|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 研發專案紀錄 | | | | Approved  by | Checked  by | | Reported  by |
|  |  | | 江明韋 |
|  | | | | | | | |
| P/N | C++ | TOPIC | Hello MFC | | Issued Date | 20150609 | |
| 1. MFC算是比較元老級的產品了，但是現今仍有許多公司持續使用著，原因是因為他曾是過去大家所大量使用的，只是C#誕生後大家就漸漸的棄暗投明了@@?? 2. MFC是微軟所推出，它可以很快地建構出一個視窗介面(相較於win32)，但相較於C#卻又慢的許多，但是MFC是一套非常具代表性的一套C++程式，不僅嚴格遵從C++繼承的風格，且與Windows的作業系統運作原理息息相關，因此在寫MFC的同時也會參照到很多Windows的作業系統。 3. 高階語言雖然學起來快且簡單，但其對底層的距離也就越遠，越容易使學習者感受不到或是難以意會底層程式的運作。 4. MFC聖經之一(深入淺出MFC)   <file:///C:/Users/alona/Desktop/Wayne/C++/Information/dissecting%20MFC%202e%20part1.pdf>   1. MFC在早期就已經提出MVC架構，這是非常了不起的架構，至今android也是遵從這個架構在執行，只是android更加優化這個架構把結構分得更清楚，各大有經驗的程式設計師也都認同並推從這樣的架構。 2. 簡易的MVC解釋: 把軟體系統分為三個基本部分：模型（Model）、檢視（View）和控制器（Controller）     （控制器 Controller）- 負責轉發請求，對請求進行處理。  （檢視 View） - 介面設計人員進行圖形介面設計  （模型 Model） - 程式設計師編寫程式應有的功能（實作演算法等等）、資料庫專家進行資料管理和資料庫 設計(可以實作具體的功能)。  每個人對MVC的解釋與使用定義皆不盡相同，有人說M「存儲資料」V「外觀」C「邏  輯判斷」，但我比較編好M:演算處理，V外觀，C邏輯判斷。  在我們公司的產品EMY中，就有導入MVC的概念XD  M:等待訊息進入並處理對應動作(如:運轉/停馬達、進入待機或休眠程式)  V:顯示燈號  C:按鈕被押後發送事件給M、或是sensor被觸發後發送事件給M，M接到事件後  就動作完全不管為什麼要動，使要C控制得好就好。  所以通常客戶回饋BUG後我們會直接思考MVC哪個地方錯了，然後針對MVC的  哪個地方去修改。   1. 要了解為何MFC嚴格遵從這個原則請先新建MFC專案: (開始>>新建專案)        1. 此時便可看到MFC建立出四個.cpp檔案，就是要程式設計師依照MVC架構去設計程式   CView>>V(View)這邊輸入視覺效果上的呈現  Capp>>C(Control)邏輯控制與判斷  CDoc>> M(Model)檔案管理與儲存  CMainFrame>>對於視窗上的呈現與功能架構(Word樣式或是一般frame樣式就是在這邊產生)   1. 完成新建專案後就會看到在專案底下產生了      1. 此時案Run 剛剛建專案時的設定就都跑出來在視窗上了   如果想拿掉部分功能或是增加，可在新建的時候在進階功能上做勾選 | | | | | | | |

金雨企業股份有限公司 管理編號: