National Taipei University of Technology

Windows Programming (Fall 2021)

Homework #6

Deadline: 12/13(Mon), before 24:00

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

從這次作業開始，我們要建構一個支援 Windows Form 和 Windows Store App 的繪圖程式。我們在 Lab 中已經實作過繪圖程式，請參考圖 1、圖 2 實作你的程式，提供繪製可變 大小的線條和矩形的功能。

- Windows Store App 的目標顯示解析度為 10.6 英寸（1366 * 768）。
- 兩個 Views(Windows Form 和 Windows Store App)必須使用完全相同的 Model。
- Model 必須記錄（存儲）所有創建的形狀，以便未來可以編輯(或重繪)這些形狀。

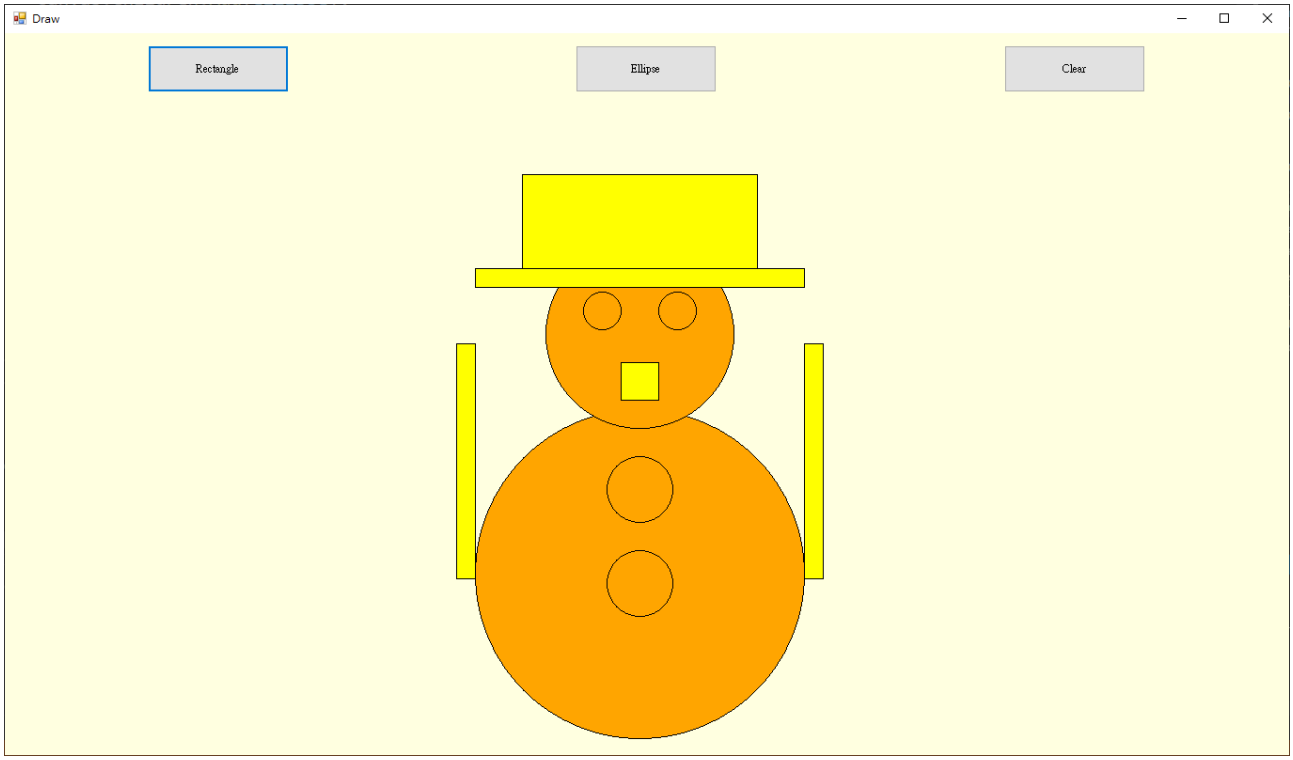

圖 1 Windows Form GUI

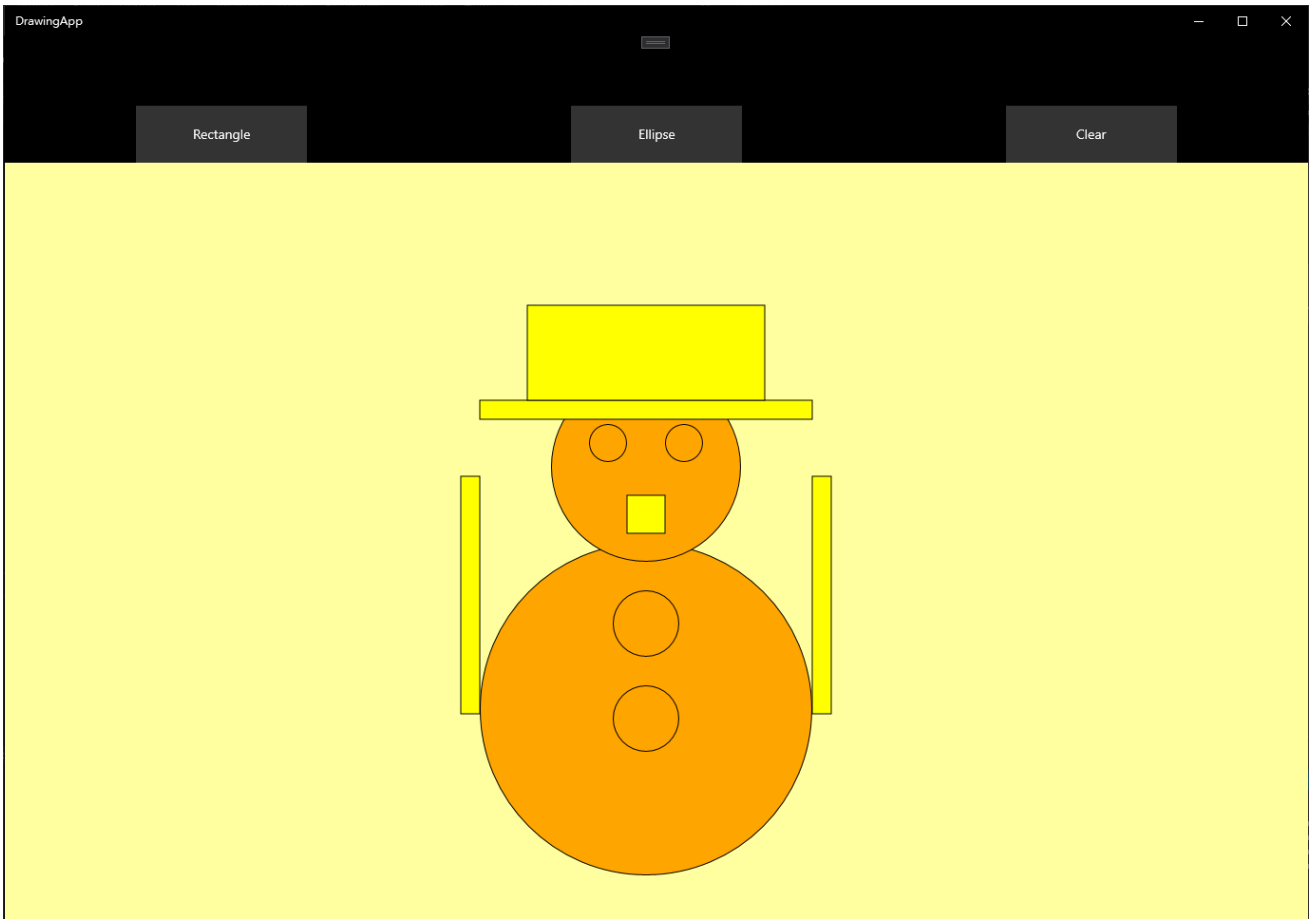

圖 2 Windows Store App

參、 題目介紹

一、[2 pts] Button

請在兩個 View 分別放上 Rectangle、Ellipse 以及 Clear 三個 Button。

點擊 Rectangle 或 Ellipse Button 時，你可以畫出對應的圖形，程式應該適當地設定各按鈕的 enable/disable。

- 點擊 Clear Button 時，清空畫面上所有圖片，並且將所有按鈕 enable。
- 畫完一個圖形後，將所有按鈕設為 enable。
- 點擊畫圖(Rectangle 或 Ellipse Button 時)按鈕時須將另一個按鈕設定為 enable，且將本身按鈕設定為 disable。

二、[4 pts] 繪製圖形

點擊畫圖(Rectangle 或 Ellipse Button 時)按鈕後，進入繪製模式，當在畫布上按下 滑鼠時，開始繪製，當滑鼠移動（拖動）時，形狀的大小會更改。當釋放滑鼠時，必須儲存形狀至 Model。另外，在釋放按鈕之前，應顯示預覽形狀。

三、[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。

四、[2 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：

1. 只有 Windows Form 需要做 GUI Testing，Windows Store App 不用。
2. 各別功能測試：寫 Test case 將畫面上三個按鈕分(畫出正方形、畫出線、清空畫布)各測試一

次。

3. 整合功能測試：寫 Test case 畫出一棟房子(如圖 1)並且清除後關閉。
4. 本次作業不要求 Assert 畫面的正確性，只要畫出來，肉眼看得到即可，下次作業會要求加入適當的 Assert。

請參考老師網站上的 Robot.cs，即可撰寫繪圖的測試。

五、[3 pts] Observer (或 DataBinding)

請使用 C# 的 event-delegation model 或 observer pattern 來實現 MVC 架構。

六、[3 pts] Adaptor Pattern

在本作業中，您需要使用同一 Model 實現兩個不同的 view，而這兩個 view 的圖形界面並不相同，請使用 Adaptor Pattern 來解決該問題。

七、[4 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 6 Class Diagram 設計。

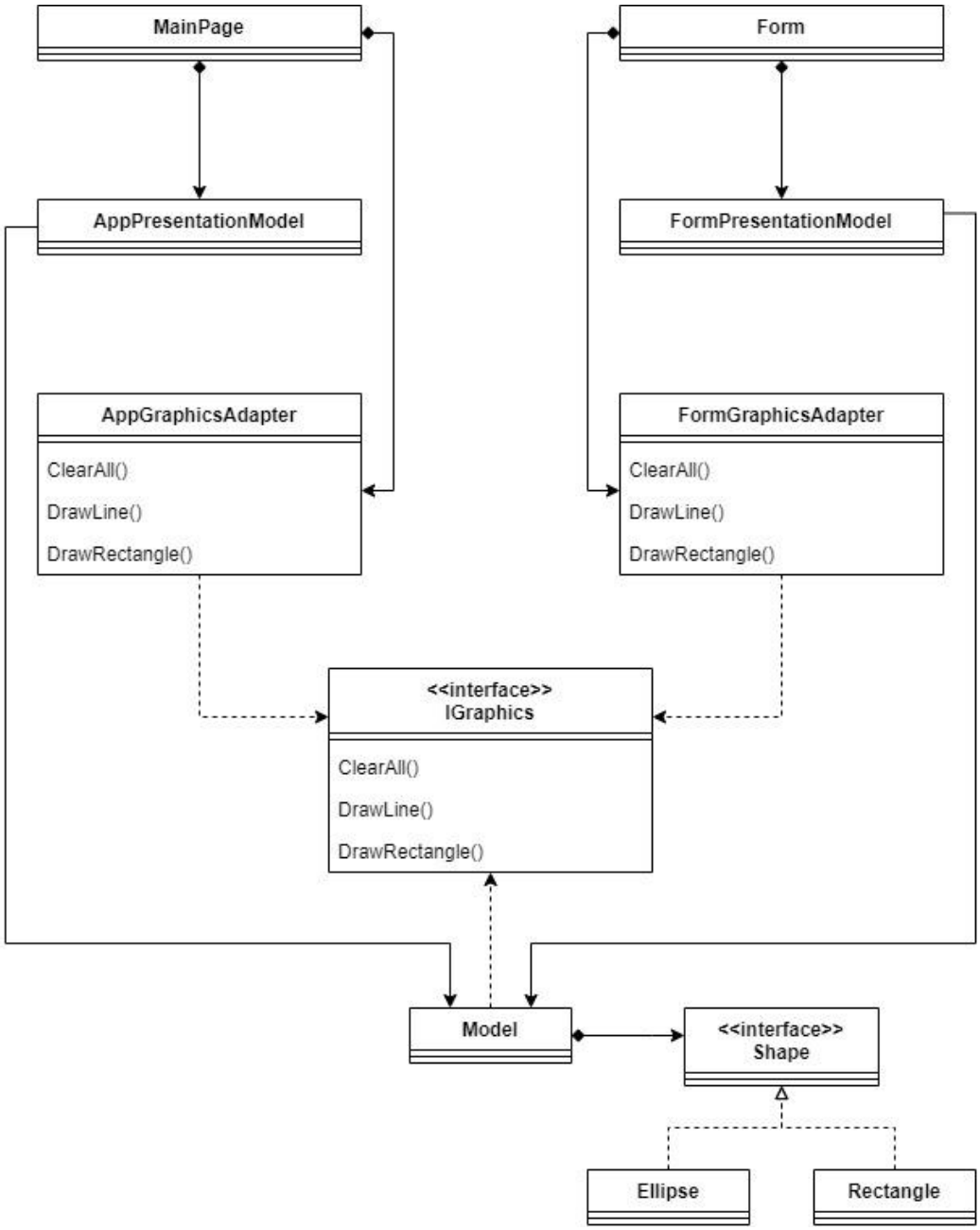

圖 6 Class Diagram

八、[8 pts] Code Quality

請使用 Dr.Smell 檢查你的程式，Code Quality 成績取決於你的程式碼是否有 bad smell，分數由你的 code smell 密度決定。請注意，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本課程之 coding standard，請修改以符合本課程之規定。

九、[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。

Additional Information

1. Drawing a rectangle in Windows Form – the following code can be used.

```
public DrawingForm()
{
    InitializeComponent();
    _canvas.AccessibleName = "_canvas";
    _canvas.Dock = DockStyle.Fill;
    _canvas.BackColor = System.Drawing.Color.LightYellow;
}
public void DrawRectangle(PaintEventArgs e)
{
    // Create pen.
    Pen blackPen = new Pen(Color.Black, 3);

    // Create rectangle.
    Rectangle rect = new Rectangle(0, 0, 200, 200);

    // Draw rectangle to screen.
    e.Graphics.DrawRectangle(blackPen, rect);
}
```

2. UI testing for drawing – add class Robot (the class can be downloaded from the instructor's web site).

   You can use the following code to do UI testing. You may also change the above class as you wish.

```
[TestMethod]
public void DrawRectangle()
{
    // BUTTON_NAME 為畫圖按鈕，CANVAS_NAME 則為 canvas 的 AccessibleName
    _robot.ClickButton(BUTTON_NAME);
    _robot.DragAndDrop(CANVAS_NAME, 500, 400, 850, 750);
    …
}
```