

作業一：「影像打地鼠」小遊戲

- 作業內容：利用 Matlab AppDesigner 撰寫擁有下列功能的小遊戲。
 - 繳交期限：4 月 10 日(週一) 24:00 前上傳至 Moodle2 系統。
 - 繳交內容：程式碼需詳細註解，程式直接以 HW1_學號命名 (例如: HW1_111xxxxx.mlapp)，以 zip 壓縮上傳。
 - 評分依據：完成度，bug 的多寡(是否有防呆)，程式碼註解詳細程度。
 - 若有問題可 e-mail 詢問色彩所劉光智同學(M11125007@mail.ntust.edu.tw)

「影像打地鼠」遊戲功能如下：

1. 執行程式，跳出選單，選取影像(如圖 1)。可參考附件 [HW1_video_demo.mp4](#) 影片範例。
2. 介面中央有排成 2x2 陣列的 4 個正方形按鈕，每隔 1.5 秒，以隨機的順序一一呈現紅色(如圖 2)。

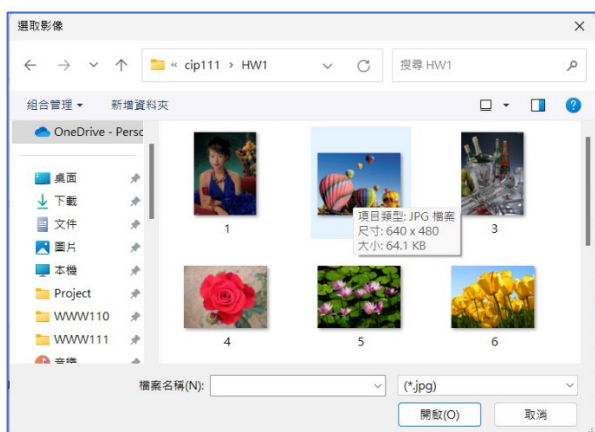


圖 1: 選取影像

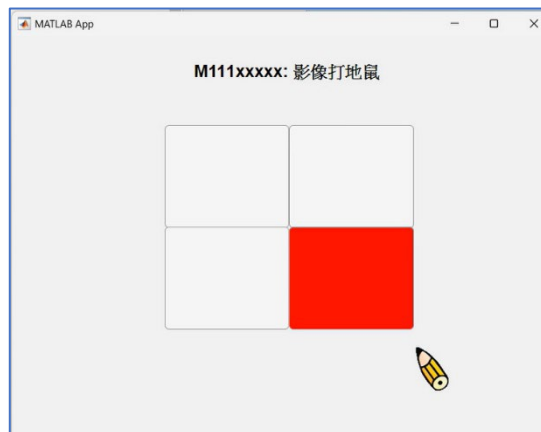


圖 2: 四按鈕一一變紅

3. 玩家的任務是盡快用滑鼠點擊變紅的按鈕。被按下的紅色按鈕，會立刻秀出影像的局部(如圖 3)，並在旁邊顯示按鈕變紅至被玩家按下的時差(單位：秒)。按下還沒變

紅的按鈕，將不會有反應。重複按已經變紅的按鈕，會顯示按鈕變紅至最後一次被玩家按下的時差。



圖 3: 盡快點擊變紅的按鈕

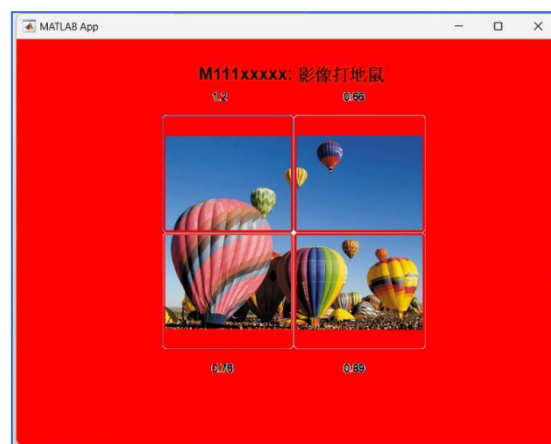


圖 4: 全部點擊後，背景閃爍

4. 四個按鈕的子影像都顯示後，也就是影像完整呈現後，介面的背景閃爍紅色三次 (圖 4)。
5. 上方帶有學號的標題隨即掉落，然後在下方現出四按鈕「變紅到玩家點擊」的時差總和(圖 5)，也就是總成績。玩家手腳如果夠快，可能在 2 秒鐘完成任務。介面在顯示成績後 4 秒鐘，自動關閉。



圖 5: 背景停止閃爍，標題掉落，下方顯示成績(總時差)

程式撰寫內容提示：

1. **Design View:** 建一個空的 Panel，將四個正方形 Button 元件，以 2x2 的陣列位置，密接排列在 Panel 的正中央(圖 6)。元件名稱分別是 Button_1....Button_4。每個按鈕有從屬的 Label 元件，用於顯示按鈕變紅到滑鼠按下之間的反應時間。另外有兩個 Label 元件(圖 6)，上方的用於顯示學號與作業名稱，下方的 Label 元件用於顯示總分。



圖 6: Design View

2. 在 Code View 裡，建立以下 **Private Properties:**

```
seq= randperm(4); % 四按鈕變紅的隨機順序，randperm(4)對數值 1~4 隨機排序
p= zeros(1,4); % 四按鈕變紅與被按下的標記
t_start= zeros(1,4); % 四按鈕變紅的(起始)時間
t_end= zeros(1,4); % 四按鈕被按下的(結束)時間
im; %影像資料
```

3. 在右方 Component Browser 下的 app，按滑鼠右鍵建立 **startupFcn Callback** (初始化回呼函式)，內容包括：

(1) 清除指令視窗：`clc`

(2) 對 `Label.Text` 做初始化(空白化)。

(3) 執行自訂函式 `func1(app)`：其功能是讀取影像，並控制按鈕變紅的順序與時間。

4. `func1(app)`的內容：

4.1 用 `uigetfile()` 讀取影像檔案的名稱與路徑。

4.2 用 `imread()` 讀取該[路徑, 檔名]的影像陣列資料到 `app.im`

4.3 用 `size()` 讀取影像高寬。

4.4 如果影像的高大於寬，四個按鈕的 `HorizontalAlignment` 屬性最好做 `right/left` 調整，使影像左右兩半密接。

4.5 如果影像的寬大於高，四個按鈕的 `VerticalAlignment` 屬性最好做 `bottom/top` 調整，使影像上下兩半密接。

4.6 建立一個跑四次的 `for` 迴圈，以 1.5 秒為間隔(`pause(0.5)`)，按照 `seq` 的順序，選擇 `switch/cases` 裡的四按鈕的 `cases`。

4.7 每個按鈕的處理(`case`)包括：

(1) 用 `BackgroundColor='r'` 將按鈕變紅。

(2) 該按鈕計時開始：用 `app.t_start` 變數與 `tic`

(3) 將該按鈕的狀態設為已變紅：用 `app.p`

5. 四個按鈕要分別建立 **Pushed_Callback function**。內容如下：

5.1 用 `if` 判斷該按鈕是否已經變紅(用 `app.p`)，如果已經變紅，執行下列動作：

5.2 用 `size()` 讀取影像的尺寸。

5.3 將按鈕的 `Icon` 屬性設為所屬的局部影像陣列(1/4 影像)。

5.4 計算 `tic`(變紅)到 `toc`(按下按鈕)的時差：以 `Button_3` 為例，程式為

```
app.t_end(3)= toc(uint64(app.t_start(3)));
```

5.5 將該按鈕的狀態設為完成影像顯示：可令 `app.p` 設定特定數值。

5.6 執行 `func2(app)`。

6. `func2(app)`的內容：

6.1 偵測 `app.p`，如果(`if`)四按鈕都已完成影像顯示，執行以下動作：

6.2 背景紅/白閃爍：用 `for` 迴圈，讓 `UIFigure.Color` 切換紅/白數次，其間要用 `pause(0.1)` 暫停，否則閃爍太快，看不出來。

6.3 學號作業名稱掉落：

(1)讀取該 `Label` 物件的 `Position` 屬性。

(2)用 `for` 迴圈，讓 `Position` 的第二筆資料加上 0 到-700 (間隔-10)移動的數值，`Label` 物件看起來會往下掉出視窗。(tip: 迴圈中要用 `pause` 暫停，才看得到掉落的過程)。

6.4 呈現總分：

(1)計算 `app.t_end` 的總和。

(2)用 `num2str()` 轉成字串。

(3)組合自訂字串，例如：'您一共花了 3.9 秒'。(4)在介面下方顯示字串。

6.5 用 `pause()` 等待 4 秒。

6.6 關閉 `app`: 用 `app.delete`