Chap X 範例

作者：王昱閔

不連續抽籤範例

老王是某間學校的老師，有一天他突發奇想，想要舉辦同樂會以鼓勵學生們這學期的努力。老王打算用抽獎的方式，來決定獎品屬於哪一位學生，但是他突然想到，中間有幾位學生轉學導致號碼變成空號。此時，老王需要一款 app 來輔助他進行抽獎，這款 app 可以讓使用者輸入區間來決定抽取的號碼，並且在使用者輸入數量時，系統會抽取相對應的數量，並且會記錄以往抽取的結果，最後可以匯出至記事本來保留此次的抽獎結果

	需求名稱	需求說明	對應課程
需求 1	輸入區間	供使用者輸入想抽取的範圍	Control TextBox Button
需求 2	清除區間	清除所輸入的區間	TextBox Button
需求 3	進入抽獎畫面	讓使用者進入進行抽獎的畫面	Control Button
需求 4	畫面警示	提醒使用者是否有輸入錯誤	Notifier
需求 5	選擇抽取數量	讓使用者輸入想抽出的人數	TextBox
需求 6	返回到上一頁	可讓使用者回到上一頁，並重新輸入想抽取的區間	Control Button
需求 7	抽籤	進行抽獎動作並且顯示在畫面中	Control Button Label
需求 8	匯出抽獎紀錄	可讓使用者匯出他的抽獎紀錄	File Button
其餘需求	變數	程式中所需要的資料可暫時存放於變數	Variables

用 App Inventor 2 設計出的程式畫面如下

左上圖為系統初始畫面，右上圖為按下的清除按鈕之後的提醒畫面,左下圖與右下圖則為輸入了兩個不同區間的畫面,在按下進入抽獎畫面的按鈕之後，會進入抽獎的畫面

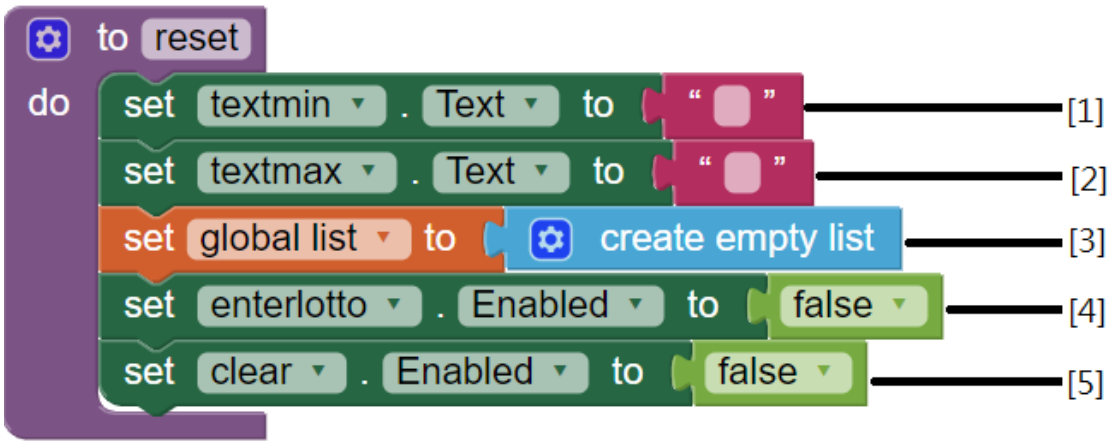
左上圖為抽籤初始畫面，右上圖為輸入抽取數量，左下圖為抽取抽取第一次，右下圖為抽取第二次，當數值為 0 時，會顯示抽籤結束

下圖為抽籤系統的設定畫面與抽籤頁面的欄位名稱

下面為的程式拼塊的講解

Screen1 部分

建立全域變數 list，將 list 設定為一個空白陣列，用於儲存執行迴圈時所產生的數值，以及提供產生亂數時所需要的清單。



建立副程式 reset：當副程式觸發時，會執行以下動作：

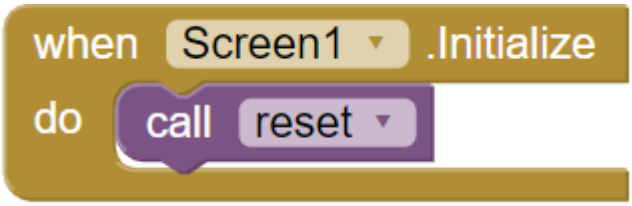
[1] 將文字輸入框 textmin 清空，以便於使用者再次輸入。

[2] 將文字輸入框 textmax 清空，以便於使用者再次輸入。

[3] 將全域變數 list 設定為空的清單，以確保之前所輸入的數值不會殘存，以保證結果的公平性。

[4] 將按鈕 enterlotto 變更為無法點擊的狀態，由於全域變數 list 已經為空值了，所以當系統跳至 Screen2 時，會無法進行抽籤的動作。為此，系統會在全域變數 list 有數值時，才會將按鈕 enterlotto 變更為可點擊狀態。

[5] 將按鈕 clear 變更為無法點擊的狀態，由於系統開啟時，並不會有任何的數值被輸入到全域變數 list 內。因此，在使用者按下按鈕 save 時，才會將按鈕 save 變更為可點擊狀態。

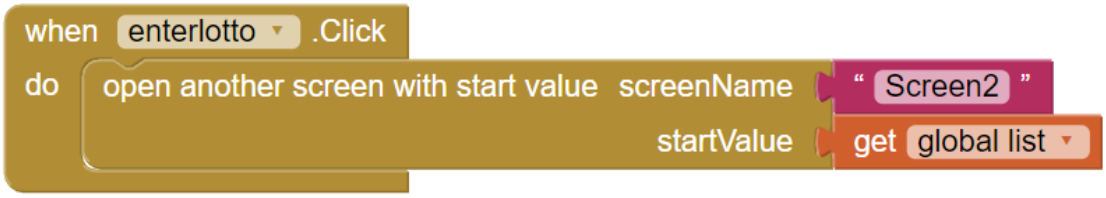


當系統進入 Screen1 時，會觸發 reset 的事件。將系統恢復成初始狀態，以避免前一次或是跳頁之後的值殘存於 Screen1 裡面影響到抽籤系統的公平性。

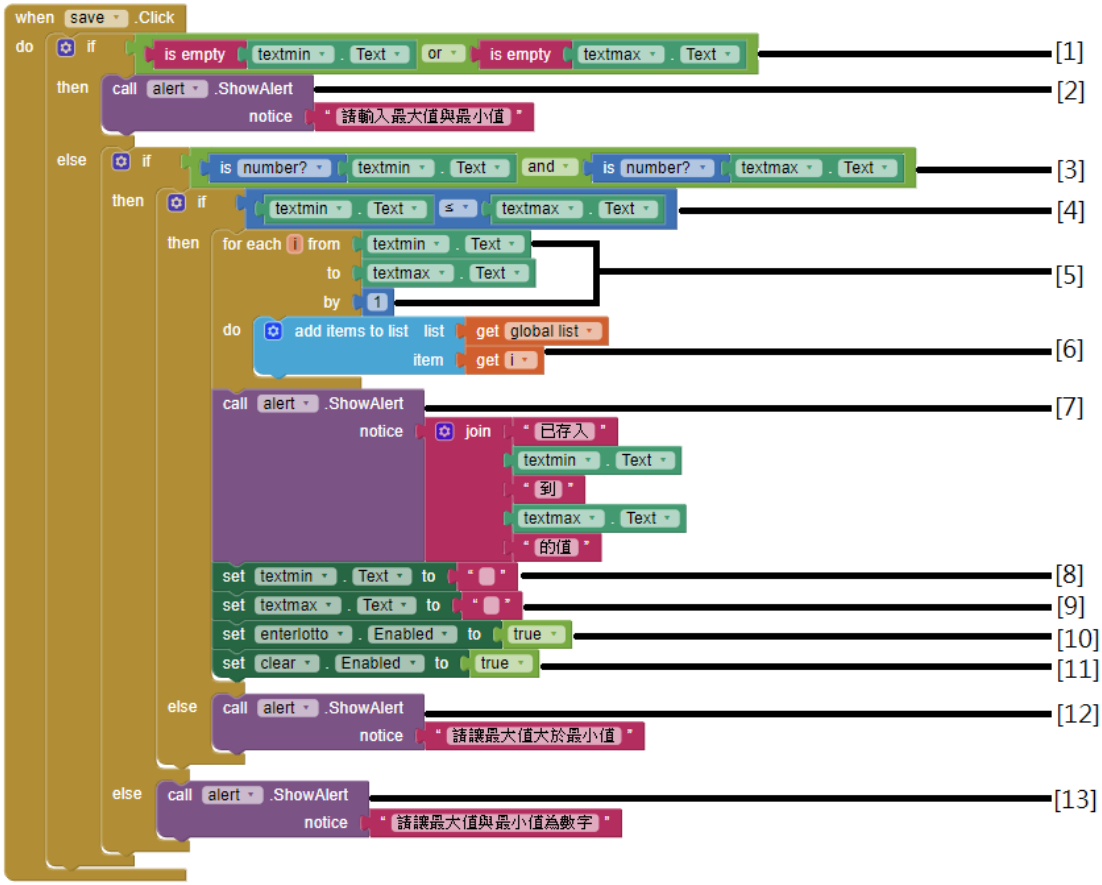
當按鈕 clear 被點擊時，會執行以下動作

[1] 觸發 reset 的事件。將系統恢復成初始狀態。此步驟主要目的為將全域變數 list 清空。

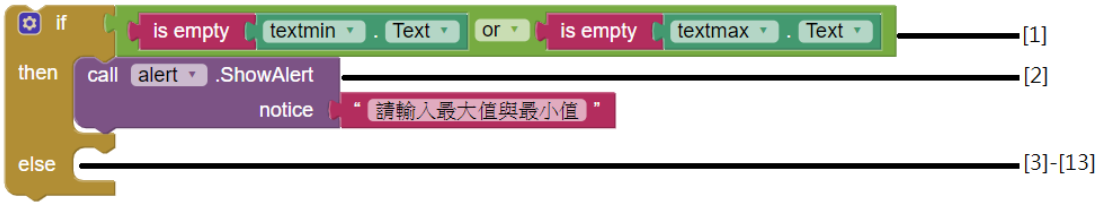
[2] 系統顯示訊息，訊息內容為" 已清除您所設定的數字" ，以便讓使用者了解到系統已經將全域變數 list 清空。



當按鈕 enterlotto 被點擊時，會執行視窗的切換，也就是跳至 Screen2，並將全域變數 list 內的數值，一同傳送至 Screen2。

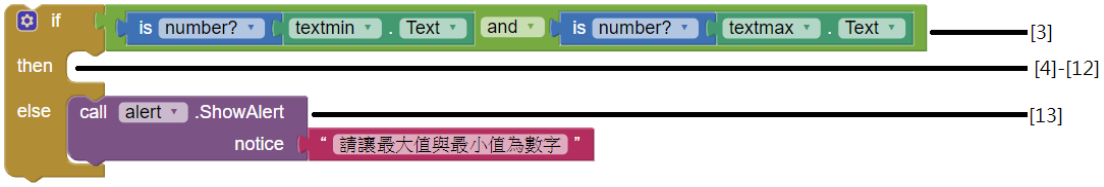


當按鈕 save 被點擊時，會執行以下動作

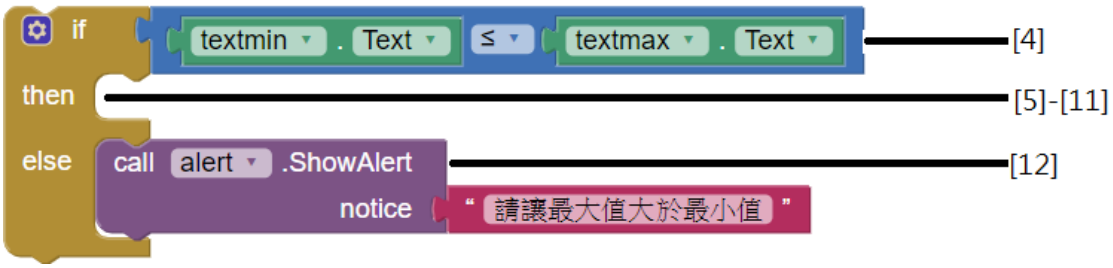


[1] 先判斷文字輸入框 textmin 與文字輸入框 textmax 是否有輸入文字，如果文字輸入框 textmin 或是文字輸入框 textmax 有一個是空白的話，則會執行步驟 [2]，若文字輸入框 textmin 或是文字輸入框 textmax 皆有文字時，會執行步驟 [3] - [13]。

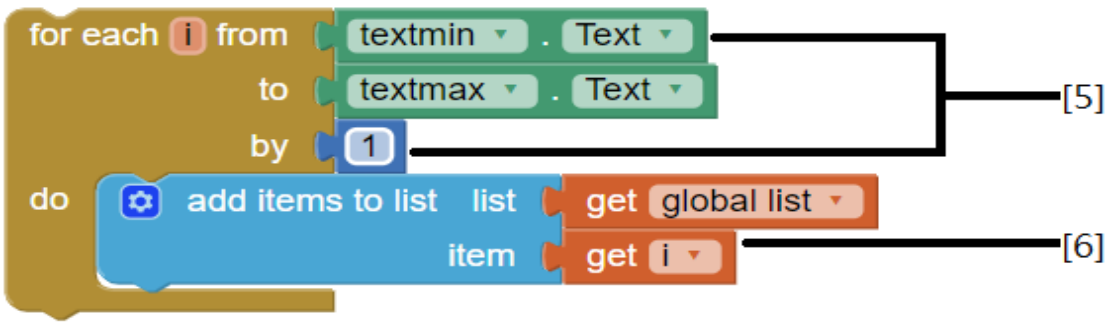
[2] 系統顯示訊息，訊息內容為" 請輸入最大值與最小值" ，以便讓使用者了解到文字輸入框 textmin 與文字輸入框 textmax 皆要有文字的輸入才行。



[3] 再來判斷文字輸入框 textmin 與文字輸入框 textmax 是否皆為數字，如果文字輸入框 textmin 與文字輸入框 textmax 的內容皆為數字時，則執行步驟[4] -[12]，若有文字輸入框 textmin 或文字輸入框 textmax 其中一邊不為數字時，則執行步驟 [13]。

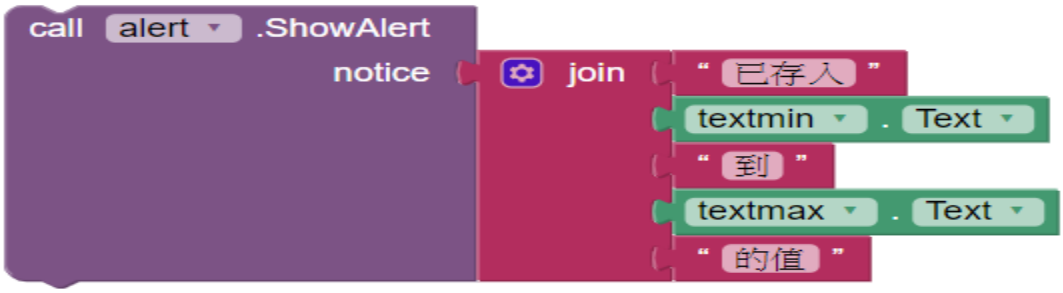


[4] 做最後的判斷，判斷文字輸入框 textmin 的內容是否小於等於文字輸入框 textmax 的內容，如果文字輸入框 textmin 的內容小於等於文字輸入框 textmax 的內容，則執行步驟 [5] - [11]，若文字輸入框 textmin 的內容大於文字輸入框 textmax 的內容，則執行步驟 [12]。

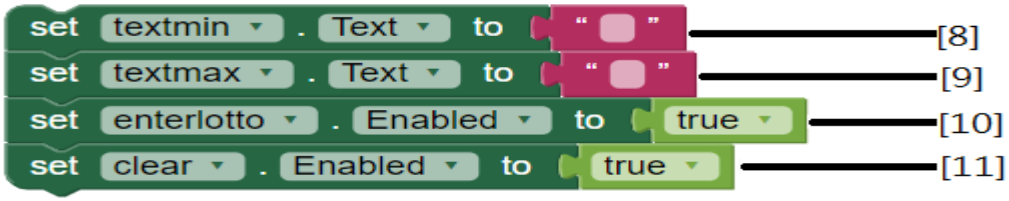


[5] 此步驟為執行迴圈動作，先設定迴圈變數 i,迴圈的起點為文字輸入框 textmin 的數值，也就是 i 的初始數值，之後，以每次累加 1 的方式進行迴圈動作，直到 i 大於迴圈的終點，也就是文字輸入框 textmax 的數值，此時，迴圈才停止動作。

[6] 將步驟 [5] 執行迴圈所產生的數值 i，新增至全域變數 list 中，也就是說全域變數 list 的內容為文字輸入框 textmin 到文字輸入框 textmax 之間的所有數值。



[7] 系統顯示訊息，訊息內容為" 已存入文字輸入框 textmin 到文字輸入框 textmax 的值" ，以便讓使用者了解到系統已經將文字輸入框 textmin 到文字輸入框 textmax 之間的所有數值新增至全域變數 list 中。

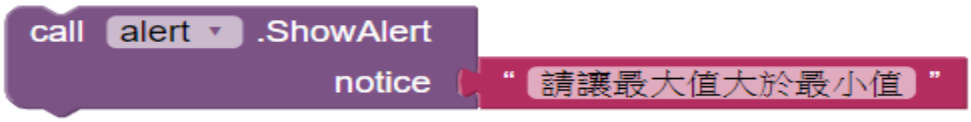


[8] 將文字輸入框 textmin 清空，以便於使用者再次輸入。

[9] 將文字輸入框 textmax 清空，以便於使用者再次輸入。

[10] 讓按鈕 enterlotto 可以被點擊，由於全域變數 list 內有數值存在，因此可以跳至 Screen2 進行抽籤的動作。

[11] 讓按鈕 clear 可以被點擊，由於全域變數 list 內有數值存在，因此可以藉由此按鈕來清空全域變數 list 內的數值。

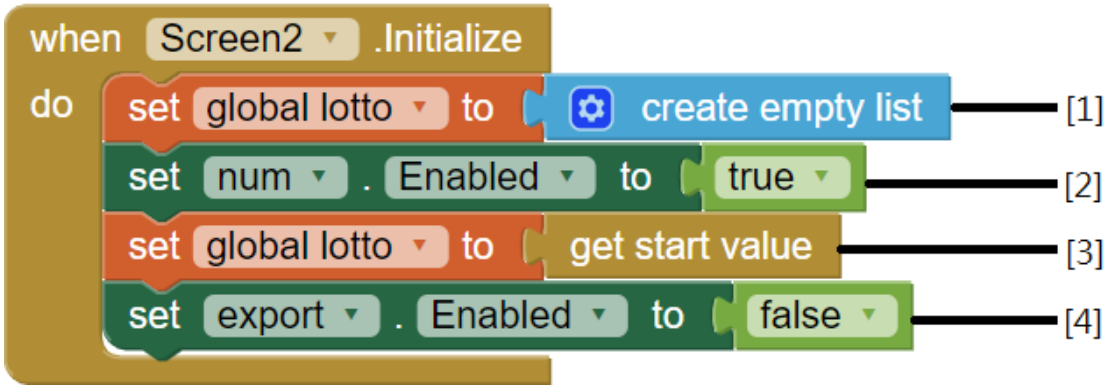


[12] 系統顯示訊息，訊息內容為" 請讓最大值大於最小值" ，以便讓使用者了解到文字輸入框 textmin 的數值必須小於或是等於文字輸入框 textmax 的數值才行。

[13] 系統顯示訊息，訊息內容為" 請讓最大值與最小值為數字" ，以便讓使用者了解到文字輸入框 textmin 的內容與文字輸入框 textmax 的內容必須皆為數字才行。

Screen2 部分

建立全域變數 lotto；將 lotto 設定為一個空白陣列，用於儲存從頁面 Screen1 所傳送的值。



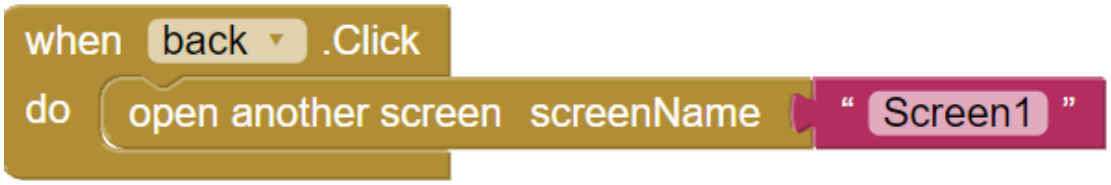
當系統進入頁面 Screen2 時，會優先執行以下動作

[1] 將全域變數 lotto 清空：為避免之前抽籤所剩餘的值殘留，影響到抽籤的公平性，所以系統在進入頁面 Screen2，會將其清空。

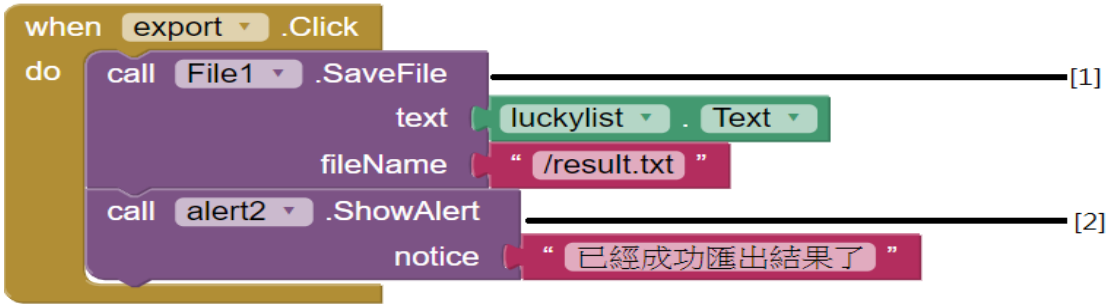
[2] 將文字輸入框 num 變成可以輸入的狀態：由於在進行抽籤的過程中，不能讓使用者一直變更想抽出的人數，因此會將文字輸入框 num 變成無法輸入的狀態，以維持整體的公平性。而在進入頁面 Screen2，會讓文字輸入框 num 可以輸入數值。

[3] 設定全域變數 lotto 的內容：由於系統在頁面 Screen1 的時候，是藉由換頁並傳送值的方式到頁面 Screen2。因此，系統利用全域變數 lotto 來接收從頁面 Screen1 所傳送到頁面 Screen2 的值(EX：頁面 Screen1 傳送 5~9 到頁面 Screen2，則頁面 Screen2 中的全域變數 lotto 的內容則為 5~9)。

[4] 將按鈕 export 設定為無法點擊的狀態：系統會在使用者抽到最後一位幸運兒時，讓按鈕 export 變成可以點擊的狀態。



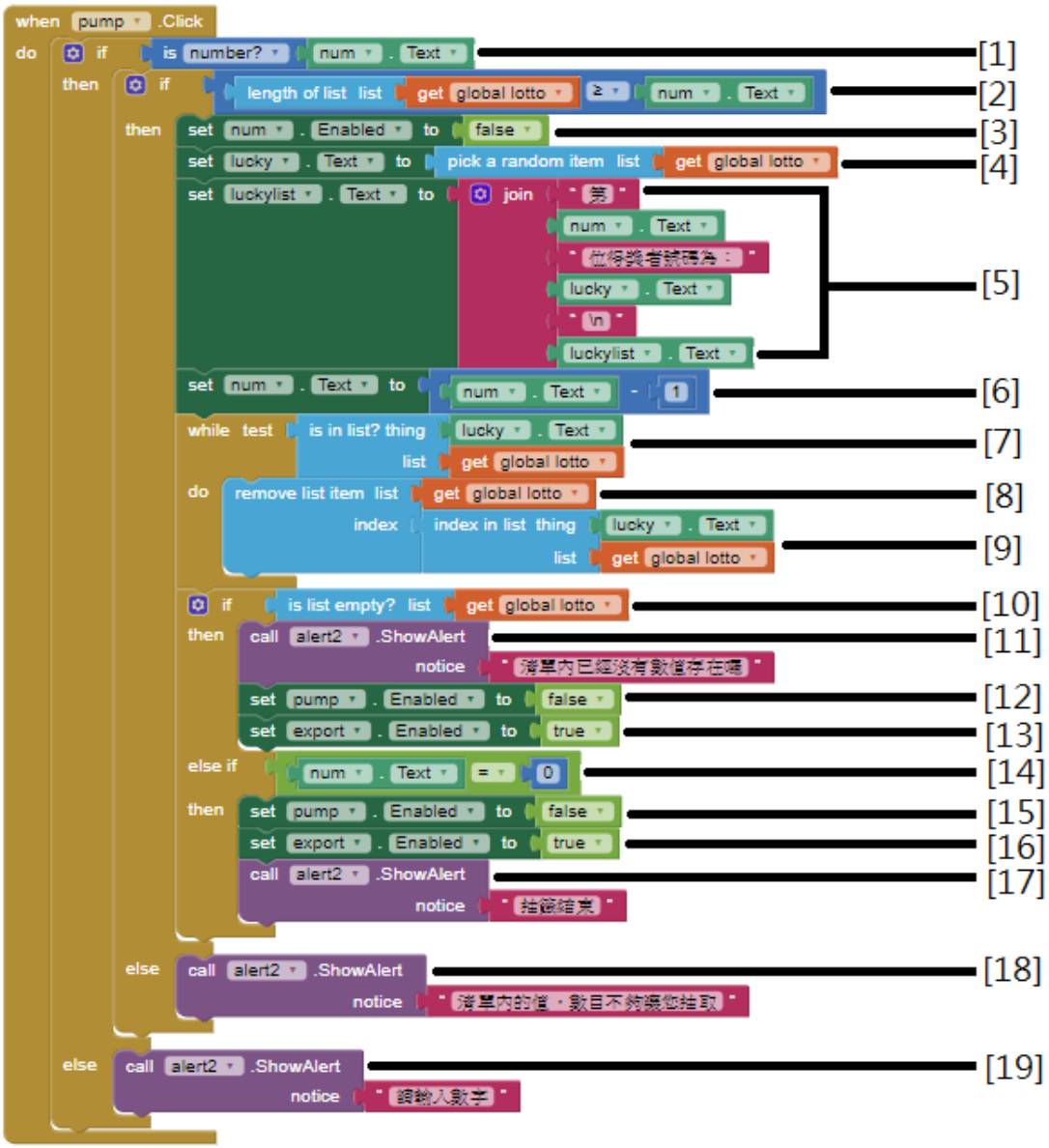
當按下按鈕 back 時，會執行跳頁：系統會跳至頁面 Screen1。



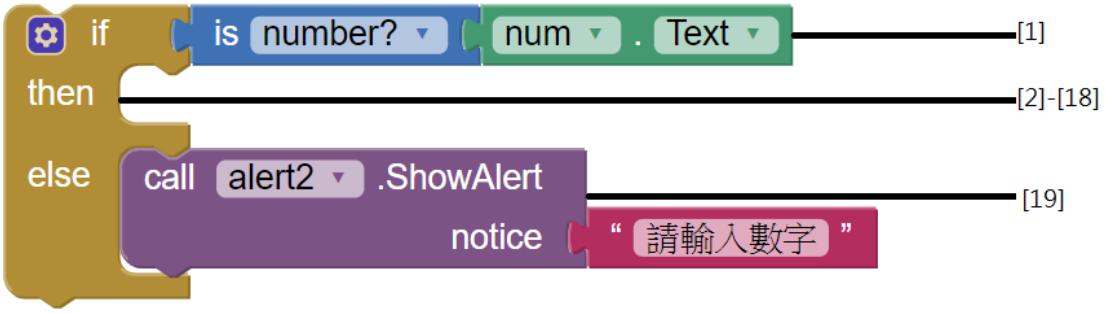
當按下按鈕 Export 時，會執行以下動作

[1] 儲存並產生檔案：系統會將標籤 luckylist 的內容，儲存至記事本 result.txt 中，並將其存放在手機內部記憶空間中，並且不會與之前的資料作合併，可避免掉使用者與之前的紀錄搞混的情況。

[2] 系統顯示訊息，訊息內容為" 已經成功匯出結果了" ，以便讓使用者了解到在標籤 luckylist 的內容已經被匯出至記事本 result.txt 中。

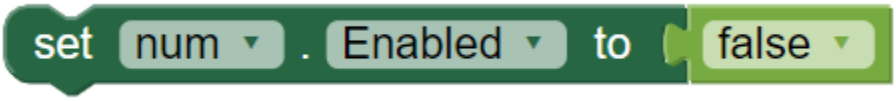


當按下按鈕 pump 時，會執行以下動作



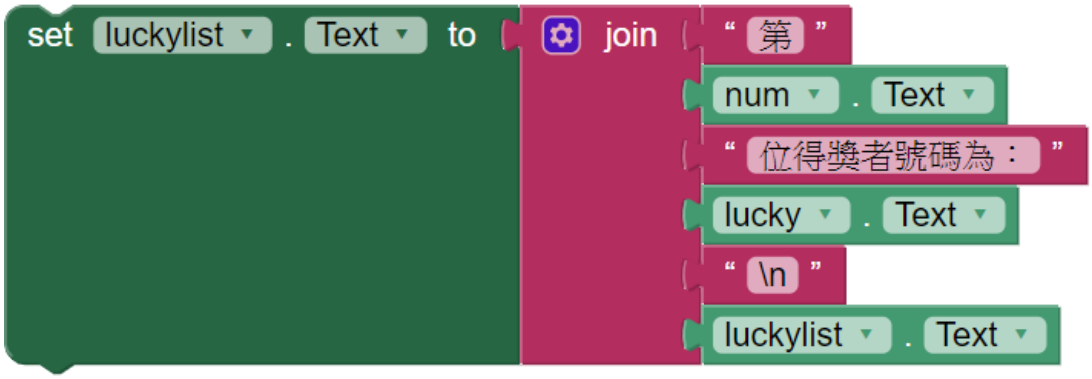
[1]先判斷文字輸入框 num 的內容是否為數字，如果是的話，執行步驟[2]-[18]；如果不是的話，則執行步驟 [19]。

[2]再來判斷全域變數 lotto 的長度(裡面數值的數量)是否足夠進行抽籤的動作，也就是說當全域變數 lotto 的長度大於等於文字輸入框 num 的數值時，表示可以進行抽籤的動作，則會執行步驟 [3] - [17]；若全域變數 lotto 的長度小於文字輸入框 num 的數值，則表示全域變數 lotto 無法供應使用者想要抽取的人數，因此會執行步驟 [18]。



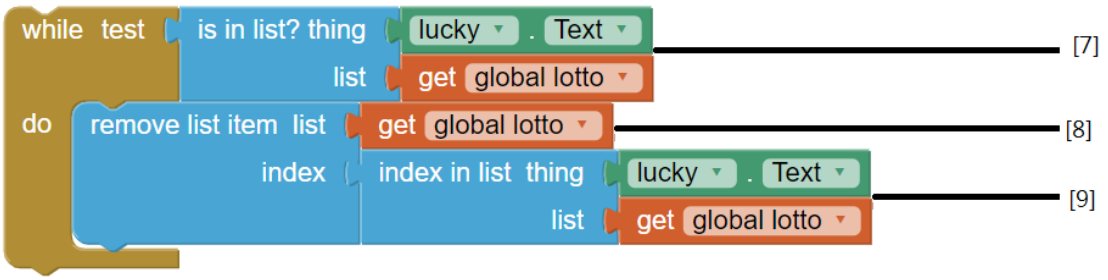
[3] 將文字輸入框 num 設定為無法輸入的狀態：由於進行抽籤的動作時，為保持抽籤結果的公平性，因此會將文字輸入框 num 設定為無法輸入的狀態。

[4] 標籤 lucky 顯示隨機產生的數值：系統會從全域變數 lotto 中，隨機抽取一數值，並將抽出來的數值用標籤 lucky 顯示。



[5] 讓標籤 luckylist 顯示過去所有的被抽出的幸運兒號碼：系統會將每次所抽出的幸運兒號碼以歷史紀錄的方式顯示在標籤 luckylist(EX：第一次抽籤結果為第 6 位得獎者號碼為：7，則下一次進行之後，標籤 luckylist 的內容則會顯示第 5 位得獎者號碼為：8，第 6 位得獎者號碼為：7，以此類推)。

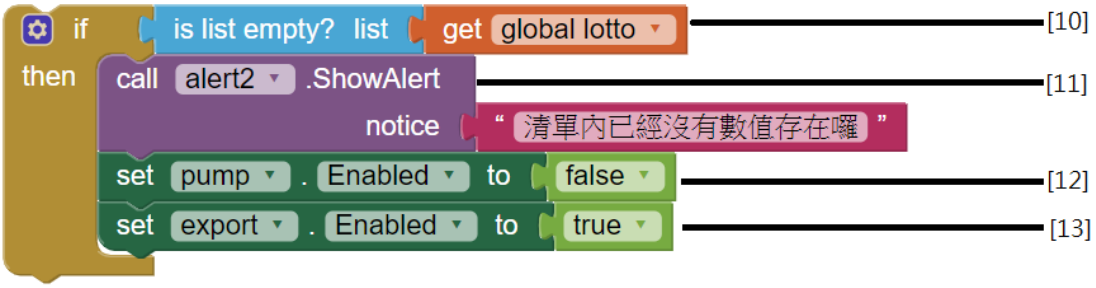
[6] 將文字輸入框 num 減去 1：藉由這一步驟，來讓使用者了解到下一次要抽出的得獎者編號。



[7] 執行無限迴圈：當系統在全域變數 lotto 中找尋到標籤 lucky 的數值時，會進行步驟 [8] - [9]，直到系統無法在全域變數 lotto 中找尋到標籤 lucky 的數值為止。

[8] 從全域變數 lotto 移除在步驟 [9] 所找到的編號。

[9] 在全域變數 lotto 中找尋到標籤 lucky 的數值所在的編號，並將他回傳至步驟 [8] 移除。



[10]當系統判斷到全域變數 lotto 中已經沒有任何數值可以進行抽籤動作的時候，系統將會執行步驟 [11] - [13]。

[11] 系統顯示訊息，訊息內容為" 清單內已經沒有數值存在囉" ，讓使用者了解到全域變數 lotto 中已經沒有任何數值可以讓使用者進行抽籤的動作。

[12]將按鈕 pump 設定為無法點擊的狀態，避免使用者繼續進行抽籤的動作，造成系統的錯誤。

[13]將按鈕 export 設定為可以點擊的狀態，讓使用者可以匯出在標籤 luckylist 上所顯示的歷史紀錄。

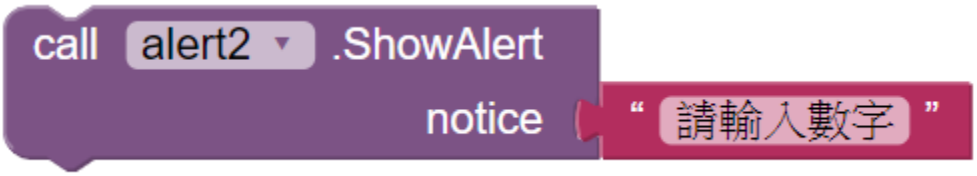
[14] 或是系統判斷到文字輸入框 num 的數值為 0 時，代表整個抽籤過程已經結束了，則系統會執行步驟 [15] - [17]。

[15]將按鈕 pump 設定為無法點擊的狀態，避免使用者繼續進行抽籤的動作，造成系統的錯誤。

[16]將按鈕 export 設定為可以點擊的狀態，讓使用者可以匯出在標籤 luckylist 上所顯示的歷史紀錄。

[17] 系統顯示訊息，訊息內容為”抽籤結束”，讓使用者了解到整個抽籤過程已經結束了，也就是已經抽完使用者想要抽取的數量了。

[18] 系統顯示訊息，訊息內容為”清單內的值，數目不夠讓您抽取”，讓使用者了解他在頁面 Screen1 所輸入的數值無法供應使用者在頁面 Screen2 想要抽取的數量，因此會讓使用者重新填寫想抽取的數量。



[19] 系統顯示訊息，訊息內容為”請輸入數字”， 讓使用者了解文字輸入框 num 的內容必須為數字才行。