

National Taipei University of Technology

Windows Programming (Fall 2022)

Homework #8

Deadline: 1/7 (Sat.), before 10:00 AM

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。學期中最多僅能申請一次作業延期(最長可以延 7 天)，若需延期，請於繳交作業截止前通知助教或老師。
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，本次新增使用 google Drive API 實作 Save & Load 的功能。

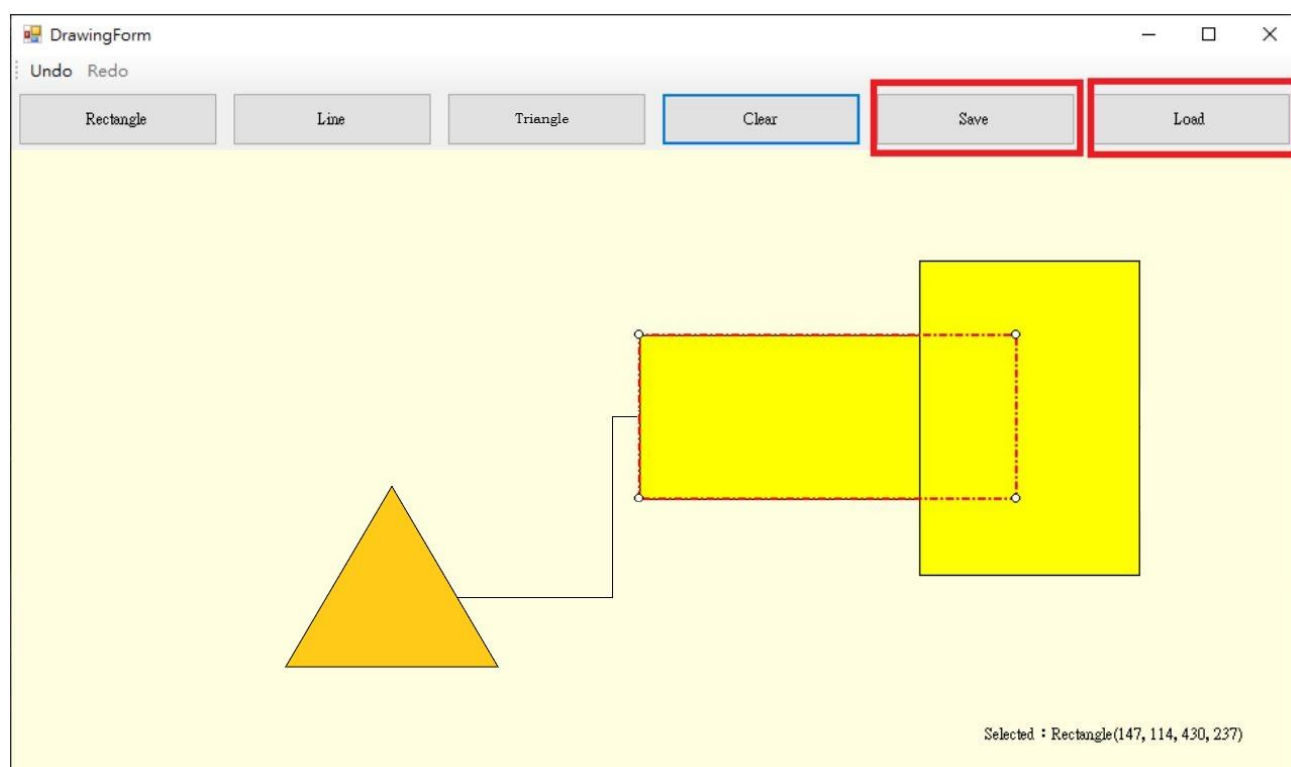


圖 1 Windows Form GUI

參、題目介紹

一、[1 pts] Factory Pattern

在此作業中，可以在 Canvas 上繪製三種形狀（線、矩形和三角形）。當繪製形狀時，請使用簡單工廠(simple factory)模式來創建不同形狀的 instance。

二、[3 pts] State Pattern

至少分為三個狀態，繪圖狀態(矩形和三角形)、繪線狀態，以及游標狀態(非繪製矩形、三角形、線時的狀態)，點擊 Shape Button (矩形和三角形) 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。 點擊 Line Button (線)時切換為畫線圖狀態，即可畫線，畫線後則自動切換回游標狀態。在游標狀態下，可以選取圖片。

三、[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。
- 你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，請於 GoogleDrive Lab 中下載檔案。**

四、[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。

五、[3 pts] Precondition

請依據上課講解 Bug and Debugging 時所建議的方式，在你的程式中做適當的 precondition 檢查，若未通過 precondition 即拋出例外。**請在 Summary 檔寫出任意一個你設計 precondition 檢查的地方，並標示其檔名、行數，及解釋檢查 precondition 的原因，未註明此項不計分。**

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、畫出三角形、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 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 **Undo** 以及 **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。
- 請參雲端連結或 teams 檔案目錄裡的 [Robot.cs](#)。

八、[2 pts] Observer (或 DataBinding)

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

請註明在 Summary 中，未註明此項不計分。

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，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。

十、[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 2 Class Diagram 設計。

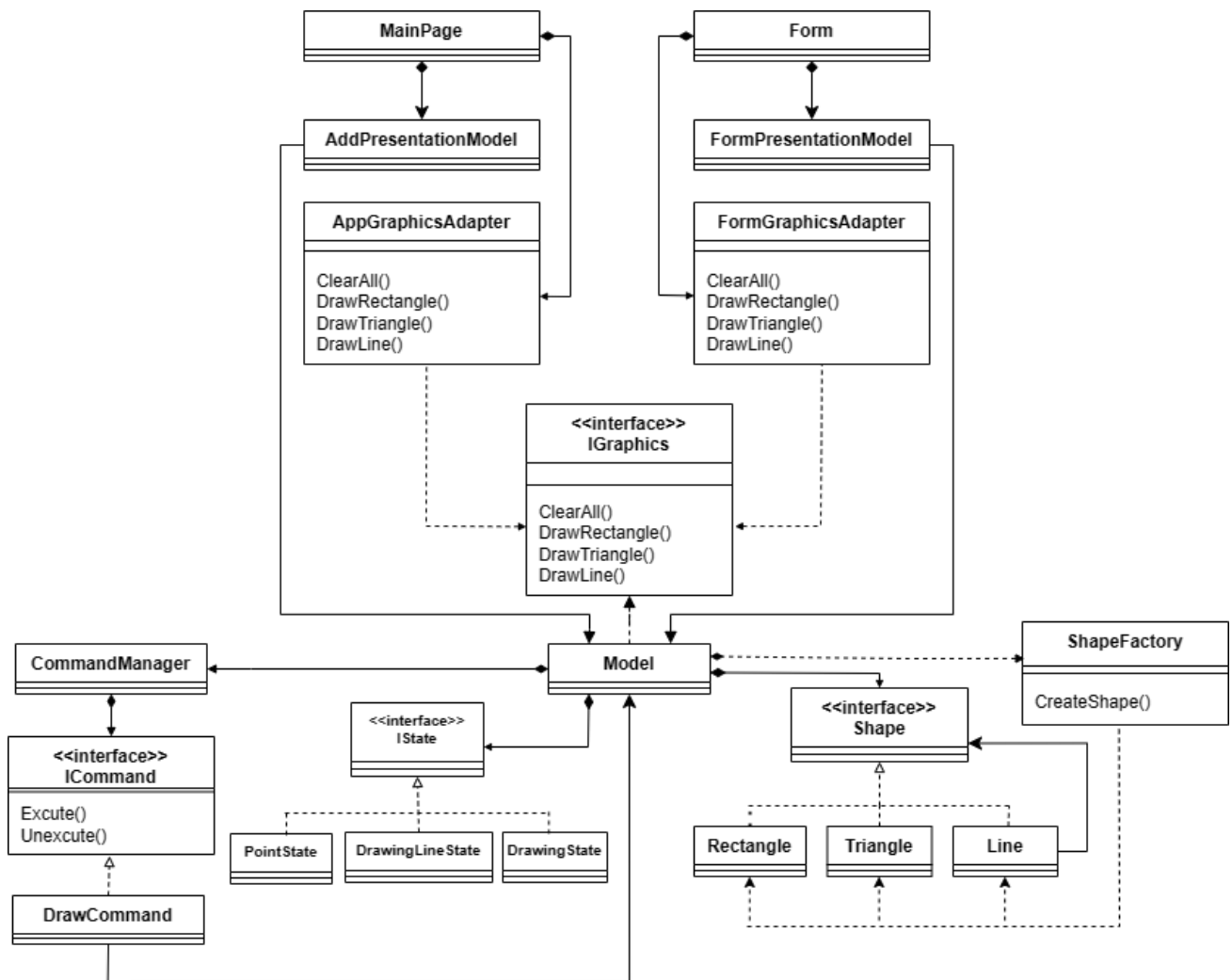


圖 2 Class Diagram

十一、[9 pts] Code Quality

請使用 Dr.Smell 檢查你的程式，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。

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);
    ...
}
```