

# The Secret

OOP期末專題

0610780 楊哲睿

0610764 陳芳瑩

0610840 劉文

# 動機與目標

The Secret，顧名思義，每個人的一生中，總有些秘密只想跟某些跟你比較親近的人分享。換句話說，在團體裡，你只想和某個特定族群以你們底下約定好的方式溝通而不讓其他人知道實際內容。

然而在網路社群媒體建立的群組中，你的一則訊息可是會讓所有在群體裡的人都看得一清二楚。比方說在體育校隊，你想要找大家一起翹課出去玩，可是，除了有隊員之外，還有教練和經理在隊伍裡。

為了不讓他們知道你的秘密行動，你便和隊員們約定好某些暗號或密碼，以示行動的細節。但是當你傳了一串密碼在群組時，教練他們也看的到你這串奇怪且毫無意義的亂碼，雖然他不會知道實際內容是什麼，但也會對你這奇怪的舉動產生懷疑。

藉此，三個都身為體育校隊的我們，決定設計一款加密程式，可以使上述之任務順利成功，同時又不讓教練起疑。

## 計畫

### 一、程式功能

- 可以在程式登入時確認身份，讓使用者僅可解密跟自己相同身分者所加密的圖片。
- 能用一段文字作為鑰匙加密你真正想講的事情
- 再把加密過後的內容隱藏在使用者使用的圖片中，之後由圖片接收者得到圖片及鑰匙，便可解密圖片得到原始內容。

## 二、預計要完成的工作

- 製作GUI圖形使用者介面
- 設計使用者類別的繼承關係、屬性與方法
- 建立資料庫儲存使用者資料
- 設計加解密文字的演算法
- 設計隱寫術 ( Steganography ) 的演算法，可以把內容藏到圖片中以及把內容從圖片中找出來

## 三、使用到的工具

- 使用Qt開發GUI程式

Qt是一個跨平台的C++應用程式開發框架 ( Application Framework )，廣泛用於開發GUI程式。



Qt是自由且開放原始碼的軟體，使用Qt開發的軟體，相同的程式碼可以在任何支援的平台上 ( 包含最常使用的Windows、Mac、Linux等，手機應用程式的開發也行 ) 編譯與執行，而不需要修改原始碼。

所以如果我們想要達成這種秘密通訊功能的軟體，讓不同作業系統的大家都能使用便是一件重要的事情，使用Qt可以幫助我們完成這件事。

- 使用SQLite資料庫儲存使用者的資料

SQLite是一種關聯式資料庫管理系統，實現了大多數SQL標準。在Qt裡面有很方便的函式庫讓我們去連結SQLite的資料庫。



- 使用OpenCV處理有關影像的事情

OpenCV的是一個跨平台的電腦視覺庫 ( Computer Vision Library )，由Intel發起並參與開發，自由軟體，可以在商業和研究領域中免費使用。



OpenCV可用於解決如人臉識別、動作識別、運動追蹤、物體識別、圖像分割等的問題。我們用OpenCV來處理隱寫術的功能。

- 使用git做版本控制

git是一個分散式版本控制軟體，也是自由軟體。我們透過git確保大家都在一樣的進度上，避免程式碼版本混亂的問題。



## 實做

### 一、開發環境設置

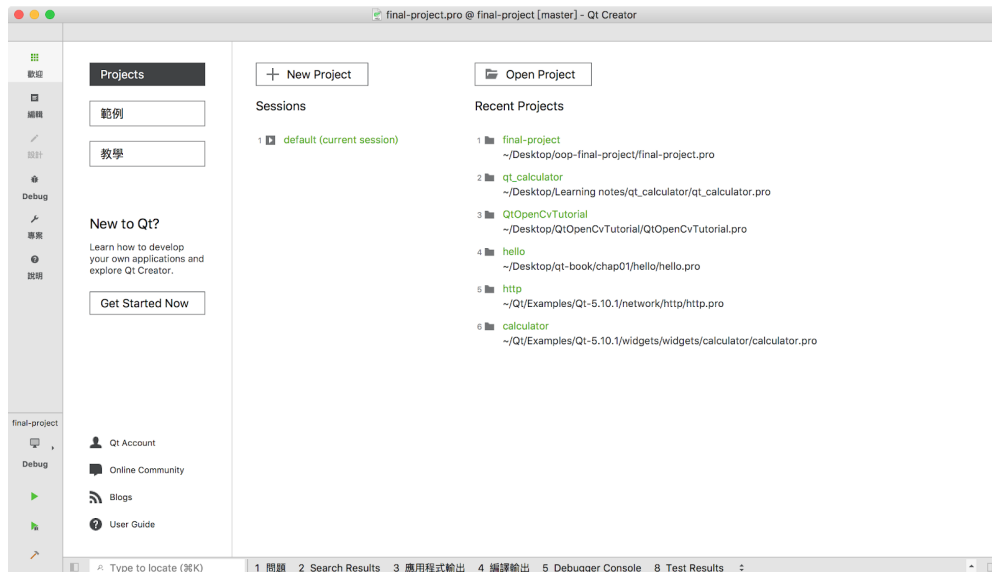
作業系統為macOS High Sierra，但因為Qt、SQLite、OpenCV跨平台的特性，在其他作業系統上應該也能正常編譯與執行。

各軟體版本如下：

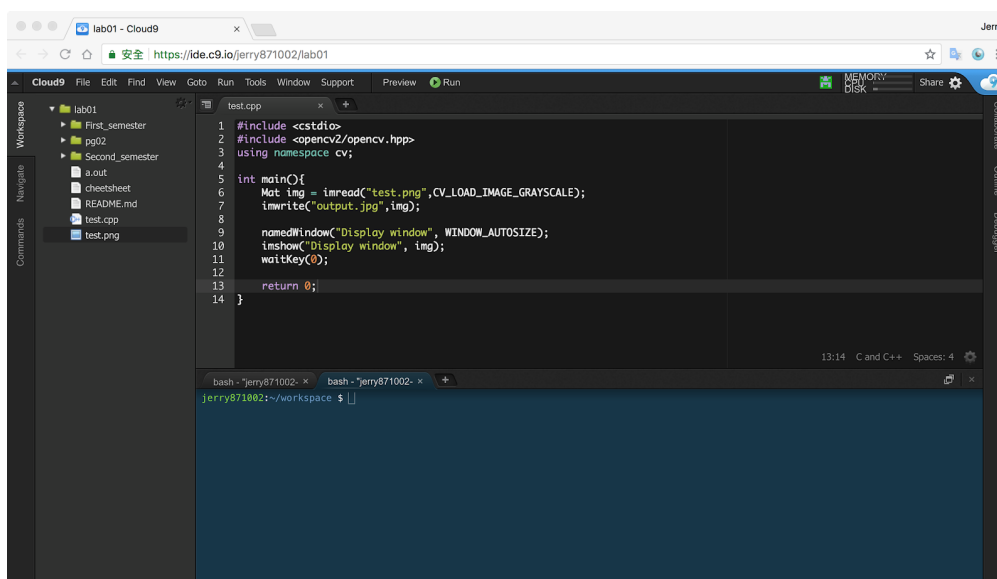
- Qt 5.10.1 ( <https://www.qt.io/download> , Open Source的版本 )
- OpenCV 3.4.1 ( <https://opencv.org/releases.html> )
- SQLite 3.24 ( <https://www.sqlite.org/download.html> )

使用Qt預設的IDE，Qt Creator作為主程式開發及測試的環境。

加密與隱寫術的演算法則是在Cloud9上撰寫與測試。



圖一、Qt Creator



圖二、Cloud9

為了讓我們的Qt專案能順利地使用OpenCV與SQLite，我們需要在專案檔中（final-project.pro）加入幾行程式：

- 為了能使用Qt SQL 模組，在專案檔.pro的QT加入sql，如下所示

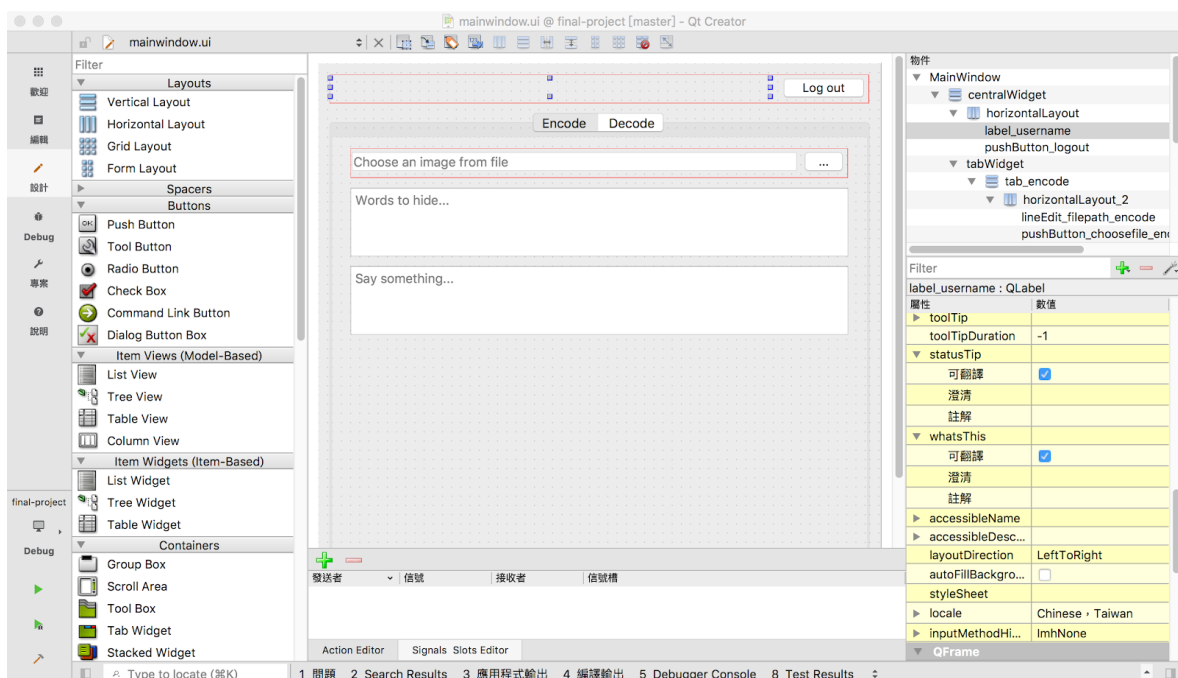
```
QT      += core gui sql
```

- 為了能使用OpenCV函式庫，需要設定include以及動態函式庫 (Dynamic-link library , DLL ) 的路徑，在我的電腦上的設定如下

```
LIBS += \  
    /usr/local/Cellar/opencv/3.4.1_5/lib/*.dylib  
  
INCLUDEPATH += \  
    /usr/local/Cellar/opencv/3.4.1_5/include
```

## 二、圖形介面設計

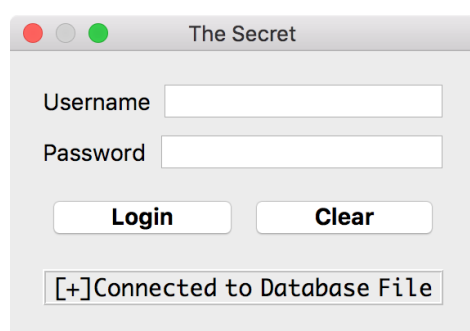
Qt Creator 提供可視化的UI編輯器，Qt Designer，可以輕易用拖拉的方式完成UI設計。透過這個工具，以及Qt中Signal與Slot的機制，設計GUI程式的難度大幅降低。



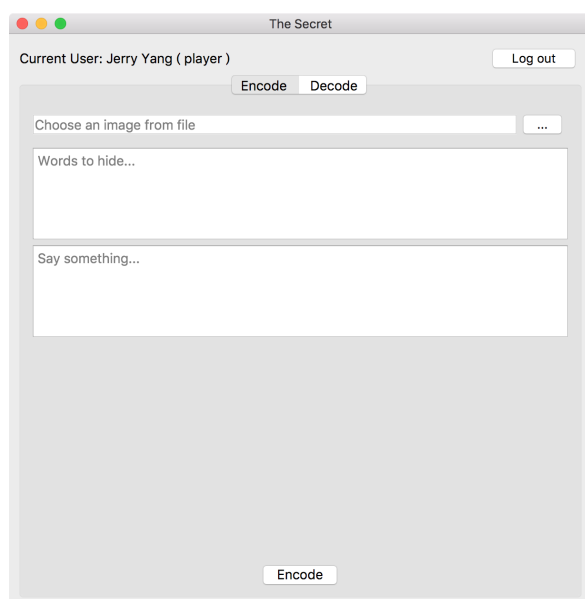
圖三、Qt Creator

其中，Signal與Slot是Qt獨有的機制，有點像聽收音機那種感覺。例如當你聽到收音機說今天會下雨（Signal），你就拿雨傘才出門（Slot）。對照到程式上來說，當某個元件發生了某件事情，我們希望另一個元件作出對應行為。比如按鈕被點擊了一下，此時會發出一個信號（Signal），這種信號像廣播一樣，如果有對象對這個信號感興趣，就用自己的的一個成員函式成為槽（Slot）來響應這個信號。

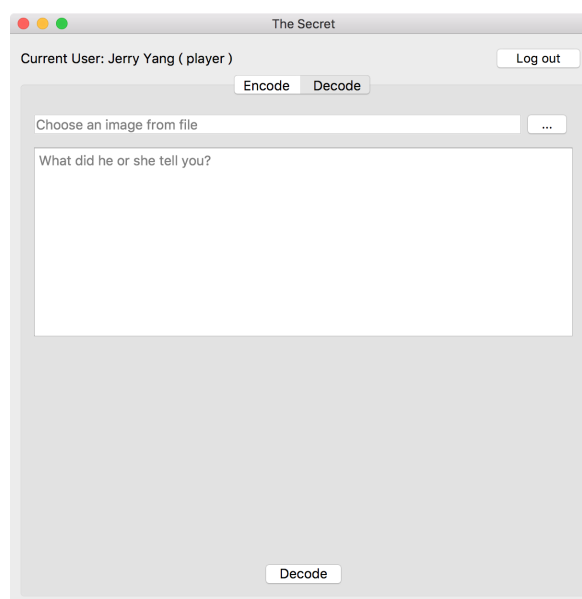
在這裡，我們設計了兩個視窗：登入視窗以及主視窗，介面如下：



圖四、登入視窗



圖五、主視窗（Encode）

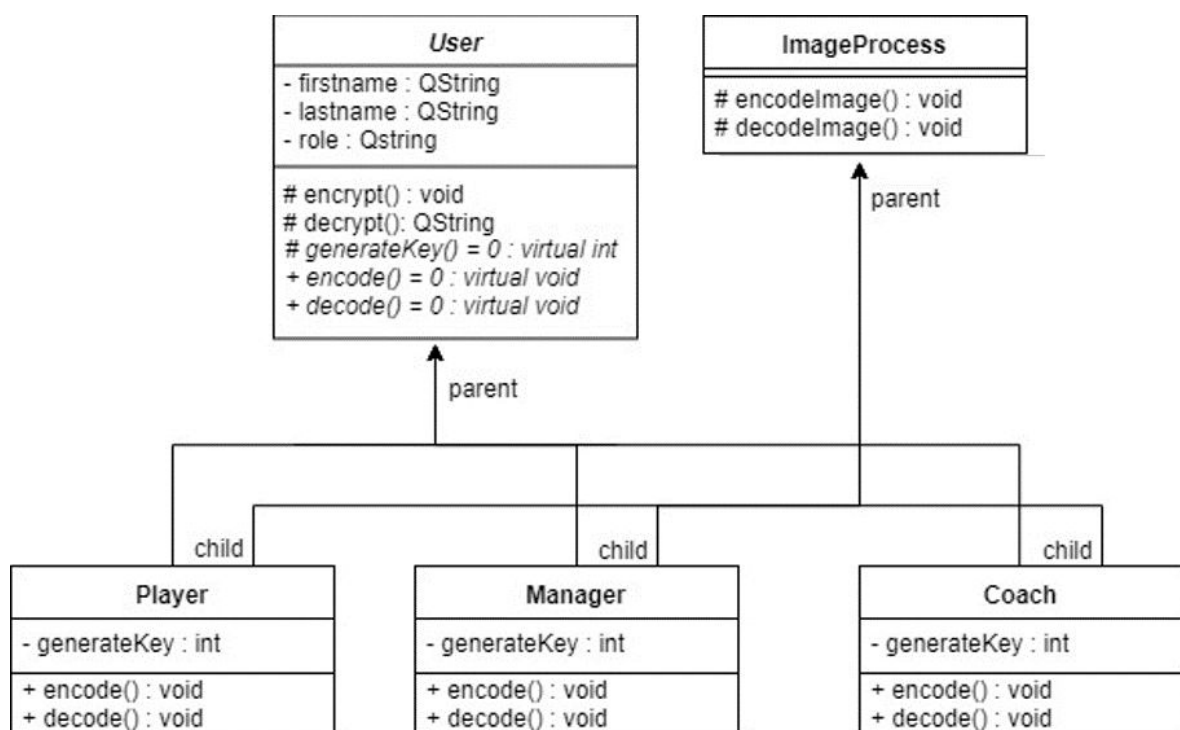


圖六、主視窗（Decode）

登入視窗跟主視窗本身都是一個類別，所有在視窗上發生的事情，例如按按鈕或輸入資料所產生的信號（Signal），都會由他們的成員函式（member function）作為槽（Slot）來處理。而登入視窗是主程式視窗的一個資料成員，在主視窗建構時也會一起被建構。

### 三、類別介紹

下圖是我們這次使用到的類別的關係圖



圖七、類別關係圖

首先，我們先介紹基本成員類別User，資料成員包含使用者的各項基本資訊，包含姓名、角色等。成員函式包含三個純虛擬函式（pure virtual function），這三個函式都跟加密及解密文字有關，因為不同角色間不能知道互相的秘密訊息，這三個函式的實作交給衍生類別（derived



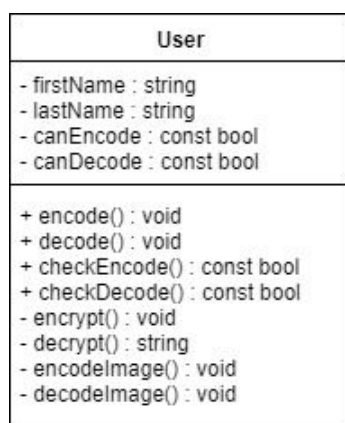
class )，也就是不同的角色類別來決定。因為含有純虛擬函式，User類別為抽象類別 ( abstract class )，不會被直接用來生成實例。

另外，我們把有關影像處理的部份封裝成另一個類別ImageProcess，因為不管角色為何，處理影像的部分都是一樣的。這樣除了可以減少重複的程式碼外，如果以後有其他有關影像處理的功能地方要擴充，也能再不改動既有的程式碼下很快的完成。

Player、Coach、Manager三個類別都繼承了User以及ImageProcess類別，他們分別實作了從User類別繼承的encode()、decode()、generateKey()函式。因為每個角色實作的方式略有不同，讓我們得以實現相同角色才能對談的功能。

事實上，在第一版程式中，為了快速實作出程式，我們只有設計一個使用者類別，所有有關加解密文字與影像處理的功能都塞在同一個類別中，是依靠很多布林變數和判斷式才能達成上述的功能，但這樣的程式碼耦合度高 ( high coupling )，要改變一個模組會引發漣漪效應 ( ripple effect )，導致其他模組也需隨之修改，因此很難去擴充新功能。

所以在接下來的版本中，我們重新調整了架構，讓一個類別只負責單一的工作。新的架構讓我們的程式變得容易維護也容易擴充，現在如果想要新增任何新的角色，都不用更改到已經存在的程式碼，跟呼吸一樣簡單。



圖八、第一版的物件圖

## 四、資料庫

資料庫的部分，我們把檔案命名為users-info.db，並在裡面建立了users\_info這個表格（table），這個表格包含了first\_name、last\_name、username、password、role等欄位，記錄了包含姓名、使用者帳號以及使用者的角色（隊員、教練或經理）等資訊，詳細的配置如下：

```
CREATE TABLE `users_info` (  
    `user_id` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,  
    `first_name` TEXT,  
    `last_name` TEXT,  
    `username` TEXT NOT NULL,  
    `password` TEXT NOT NULL,  
    `can_encode` NUMERIC NOT NULL,  
    `can_decode` NUMERIC NOT NULL,  
    `role` TEXT CHECK(role == "player" OR role == "coach" OR  
role == "manager")  
);
```

當我們在登入視窗輸入完帳號密碼，按下「登入」按鈕時，就會去查找資料庫裡面是否有登錄該使用者的資料，大略的過程如下：

```
void LoginDialog::on_pushButton_login_clicked()  
{  
    //.....  
  
    // 搜尋使用者資料  
    QSqlQuery qry;  
    if (qry.exec("SELECT username, password, first_name,  
last_name, role, can_encode, can_decode FROM users_info WHERE  
username=\'" + username + "\" AND password =\'" + password +  
"\'"))  
    {  
        if (qry.next()) // 有找到符合條件的使用者  
        {  
            //.....  
            // 根據角色的不同，建構出不同類型的物件  
            if (role == "player")
```

```



        parent->currentUser = new Player(略);
    else if (role == "coach")
        parent->currentUser = new Coach(略);
    else if (role == "manager")
        parent->currentUser = new Manager(略);
    //.....
}

```

其中，currentUser是主視窗的資料成員，他是指向類別User的指標（User\*），它指向的實體在登入時建構，登出時解構。我們在這邊把它動態繫結（Dynamic Binding）到Player、Coach或Manager類別上，透過虛擬函式（virtual function），這樣當使用者在執行加密或解密等動作時，會因為角色的不同，而有不同的行為。

## 五、加密演算法

利用使用者輸入的文字製成鑰匙（公鑰，Public key），再利用XOR的自反性（可逆特性）執行加密，也就是所謂的對稱金鑰加密（Symmetric-key algorithm）。

	0100	1011	N		0100	0100	D		
XOR	0000	1010			XOR	0000	1010		
	0100 0100			D		0100 1011			N

圖九、XOR加解密示意圖

以下程式簡單展示了我們加密的過程：

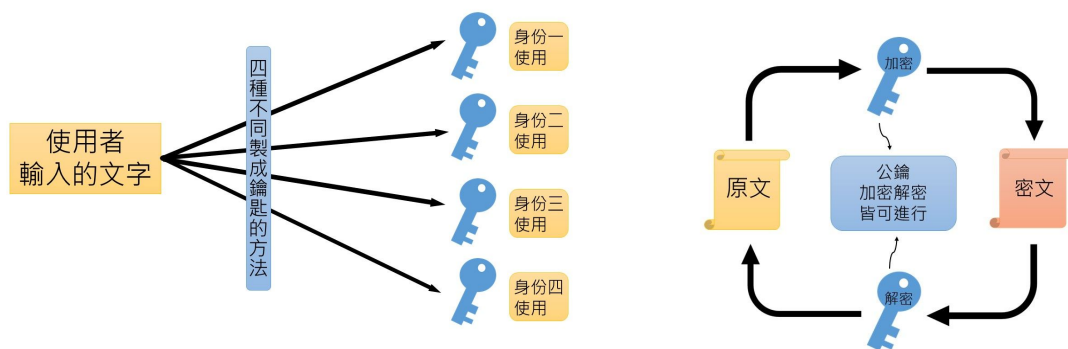
```

// Make a key by messageKey
int key = generateKey(messageKey);

```

```
// Encrypt the message
for (unsigned long i = 0; i < messageToHide.length(); i++)
    messageToHide[i] = messageToHide[i] ^ key;
```

類別Player、Coach及Manager中的函式generateKey是其三者間的主要差別，由於他們接受到使用者輸入的訊息後，產生鑰匙的方法不同，因此其產生的鑰匙只能將同類別產生的密文轉換成正確的原文，否則將得到亂碼。



圖十、加解密示意圖

## 六、隱寫術

- 概念：

為了將文字隱藏到圖像中，我們把字串分解成一個一個一個字元（char，8位元），並將8位元中的每一位元依序儲存在像素值（紅色，綠色，藍色）的最低有效位（Least Significant Bit，LSB）中。最低有效位是二進位數字中的最低位（第0位，權值為 $2^0$ ）。因為改變像素值的LSB不會對圖像產生巨大的差異，以這種方式操縱像素，該圖像用肉眼看起來仍然會跟改變前相似。

- 操作：

我們將前一段，已經加密過的文字，放到圖片裡面進行加密。首先，用OpenCV的函式庫的Mat型態來儲存圖像，Mat是OpenCV訂定的資

料型態，代表的是矩陣 ( Matrix ) 前三個字母，因為影像其實也可以看成是一個二維陣列。Mat最基本的成員有長 ( row )、寬 ( col )、通道數 ( 灰階圖為1，彩色圖為3 ) 等資訊。接著，我們再用Vec3b這種vector去裝每一個像素中代表藍紅綠元素的值，每一個元素值用uchar ( unsigned char, 0~255 ) 來儲存。

例如下面這段程式，我們用它來遍歷一張圖片中所有像素，並將每個像素用Vec3b來儲存，將來可以對它進行操作：

```
// 讀取影像
Mat image = imread(filename);
//.....
for (int row = 0; row < image.rows; row++)
{
    for (int col = 0; col < image.cols; col++)
    {
        for (int col = 0; col < image.cols; col++)
        {
            // stores the pixel details
            Vec3b pixel = image.at<Vec3b>(Point(row, col));
            // 接下來用pixel.val[color]去對像素做操作
            //.....
        }
    }
}
```

接下來我們開始讀取資料中的字元 ( 就是那些被加密後的文字 )，每個字元用8位元來代表，將此8位元的資料藏進Vec3b每個元素的LSB ( 顏色通道的最後一位元 )。把訊息藏入像素中的圖解如下：

( 備註：Vec3b將通道0定義成藍色通道，通道1為綠色通道，通道2為紅色通道，每一個通道有一個0~255的值 )

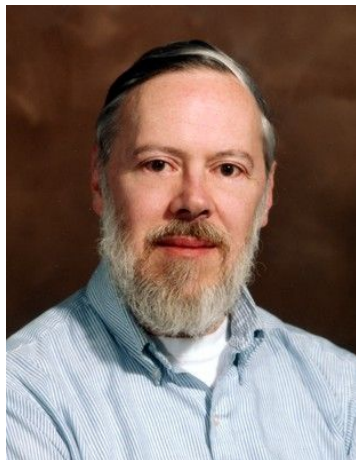
```
pixel.val[0] = 0 1 0 0 1 0 1 1
pixel.val[1] = 0 0 0 0 1 0 1 0
pixel.val[2] = 0 1 0 0 0 1 0 0
```

(藏入111的訊息後)

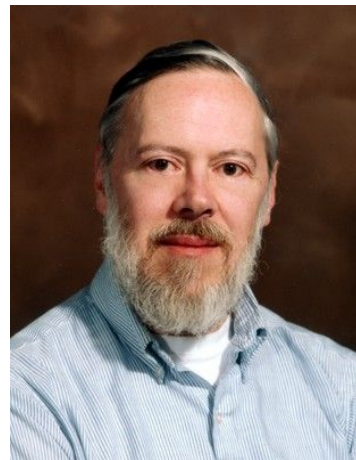
```
pixel.val[0] = 0 1 0 0 1 0 1 1
pixel.val[1] = 0 0 0 0 1 0 1 1
pixel.val[2] = 0 1 0 0 0 1 0 1
```

因為是改變每個色彩元素的最低位元，所以肉眼幾乎看不出來有所改變。

以下是我們加密前，與加密後的圖片：



圖十一、加密前



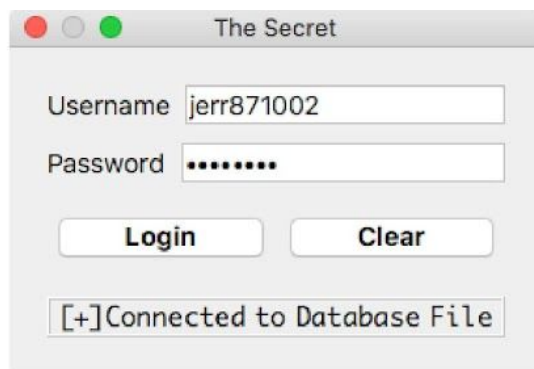
圖十二、加密後

完全看不出來差在哪！

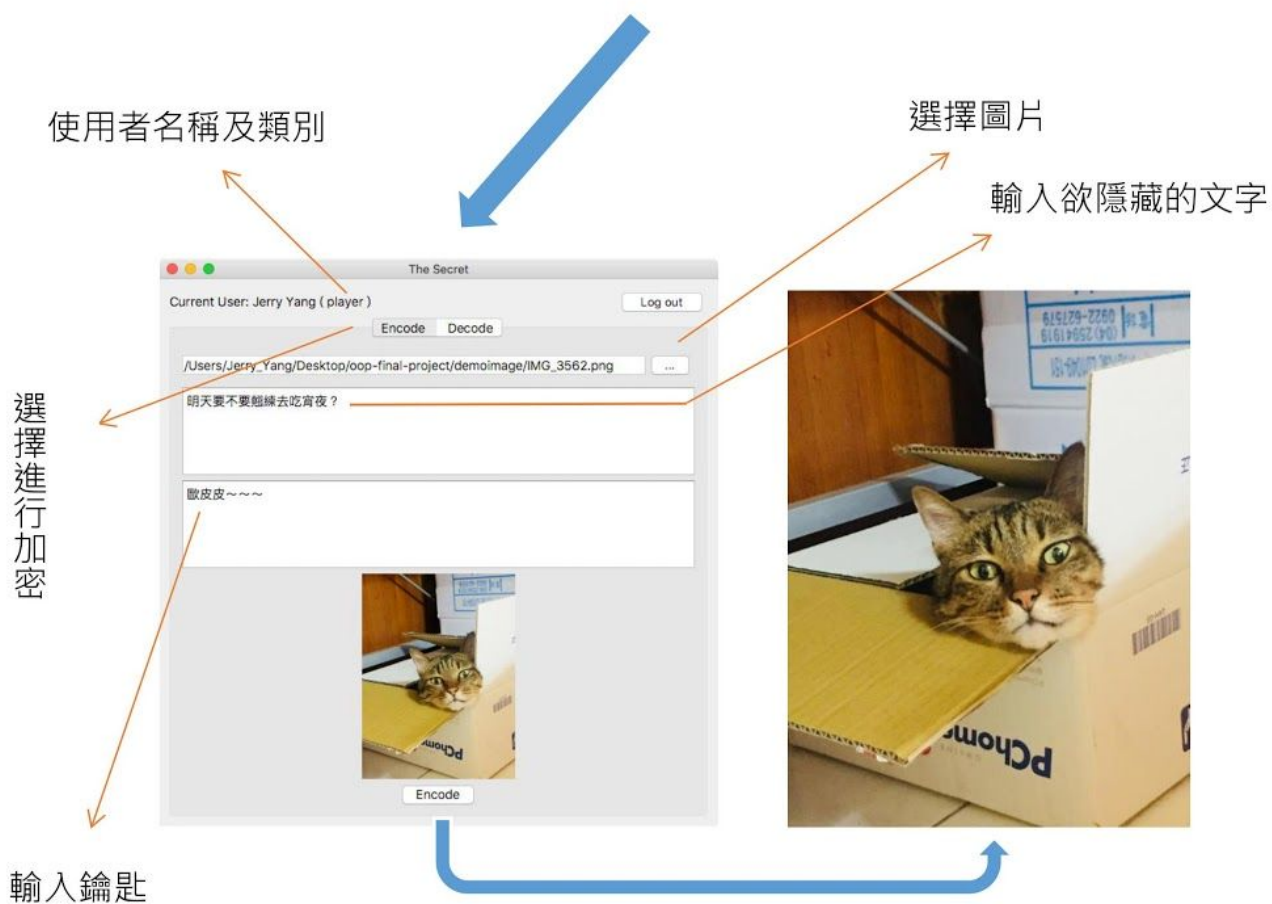
- 解密:

為了從圖像中提取消息，我們將依序提取像素值 ( RGB ) 中的LSB，當讀到'\0'時，則跳出迴圈，如此一來我們可以得到隱藏在其中的密文。最後再交給前面提到解密文字的演算法，便可以得到我們想要的訊息。

# 執行畫面節錄



登入畫面輸入帳號及密碼



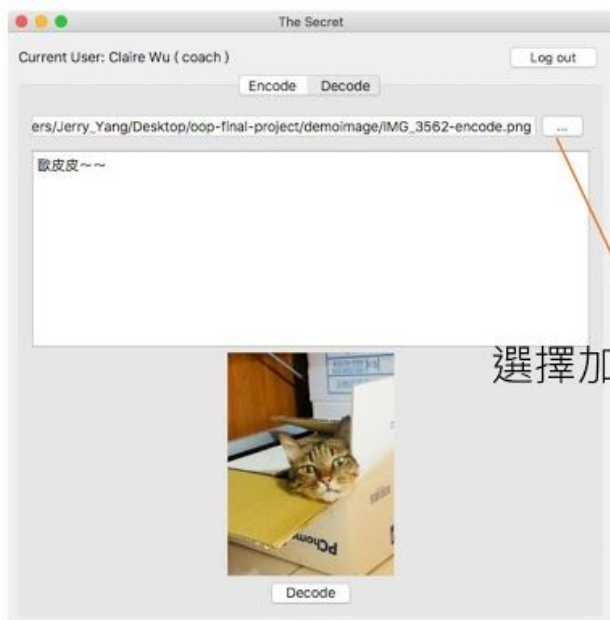
選取圖片，輸入

在原本選取圖片的資料夾中

欲加密的內容並進行加密

產生加密後的圖片

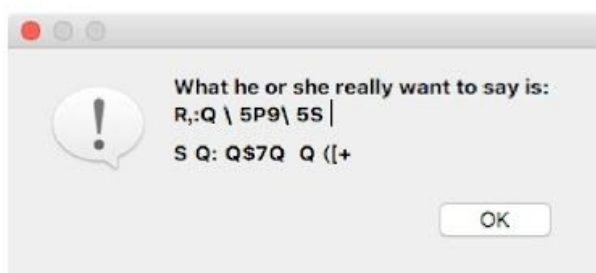




選擇加密過的圖片

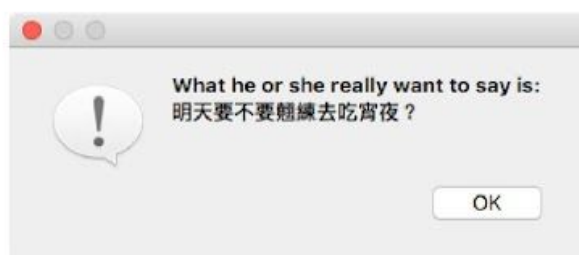


輸入鑰匙



如果對應的類別不相同

會出現亂碼



如果對應的類別相同

隱藏的文字將會顯示出來



## 分工

	楊哲睿	陳芳瑩	劉文
圖形介面	X	X	X
使用者類別設計	X		
資料庫	X		
加密演算法			X
隱寫術		X	
專題報告	X	X	X
玩貓咪		X	
Swimming			X

## 感想

### ● 楊哲睿

這個專題是我第一次完整地做出一個專案，對我來說是很新的體驗。它讓我學習到很多課堂上沒有教的東西，比如如何做出GUI程式、如何使用第三方函式庫以及如何結合資料庫，還有最重要的，如何跟別人一起合作寫程式。我也在設計使用者類別時，實際體會了物件導向中，封裝、繼承、多型的特性帶來的好處。應用課堂上教的概念，我在第一版完成後，又重構了我們的程式，讓他變得精簡有效。這些東西都是要實際動手操作後才會發現的，我在這些過程中收穫良多。

最後我要感謝我的組員們一直聽我撈叨還有完成我有點多的要求，是因為他們的幫忙才能一起完成這個雖然看起來沒什麼用（？）但是很好玩的程式。

## ● 陳芳瑩

個人覺得這個主題跟其他組的題目很不一樣，很酷也還算實用，我是負責encode/decode的部分，第一次使用OpenCV的函式庫，也是第一次處理圖片的檔案，在查資料上面花很多時間，但也算是長了不少知識，學到新的東西我覺得很開心。整個做專題的過程都按著當初的規畫走，不會有狗急跳牆的感覺，壓力比較不會那麼大，可以有時間好好思考更適合的方式來完成著個程式。

由於我們三個都在校隊，背景很相似，加了我們都很有共鳴的情境，讓整個專題變得更有趣～～雖然我是個快樂小夥伴，自己的工作之餘，幫不太上甚麼忙，還是感謝大家我們一起完成啦～

## ● 劉文

以往打程式都是上機時完成題目，所以對我來說完成這個專案是一個很棒而且很新穎的體驗，我覺得我主要學到的其中一點是string的使用。為了要使這個專案也可以讀取中文，我做了不少測試與狀況模擬，這與在上機時間內完成題目相較之下是完全不同的體驗。除此之外，為了完成關於成員類別的設置，哲睿使用了許多這學期學到的繼承、多型等等概念，再聽他詳細的講解後，使我對物件的概念也有了更進一層的了解。

很感謝兩位隊友容忍我經常性的搞不清楚狀況，適時提供給我協助，讓我們得以順利完成專案。

# 問題與討論

目前我們的程式只能隱寫到PNG檔裡面，未來可能要研究怎麼讓大家廣泛使用的JPG檔也能使用這個程式。

## 參考資料

- hitanshu-dhawan/ImageSteganography, GitHub  
這個在GitHub上的專案對我們理解隱寫術的幫助很大  
<https://github.com/hitanshu-dhawan/ImageSteganography>
- 阿洲的程式教學  
這個網站裡關於Qt與OpenCV的教學讓我們能對他們快速上手  
<http://monkeycoding.com/>
- C++ Qt SQLite-Login With Database, YouTube  
這個影片教我們怎麼創建資料庫並連結到我們的程式中  
<https://www.youtube.com/watch?v=cc06D3wuTn4>