**Code Document for WYSIWYG Configurator Application**

**1. Overview**

The WYSIWYG Configurator application allows users to drag-and-drop UI elements (Rectangle, Ellipse, and TextBlock) into a design area. It also provides property editing functionality, including modifying dimensions, color, text, font size, and opacity.

The codebase is organized into the following files:

| **File** | **Purpose** |
| --- | --- |
| MainWindow.xaml | Defines the UI layout for Toolbox, Design Area, and Property Box. |
| MainWindow.xaml.cs | Contains the logic for drag-and-drop, element selection, and property editing. |
| ToolboxContainer.cs | Manages the Toolbox functionality for drag-and-drop. |
| DesignAreaContainer.cs | Handles element addition, selection, and overlay logic in the Design Area. |
| PropertyContainer.cs | Dynamically generates property fields for the selected element. |

**2. Classes and Their Descriptions**

**MainWindow (MainWindow.xaml.cs)**

The entry point and core logic for the application.

| **Member** | **Type** | **Description** |
| --- | --- | --- |
| \_selectedElement | UIElement | Tracks the currently selected element. |
| \_selectionOverlay | Rectangle | A visual overlay used to highlight the selected element. |
| Toolbox\_PreviewMouseDown | Method | Handles drag-and-drop initiation from the Toolbox. |
| DesignArea\_Drop | Method | Handles adding elements to the Design Area when dropped. |
| SelectElement | Method | Selects an element, displays the overlay, and updates the Property Box. |
| UpdatePropertyBox | Method | Dynamically populates the Property Box based on the selected element's type. |
| AddPropertyField | Method | Adds a property field (e.g., Width, Height) for editing. |
| AddColorProperty | Method | Adds a color property field for shapes and text. |
| AddOpacityProperty | Method | Adds an opacity property field with validation (0-1). |

**ToolboxContainer (ToolboxContainer.cs)**

Manages drag-and-drop functionality for Toolbox items.

| **Member** | **Type** | **Description** |
| --- | --- | --- |
| \_toolbox | ListBox | The ListBox containing Toolbox items (Rectangle, Ellipse, TextBlock). |
| Toolbox\_PreviewMouseDown | Method | Detects mouse events and initiates drag-and-drop. |
| ItemDragged | EventHandler | An event triggered when an item is dragged. |

**DesignAreaContainer (DesignAreaContainer.cs)**

Handles the Design Area where elements are dropped and manipulated.

| **Member** | **Type** | **Description** |
| --- | --- | --- |
| \_designArea | Canvas | The workspace where elements are dropped and displayed. |
| \_selectionOverlay | Rectangle | An overlay rectangle to highlight the selected element. |
| AddElement | Method | Adds a new UI element (Rectangle, Ellipse, TextBlock) to the Design Area. |
| SelectElement | Method | Updates the overlay to indicate the currently selected element. |

**PropertyContainer (PropertyContainer.cs)**

Generates dynamic property fields for the selected element.

| **Member** | **Type** | **Description** |
| --- | --- | --- |
| \_propertyBox | StackPanel | The container where property fields are displayed dynamically. |
| DisplayProperties | Method | Dynamically generates property fields based on the selected element's type. |
| AddPropertyField | Method | Adds a property field (e.g., Width, Height) for editing. |
| AddColorPicker | Method | Adds a color picker property for shapes and text. |

**3. Properties and Editable Fields**

The following properties are dynamically generated and can be edited in the Property Box:

| **Element Type** | **Editable Properties** |
| --- | --- |
| Rectangle | Width, Height, Fill Color, Opacity |
| Ellipse | Width, Height, Fill Color, Opacity |
| TextBlock | Text, Font Size, Foreground Color, Opacity |

**4. Code Flow Overview**

1. **Toolbox Drag-and-Drop**:
   * The Toolbox\_PreviewMouseDown method detects a drag event for the selected Toolbox item.
   * The item is dragged to the Design Area and added via the DesignArea\_Drop method.
2. **Element Addition**:
   * When an item is dropped, it is created dynamically (Rectangle, Ellipse, TextBlock) and added to the Design Area.
3. **Element Selection**:
   * Clicking on an element in the Design Area triggers DesignArea\_MouseLeftButtonDown.
   * The SelectElement method displays the selection overlay and updates the Property Box.
4. **Dynamic Property Editing**:
   * UpdatePropertyBox generates fields such as Width, Height, Color, and Opacity dynamically based on the selected element.
   * User input is validated, and changes are applied immediately.

**5. Known Issues Fixed**

* The first item from the Toolbox no longer drops automatically.
* The selection overlay now always appears on top using Canvas.SetZIndex.
* The TextBlock now properly displays the selection overlay.

**6. Extensibility**

The current implementation allows easy extension:

* New UI elements can be added by modifying DesignArea\_Drop.
* Additional properties can be supported by extending UpdatePropertyBox and PropertyContainer.

**Next Steps**

Please review the updated code and executable. The application is now stable and resolves all previously reported issues. Let me know if further refinements are needed or if you'd like to discuss additional features.