人乃至孩童皆能輕易操作之目的以求電子設備能更普及於人們生活中。

第二章 開發環境、硬體、軟體介紹

2-1 開發環境簡介

本專題使用美國 The MathWorks 公司出品的商業數學軟體 MATLAB_R2020B，並於 MAC 的 10.15.5 (19F101) 版本系統之基著上開發。

2-2 硬體環境介紹

本專題之開發硬體為：筆記型電腦 Apple-MacBook Pro (15-inch, 2018)。



(圖 2-2-1) MacBook Pro (15-inch, 2018)

搭載 2.6 GHz 6 核心 Intel Core i7 處理器，記憶體配備為 16 GB 2400 MHz DDR4，顯示卡則為 Intel UHD Graphics 630 1536 MB。



(圖 2-2-2) 硬體環境

2-3 軟體環境介紹

本專題使用由美國 The MathWorks 公司出品的商業數學軟體 MATLAB_R2020B 開發。



(圖 2-3) MATLAB_R2020B

MATLAB 是一種用於演算法開發、資料視覺化、資料分析以及數值計算的進階技術計算語言和互動式環境。除矩陣運算、繪製函數/資料圖像等常用功能外，MATLAB 還可用來建立使用者介面，以及呼叫其它語言 (包括 C、C++、Java、Python、FORTRAN) 編寫的程式。

MATLAB 主要用於數值運算，但利用為數眾多的附加工具箱，它也適合不同領域的應用，例如控制系統設計與分析、影像處理、深度學習、訊號處理與通訊、金融建模和分析等。

第三章 研究方法及步驟、GUI 設計介紹

3-1 繪製初期圖像

首先直接以陣列的形式儲存摩天輪底座之圖像數值數據。

```
%Btm  
x_btm_tri = [-3.5 0 3.5 -3.5];  
y_btm_tri = [-10 0 -10 -10];
```

(圖 3-1-1) 繪製底座之程式碼

先預先寫出複製摩天輪中車廂的複製函式，其中下圖中的變數 m 控制車廂的總數量，因為車廂會被定位在後續定義的圓形旋轉軌道上，故將圓的 360 度除以車廂總數以計算每個車廂之角度差變數 d 。

變數 A 為線性轉換之矩陣，將變數 d 帶入其中以計算所有車廂在相對角度中的實際位置以便繪製。

```
%Copy.  
m = 8; %Number of boxes.(GUI)  
d = 2 * pi / m; %Interval(間隔) angle.  
p_box = 7; p_tri = 3; %Set the number of coordinate points.  
  
A = [cos(d) sin(d); %Set location of other boxes.  
     -sin(d) cos(d)]; %線性轉換矩陣 (Clockwise)
```

(圖 3-1-2) 車廂線性轉換之程式碼

摩天輪的旋轉軌道也在此先繪製完成，將每個點的間距控制在 0.1 已讓其在視覺上看似圓形。注意，如果間距過大會繪製出具稜角的圖形，這樣的結果並非此處的需求。

```
%Circle  
theta=0:0.1:2*pi;  
Circle_r = 7.5;  
Circle1=Circle_r*cos(theta);  
Circle2=Circle_r*sin(theta);
```

(圖 3-1-3) 繪製運動軌跡之程式碼

在初步製圖過程，先設定第一個車廂之繪製數值，以便後續利用線性轉換矩陣作運算繪製其餘車廂。

```
%Plot first box.  
X_box(1, :) = [0 0 -1 -1 1 1 0];  
Y_box(1, :) = [0 5 5 7 7 5 5];  
  
X_tri(1, :) = [-1 0 1];  
Y_tri(1, :) = [7 7.5 7];
```

(圖 3-1-4) 繪製第一個車廂之程式碼

3-2 定位初始車廂

使用 for 迴圈將初始車廂數據依照需求車廂數量 (變數 m) 重複乘以前面宣告的線性轉換矩陣

(變數 A) 後儲存，迴圈結束後將所有車廂的初始位置繪製出來。

```
for i = 2 : m
    X_box(i, :) = A(1, :) * [X_box(i-1, :); Y_box(i-1, :)];
    Y_box(i, :) = A(2, :) * [X_box(i-1, :); Y_box(i-1, :)];

    X_tri(i, :) = A(1, :) * [X_tri(i-1, :); Y_tri(i-1, :)];
    Y_tri(i, :) = A(2, :) * [X_tri(i-1, :); Y_tri(i-1, :)];
end

plot(X_box', Y_box', 'b', 'LineWidth', 3);
plot(X_tri', Y_tri', 'b', 'LineWidth', 3);

axis([-12 12 -12 12]),axis square;
```

(圖 3-2-1) 繪製各車廂初始位置之程式碼

3-3 優化豐富度之前置作業

實際上程式的初步構造在專題開始沒多久就幾乎已經完成，故團隊擬優化程式之豐富度，同時在後期 GUI 介面設計上亦能展現更多元件功能實作。

團隊在這摩天輪遊樂園中加入了歡快的音樂與來往的車輛以營造活絡的氛圍。車輛的部分暫先繪製其初始位置，在後續會增添移動的視覺效果。

```
%Sound
[y, Fs] = audioread("Matlab1_1_Topic.mp3");
sound(y, Fs);
```

(圖 3-2-1) 背景音樂導入之程式碼

```
x_cars1 = [14 12.5 12.5 13.5 14.5 16.5 17.5 18.5 18.5 17];
y_cars1 = [-10 -10 -9 -9 -7.5 -7.5 -9 -9 -10 -10];
x_cars2 = [15 16];
y_cars2 = [-10 -10];

x_win1 = [13.8 14.6 15.5 15.5 13.8];
y_win1 = [-9 -7.8 -7.8 -9 -9];
x_win2 = [15.7 15.7 16.4 17.2 15.7];
y_win2 = [-9 -7.8 -7.8 -9 -9];

wheel_r = 0.5;
x_wheel1 = [14.5+wheel_r*cos(theta)];
y_wheel1 = [-10+wheel_r*sin(theta)];
x_wheel2 = [16.5+wheel_r*cos(theta)];
y_wheel2 = [-10+wheel_r*sin(theta)];

p_wheel3 = 2;

x_wheel3_1_1 = [14 15];
y_wheel3_1_1 = [-10 -10];
x_wheel3_1_2 = [14.25 14.75];
y_wheel3_1_2 = [-9.55 -10.45];
x_wheel3_1_3 = [14.75 14.25];
y_wheel3_1_3 = [-9.55 -10.45];

x_wheel3_2_1 = [16 17];
y_wheel3_2_1 = [-10 -10];
x_wheel3_2_2 = [16.25 16.75];
y_wheel3_2_2 = [-9.55 -10.45];
x_wheel3_2_3 = [16.75 16.25];
y_wheel3_2_3 = [-9.55 -10.45];
```

(圖 3-2-2) 繪製各車輛初始位置之程式碼

3-3 繪製摩天輪與車輛動畫

程式的部分進入到最後階段，在此我們先預設一些控制項，部分控制項在 GUI 上 call back 實作上會予使用者自由調整數據以改變影像輸出結果，使用者在後續能夠調整動畫的速度、轉動的次數等。

另外值得一提的是，團隊為了能讓動畫更趨有趣，在此增加了摩天輪與車輛在每次轉動中變色的功能，配合迴圈中的隨機數值能讓每次轉動都能隨機變色。

```
n = 100; %Number of rotations(轉動) once time.  
angle = 2 * pi / n;%Set turn angle.  
B = [cos(angle) sin(angle);  
     -sin(angle) cos(angle)]; %線性轉換矩陣 (Clockwise)  
  
color = ['m', 'c', 'g'];  
color_car = ['c', 'g', 'm', 'r'];  
ci = 0;  
  
times = 3; %(GUI)
```

(圖 3-3-1) 控制項設置之程式碼

在此進入動會繪製階段，於第一層迴圈中加入隨機變數以帶入第二層迴圈中的參數，以使摩天輪與車輛顏色能隨機變換外，還能使車輛行駛產生隨機速度。

```
%Start turnig. Every turn "angle" angle.
for i = 1 : times

    ci = ci + 1;
    ran_car_color1 = randi(4);
    ran_car_color2 = randi(4);
    ran_car_color3 = randi(4);
    ran1 = randi([-3,-2],1);
    ran2 = randi([-1,0],1);
    ran3 = randi([1,2],1);
    fast1 = randi([35,100],1);
    fast2 = randi([200,300],1);
    fast3 = randi([600,1000],1)
```

(圖 3-3-2) 隨機顏色、速度之程式碼

每次迴圈中轉換至下一個座標點並重新賦值以利下次的運算並繪製摩天輪。

```
j = 1 : n

hold on;
axis([-12 12 -12 12]),axis square;

x_temp_box = B(1, :) * [reshape(X_box', 1, p_box*m);
    reshape(Y_box', 1, p_box*m)];
y_temp_box = B(2, :) * [reshape(X_box', 1, p_box*m);
    reshape(Y_box', 1, p_box*m)];
x_temp_tri = B(1, :) * [reshape(X_tri', 1, p_tri*m);
    reshape(Y_tri', 1, p_tri*m)];
y_temp_tri = B(2, :) * [reshape(X_tri', 1, p_tri*m);
    reshape(Y_tri', 1, p_tri*m)];

%New coordinate of x,y.
x_new_box = reshape(x_temp_box, p_box, m)';
y_new_box = reshape(y_temp_box, p_box, m)';
x_new_tri = reshape(x_temp_tri, p_tri, m)';
y_new_tri = reshape(y_temp_tri, p_tri, m)';
```

(圖 3-3-3) 摩天輪轉換繪製之程式碼

而車輛位移的部分，利用數值遞減的方式使其由右方物移至左方，並利用第一層回圈之隨機變數做速度參數之運算使其每次運算時會有不同的遞減數據以產生隨機速度。

```
%Car1 speed
x_cars1 = x_cars1 - fast1/n;
x_cars2 = x_cars2 - fast1/n;

x_win1 = x_win1 - fast1/n;
x_win2 = x_win2 - fast1/n;

x_wheel1 = x_wheel1 - fast1/n;
x_wheel2 = x_wheel2 - fast1/n;

x_wheel3_1_1 = x_wheel3_1_1 - fast1/n;
x_wheel3_1_2 = x_wheel3_1_2 - fast1/n;
x_wheel3_1_3 = x_wheel3_1_3 - fast1/n;

x_wheel3_2_1 = x_wheel3_2_1 - fast1/n;
x_wheel3_2_2 = x_wheel3_2_2 - fast1/n;
x_wheel3_2_3 = x_wheel3_2_3 - fast1/n;
```

(圖 3-3-4) 車輛速度亂數運算之程式碼

```
%Car1
plot(x_cars1, y_cars1 + ran1, color_car(ran_car_color1), 'LineWidth', 3);
plot(x_cars2, y_cars2 + ran1, color_car(ran_car_color1), 'LineWidth', 3);

plot(x_win1, y_win1 + ran1, color_car(ran_car_color1), 'LineWidth', 1.5);
plot(x_win2, y_win2 + ran1, color_car(ran_car_color1), 'LineWidth', 1.5);

plot(x_wheel1, y_wheel1 + ran1, color_car(ran_car_color1), 'LineWidth', 3);
plot(x_wheel2, y_wheel2 + ran1, color_car(ran_car_color1), 'LineWidth', 3);

plot(x_wheel3_1_1, y_wheel3_1_1 + ran1, color_car(ran_car_color1), 'LineWidth', 1.5);
plot(x_wheel3_1_2, y_wheel3_1_2 + ran1, color_car(ran_car_color1), 'LineWidth', 1.5);
plot(x_wheel3_1_3, y_wheel3_1_3 + ran1, color_car(ran_car_color1), 'LineWidth', 1.5);

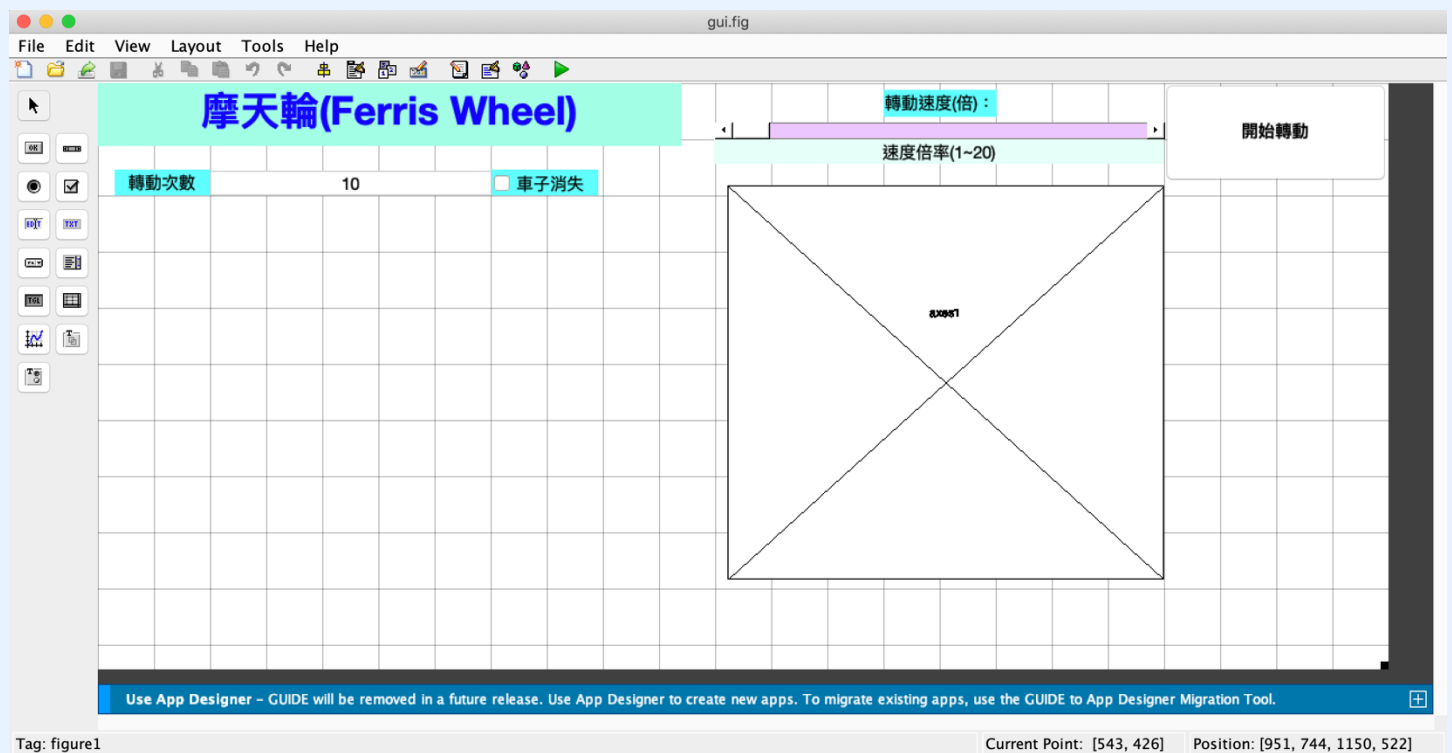
plot(x_wheel3_2_1, y_wheel3_2_1 + ran1, color_car(ran_car_color1), 'LineWidth', 1.5);
plot(x_wheel3_2_2, y_wheel3_2_2 + ran1, color_car(ran_car_color1), 'LineWidth', 1.5);
plot(x_wheel3_2_3, y_wheel3_2_3 + ran1, color_car(ran_car_color1), 'LineWidth', 1.5);
```

(圖 3-3-5) 車輛位移之程式碼

3-4 GUI 介面外觀設計

在 GUI 介面中，團隊除放入基本的 Static Text 元件做簡易的控制項說明及標題與 Axes 元件顯示繪製結果外，更放入 Edit Text 予使用者輸入整數數值控制轉動次數。

另外置入了 Slider 元件使使用者能倍化動畫速度，如果覺得車輛礙眼，也貼心的提供能取消車輛顯示的 Check Box 元件。

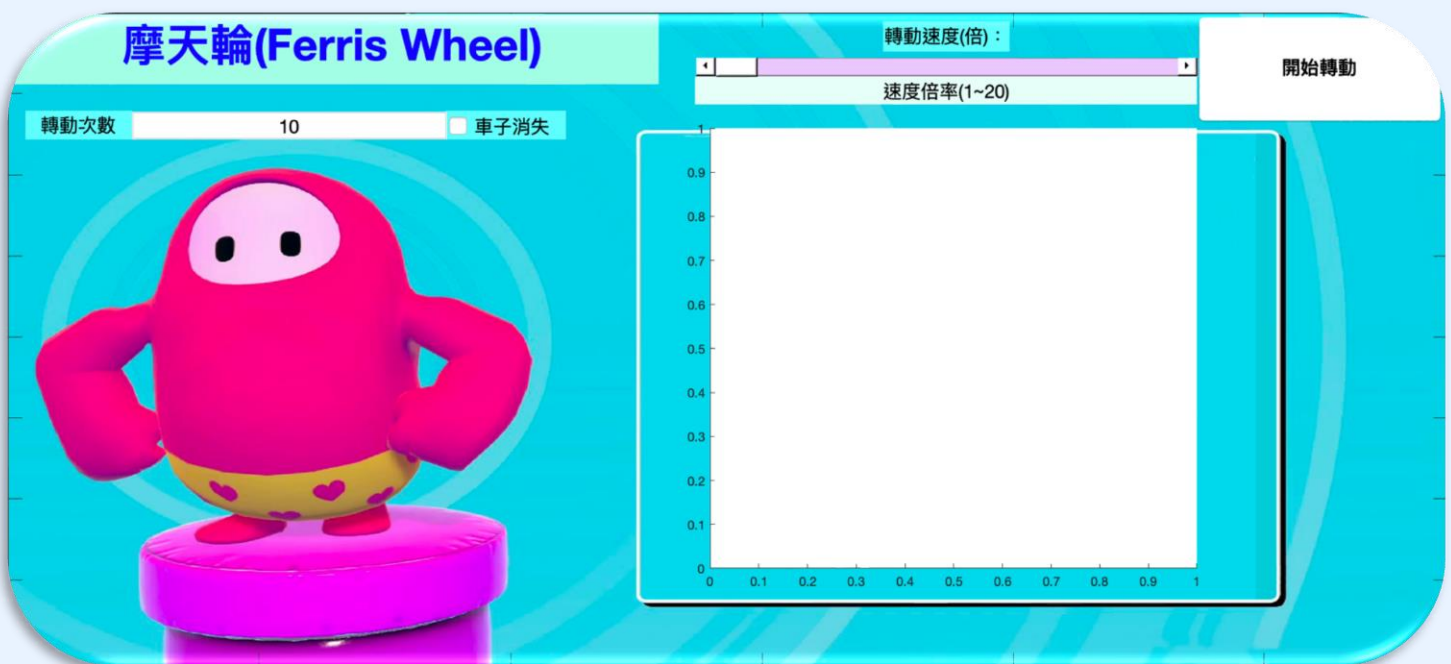


(圖 3-4-1) GUI 介面外觀設計

在 GUI 初始創建的 function 中，團隊導入了背景圖片以增加介面的活潑感。

```
function figure1_CreateFcn(hObject, eventdata, handles)
    ha=axes('units','normalized','pos',[0 0 1 1]);
    uistack(ha,'down');
    background=imread('background.jpg');
    image(background);
    colormap gray
    set(ha,'handlevisibility','off','visible','on');
    % hObject    handle to figure1 (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    empty - handles not created until after all CreateFcns called
```

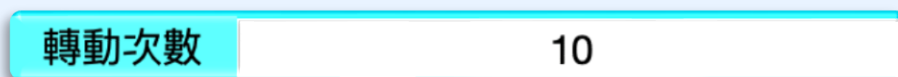
(圖 3-4-2) GUI 介面背景之程式碼



(圖 3-4-3) GUI 介面外觀成果

3-5 GUI 介面 call back 嫁接

使用者能藉由在轉動次數控制項中輸入整數次數以實現對次數的控制，在下圖程式碼中能看到其取得 String 參數並利用 str2num 函式將其轉換為數值型態存入，最後將其帶入 value 參數，此參數在後續將被提取出來於程式碼中做次數控制之應用。



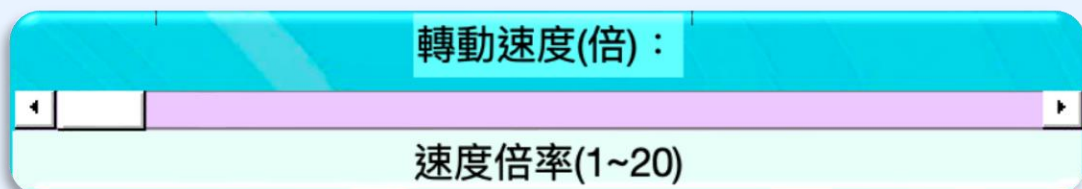
(圖 3-5-1) 轉動次數控制之介面

```
function times_edit_Callback(hObject, eventdata, handles)
% hObject      handle to times_edit (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of times_edit as text
%        str2double(get(hObject,'String')) returns contents of times_edit as a double
times_edit_value = str2num(get(hObject,'String'));
set(handles.times_edit,'value', times_edit_value);
```

(圖 3-5-2) 轉動次數控制之程式碼

在轉動速度控制項中，可以運用拉桿調整轉動速度之倍率，並於程式碼中提取拉桿之數值 Value，並用 set 函式替換下列 Text 元件顯示之字幕，使使用者能更直覺式的操作其倍率。



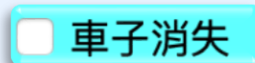
(圖 3-5-3) 轉動速度控制之介面

```
% --- Executes on slider movement.
function speed_slider_Callback(hObject, eventdata, handles)
% hObject      handle to speed_slider (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range of slider
slider_value = get(hObject,'Value');
set(handles.speed_text,'String', slider_value);
```

(圖 3-5-4) 轉動速度控制之程式碼

Check Box 能控制車輛是否顯示。程式碼中提取其 Value 做布林值開關，置於按鈕元件中一開關狀態執行不同程式碼。



(圖 3-5-5) 車輛顯示控制之介面

第四章 實驗結果



(影片 4-1) 轉動次數控制

(影片 4-2) 轉動速度控制

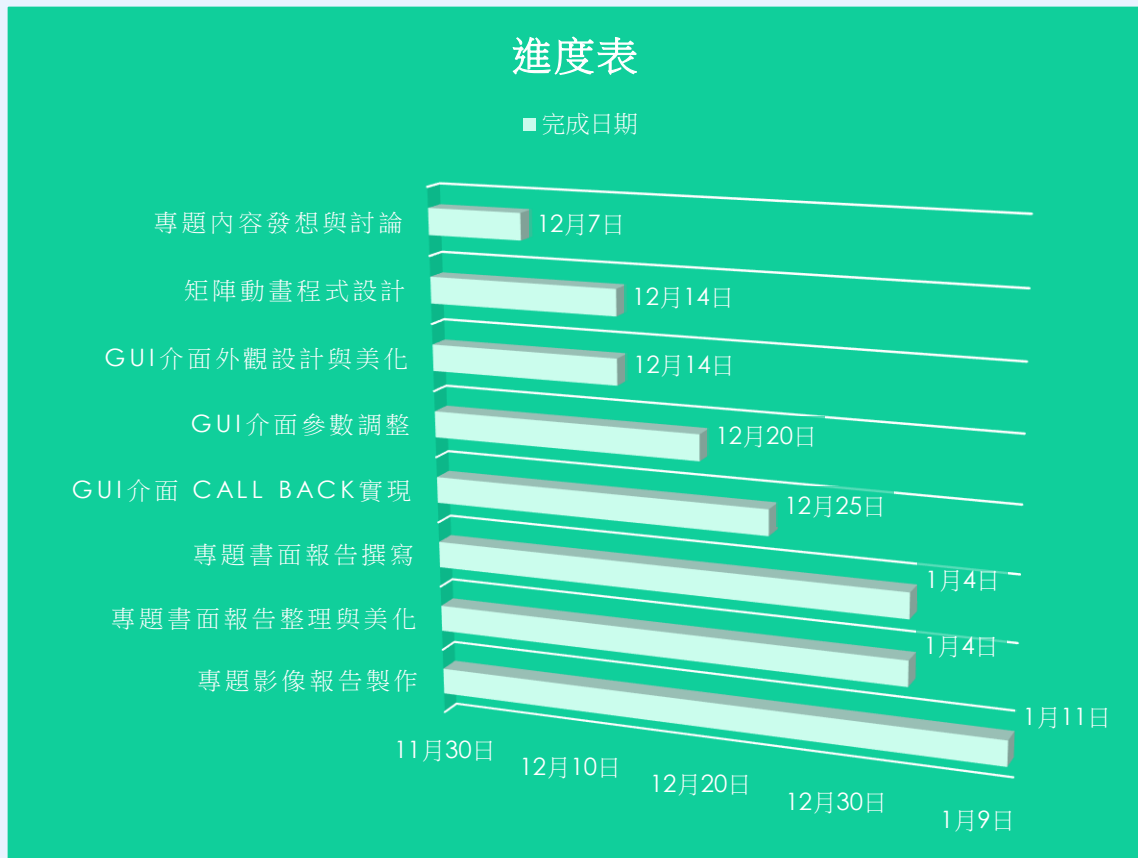


(影片 4-3) 車輛顯示控制

第五章 小組成員工作分配表

人員	工作分配
郭諺勳	矩陣動畫程式設計 GUI 介面 call back 實現 專題影像報告製作 專題書面報告整理與美化
顏宏宇	專題書面報告撰寫 專題書面報告整理與美化 專題書面報告文獻引用
林書雋	GUI 介面外觀設計與美化 GUI 介面參數調整
劉子萱	專題書面報告撰寫 專題內容發想與討論
呂偉宏	GUI 介面外觀設計 專題內容發想與討論
陳昇新	GUI 介面外觀設計 專題內容發想與討論

第六章 進度表 (直條圖)



第七章 計畫配合事項及限制

開始

團隊共同討論並擬定專題所要需研討方向及方法，再者進行所需的工作分配事項，最後進行以上討論之工作進度與分配並將結果以錄影報告完成專題。

過程

各組員依照所分配的工作利用上課時間跟與假日時間進行程式的編寫、GUI 介面的設計與美化與書面資料的撰寫。如遇各個領域無法解決之困難點，則於上課時間團隊聚集之機會相互討論並於網路上找尋解決方法。

結果

團隊全員於本學期結束前完成所分配的工作。專題完成後整體檢查是否有不完整的地方並加以修改補充。組員間詳細的討論是否有能力可及之前提下能、改進的部分，其後進行錄影與報告並於規定日期前上傳繳回給老師。

困難

研究與執行的過程中遇到了不同作業系統之相容性問題、程式部分無法理解亦或使用者介面設計上所遇到的難題，例如不會修改數據等。但在組員之間的互相幫忙下團隊得以順利解決問題，部分組員參與修改與討論，剩餘組員則負責查找相關資料，在難題面前組員們發揮團隊分工的精神以完成這份專題研究。

第八章 結論與心得

當團隊一開始聽到要做專題的時候，因為是首次接觸專題製作，故沒有自信認為無法完成，然而確定主題並著手實際研究後，才發現其實並不難。從最初只會比較基礎的程式碼，慢慢的翻課本學習，並上網查網路資源把團隊的作品變得更好、更加豐富，組員們不但學習到了 MATLAB 的運用，更甚如何活用，也讓團隊熟悉的這個語言的寫法以及操作。

本團隊發現 MATLAB 這個語言與目前所學的 python 與眾不同，部分在 python 寫出的程式在 MATLAB 變得不一樣，諸如起始值為 1 與 0 的差別等的差異不在話下，必需要找更多的資料以吸收更多的知識來解決當下遭遇的各種問題。

本專題所製作的動畫，也讓組員們苦惱，在製作的過程中不僅要用到函式繪圖，也要將程式輸出的圖與 GUI 做結合，要讓動畫中的摩天輪轉動其實不難，但是在寫程式時須深刻理解 matlab 中的矩陣相乘與矩陣轉換之間的關係。另外利用人體視覺暫留的現象使圖像一格一格的形成連貫的動作才會變成團隊現在所看到的小動畫。

透過這次的專題，也凝聚了團隊組員團結的心，組員間的分工也非常重要，如果能將各自的專長放在對的地方去發揮，並彌補彼此的不足，更可以有效率的去增加做事的效率，雖然是一個簡單的動畫，但亦是團隊團隊的心血結晶，大家聚在一起腦力激盪就能創造出一副好的作品，雖然這次的作品並不像市面上那些作品那麼的完美，但當團隊完成他時所得到的成就感也不亞於其他的作品。

第九章 未來展望

團隊希冀在未來之中可以將此作品做更完善、精美，未來有機會的話也想跟有美術方面專長的人才共同合作，由其來彌補團隊美術方面的不足。美術專才來繪畫出團隊動畫裡各式各樣的美術圖，來達到團隊理想中的樣貌，也讓團隊的作品變得如市面上之作品般精緻。

組員們藉此次經驗吸收了更多程式相關知識，期許在未來能做出更多更好也更大的動畫並融合更優異的 GUI 介面予使用者更直觀的體驗。甚至如同上述與美術專才團隊合作，能在 Android 等平台上架簡易上手的小遊戲以紓緩生活中緊繃的情緒。

MATLAB 這個程式還有許多等著團隊去學習，在未來的學習與團隊合作中變得更強大，這就是團隊對於未來不管是這個作品，還是團隊的個人能力，所要邁進的方向。