# Application architecture and features

## Application information

* Solution/Project Name: Canvas Practice
* Description: For users to draw/erase/select graphs on canvas. Configuration import/export functionality is provided.
* Github

## Application architecture

### Directory Hierarchical Tree

Generic file and its usage will not be mentioned. (e.g., App.xaml, AssemblyInfo.cs).

|  |  |
| --- | --- |
| ├─Behavior |  |
| │ ClosePopupBehavior.cs  │ | For children buttons to determine whether to close parent Popup or not |
| ├─Common |  |
| │ AdvancedInvokeCommandAction.cs | Implement TriggerAction to send the command parameter and event argument simultaneously |
| │ BindingProxy.cs | For the use of xaml hierarchical binding |
| │ ClosePopupBehaviorService.cs | Attach Property for ClosePopupBehavior.cs |
| │ NotifyPropertyChangedBase.cs  │ | Generic class for INotifyPropertyChanged implement |
| ├─Model |  |
| │ CompositeCommandParameter.cs | Model for AdvancedInvokeCommandAction; combination of command parameter and event argument |
| │ ShapeAttribute.cs | Shape Properties |
| │ ShapeType.cs | Predefined attributes for Shape Types (e.g., Key, System.Type, Icon) |
| │ UXMode.cs | Model of Three edit mode (Draw, Erase and Select) |
| │ |  |
| ├─Resource  │ | Resource files |
| ├─Style |  |
| │ Button.xaml  │ | Button Style |
| ├─View |  |
| │ MainWindow.xaml | Describe UI elements |
| │ MainWindow.xaml.cs  │ | Code behind for MainWindow.xaml. Dynamically create UI elements and Binding setting |
| └─ViewModel |  |
| MainViewModel.cs | 1. Handle Mouse events of UI elements 2. Implement and handle the back-end logic 3. Read/write config file |

### MVVM Structure

#### ViewModel

1. Catch Canvas Mouse event (mouseDown/Move/Up).
2. Catch Shape Mouse event.
3. Handle model data.
4. Handle the ui status (e.g., currently focused graph, ux mode, shape type)
5. Invoke the ui edit process (create, remove, etc.).
6. JSON File import/export.

#### View – xaml

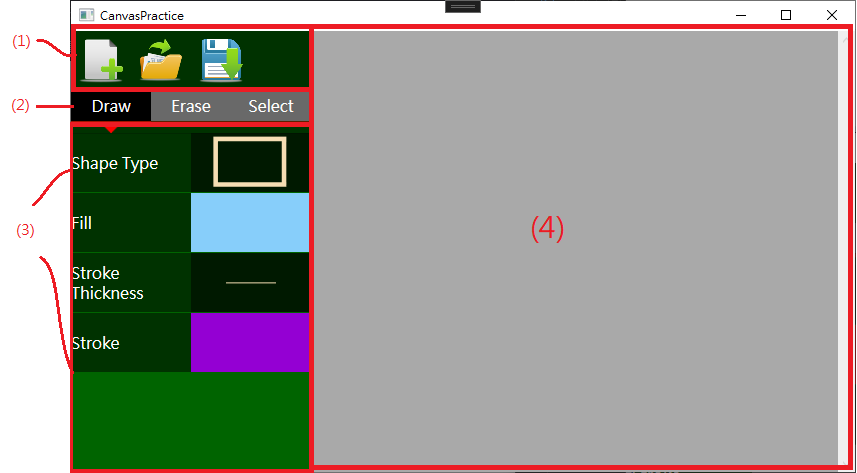
1. Describe the startup ui.

#### View – xaml.cs

1. Create/remove UIs and setup
   1. Properties
   2. Data binding
   3. Invoke command setting
   4. Animation
   5. Popup IsOpen status handling

## Application Layout

### Application Layout Overview



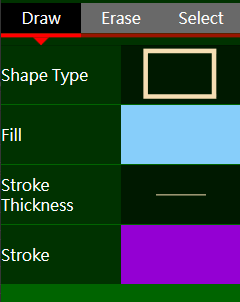
1. 檔案選單 (File IO Menu): For file saving/ loading manipulation
2. 主功能選單 (UX Mode Menu): Graph editing options
3. 屬性選單 (Graph Attribute Menu): Graph appearance/ attribute setting
4. 畫布 (Canvas)

### File I/O Menu



1. New File: Clear all objects on the canvas and start a new one
2. Open File: Open the \*.cvs file to load the previous setting
3. Save File: Save all the graphs attribute on the canvas to disk.

### UX Mode Menu



1. Draw Mode: Drag or click on canvas to generate a new graph.
   * Rectangle, Ellipse: Mouse Press, drag and release
   * Triangle: Click three vertices on canvas to create
2. Erase Mode: Select one graph and then press Delete key to remove it.
3. Select Mode:
   * Drag on graph to translate
   * Select one graph and change the appearance

### Graph Attribute Menu

|  |  |  |  |
| --- | --- | --- | --- |
|  | Draw | Erase | Select |
| Shape Type | Choose the preferred attributes before draw | N/A | N/A |
| Fill | Change the appearance of drawn graph |
| Stroke Thickness |
| Stroke |