Simple CSS Button Generator

Usage

Simple CSS Button Generator is a GUI application that can generate a CSS(Cascading Style Sheets) code or export it to a HTML file including a CSS code within it. Users can create new button with a simple style or choose from the provided template button by clicking on white cards with button within them. Users can scroll down the screen to see all templates.

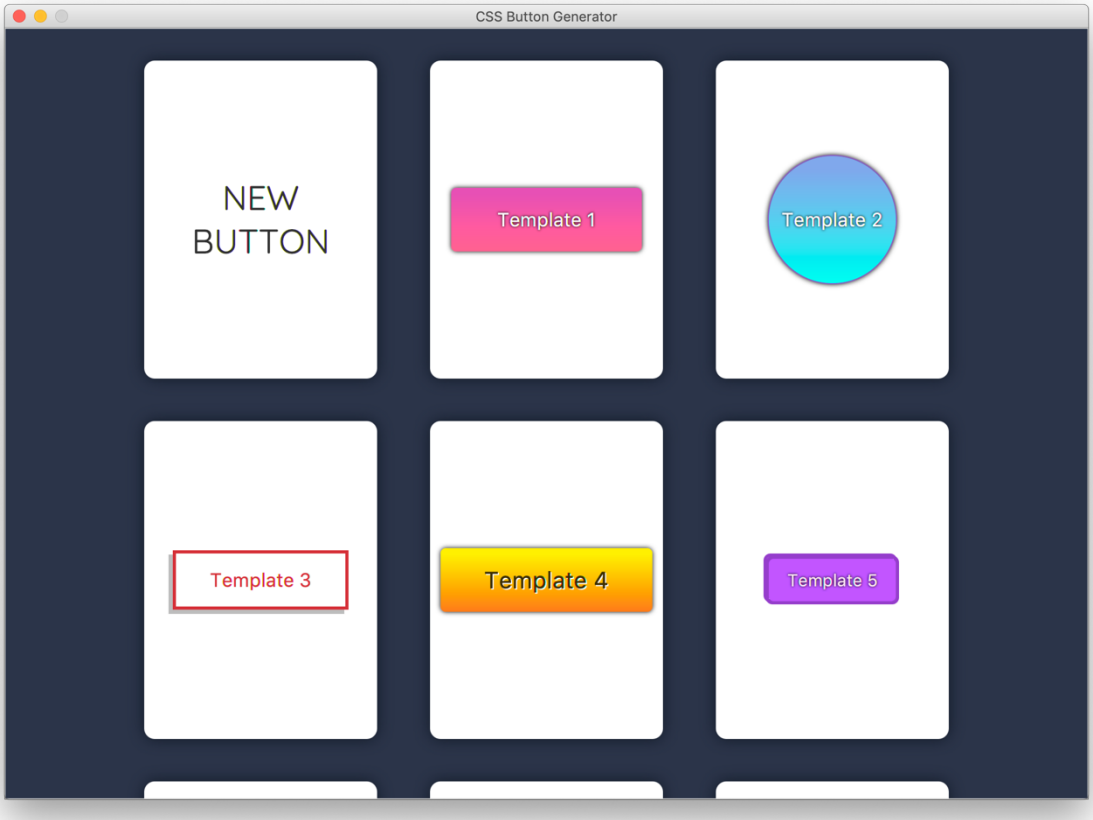

Figure 1 : The initial GUI of the application

6031055321
Weerayut Thinchanlong

Whether users click on a ‘NEW BUTTON’ card or the template card, the screen will be changed to a button generator showed in the Figure 2 below. Users can adjust the button’s properties , which are label, size, background, border, text shadow, and box shadow, on the scrollable left side pane of the application.

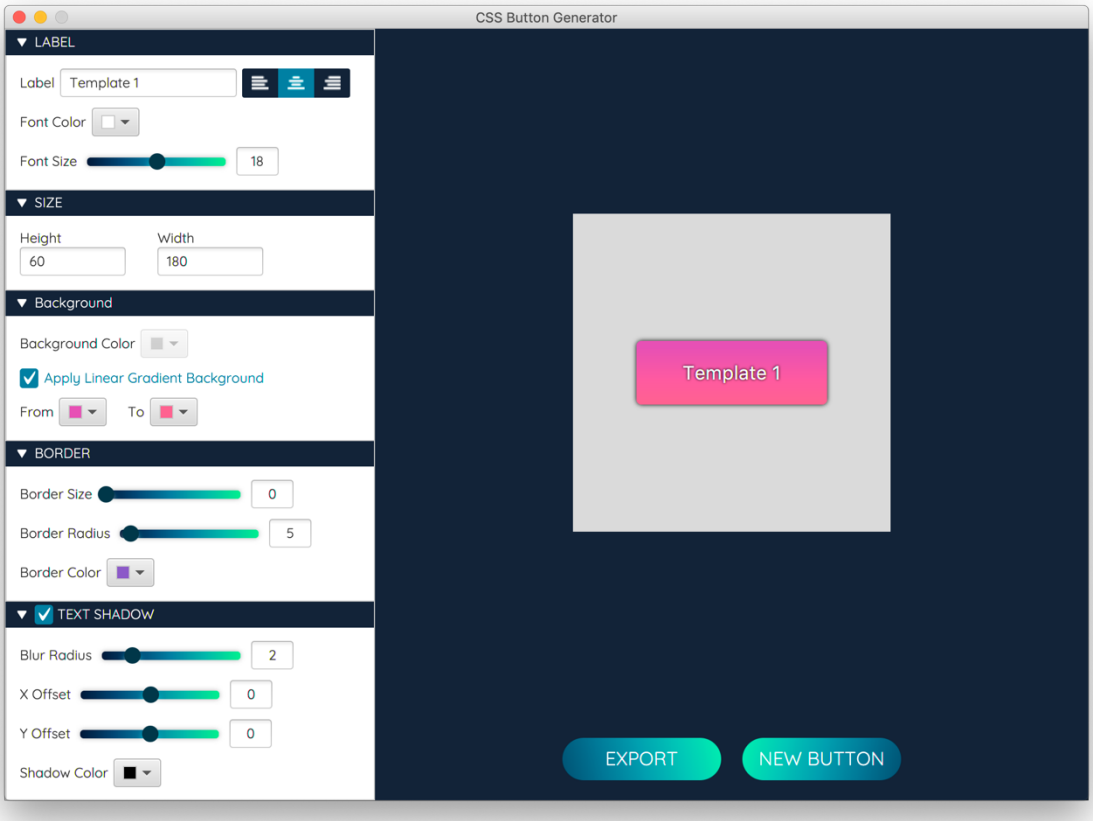

Figure 2 : The GUI of the application when users clicks on the cards

In the label panel, users can change the text showing in the button and set the position of the text, change the text color, and change the text size, noted that the minimum and maximum text size are 0pixel and 36pixels respectively.

In the size panel, users can change the width or height of the button, noted that minimum and maximum value of both width and height are 0pixel and 200pixels respectively.

In the background panel, users can set the background to solid color or top-down linear gradient color by ticking the ‘Apply Linear Gradient Background’ checkbox.

In the border panel, users can change the outer border size, color, or radius of the button.

In the text shadow panel, users can apply this effect by ticking the checkbox preceding the panel title and change the text shadow blur radius, whose minimum and maximum value are 0pixel and 10pixels respectively, the text shadow offsets, whose minimum and maximum value are -5pixels and 5pixels respectively, or the text shadow color.

In the box shadow panel, users can apply this effect by ticking the checkbox preceding the panel title and change the button shadow blur radius, whose minimum and maximum value are 0pixel and 15pixels respectively, the button shadow offsets, whose minimum and maximum value are -10pixels and 10pixels respectively, or the button shadow color.

When the button's properties are changed, the button on the right-side pane is changed simultaneously. In other words, the right side pane shows the button adjusted by users.

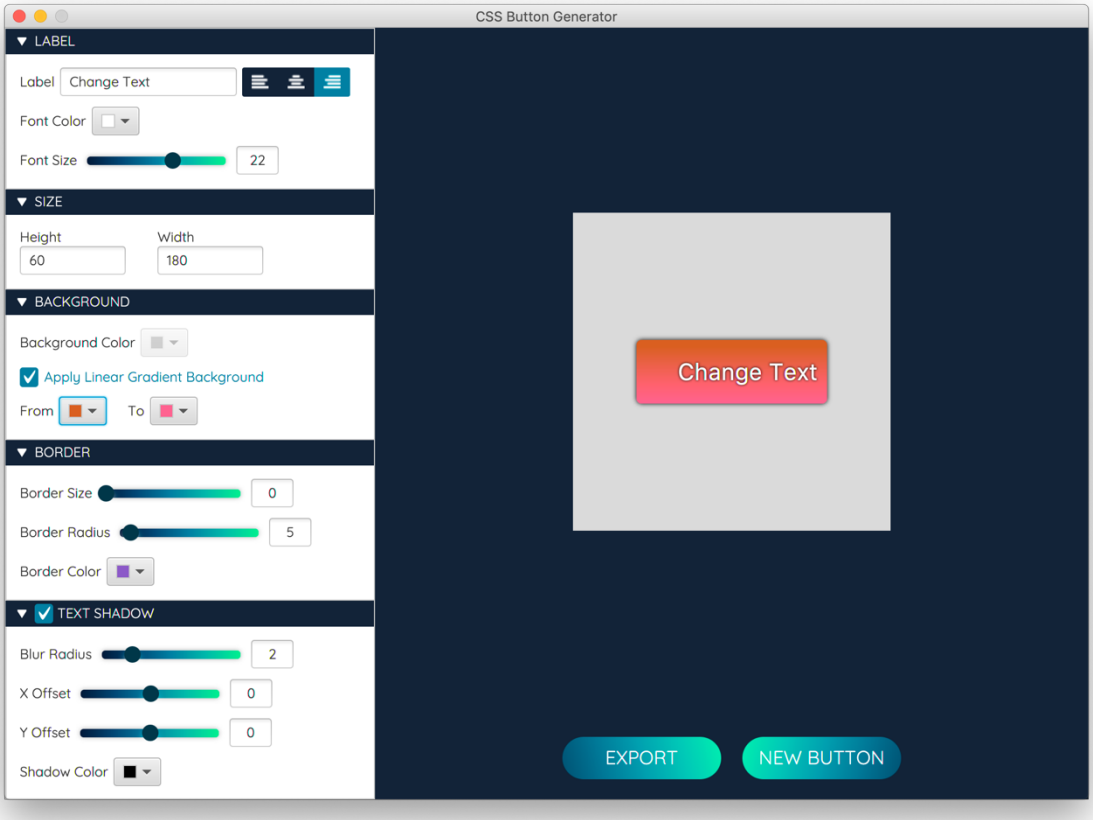

Figure 3 : The GUI of the application after users adjusting some of the button's properties

When users are done adjusting the button, users can generate the CSS code or export it to HTML file by clicking on **EXPORT** button but if users click on the **NEW BUTTON** button the screen will be changed to the initial GUI of the application showed in the Figure 1.

After clicking on the **EXPORT** button, the export dialog box will appear. To hide it just click the Close button.

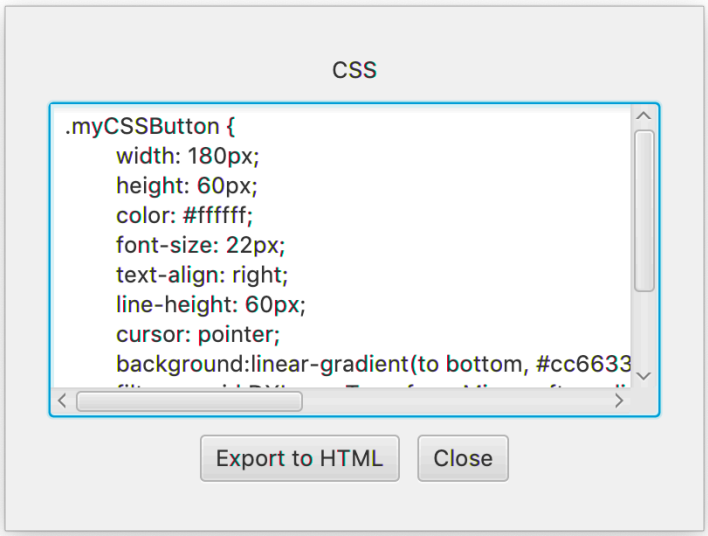

Figure 4 : The export dialog box

From Figure 4 above, if users click on the **Export to HTML** button, users can save HTML file in the preferred directory.

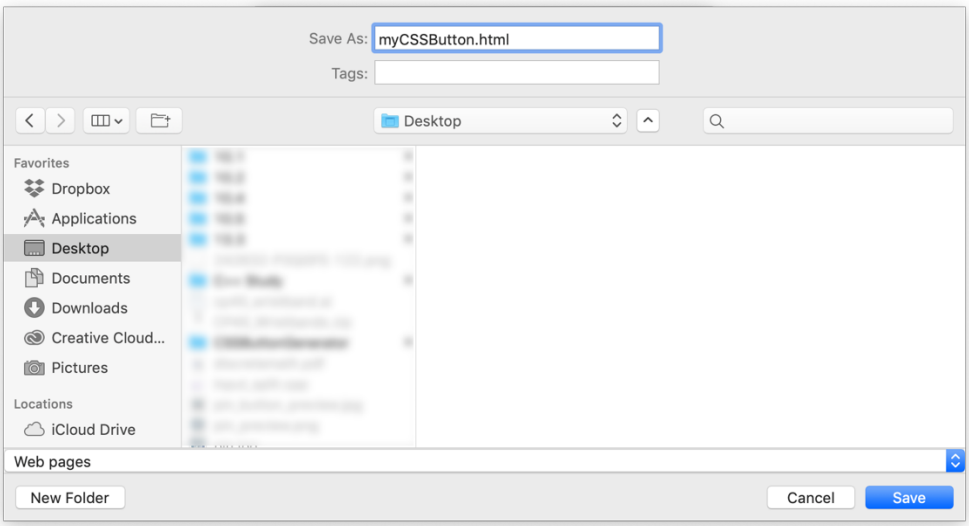

Figure 5 : Directory Chooser appears after clicking on the **Export to HTML** button

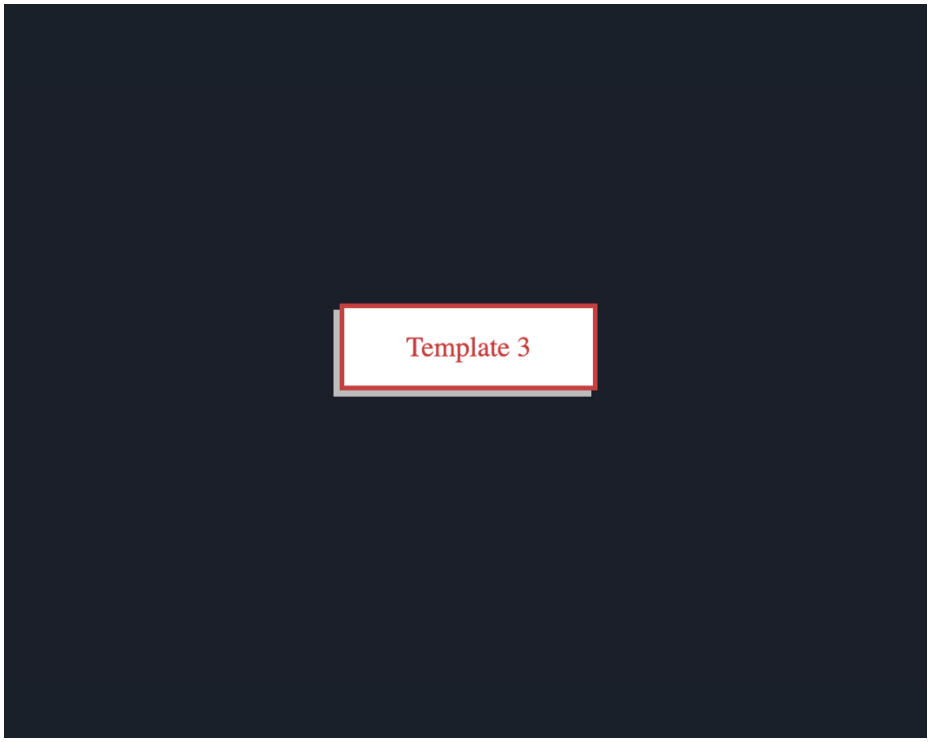

Figure 6 : What it looks like after opening the HTML file , saved from the application, in browser

Implementation Details

The diagram of this program is illustrated in Figure 7 below.

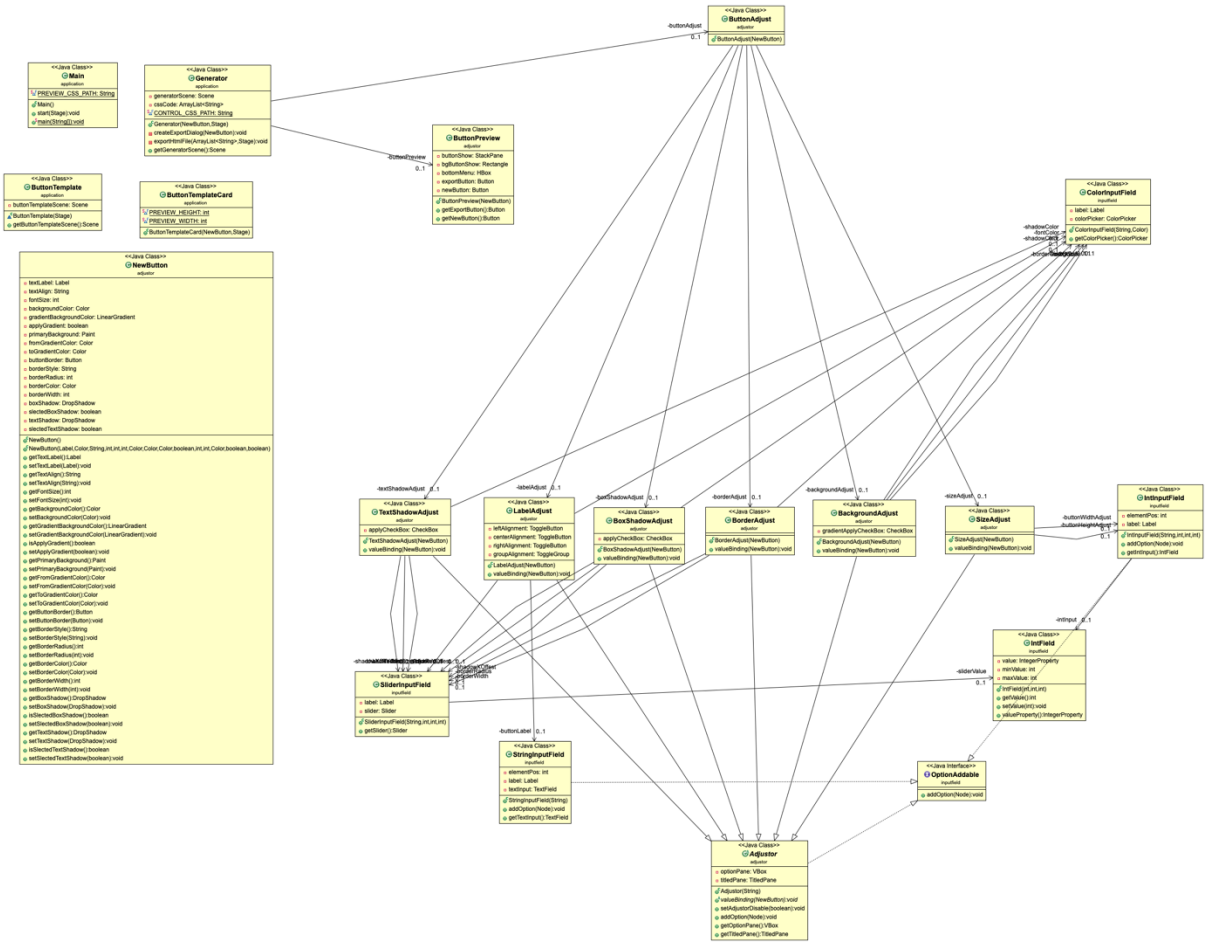

Figure 7 : The UML diagram of the program.

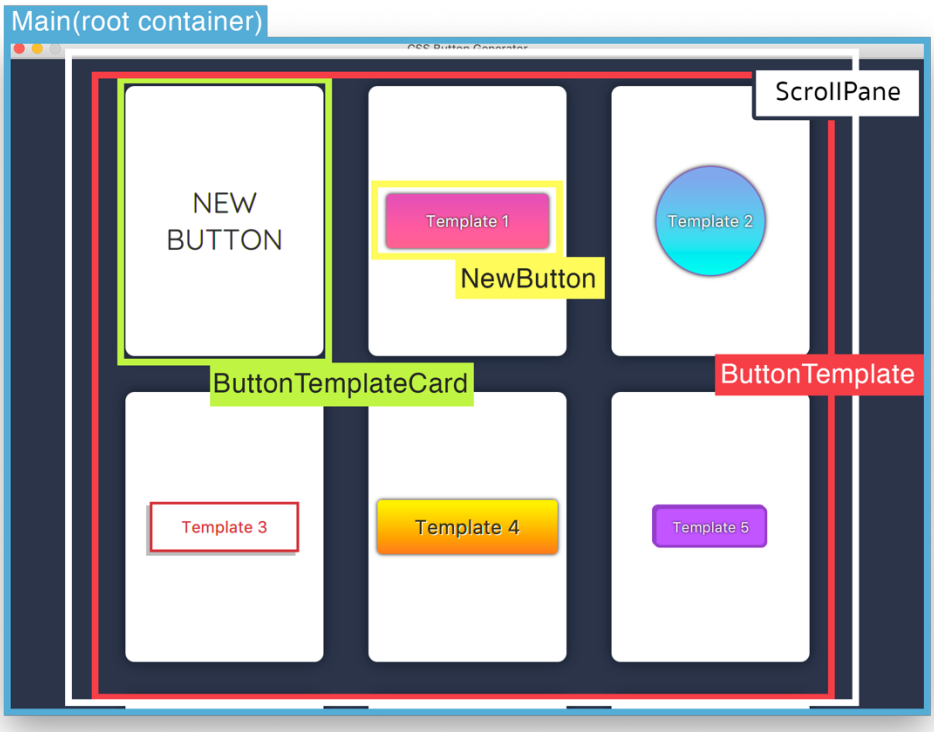

Figure 8 : The outline showing the core structure of the button template scene.

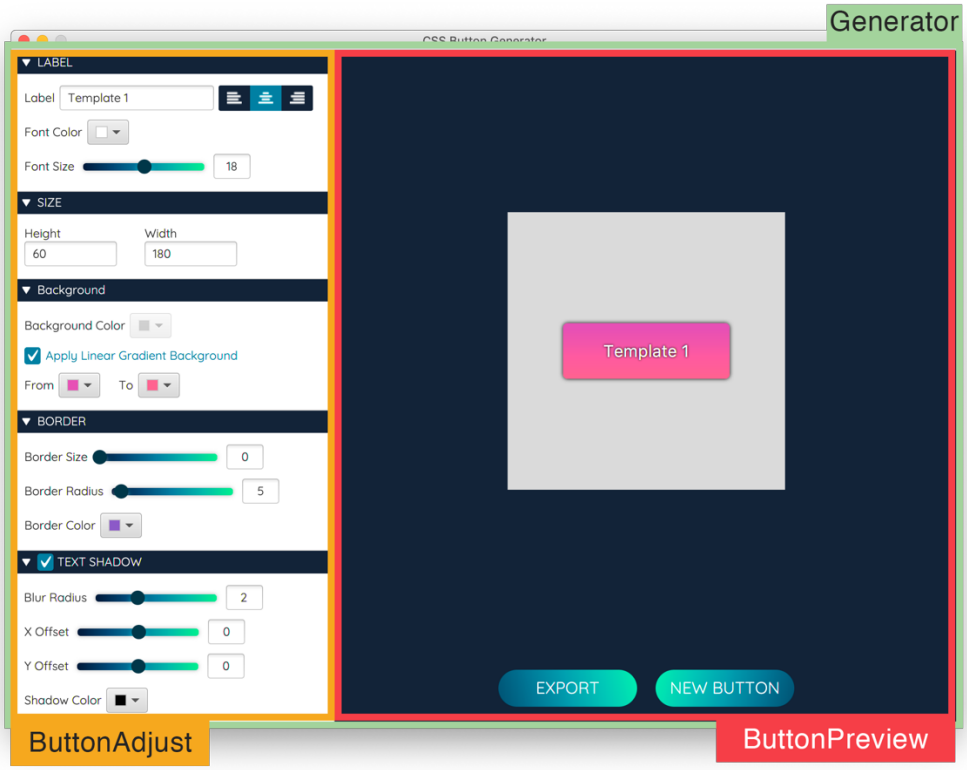

Figure 9 : The outline showing the core structure of the button generator scene.

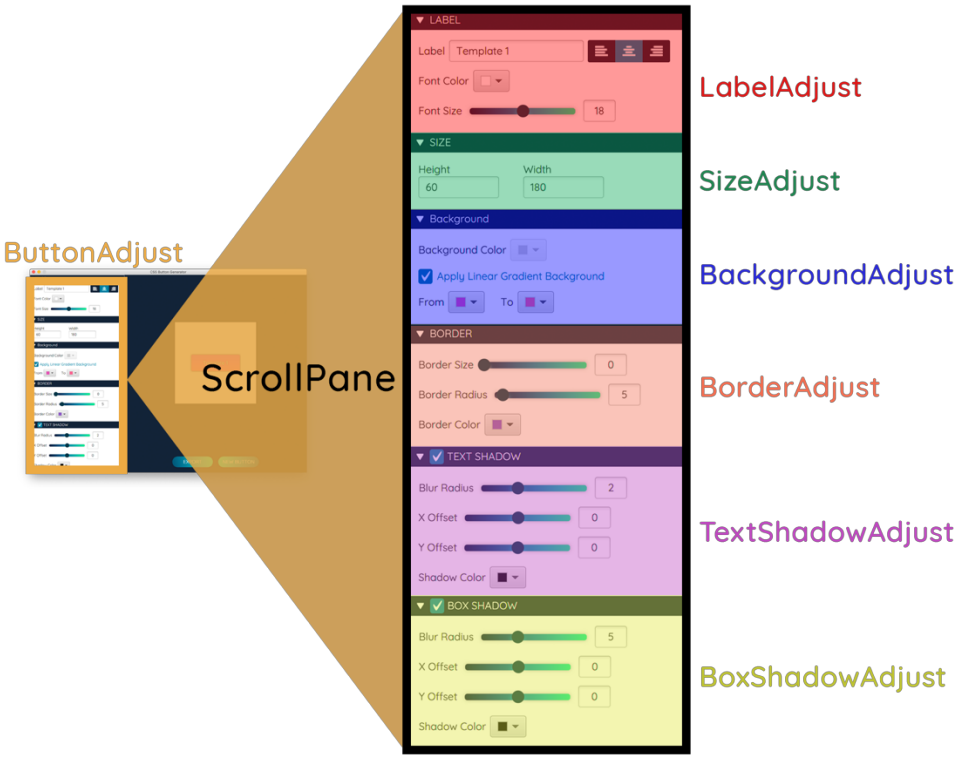

Figure 10 : The core structure and the sub-adjustor of ButtonAdjust.

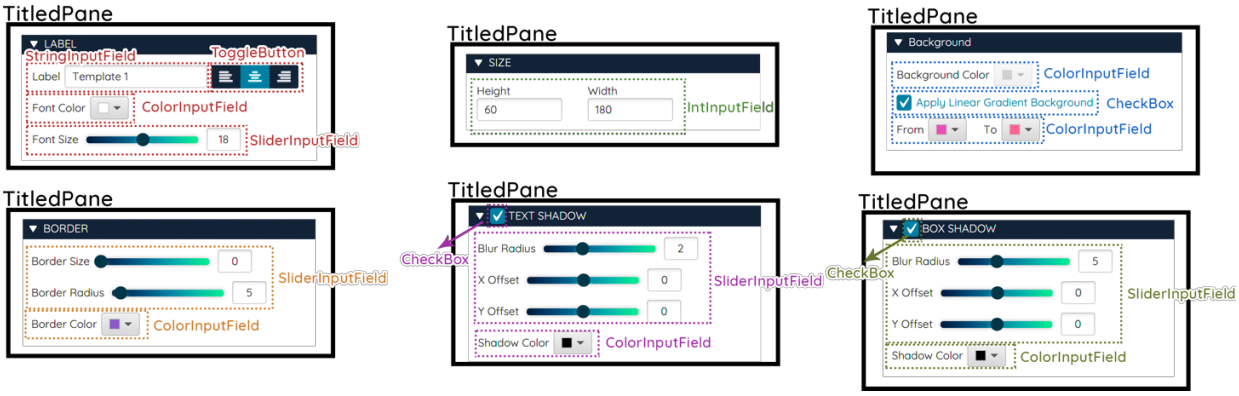

Figure 11 : The outline showing adjustors' components and how each of them, from left to right, top to bottom, LabelAdjust, SizeAdjust, BackgroundAdjust, BorderAdjust, TextShadowAdjust and BoxShadowAdjust., is constructed.

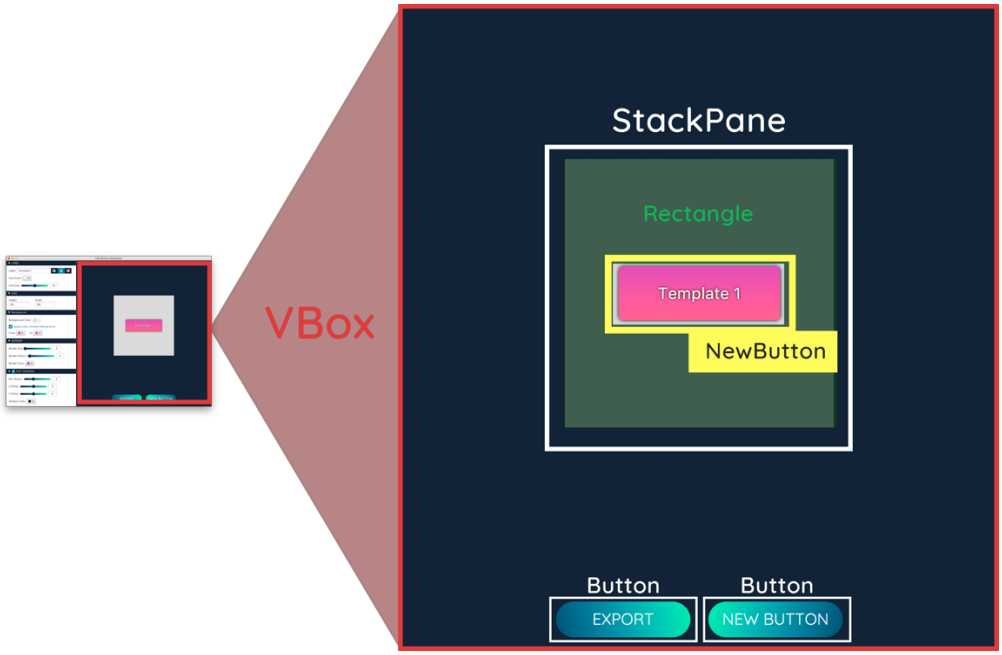

Figure 12 : The outline showing the core structure of button preview scene.

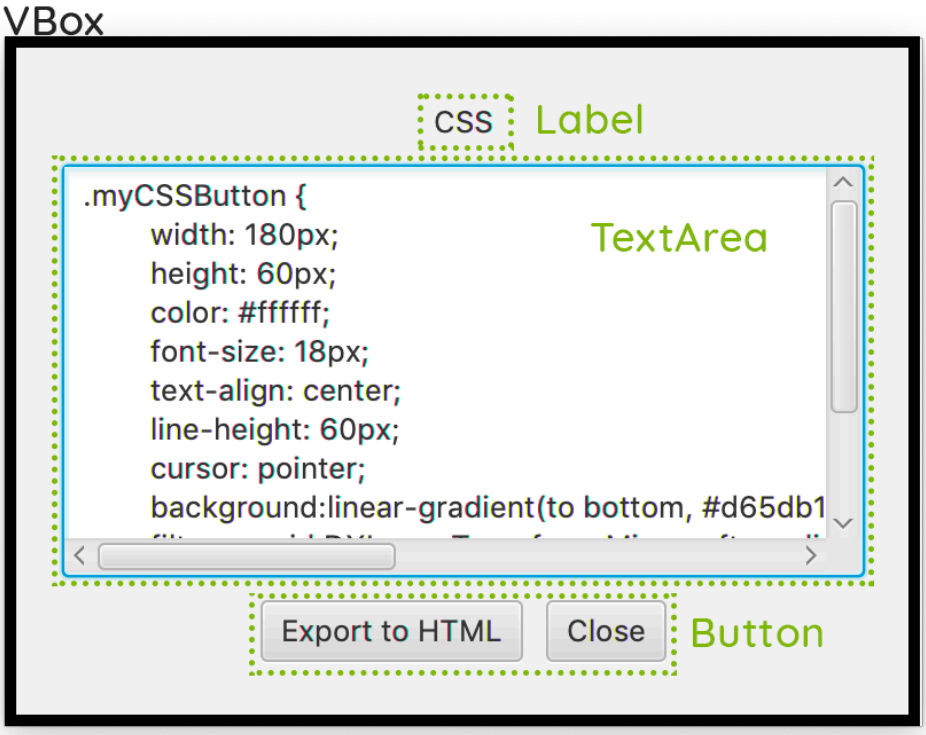

Figure 13: The outline showing the core structure and components of the export dialog box.

Figure 14 : The outline showing how the StringInputField is constructed.

Figure 14 : The outline showing how the IntInputField is constructed.

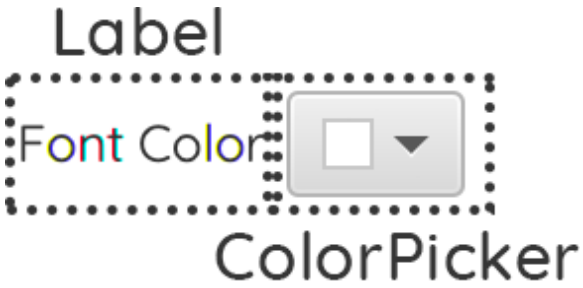

Figure 15 : The outline showing how ColorInputField is constructed.

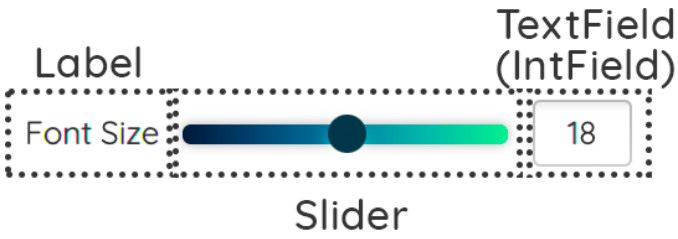

Figure 16 : The outline showing how the SliderInputField is constructed.

** Noted that Access Modifier Notations can be listed below*

+ (public)

 # (protected)

- (private)

static will be underlined.

final will be red.

abstract will be italic.

2.1 Class *adjustor.Adjustor* implements OptionAddable

2.1.1 Field

- VBox optionPane	VBox for containg components in adjustors.
- TitledPane titledPane	TitledPane to make adjustor panel.

2.1.2 Constructor

+ Adjustor(String title)	Initializes the Adjustor - Instantiate optionPane as a VBox with insets padding of 5. - Instantiate titledPane as a TitledPane -Set its title to String title. -Set its width to 350; -Set its content to optionPane

2.1.3 Method

+ *void valueBinding(NewButton button)*	Bind the values of the button, which is instance of Class NewButton.

2.2 Class adjustor.BackgroundAdjust extends Adjustor

2.2.1 Field

- CheckBox gradientApplyCheckBox	CheckBox determines whether primaryBackground is gradient or solid color.
- ColorInputField bgColor	ColorInputField used for adjusting button's background color.
- ColorInputField fromColor	ColorInputField used for adjusting the first gradient color.
- ColorInputField toColor	ColorInputField used for adjusting the second gradient color.

2.2.2 Constructor

+ BackgroundAdjust (NewButton button)	Initializes the BackgroundAdjust - Instantiate gradientApplyCheckBox as a CheckBox with "Apply Linear Gradient Background" title.

	- Instantiate bgColor as a ColorInputField with “Background Color” title and set its current color to button’s background color. - Instantiate fromColor as a ColorInputField with “From” title and set its current color to first gradient color. - Instantiate toColor as a ColorInputField with “To” title and set its current color to second gradient color.

2.2.3 Method

+ void valueBinding(NewButton button)	Bind the values of the button, which is instance of Class NewButton, using ChangeListener to set the background to its current color or gradient color.

2.3 Class adjustor.BorderAdjust extends Adjustor

2.3.1 Field

- SliderInputField borderWidth	SliderInputField for adjusting button’s border width.
- SliderInputField borderRadius	SliderInputField for adjusting button’s border radius.
- ColorInputField borderColor	ColorInputField for determining the button’s border color.

2.3.2 Constructor

+ BorderAdjust (NewButton button)	Initializes the BorderAdjust - Instantiate borderWidth as a SliderInputField with “Border Size” title -set its current width to button’s border width. -set its minimum value and maximum value to 0 and 5 respectively. - Instantiate borderRadius as a SliderInputField with “Border Radius” title

	-set its current radius to button's border radius. -set its minimum value and maximum value to 0 and 100 respectively. - Instantiate borderColor as a ColorInputField with "Border Color" title and set its current color to button's border color. -Call valueBinding method and add all the input fields to optionPane.

2.3.3 Method

+ void valueBinding(NewButton button)	Bind the values of the button, which is instance of Class NewButton, using ChangeListener to set the button's border properties to their current properties.

2.4 Class adjustor.BoxShadowAdjust extends Adjustor

2.4.1 Field

- CheckBox applyCheckBox	CheckBox determines whether button applies a box shadow effect or not.
- SliderInputField blurRadius	SliderInputField for adjusting button's box shadow radius.
- SliderInputField shadowXOffest	SliderInputField for adjusting the horizontal position of button's box shadow.
- SliderInputField shadowYOffest	SliderInputField for adjusting the vertical position of button's box shadow.
- ColorInputField shadowColor	ColorInputField for determining the button's box shadow color.

2.4.2 Constructor

+ BoxShadowAdjust (NewButton button)	Initializes the BoxShadowAdjust - Instantiate applyCheckBox as a CheckBox with no title.

	- If the button applies the box shadow effect, then set applyCheckBox to be selected and set the adjustor to not be disable. - Otherwise, set applyCheckBox to be unselected and set the adjustor disable. - Instantiate blurRadius as a SliderInputField with “Blur Radius” title -set its current blur radius to the button’s box shadow radius. -set its minimum value and maximum value to 0 and 15 respectively. - Instantiate shadowXOffest as a SliderInputField with “X Offset” title -set its current horizontal position to the button’s box shadow horizontal position. -set its minimum value and maximum value to -10 and 10 respectively. - Instantiate shadowYOffest as a SliderInputField with “Y Offset” title -set its current vertical position to the button’s box shadow vertical position. -set its minimum value and maximum value to -10 and 10 respectively. - Instantiate shadowColor as a ColorInputField with “Shadow Color” title and set its current color to button’s box shadow color. -Set titledPane graphic with a applyCheckBox -Call valueBinding method

2.4.3 Method

+ void valueBinding(NewButton button)	-Bind the values of the button, which is instance of Class NewButton, with applyCheckBox, blurRadius, shadowXOffest, and shadowYOffest using Bind Property to set the button's box shadow properties to their current properties. -Add ChangeListener to applyCheckBox to determine whether it's selected or not. - If selected, applies the box shadow effect to the button, set applyCheckBox to be selected ,and set the adjustor to not be disable. - Otherwise, applies no effect to the button, set applyCheckBox to be unselected and set the adjustor disable.

2.5 Class adjustor.LabelAdjust extends Adjustor

2.5.1 Field

- StringInputField buttonLabel	StringInputField for changing the text on the button.
- SliderInputField fontSize	SliderInputField for adjusting button's font size.
- ColorInputField fontColor	ColorInputField for determining the button's font color.
- ToggleButton leftAlignment	ToggleButton for set the text alignment of the button to left.
- ToggleButton centerAlignment	ToggleButton for set the text alignment of the button to center.
- ToggleButton rightAlignment	ToggleButton for set the text alignment of the button to right.
- ToggleGroup groupAlignment	ToggleGroup for grouping alignment toggle buttons together.

2.5.2 Constructor

+ LabelAdjust (NewButton button)	Initializes the LabelAdjust - Instantiate buttonLabel as a StringInputField with “Label” title -set its current text to the button’s label. - Instantiate fontSize as a SliderInputField with “Font Size” title -set its current font size to the button’s font size. -set its minimum value and maximum value to 0 and 36 respectively. - Instantiate fontColor as a ColorInputField with “Font Color” title and set its current color to button’s font color. - Instantiate leftAlignment, centerAlignment, rightAlignment as a ToggleButton - Set their graphic to left-alignment, center-alignment and right-alignment icon respectively. - Set their cursor to hand icon. - Set their toggle group to groupAlignment. - Add mouse event filter to them to prevent untoggled. - Instantiate groupAlignment as a ToggleGroup - Instantiate alignmentAdjust as a HBox and add the toggle buttons to it. -Call valueBinding method and add all the input fields to optionPane.

2.5.3 Method

+ void valueBinding(NewButton button)	-Bind the values of the button, which is instance of Class NewButton, with buttonLabel and fontColor using Bind Property to set the button's label and font color to its current text and color. -Add ChangeListener to fontSize and groupAlignment to set the font and alignment of the button's label.

2.6 Class adjustor.SizeAdjust extends Adjustor

2.6.1 Field

- IntInputField buttonHeightAdjust	IntInputField for adjusting the button's height
- IntInputField buttonWidthAdjust	IntInputField for adjusting the button's width

2.6.2 Constructor

+ SizeAdjust (NewButton button)	Initializes the SizeAdjust - Instantiate buttonHeightAdjust as a IntInputField with "Height" title -set its current value to the button's height. -set its minimum value and maximum value to 0 and 200 respectively. - Instantiate buttonWidthAdjust as a IntInputField with "Width" title -set its current value to the button's width. -set its minimum value and maximum value to 0 and 200 respectively. - Instantiate groupOption as a HBox and add the buttonHeightAdjust and buttonWidthAdjust to it.

	-Call valueBinding method and add groupOption to optionPane.

2.6.3 Method

+ void valueBinding(NewButton button)	-Bind the values of the button, which is instance of Class NewButton, with buttonHeightAdjust and buttonWidthAdjust using Bind Property to set the button's width and height.

2.7 Class adjustor.TextShadowAdjust extends Adjustor

2.7.1 Field

- CheckBox applyCheckBox	CheckBox determines whether button applies a text shadow effect or not.
- SliderInputField blurRadius	SliderInputField for adjusting button's text shadow radius.
- SliderInputField shadowXOffest	SliderInputField for adjusting the horizontal position of button's text shadow.
- SliderInputField shadowYOffest	SliderInputField for adjusting the vertical position of button's text shadow.
- ColorInputField shadowColor	ColorInputField for determining the button's text shadow color.

2.7.2 Constructor

+ TextShadowAdjust (NewButton button)	Initializes the TextShadowAdjust - Instantiate applyCheckBox as a CheckBox with no title. - If the button applies the text shadow effect, then set applyCheckBox to be selected and set the adjustor to not be disable. - Otherwise, set applyCheckBox to be unselected and set the adjustor disable. - Instantiate blurRadius as a SliderInputField with "Blur Radius" title

	-set its current blur radius to the button's text shadow radius. -set its minimum value and maximum value to 0 and 15 respectively. - Instantiate shadowXOffest as a SliderInputField with "X Offset" title -set its current horizontal position to the button's text shadow horizontal position. -set its minimum value and maximum value to -10 and 10 respectively. - Instantiate shadowYOffest as a SliderInputField with "Y Offset" title -set its current vertical position to the button's text shadow vertical position. -set its minimum value and maximum value to -10 and 10 respectively. - Instantiate shadowColor as a ColorInputField with "Shadow Color" title and set its current color to button's text shadow color. -Set titledPane graphic with a applyCheckBox -Call valueBinding method

2.7.3 Method

+ void valueBinding(NewButton button)	-Bind the values of the button, which is instance of Class NewButton, with applyCheckBox, blurRadius, shadowXOffest, and shadowYOffest using Bind Property to set the button's text shadow properties to their current properties.

	-Add ChangeListener to applyCheckBox to determine whether it's selected or not. - If selected, applies the text shadow effect to the button, set applyCheckBox to be selected ,and set the adjustor to not be disable. - Otherwise, applies no effect to the button's label, set applyCheckBox to be unselected and set the adjustor disable.

2.8 Class adjustor.ButtonAdjust extends ScrollPane

2.8.1 Field

- LabelAdjust labelAdjust	LabelAdjust for adjusting button's label.
- SizeAdjust sizeAdjust	sizeAdjust for adjusting button's size.
- BackgroundAdjust backgroundAdjust	backgroundAdjust for adjusting button's background.
- BorderAdjust borderAdjust	borderAdjust for adjusting button's border.
- TextShadowAdjust textShadowAdjust	textShadowAdjust for adjusting button's text shadow effect.
- BoxShadowAdjust boxShadowAdjust	boxShadowAdjust for adjusting button's box shadow effect.

2.8.2 Constructor

+ ButtonAdjust (NewButton button)	Initializes the ButtonAdjust - Set its size to 350, 350. - Set its scroll bar transparent. - Instantiate labelAdjust as a LabelAdjust. - Instantiate sizeAdjust as a SizeAdjust. - Instantiate backgroundAdjust as a BackgroundAdjust. - Instantiate borderAdjust as a BorderAdjust. - Instantiate textShadowAdjust as a TextShadowAdjust. - Instantiate boxShadowAdjust as a BoxShadowAdjust.

	- Instantiate adjustCollection as a VBox and add all adjustors to it. - Set its content to adjustCollection.

2.9 Class adjustor.ButtonPreview extends VBox

2.9.1 Field

- StackPane buttonShow	StackPane for stacking edited button and a background that lies under the button.
- Rectangle bgButtonShow	bgButtonShow for lying under the button.
- HBox bottomMenu	bottomMenu for grouping and arrange exportButton and newButton horizontally.
- Button exportButton	Button for calling an export dialog box.
- Button newButton	Button for going back to initial GUI of the application.

2.9.2 Constructor

+ ButtonPreview (NewButton button)	Initializes the ButtonPreview - Set its alignment to center. - Set its bottom insets to 20. - Add button-show style class to it. - Instantiate buttonShow as a StackPane. - Set its height to 680. - Add bgButtonShow, buttonBorder(from button), and button respectively. - Instantiate bgButtonShow as a Rectangle. - Set its size to 300, 300. - Set its background color to #DADADA. - Instantiate bottomMenu as a HBox. - Set its alignment to center. - Set its padding to 20. - Add exportButton, and newButton respectively. - Instantiate exportButton

	and newButton as a Button. - Set their text to “EXPORT” and “NEW BUTTON” respectively. - Set their size to 150, 70. - Set their style class to ”export-button” and “new-button” respectively. - Set their cursor to hand icon.

2.9.3 Method

+ Getters	Getters for only exportButton and newButton

2.10 Class adjustor.NewButton extends Button

2.10.1 Field

- Label textLabel	Label on the button.
- String textAlign	Determine what text alignment it is.
- int fontSize	Determine button’s label font size.
- Color backgroundColor	Determine button’s solid background color.
- LinearGradient gradientBackgroundColor	Determine button’s gradient background.
- boolean applyGradient	Determine if the button applies the linear gradient background or not.
- Paint primaryBackground	Determine what is the main background for button.
- Color fromGradientColor	Determine the first gradient color.
- Color toGradientColor	Determine the second gradient color.
- Button buttonBorder	To be the fake button’s border because the button cannot implement outer border.
- String borderStyle	Determine the button’s border style.
- int borderRadius	Determine the button’s border radius.
- Color borderColor	Determine the button’s border color.
- int borderWidth	Determine the button’s border width.
- DropShadow boxShadow	Determine the button’s box shadow effect.
- boolean slectedBoxShadow	Determine if the button applies the box shadow effect or not.
- DropShadow textShadow	Determine the button’s text shadow effect.
- boolean slectedTextShadow	Determine if the button applies the text shadow effect or not.

2.10.2 Constructor

+ NewButton ()	Initializes the NewButton - Set its width size to 180 and height to 60. -Set its background color to primaryColor. -Set its border radius to borderRadius. -Set its graphic to textLabel. -Initialize all fields. - Set textAlign to “center”. - Set textLabel text to “BUTTON”. - Set primaryBackground to backgroundColor. - Set all the properties of buttonBorder like button. - Set borderRadius to 5. - Set borderWidth to 2. - Set borderColor to Color.BLACK. - Set slectedBoxShadow and selectedTextShadow to false.
+ NewButton(Label textLabel, Color textColor, String textAlign, int fontSize, int buttonWidth, int buttonHeight, Color backgroundColor, Color fromGradientColor, Color toGradientColor, boolean applyGradient, int borderWitdh, int borderRadius, Color borderColor, boolean slectedBoxShadow, boolean slectedTextShadow)	Initializes the NewButton -Initialize all fields. - Set textLabel text to textLabel. - Apply the box shadow effect to the button if slectedBoxShadow is true. - Apply the text shadow effect to the button if slectedTextShadow is true.

2.10.3 Method

Getters & Setters	Every fields has its getter and settes.

2.11 Class inputfield.ColorInputField extends GridPane

2.11.1 Field

- Label label	Label for the input.
- ColorPicker colorPicker	ColorPicker for choose color.

2.11.2 Constructor

+ ColorInputField(String labelText, Color initialColor)	Initializes the ColorInputField - Set its Vgap and Hgap to 5. - Add label to 0, 0 position and colorPicker to 1, 0 position to it. - Instantiate label as a Label with “labelText”. - Instantiate colorPicker as a ColorPicker - Set its current color to initialColor. - Set its width to 45.

2.11.3 Method

+ Getters	Getters only for colorPicker.

2.12 Class inputfield.IntInputField extends GridPane

2.12.1 Field

- Label label	Label for the input.
- IntField intInput	IntField receive only integer values.

2.12.2 Constructor

+ IntInputField (String labelText, int minValue, int maxValue, int initialValue)	Initializes the IntInputField - Set its Vgap to 1 and Hgap to 5. - Add label to 0, 0 position and intInput to 1, 0 position to it. - Instantiate label as a Label with “labelText”. - Instantiate intInput as a IntField with required parameters.

2.12.3 Method

Getters	Getters only for intInput.

2.13 Class inputfield.SliderInputField extends GridPane

2.13.1 Field

- Label label	Label for the input.
- Slider slider	Slider for adjusting values.
- IntField sliderValue	IntField receive only integer values.

2.13.2 Constructor

+ SliderInputField(String labelName,int initialValue, int minSliderValue, int maxSliderValue)	Initializes the SliderInputField - Set its Vgap and Hgap to 5. - Add label to 0, 0 position, slider to 1, 0 position and sliderValue to 2, 0 position to it. - Instantiate label as a Label with “labelName”. - Instantiate slider as a Slider with required parameters. - Add ChangeListener to round the value up. - Set its min, max and current value with provided parameters. - Instantiate sliderValue as a IntField with required parameters. - Bind the value of sliderValue with the slider’s value. - Set its width to 40. - Set its alignment to BASELINE_CENTER.

2.13.3 Method

Getters	Getters only for slider.

2.14 Class inputfield.StringInputField extends GridPane implements OptionAddable

2.14.1 Field

- Label label	Label for the input.
- TextField textInput	Receive text
- int elementPos	To store the next position for component to be added.

2.14.2 Constructor

+ StringInputField(String labelText)	Initializes the StringInputField - Set its Vgap and Hgap to 5. - Add label to 0, 0 position and intInput to 1, 0 position to it. - Instantiate label as a Label with “labelText”. - Instantiate textInput as a TextField. -Set the value of elementPos to 2.

2.14.3 Method

+ addOption(Node option)	Add the component next to the StringInputField
Getters	Getters only for textInput.

2.15 Class inputfield.IntField extends TextField

2.15.1 Field

- IntegerProperty value	To be gotten and set.
- int minValue	Receive text
- int maxValue	To store the next position for component to be added.

2.15.2 Constructor

+ IntField(int minValue, int maxValue, int initialValue)	Initializes the IntField - Initializes minValue and maxValue - Instantiate value as an SimpleIntegerProperty with initialValue.

	- Implement ChangeListener to check that the input is in the correct range. - Implement addEventFilter to prevent the alphabetic inputs.

2.15.3 Method

+ addOption(Node option)	Add the component next to the StringInputField
+ int getValue	Return an int value of the IntegerProperty.
+ void setValue(int newValue)	Set a value of the IntegerProperty.
+ IntegerProperty valueProperty	Return IntegerProperty.

2.16 Interface OptionAddable

2.16.1 Method

+ addOption(Node option)	Add the component to the Pane.

2.17 Class application.ButtonTemplateCard extends stackPane

2.17.1 Field

int PREVIEW_HEIGHT = 300	Card's height
int PREVIEW_WIDTH = 220	Card's width

2.17.2 Constructor

+ ButtonTemplateCard(NewButton button, Stage primaryStage)	Initializes the ButtonTemplateCard - Set its size to PREVIEW_HEIGHT and PREVIEW_WIDTH. - Set its alignment to center. - Add "previewCard" style class to it. - Implement setOnMouseEntered to play the fade transition animation. - Implement setOnMouseExited to set the overlay and selectText disable. - Implement setOnMouseClicked to set the scene to generator with given button. - Add the final variables to it.

	- Instantiate selectText as a Label with "SELECT THIS TEMPLATE" text. - Instantiate overlay as a Region. - Set its size to card's size. - Set it invisible. - Instantiate overlayFade, textFade as a FadeTransition with 0.5s and 0.4s duration and bind them with overlay and selectText respectively.

2.18 Class application.ButtonTemplate extends ScrollPane

2.18.1 Field

- Scene buttonTemplateScene	Scene to be gotten.

2.18.2 Constructor

+ ButtonTemplate(Stage primaryStage)	Initializes the ButtonTemplate - Set its to fit the width. - Set its scroll bar to invisible. - Add all the ButtonTemplateCard instances to it. - Set its content to gridPane - Instantiate gridPane as a GridPane - Set its Vgap to 40 and Hgap to 50. - Sets its top and bottom padding to 30. - Add "buttonTemplatePage" style class to it. - Instantiate button1 – button 9 as a NewButton with different style. - Instantiate btn1preview – btn9preview as a ButtonTemplateCard.

	- Instantiate buttonTemplateScene as a Scene with instance of this class to be its root. - Add a stylesheet from Main.PREVIEW_CSS_PATH to it.

2.18.3 Method

Getters	Getters only for buttonTemplateScene.

2.19 Class application.Generator extends HBox

2.19.1 Field

- Scene generatorScene	Scene to be gotten.
- ButtonPreview buttonPreview	For showing being adjusted button.
- ButtonAdjust buttonAdjust	For showing the adjustors.
- ArrayList<String> cssCode	Store each line of CSS code.
String CONTROL_CSS_PATH	controlStyleSheet.css path

2.19.2 Constructor

+ Generator (NewButton button, Stage primaryStage)	Initializes the Generator - Add buttonAdjust and buttonPreview to it respectively. - Instantiate generatorScene as a Scene with instance of this class to be its root. - Add a stylesheet from CONTROL_CSS_PATH to it. - Instantiate buttonPreview as a ButtonPreview with required parameter. - Get the exportButton from this object and implement setOnAction to call createExportDialog method. - Instantiate buttonAdjust as a ButtonAdjust with required parameter. - Instantiate btnTemplate as a ButtonTemplate and implement setOnAction

	to set scene of the primaryStage to be the buttonTemplateScene from btnTemplate. - Instantiate cssCode as a ArrayList<String>

2.19.3 Method

+ void createExportDialog(NewButton button)	This method clears the cssCode and adds the new CSS code to it and create an export dialog box.
+ exportHtmlFile(ArrayList<String> cssCode, Stage dialog)	This method create File Chooser for saving html file to preferred directory.
Getters	Getters only for generatorScene.

2.20 Class application.Main extends Application

2.20.1 Field

static final String PREVIEW_CSS_PATH	previewStyleSheet.css path

2.20.2 Method

+ void start(Stage primaryStage)	The main entry point of the JavaFX application. - Load font Quicksand-Regular.ttf. - Instantiate root as a VBox - Instantiate btnTemplate as a ButtonTemplate to be the initial GUI of the application. - Instantiate btnTemplateScene as a Scene and make root(VBox) to be its root. - Set the stage's scene to be the btnTemplateScene. - Set the stage's width to 1024 and height to 750 and make it unresizeable. - Set the stage's name to CSS Button Generator.
+ void main(String[] args)	An entry point of the application.

Git project : https://github.com/enwee1998/SimpleCSSButtonGenerator