

# National Taipei University of Technology

Windows Programming (Fall 2019)

Homework #8

Deadline: 1/3 (Fri), before 24:00 PM

## 壹、注意

請遵守以下規則，否則單次成績以 0 分計算

1. 準時繳交作業，逾時不候
2. 不准抄襲他人作業，請自己完成

## 貳、主題

根據上一次的作業內容，本次新增一個 Resize Shape 的功能，以及使用 google Drive API 實作 Save & Load 功能。

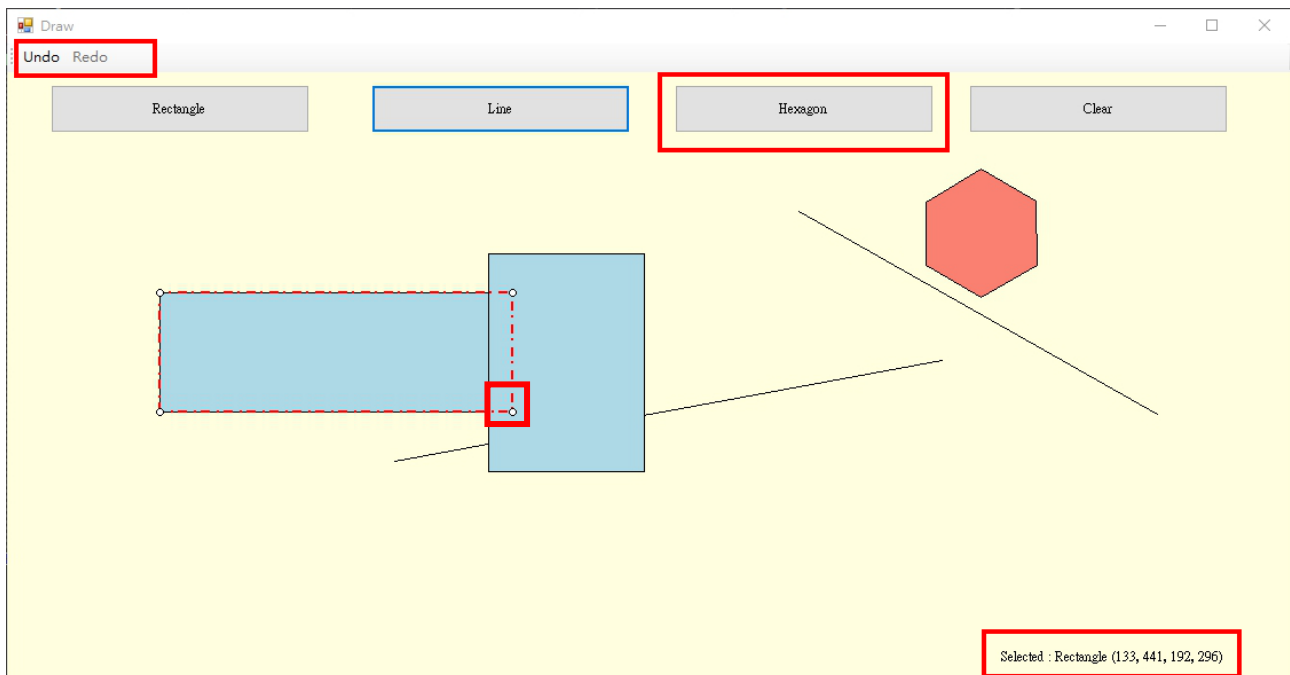


圖 1 Windows Form GUI

## 參、題目介紹

### 一、[4 pts] Resize Shape

使用者選取到畫布上的圖片後，點擊右下角的白點並拖曳時，可以放大或縮小圖形。

- 請根據圖片上選去後的右下角白點去做 Resize。
- 只需要右下角白點即可，如圖 1。
- 放大或縮小後，圖片的左上角座標不變。

### 二、[1 pts] Factory Pattern

在此作業中，可以在 Canvas 上繪製三種形狀（線、矩形和六邊形）。當繪製形狀時，您需要使用（簡單）工廠模式來創建不同形狀的 instance。

### 三、[3 pts] State Pattern

至少分為兩個狀態，繪圖狀態以及游標狀態，點擊 Shape Button 時切換為繪圖狀態，即可畫圖(增加圖形)，畫完圖片後，則自動切換回游標狀態。在游標狀態下，可以選取圖片以及 Resize。

### 四、[6 pts] Save and Load

新增兩個 button 分別為 Save 以及 Load。當按下 Save 時，跳出對話框詢問是否確定要儲存，若選擇儲存，則將所有圖形(Model 的所有資訊) 儲存至 google 雲端硬碟上。當按下 Load 時，跳出對話框詢問是否要重新載入(Load)，確認後將雲端上儲存的圖形 Load 回來並且顯示該儲存的圖形。

- 儲存功能將所有圖形儲存到 Google 雲端硬碟中。
- 讀取功能自 Google 雲端硬碟讀取所有圖形。
- 您可以自己安排或定義圖形的儲存格式。
- 上傳(save)圖形時應使用 **非同步方法**，UI 在上傳期間應該可以繼續操作。相反地，在載入(load) 圖形時，應使用 **同步方法**，UI 暫停動作，避免使用者在正確讀取文件之前進行任何操作。
- **請注意，GoogleDriveService.cs 僅為使用範例，供參考用，寫作業時應該依據你自己的需求修改之。**

### 五、[3 pts] Redo and Undo

你需要使用 Command Pattern 來實作 redo 和 undo 功能。

- 如果是第一步，undo button 為 disable。
- 如果是最後一步，redo button 為 disable。
- 畫圖需 redo、undo，點選不用。
- Resize Shape 必須能 redo、undo。

### 六、[6 pts] Unit Testing

請為每個 Model class 的每個 method 撰寫 test case，且必須全部通過測試，Unit Test 是根據 test-coverage 去評分。。

### 七、[5 pts] GUI Testing

請使用自動化 UI 測試為你的 GUI 編寫測試，測試案例(Test case)的規定如下：

- 只有 Windows Form 需要做 GUI Testing。
- 請**單獨**將畫面上各繪圖按鈕分別測試一次(畫出正方形、畫出線、畫出六邊形、清空畫布)。當畫出一個形狀後，請點選這個形狀，並 Assert 畫面上的 Label 所顯示的值與所繪的圖形吻合(assert 圖 1 的 Label 為正確的)。
- **請單獨測試 Resize 功能，當選取圖片將其 Resize 後，選取該圖片並 Assert 新的座標是正確的。**
- **請單獨測試 Redo、Undo 功能，並進行適當的 Assert。**
- 請整合所有按鈕畫出一個**具體的圖形**，過程中必須用到 **Resize**，以及 **Undo、Redo**，並執行適當的 Assert，然後**上傳**至 Google drive，**清除**畫面，再重新至 Google drive **載入**圖形，並 **Assert** 圖形是正確的，最後關閉。
- 請於老師網站下載 Robot.cs 以及 DrawingTestTool.cs

#### 八、[3 pts] MVC Pattern

你的設計將根據 MVC 架構評分，讓 UI 盡可能薄，必須將 UI 以及 Model 切開，強制實行單向依賴，建議遵循圖 1 Class Diagram 設計。

#### 九、[2 pts] Adaptor Pattern

在本作業中，您需要使用同一 Model 實現兩個不同的 view。這兩個 view 具有不兼容的圖形界面。請使用 Adaptor Pattern 來解決該問題。

#### 十、[2 pts] Observer pattern and event-delegation model

您需要使用帶有 C# event-delegation model 的 observer pattern 來實現 MVC 架構。

#### 十一、[9 pts] Code Quality

Code Quality 將取決於你的程式碼是否有 bad smell，你的分數將取決於你的 code smell 密度。當你使用 Windows Form Designer 產生程式碼時，其預設的變數命名、副程式命名等可能違反 coding standard，請修改以符合本課程之規定。

#### 十二、[2 pts] Summary

每一次寫完作業時都必須完成 homework summary，必須填寫本次作業花費時間，請在網站下載範本檔。

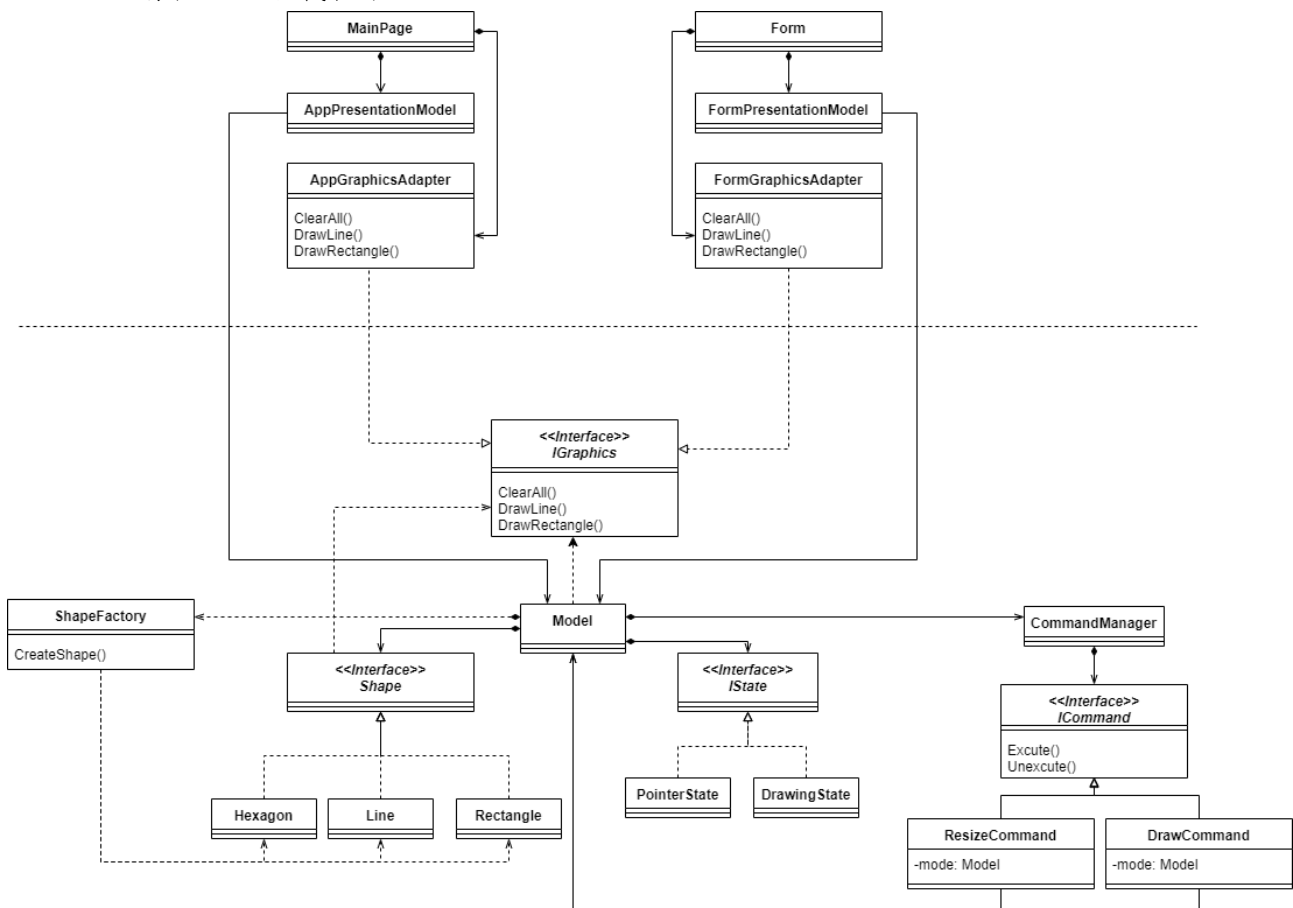


圖 2 Reference class diagram

### *Additional Information*

The following information may be useful to your homework. If you need more details, please visit MSDN website.

1. Draw Rectangle in Windows Form - You can use the following codes to draw diamond in Windows Store Form.

```
myRect = new System.Windows.Shapes.Rectangle();  
myRect.Stroke = System.Windows.Media.Brushes.Black;  
myRect.Fill = System.Windows.Media.Brushes.SkyBlue;  
myRect.HorizontalAlignment = HorizontalAlignment.Left;  
myRect.VerticalAlignment = VerticalAlignment.Center;  
myRect.Height = 50;  
myRect.Width = 50;  
myGrid.Children.Add(myRect);
```

For more information, please visit MSDN :

<https://docs.microsoft.com/en-us/dotnet/api/system.drawing.graphics.fillpolygon>

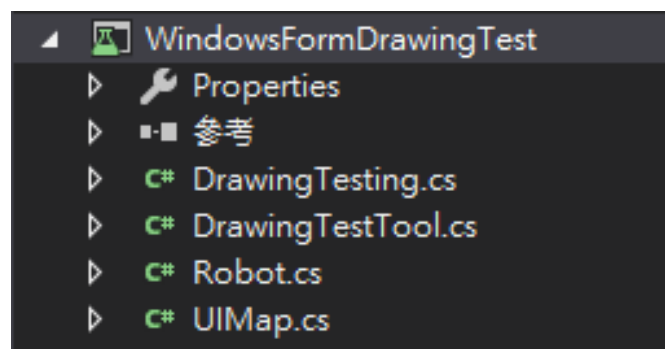
2. Draw Rectangle in Windows Store App - You can use the following codes to draw diamond in Windows Store App.

```
rectangle = new Rectangle();  
InitializeShape(rectangle, 0, 250, 150, 150, _blueColor);  
_canvas.Children.Add(rectangle);
```

For more information, please visit MSDN :

<https://docs.microsoft.com/en-us/uwp/api/windows.ui.xaml.shapes.polygon>

3. Add Two Class Robot and DrawingTestTool - You can use the following class to Do UI testing.



4. Drawing UI testing - You can use the following codes to Do UI testing.

```
[TestMethod]
public void DrawingRectangle()
{
    // PANEL_ACCESSIBLENAME為Panel的AccessibleName
    DrawingTestTool.DrawRectangle(PANEL_ACCESSIBLENAME, 50, 150, 50, 150);
    Assert.IsTrue(true);
}
```