

# 105-1 計算機程式(EE1004) 謝宏昀班

## Final Report

組別：第三組

專題名稱：iGroc

組員：簡廷翰、李矜毓、趙冠豪

### 一、程式中所有類別 ( Class ) 之間關係及功能說明：

#### 1. main

創造首頁，進入首頁迴圈。

#### 2. Core

constructor 初始化所有基本介面：windows、renderer、event。

mainloop 為首頁迴圈，一開始先秀一段入場動畫，再判斷所有滑鼠動。

close 為關閉 SDL 的函式。

#### 3. Genie

constructor 初始化所有基本介面：windows、renderer、event，還有比較不一樣的精靈動畫 ( moodPicture ) 初始化。

mainloop 為精靈頁面迴圈。

RollingEyes 為滑鼠放上精靈頁面時的功能——眼睛看著滑鼠方向。

CountDown 為倒數計時器的函式。

TrueClock 為指針時鐘的函式。

Timer 為計時器的函式。

RandomMove 為精靈視窗隨機移動的函式。

Drag 為拖曳精靈視窗的函式。

Animation 為精靈動畫的函式。

close 為關閉 SDL 的函式。

#### 4. Texture

constructor 有三種，一是空的建構子，在宣告類似東西時搭配 Load 用 for 迴圈宣告比較方便；二是普通的建構子，輸入包含圖片名稱跟顯示器，

三是給 Button 用的建構子；輸入包含圖片名稱及顯示器外，還有圖片大小。

在 destructor 中，我們將 SDL\_Texture 給 free 掉。

Initialize 為後來加的初始化函式，搭配上空的建構子初始化精靈動畫圖片，如此一來可以將讓這些圖片在一開始就被初始化好。

Load 為製作 Texture 的函式。

Show 為將圖片丟上顯示器的函式，有三種 Show 的 overloading，一是普通的，可以決定要在螢幕哪個位置、長寬秀出來；二是添加了選轉功能的，可以決定轉軸及轉動角度三是為了 Button 設計的；可以再使用者觸發按鈕時給適當的回饋。

setAlpha 為設定透明質的函式，做動畫時很好用。

setRECT 為設定圖片位置的函式。

setClip 為設定 Button 及 Animation 的函式。

## 5. Button

Button 繼承了 Texture 的所有 member，在初始化時跟普通圖片比較不同，需要做切 state 的動作，所以需要圖片長寬。然後多了一個 handleevent 的函式，用來偵測滑鼠動作。

這樣的程式編排，當初是想採垂直式的呼叫：一開始叫出首頁頁面，而關於此頁面的動作都在 mainloop() 裡面執行，像是做動畫、點擊選單等等。當選擇精靈後，會 new 一個 Genie 出來，進入精靈介面，而程式碼就進入了精靈的 mainloop() 裡面，並關掉首頁介面。

但後來其他組員覺得其實不必要這樣，應該將 mainloop() 裡的東西搬出來 main，這樣在撰寫程式時比較有一個大體的容貌，不用一直切換 class 這樣，會比較直覺一點。這是我最初在設計時就犯的錯，後來因為時間問題就沒更改了。

## 6. Time

2017/1/16 更新前

constructor 初始化時間 ( 00:00:00 )。

SetHour、SetMinute、SetSecond 分別設定時、分、秒。

ResetTime 重置時間 ( 00:00:00 )。

GetTime 以 00:00:00 的格式回傳時間。

GetTimeDigit 根據輸入值回傳時、分或秒的十位數或個位數。

MinusTime 減去時間。

PlusTime 加上時間。

2017/1/16 更新後

將改為 GetTime 改為 TimeZero ( 回傳時間是否為 00:00:00 )。

注記：原本會寫成 GetTime 那樣是因為當初原本是用 SDL\_ttf 將時間輸出到介面上，但後來我們設計自己的數字圖片，於是更新的時候便決定把 ttf 時代的影子改掉。

## 二、關於 Time 的封裝說明 ( 可參見 Time.hpp )

由於計時及倒數計時都會用到時間的儲存，因此便將此部分提出來成為一個 class。又 ttf 的舊時代已不復存在，所以此處僅就更新過後的 time 做說明。

```
class iTime{
public:
    iTime();
    ~iTime();
    bool SetTimeDigit(int,int);
    void ResetTime();
    int GetTimeDigit(int);
    bool TimeZero();
    void MinusTime(long temp);
    void PlusTime(long temp);

private:
    int hour;
    int minute;
    int second;
};
```

時、分、秒為時間儲存中最基本的元素，但此處將它們設為 private，要設置或者抓取時間皆要透過 public 中的函式 ( SetTimeDigit、GetTimeDigit ) 來進行。為什麼會這樣設計呢？這是因為我們是使用自己設計的數字圖片，因此設定及抓取時間的時候是一位一位進行的。為了方便 Genie 中 countdown 及 timer 的使用，便將 time 設計成如此。

另外，TimeZero 和 MinusTime 是供 countdown 使用，PlusTime 則是供 timer 使用，因為我認為這是屬於 time 的相關功能，如果以後要擴充功能也可能使用

到，因此便放在這個 class 裡面，而不在 countdown 和 timer 中再行處理。

### 三、Texture 的建構子和解構子（可參見 Texture.hpp）

當初為我們重寫 tutorial 的程式 LTexture 的原因，是我覺得他廢話太多而且功能不齊全，所以我們在一開始花比較多時間在認識 SDL 這東西，不過寫完 Texture 後在寫主程式碼時就很方便了，也讓主程式碼顯得非常簡潔。在宣告 iTexture 時只需要給圖片名稱，完整路徑在建構時會幫你加好。而 Animation 及 Button 的初始化則是需要加入分鏡大小提供 setClip 來切圖片，也讓我們在宣告 Animation 及 Button 時非常方便。如下：

```
iTexture Test("Test", rR);           //宣告一個叫Test的圖片，名字為"Test"  
iButton Test("Test", rR, 50, 50);    //宣告一個叫Test的按鈕，名字為"Test"
```

至於解構子，我們將 SDL\_Texture 給 free 掉，以免造成記憶體流失的現象。

### 四、Button 的繼承與 Texture 的 member function 實作（可參見 Button.hpp）

因為 Button 跟 Texture 基本上相同，只是多了 state 跟 handleevent 而已，所以就讓 Button 繼承了 Texture 的所有成員，讓程式碼精簡不少。而在 Texture 內我們打包了很多 SDL 基本的函式來實作 function，使我們在初始化及做動畫時非常輕而易舉，這邊舉 show(SDL\_Renderer\*, int, int, int, int, double, int, int) 為例：第一個參數為顯示器，接著的 4 個 int 為圖片位置，double 為要旋轉的角度，最後兩個 int 為旋轉中心，如此一來，若想要做旋轉的動畫，便只要隨時改變 double 的值即可，非常輕鬆。如下為秒針旋轉的做法（將現在時間的秒乘上 6 度，即為秒針應該待的角度）：

```
second_hand.Show(rR_TrueClock, 294, 58, 12, 250, (6*Date_information_INIT_number[2]), 8, 242);
```

※程式碼可參見 Geine.cpp 中的 TrueClock()

### 五、特別的函式補充

Genie 的造型視窗，是我們在一開始花了不少時間研究出來的，他是在 SDL2.0.5 版本中的一項新功能，可以借圖片指定色碼來切去不要的視窗，這也讓我們移動視窗時增添了不少泡泡的漂浮感。至於特別的程式設計與實作技巧我們很像沒有了 QQ

## 六、程式分工

簡廷翰：介面設計、Texture(class)、Button(class)、Core(class)、Genie(class)、RollingEyes 程式部分撰寫

李羚毓：報告版面配置、Time(class)、CountDown、Timer、RandomMove、Animation 程式部分撰寫

趙冠豪：圖片繪製、美工設計、TrueClock 程式部分撰寫、簡報及影片製作

## 七、心得

從一開始決定專題的主題就花了一些時間，因為我們不想和大部分組別一樣做遊戲，想做一些會讓人印象深刻的東西。一開始的選項有台大附近的美食抽抽樂、定時提醒的 app 等，後來才決定要做桌面上的功能小精靈。我們覺得這個主題似乎不會很難做，一開始想了很多很酷炫的功能，也想做很多隻不同的精靈、主題，但當我們開始做的時候才發現似乎不是這麼一回事。我們第一個遇到的問題是視窗的透明。我們把 render 的圖背景用透明，但是出來的會是視窗終端機的顏色，所以就會有黑黑醜醜的部分，後來我們問了助教、教授，自己也研究了很久，才在一個日本的網頁找到了解決的方法，可以把視窗的一部份截掉。又因為我們的精靈有一些動作，所以裁的形狀不能是精靈的形狀，所以我們又把精靈畫進了泡泡裡面，才終於解決了視窗不是方方正正的問題。我們的 project 跟其他組的大方向差蠻多的，我們的偏向介面設計，要做出使用者用起來覺得舒服的畫面，所以我們花很多心思在於畫面的排版、圖形的設計，功能的分配等等。我們做的桌面小精靈需要很多層的視窗，藉由不同 button 連接不同的畫面。此外我們的圖形都是親自繪製的，希望做出很有特色的專題。

## 八、更新

更新檔網址同之前繳交，放在「20170116 更新」的資料夾中。

網址：<https://drive.google.com/open?id=0BwMQWg5qExYGU1BGVnNxVmZCQkk>