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。

該函式的好處是，由於CMovingBitmap內本來就有表示位置的矩形，因此只要將矩形取出，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，也只要修正矩形的範圍就好。

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：CDialog對話在進行的時候，該如何讓文字逐字顯示？

解決方法：將CDialog讀取到的字串切成長度為k的子字串，而k的值隨著時間逐漸增加，而在在輸出文字的時候選擇子字串輸出。

問題十五：地圖上的小怪物如何重生？

解決方法：當子彈與小怪物碰撞時，將刪除小怪物並將其從vector中移除，與此同時在vector中new一個新的小怪物，並賦予新的小怪物一亂數時間，當時間到的時候才讓小怪物顯示在地圖上，如此就有一種重生的效果。

問題十六：CAudio對使用何種音效的定義寫在mygame.cpp，若要在mygame.cpp以外的地方使用要把該定義傳來傳去很不方便。

解決方法：改寫CAudio，將使用到的音效以map的方式儲存起來，並利用多型新增Play(string)，令CAuido可以使用字串當map的key來使用音效。

問題十七：CAudio每次多一條音效時都要多寫一行載入程式來載入音效，挺麻煩的。

解決方式：使用dirent.h取得指定資料夾中的所有檔名，如此一來就可以一次載入所有音效，每當新增一條音效的時候只要將音效放進指定的資料夾即可自動載入。

dirent.h為unix系統下，C/C++自帶的library，能夠操作資料夾，但windows的library並不支援，因此本程式使用他人改寫自可相容windows的dirent.h。

github：<https://github.com/tronkko/dirent>

問題十八：CDialog對話每次新增新的文本(txt)或是人物頭像時都要多寫一行程式來載入文本或圖片，很不方便。

解決方法：呈問題十七之解，透過資料夾操作取得指定資料夾的所有檔名，一次性載入所有文本、人物頭像，如此一來只要將檔案放進指定的資料夾就可以使用。

問題十九：製作完Boss以及NPC之後，已經確定所有new出來的空間都有正確delete掉，但還是莫名其妙多出很多memory leaks。

解決方法：因為Boss以及NPC是virtual class，底下有個別的子類別繼承他們倆，而製作的時候忘記virtual class的deconstructor也要變成virtual function，否則只會解構父類別，而不會解構到子類別。

問題二十：CMovingBitmap只能讀bmp，但又想要讓結局的圖片有透明效果

解決方法：CMovingBitmap在讀取的那一瞬間就已經決定好圖片的透明色、底色，即使套用CDC的alphaBlend，也只會得到背景是黑色的透明色，因此在做有透明效果的圖片時採用MFC原本的CBitmap，並搭配CDC的alphaBlend轉換透明度，缺點是硬體設備不夠好的電腦執行速度可能會有點慢。

問題二十一：想要在關掉程式後依然能夠存取已經擁有的結局。

解決方法：在離開遊戲的時候將「目前擁有的結局」透過fstream寫入txt檔儲存。

問題二十二：能在擁有結局的頁面中透過點擊的方式看結局的劇情。

解決方法：擁有結局的頁面中放入自行製作的CButton，令其具有點擊事件，並在點擊的時候判斷是否擁有該結局，如果有，則透過CEndManager播放對應的結局劇情。

問題二十三：  
兩個.h互相引入對方，例如想在role所在的CEraser中，引入投擲物CBall，又想在CBall中取用role目前的位置，就需要交互引入兩個標頭檔。

解決方法：根據上述例子，交互引入對方後，在宣告class CBall之前使用先前置宣告class CEraser即可。

問題二十四：string如何以空白切割字串？

解決方法：透過C++的stringstream字串符流，將字串以<<運算子添加到stringstream，全部添加完後，再用>>運算子添加回vector<string>，最後的vector<string>即是分割完後的字串vector。

問題二十五：做地圖編輯器的時候，想利用工具列做出新增物件/載入地圖等等功能，但工具列實作在gameView.cpp中，跟mygame.cpp似乎沒有交集，因此在mygame.cpp取得不到透過工具列得到的檔名。

解決方法：以新增物件為例，當透過工具列新增物件的時候，將新增的物件複製到指定的資料夾中，同時在該資料夾以fstream寫下txt，其內容為「新增的物件種類」以及「該物件的檔名」，兩字串以空白分割。

新增完txt後，在mygame.cpp中檢查「是否有新增的txt」，如果有，則分析該txt的內容，並將其新增的物件添加到地圖編輯器中，同時刪掉新增的txt。

問題二十六：地圖編輯器中新增的物件想要進行拖曳以及刪除操作。

解決方式：在地圖編輯器中，當滑鼠點擊螢幕時，抓取「點擊到的物件」，如果按住左鍵並移動，則讓抓取的物件隨著滑鼠座標移動，如果按下delete鍵，則將物件從刪除。需要注意的是，抓取的物件移動時設定的位置不能是滑鼠的位置，應該要設成滑鼠位置減去「點擊瞬間，滑鼠位置與選擇物件的座標差」。

問題二十七：新增/編輯好的地圖要存檔。

解決方式：將地圖編輯器中新增/編輯好的地圖，其地圖上的物件轉換成遊戲中使用的地圖CBlockMap，並使用CBlockMap內的函式新增/覆寫地圖資訊。

問題二十八：編輯好地圖後，希望能直接測試，不用關掉程式讓地圖重新載入。

解決方法：此問題來自於原本載入地圖發生在開啟程式的瞬間，因此如果在地圖編輯器修改完地圖，重新進入遊戲時是不會重新載入地圖的。而如果改成每次進入遊戲都重新載入地圖，又感覺略沒效率，因此我們採用：在存檔地圖的同時，紀錄「被修改過的地圖編號」，在進入遊戲的時候，只重新創建被修改過的地圖，替換掉原本的地圖。