組員名單：施順耀 林奕廷 李昊澤

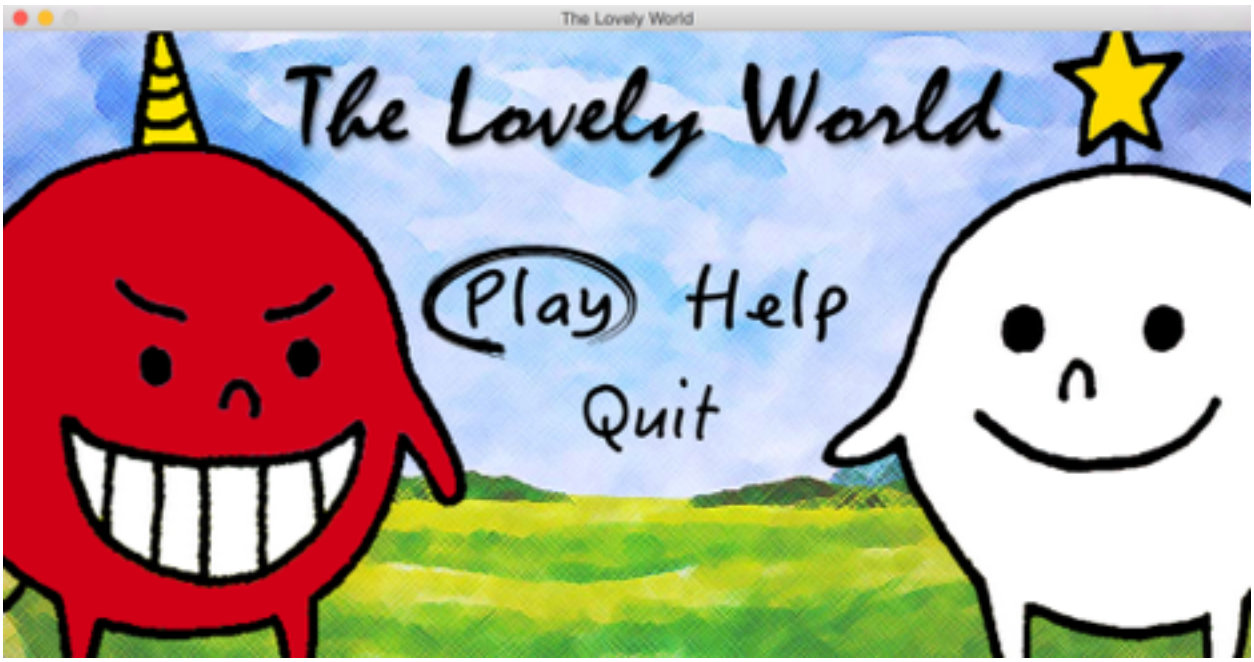
Third-party函式庫

1. SDL2
功能：開啟圖形介面、將圖片繪製在介面上
2.SDL2_image
功能：載入非.bmp檔的圖片（此專題皆使用png）
3.SDL2_mixer
功能：載入音樂

操作與使用

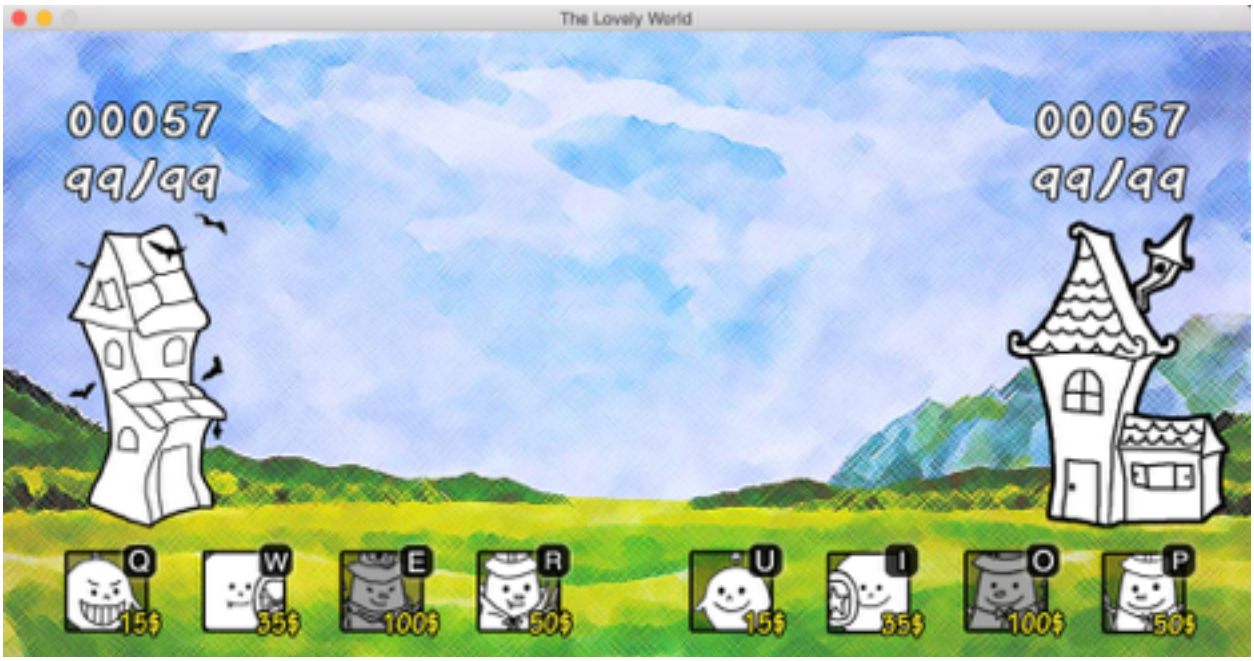
輸入法請調成英文

遊戲進入畫面：使用方向鍵選取，按enter鍵進入。

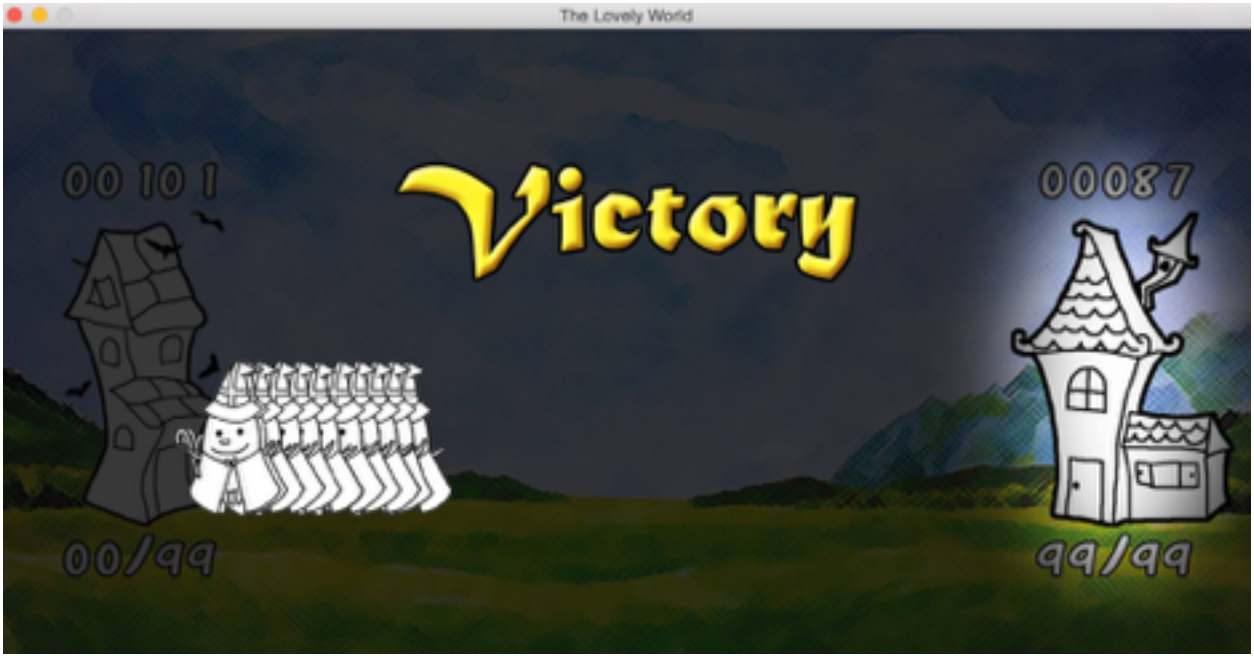


首先介紹help畫面：進入help後一共包含五個頁面，使用左右鍵切換，其中包含遊戲簡介和怪物特色、數值、介紹，並且使用enter鍵退出，回到主頁面。

接著進入play畫面：左（暗黑）右（光明）兩方在遊戲設定中需要利用資源（城堡最上方數字）召喚怪物入侵敵方陣營，每隻怪物對城堡有不同數值的傷害，一旦將敵方的城堡攻陷（城堡上方第二個數字歸零）便能獲得勝利。



最後介紹play畫面時怪物的召喚方式：如介面下方所示，一共有四種怪物，
左方陣營使用q、w、e、r召喚，右方陣營使用u、i、o、p召喚，



並且每隻怪物可以進行兩次的升級，左方陣營相對應的按鍵為a、s、d、f，右方則為h、j、k、l。
當資源足夠召喚怪物時，怪物將會亮起，反之維持暗色。
而遊戲進行中時，可按m鍵暫停音樂。

怪物升級後會有不同的外觀，升級一次後，怪物將會出現半透明的顏色，甚至有外型上的改變，同時，升級後的建築物外觀也會在相對應位置產生顏色。

屋頂：q、u

窗戶和門：w、i

其餘牆面：e、o

有門的牆面：r、p

再升級第二次後，怪物的著色和建築物的塗色將轉為純色，因此，玩家可由怪物的樣貌或建築物外觀分辨怪物的等級。

勝利後可以按下enter關閉程式或是直接點擊左上角紅點關閉程式。

遊戲設計

這個類似塔防的遊戲，主要的設計在於，各個不同功能的生物從被召喚到死亡或到達目的地間的各式各樣的判斷，因生物的大小、功用的不同，在彼此碰撞時，也要處理各種的狀況。

因應這樣的需求，程式裡面的架構主要是在整個大迴圈中，每一次都呼叫各式各樣的Print函數，而在每種Print函數裡增加參數，使得遊戲能在每一個下個時刻，印出下一個動作，而在各個生物方面，由其ChooseStage去判斷淡入、淡出、移動、死亡、攻擊、攻擊動畫，函數裡也都有對應的變數去控制透明度、不同的動畫。

而在控制各個生物的部分，我們採取一次宣告一定數量的生物陣列，我們僅儲存一次圖片，之後利用指標讓剩下的同種生物指到同一區塊，以此來避免我們使用這方法可能導致佔據過多記憶體的問題，之後，我們以兩個參數（x_f, x_c, x=生物種類）紀錄在場上的第一隻和最後一隻去得知在場上的是哪些生物，好讓Print函數正常運作，當使用者按下對應的召喚按鍵時，就會Push（x_c+1）一隻對應的生物，到存在在場上的生物的Queue，而死亡，或到達終點時，做完對應的動畫（淡出）或數值紀錄後，就Pop(x_f-1）。而在這兩個參數x_f, x_c做加減的原因在於，當使用者召喚了一隻新的生物，那必定是那種生物在場上的最後一隻，所以x_c+1，而由於我們程式在設計上，都會打某一種類最前排的那隻生物，所以不論是到達終點，或者死亡，我們都可以確定那是發生在最早被召喚出來的那一隻，所以x_f-1。

在碰撞函數的設計上，最基本的是判斷雙方的橫向座標，當有重疊時則將兩個物件的座標調整到剛接觸的時候，判斷碰撞到的東西的生命值(hp)大於0，目的在於確認碰撞到的東西還是活著的，而透明度是否=255，目的在於判斷碰撞到的物件是否還在召喚中（淡入），同時，即便上述條件都達成時，也要確保目前的物件的座標不會超出原本召喚時的橫向座標，而在魔法師或牧師的情形下，則要多考慮一個攻擊或補血距離的參數（M_Range or H_Range）。

魔法師和牧師的碰撞函數上，還多了一個設定攻擊或補血動畫的縱向以及橫向的座標，針對攻擊或補血的目標，產生相對應的攻擊或補血動畫。

牧師的碰撞判斷上又較魔法師複雜了一點，因為他必須同時判斷友方以及敵方的碰撞，在我們的設計中，我們讓他在自己損血時，先對自己補血，而當範圍內的友方目標血量小於其最大血量時，對那個友方補血，而在對敵隊的判斷時，就是一般的碰撞判斷。相對於魔法師的攻擊動畫，牧師的補血動畫多了一點設計，我們讓他在一次完整的補血動畫後，才會在動畫上進行目標的轉換。

事實上，我們也有在圖層順序上稍作琢磨，例如較為大型的生物應該適合放在比較後面，以免擋住其他生物，而先召喚出來的生物適合放在前面，這麼一來當它死亡時，玩家便能看到死亡的動畫，攻擊動畫（法師的法術）也有稍作設計，我們讓它能直接在攻擊目標的前面一個圖層，如此一來，攻擊動畫的擬真度又會更加提升，結算畫面的部分，也可以看出其中有圖層順序的變化，此外，我們也在出現結算畫面時，雙方維持動畫停止損血，來達到有更好效果的勝利畫面。

另外，我們的生物有升級的設定，這部分可以從它本身的或建築物的變色看出來，而這方面的設定，我們讓它和建築物連動，這樣設計的意義在於，即便使用者還沒召喚出那一類的生物，他仍然可以從建築物的變色得知有沒有升級成功。

綜合上述幾點所述，再加上我們不管升級或召喚都需要消耗的資源，可以發現這些東西彼此都有相關，因此我把他們全部加在同一個class（potatoqueue）裡

面，以便放一些變數或函數，控制彼此的關係。

此外，我們還寫了兩個基本函數，運用SDL2裏已經寫好的函數和物件，達成較直觀且方便載入圖片的方式，而在開始畫面、Help畫面，所運用到的方法基本上都是基本的載入畫面以及畫面切換。

在外觀的感官享受部分，可以看出我們設計了相當多的動畫，幾乎每一個生物的階段都是動畫，甚至我們還加入了音樂。

開發過程

在遊戲設計部分，最困難的部分在於碰撞判定及其事後影響的設計，這不僅要考慮都碰撞兩者間的各種狀態下產生的影響，還要有正確的結果，甚至我們必須不斷測試或思考，究竟對於這個遊戲來說，哪一種碰撞效果是較為正常、合理，甚至是有趣的，這部分尤其是在魔法師、牧師上更為明顯，我們專題開發後期，多數都在這個函數中做調整，再慢慢的配合修正後的函數，修改其他部分。

碰撞方面，在過程中我們卡住最久的是，事實上雙方的碰撞函數並不是完全同型的，原因在於在potatoqueue::Print函數中，先印出來的都是右邊那方，而同時也先進行碰撞的判定，因此有可能發生在同一個迴圈，右邊那方因為判定沒有碰撞後走進碰撞範圍內，導致接下來左邊那方判定碰撞，而先展開攻擊，這麼一來，對於遊戲效果方面，就會發生明明雙方召喚了一樣的生物，但卻有可能發生某一方先死亡的情況，因此在右邊那方的碰撞函數上，判定碰撞範圍多了一次的移動距離，如此一來，便能解決原本的問題。

圖片部分，我們採取全自製的方式，所有的圖片都是先經過手繪畫出原型，再利用電腦畫出動畫，或者做一些細部的效果調整，再畫出動畫的部分，總是要經過不斷的測試，必須同時兼顧可行性以及效果，僅僅在畫某一種生物的某一階段的動畫，往往花費了3~4個小時，而全部做下來，至少也花費了數十個小時。

音樂部分，我們也是自行創作、自行錄音，我們有兩首音樂，皆是純鋼琴的抒情音樂，這部分也是花了我們很多時間，從創作、調整、練習，到最後真正錄出來，而光是最後的練習和錄音，就花了兩天，再加上前面的構思，真的花了不少時間，可惜的是我們對專業的音樂軟體不熟，沒有後製過，錄音設備也很簡陋，使用手機錄音，這部分有點可惜。

雖然圖片和音樂可能在這個專題的評分上沒什麼幫助，或甚至完全沒有評分，但我想以一個遊戲的角度來說，有完整的音樂和圖片，對於整個感官效果也是必要且很有助益的。

工作分配

1.程式碼撰寫

主要架構設計與撰寫：施順耀

生物物件的基本設定：林奕廷

基本函數的設計：李昊澤

2.圖片
設計&繪製：林奕廷

Help文案撰寫：李昊澤

3.音樂

創作&錄音：施順耀

4.遊戲平衡與效果調整

遊戲測試：施順耀 林奕廷 李昊澤

參數調整：李昊澤

生物彼此的對應關係以及攻擊設定：施順耀 林奕廷

程式Debug：施順耀

版本轉換(mac->windows）：李昊澤