問題一：學期初執行程式時，程式偶爾會跳出存取被拒，導致不能執行。

解決方法：經過反覆測試後發現問題出在我的電腦有一個遊戲有防外掛系統，因此我刪除該遊戲永絕後患，問題就解決了。

問題二：MFC中很多原生函式都只接受char\*型態，但寫C++程式時常以string為主。

解決方法：自行做一函式將string轉為char\*：先用new關鍵字配置一些空間給char\*，在將string的內容複製給char\*並回傳，要注意的是，因為有new，所以新創建的char\*在使用完後必須delete，否則會memory leaks。

問題三：在做地圖的卷軸效果的時候，原本預期達到的效果是人物走→走到一半換成地圖捲動→地圖捲到底時，在換人物走，但有時候會發生人物走到一半地圖卻沒有捲動。

解決方法：原本的判斷是計算人物的座標是否達到開始捲動的值(達到時地圖捲動)，以及判斷地圖是否捲到底(捲到底則地圖停止捲動，人物開始行走)，後來新增Camera，在人物行走的時候修正Camera的值，並透過Camera的值判斷地圖是否該開始捲動。

問題四：地圖捲動的時候，地圖上的物件沒有跟著捲動。

解決方法：將物件的座標分成兩種：實際上的座標、物件在螢幕上的座標，物件在自主移動的時候更改自己實際上的座標，而物件在螢幕上的座標，則透過實際座標減去Camera的值求得。

問題五：每次要更改物件顯示的優先權很麻煩，(例如物件A要顯示在物件B上面，就要在OnShow()將物件A的程式碼放在物件B下面) 。

解決方法：製作layerManager負責管理圖層，該Manager為vector<圖片>陣列，只需將個別物件的圖片放入對應陣列裡的vector，再將此vector陣列由小到大輸出即可，舉個例子：設定物件A的圖層為1，物件B為2，如此將物件A放入vector<圖片>[1]中，物件B放入vector<圖片>[2]中，而輸出的時候從vector[0]輸出到vector[9]，這樣一來，圖層小的圖片自然會被圖層大的圖片蓋住。

問題六：有時候需要計時，時間到執行某個動作。

解決方法：將老師原本的時間計算統合整理成一類別：CTimer，CTimer會計算出要「跑過多少個執行緒」，使用時透過其函式CountDown()將計數-1，同時使用IsTimeOut()判斷是否已經計時完成，若計時完成，則執行自定義的程式，同時透過ResetTime()回復Timer的計數。

問題七：每次碰撞都要取出點座標進行判斷，不僅麻煩也容易出錯。

解決方法：在製作的過程中，我們發現MFC自帶CRect，用來表示矩形的位置，因此我們製作函式IsPointInRect(CPoint, CRect)，傳入點座標與矩形，用來判斷「該點是否在目標矩形內」。然後擴展成另一個函式IsRectCollision(CRect,CRect)，判斷「矩形是否與矩形碰撞」，其概念為：將矩形A的四個座標取出來，透過IsPointInRect判斷，如果四個點中有一個在目標矩形中，那麼兩矩形就形成碰撞。

該矩形的好處是，由於CMovingBitmap內本來就有表示位置的矩形，因此只要將該矩形取出，就可以簡單的判斷兩個圖片(或物件)是否碰撞。如果要更改碰撞範圍，也只要修正自己的矩形就好。

問題八：有時候執行程式時，會馬上彈出CDDraw的某個東西為nullptr。

解決方法：發生該問題時，大多數原因是出自在default constructor中Load圖片，解決方式很簡單，將Load圖片從default constructor移出，另外做個LoadImg的函式，並在遊戲的OnInit()呼叫LoadImg來載入圖片。

問題九：CGameStateInit的OnInit跟OnBeginState()是反的，因此在OnInit()載入圖片，在OnBeginState()進行初始化的時候，會因為「圖片還沒被載入」而使程式崩潰。

解決方法：逆著爬回去發現，能力所及解決不了，因此我們架空OnInit()，將所有程式碼移到OnBeginState()，並使用布林變數判斷「哪些部分只須要做一次」(像是載入圖片、音效，只需做一次)。

問題十：製作對話的時候，發現如果將所有字串都硬設在程式裡，要修改時相當不容易。

解決方法：透過fstream開啟指定的txt檔，並將其內容讀給自定義的類別CDialog，同時透過CDialogManager管理所有的CDialog，要開啟對話的時候透過CDialogManager開啟指定的CDialog即可。

問題十一：很多情況下都會觸發到對話，如果每個要觸發對話的物件都必須擁有CDialogManager的指標來控制，在寫程式上會有些不方便(因為要把指標傳來傳去，常常傳到最後不知道自己在幹嘛) 。

解決方法：參考前人所作之CAudio，將CDialogManager，static化，令其能在所有地方都能被呼叫。

問題十二：CDialog對話所讀取的字串如果太長，在顯示的時候會超過遊戲畫面。

解決方式：限制每一行最大字數(bytes數)，超過該最大字數就換行，注意這裡的換行並不是字串加上\n，而是「換一個char\*」給CDC輸出，如此才能確保位置正確。

問題十三：承問題十二，CDialog對話在換行時，由於中文和英文所使用空間不同(ASCII下中文2bytes，英文1byte)，在切割字串時若正好切到中文，則輸出時會顯示亂碼。

解決方法：該問題原因來自於，因為中文要兩個bytes在一起才能正常顯示，如果正好將中文切成2bytes，第一行的末尾拿到第一個byte，第二行開頭拿到第二個byte，則該中文的編碼則沒有連續，因此產生亂碼。解決方法是判斷「如果拿了這個中文字，會不會導致這行的總bytes數」，如果會超標，則把該中文字放到下一行顯示。

問題十四：