|  |  |
| --- | --- |
| 技術 | 說明 |
| 檔案讀寫 | 以fstream讀取、寫入檔案。 |
| 分割字串 | 透過stringstream，以空白將字串切割，並能得到切割後的子字串。 |
| 字串轉換數字 | 透過stringstream讀取字串，並將其輸出給整數型態，從而得到數字。 |
| 點與矩形碰撞 | 透過點的座標與矩形的座標，判斷點是否在矩形中。  將此函數擴展，可以得到矩形與矩形碰撞(矩形的四個點，任一點在目標矩形中，即與目標矩形碰撞)。 |
| 圖層 | 將圖片/動畫/動作等賦予圖層，並交由圖層管理器(layerManager)統一顯示。  其概念為：將圖片/動畫/動作放入對應圖層(index)的陣列裡面的vector，而圖層管理器在輸出的時候，從陣列index為0開始輸出圖片，直到陣列index為最大圖層數時停止，如此一來，圖層叫小的圖片/動畫/動作會先被呼叫顯示，而圖層大的則後呼叫顯示。  為了方便將圖片/動畫/動作加入圖層管理器，圖層管理器設置為static，令其能在任何地方被呼叫。 |
| 對話 | 對話能讀取文本中的字串，並由對話管理器(CDialogManager)統一管理所有的對話。  其概念為：對話主體(CDialog)能夠以fstream讀寫檔案，配合字串分析，取得對話文本中「誰說了什麼話」，而對話管理器CDialogManager統一管理所有的對話，並提供介面使管理的對話能執行開始對話(Start)、結束對話(Stop)、下一句(Next)、顯示對話(showText)。  其中開始對話以指標取得指定的CDialog，並進行初始化設定，結束對話則將指標還為nullptr。顯示對話為主要部份，會取得指定的CDialog的字串，並將其以最大字數(bytes數，ASCII體系，中文2bytes，英文1bytes)切割為個別的子字串，再將切割完的子字串交由CDC繪出。  對話同時也具有逐字顯示的效果，是利用計時器，每隔0.1秒改變指定字串的長度來達成。  為了方便讓對話能在任何地方被觸發，因此對話管理器設成static。 |
| 結局 | 結局利用圖片+對話的方式呈現，同樣由結局管理器(CEndManager)統一管理所有結局。  其概念為：一個結局具有多組圖+多組對話，一張圖片會搭配一個對話，當圖片播放完時代表結局結束。結局管理器(CEndManager)提供介面使管理的結局能執行開始結局(Start)、結束結局(Stop)，同時透過fstream讀取與寫入「擁有哪些結局」，使得擁有的結局不會因為關閉程式而遺失。  結局中使用的圖片帶有淡入淡出的透明度效果，其透明度以MFC自帶的CBitmap以及CDC內的alphaBlend函式達成。 |
| 透明圖片 | 使用MFC自帶的CBitmap以及CDC內的alphaBlend函式達成，CBitmap雖然本身並不帶透明度，但可以將其傳給alphaBlend，並在alphaBlend額外賦予透明度，將CBitmap以帶透明的效果顯示。 |
| 資料夾操作 | 能夠開啟指定的資料夾，並讀取資料夾內所有的檔名，或是取得資料夾內的檔案總數目。  資料夾操作藉由dirent.h達成，dirent.h為unix系統下，C/C++的library，但很不幸的Windows並不支援，因此我們使用他人改寫能相容於Windows系統的dirent.h。  程式中透過while迴圈以及readdir()操作，讀取指定資料夾內所有的檔名，利用此方法，我們成功自動化讀取音效音樂、對話文本、對話頭像、結局等等需要大量Load指令的檔案，使用上更為人性。  Github：<https://github.com/tronkko/dirent> |
| 滑鼠滾輪事件 | afxwin中已經有定義滑鼠滾輪事件onMouseWheel()了，因此只需在CGame中新增virtual function：OnMouseWheel，即可在CGameStateInit、CGameStateRun、CGameStateOver中使用滑鼠滾輪事件，另外，滾輪事件中傳遞的short參數zDelta為判斷滾輪是上滾或是下滾，zDelta > 0代表滾輪上滾，zDelta < 0代表滾輪下滾。 |
| 複製檔案 | MFC使用原生函數CopyFile  CopyFile(原檔案路徑, 目的地路徑, 是否不覆蓋檔案) |
| 刪除檔案 | MFC使用原生函數DeleteFile  DeleteFile(檔案路徑) |
| 工具列 |  |