

UI 開發注意事項

一、命名規則

1. 會被程式呼叫的 widget 名稱必須為 Unique
2. Layer : (全小寫 : layername)Layer
Ex: mainLayer
3. 產生 layer code : layer_(全小寫 : layername).c
Ex: layer_main.c
4. Widget : (全小寫 : layername)+(開頭大寫 : 功用)+(Widget)
Ex: mainVolumeIcon
5. Export Function : (widget 名稱開頭大寫)+(動作)
Ex: MainVolumeButtonOnPress

二、提升效能

1. 使用 ituTextSetString 在 text/textbox 產生字串會經過 render 的動作，如此會拖慢效能，盡量以預先產生圖片的方式來規避
2. 由於 ituSceneFindWidget 會降低效能，因此只要在開頭做一次即可
Ex: if(!mainVolumeIcon){ Do ituSceneFindWidget() }
3. 若用到數學計算，盡量用查表的方式對應
Ex: sin cos 把數值先寫成表，運算時用查表的方式去做運算
4. 避免在畫 UI 的線程(以下統稱 main 線程)裡做太多事
Ex: 讀寫外部訊號，或是 open file 開檔讀寫等，這些動作都會對畫 UI 造成影響
處理方式：移到另外的線程處理，線程裡必須有適當的 sleep，讓系統有機會做 content switch 在不同線程做切換。
5. itu 的 APIs 只能在畫 UI 的線程(main 線程)裡呼叫，不能在其他線程中呼叫，會造成死機
處理方式：如果在其他線程裡有呼叫 itu APIs 的需要，可利用 event 或 queue 的方式通知 main 線程來做。這部份可參考 scene.c 中有關 key 的處理方式
6. UI layer 裡的 OnTimer function，其 return 值是有意義的，當 return true 時會觸發重新畫 UI 的動作，如果太頻繁就會影響 UI 操作的流暢度
處理方式：只有在有需要重畫 UI 的時候，才 return true，否則一律 return false

7. 在 UI 畫面 Frame Rate 可設 60 FPS，但如果進入影像播放畫面，必須降為 30 FPS
8. 中斷 ISR 裡如有印訊息的需求，不能使用 printf，必須改用 ithPrintf，ithPrintf 才是 reentrant function
9. 檢查 nor cache 與 font cache 是否開足夠大，一般建議值為 nor cache: 0x200000，font cache: 0xFA000

三、撰寫程式小技巧

1. 在 layer code 中盡量不要有其他 layer 的 widget 或 function，保持乾淨
2. Layer 的 AlwaysVisible 為 true 並且 Visible 為 true 則不會有 OnEnter 事件，若需要 ituSceneFindWidget 可在 OnEnter logoLayer 時做
3. 使用 ituTextSetString 產生中文並且在 win32 模擬，在 code 開頭加入 `#pragma execution_character_set("utf-8")` 才可顯示出正確字串

四、GUIDesigner 小技巧

1. 字型統一用 文泉驛等寬微米黑
2. Icon 要有 transform 的效果，就不能設 stretch 屬性為 true
3. Widget 有載入圖片，可在點擊 System.Drawing.Bitmap 後鍵入 Delete 清除
4. 名稱修改時，不要用左右鍵調整游標，只能用 Home、End 和滑鼠