

```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.geometry.Insets?>
4 <?import javafx.scene.control.Button?>
5 <?import javafx.scene.control.CheckBox?>
6 <?import javafx.scene.control.ComboBox?>
7 <?import javafx.scene.control.Label?>
8 <?import javafx.scene.control.ListView?>
9 <?import javafx.scene.control.TextField?>
10 <?import javafx.scene.layout.ColumnConstraints?>
11 <?import javafx.scene.layout.GridPane?>
12 <?import javafx.scene.layout.HBox?>
13 <?import javafx.scene.layout.RowConstraints?>
14 <?import javafx.scene.layout.StackPane?>
15 <?import javafx.scene.layout.VBox?>
16 <?import javafx.scene.shape.Rectangle?>
17 <?import javafx.scene.text.Font?>
18
19 <GridPane alignment="TOP_CENTER" hgap="10.0" maxHeight="360.0" maxWidth=
  ="560.0" minHeight="-Infinity" minWidth="-Infinity" prefHeight="360.0"
  prefWidth="560.0" style="-fx-background-color: #fff;" vgap="20.0" xmlns
  :fx="http://javafx.com/fxml/1" xmlns="http://javafx.com/javafx/17" fx:
  controller="com.example.vphw05.HelloController">
20   <columnConstraints>
21     <ColumnConstraints hgrow="SOMETIMES" maxWidth="200.0" minWidth="10.
  0" prefWidth="200.0" />
22     <ColumnConstraints hgrow="SOMETIMES" maxWidth="120.0" minWidth="
  10.0" prefWidth="120.0" />
23     <ColumnConstraints hgrow="SOMETIMES" maxWidth="80.0" minWidth="10
  .0" prefWidth="80.0" />
24     <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0" prefWidth="100
  .0" />
25   </columnConstraints>
26   <rowConstraints>
27     <RowConstraints maxHeight="210.0" minHeight="10.0" prefHeight="210.
  0" vgrow="SOMETIMES" />
28     <RowConstraints maxHeight="110.0" minHeight="10.0" prefHeight="30.0
  " vgrow="SOMETIMES" />
29   </rowConstraints>
30   <children>
31     <VBox prefHeight="200.0" prefWidth="100.0">
32       <children>
33         <Label maxHeight="20.0" minHeight="20.0" prefHeight="20.0"
  prefWidth="210.0" text="Font:" />
34         <TextField fx:id="tfFont" editable="false" />
35         <ListView fx:id="lvFont" prefHeight="200.0" prefWidth="200.
  0" />
36       </children>
37     </VBox>

```

```

38     <VBox prefHeight="200.0" prefWidth="100.0" GridPane.columnIndex="
39     1">
40         <children>
41             <Label minHeight="20.0" prefHeight="17.0" prefWidth="121.0"
42             text="Style:" />
43             <TextField fx:id="tfStyle" editable="false" />
44             <ListView fx:id="lvStyle" prefHeight="200.0" prefWidth="200
45             .0" />
46         </children>
47     </VBox>
48     <VBox prefHeight="200.0" prefWidth="100.0" GridPane.columnIndex="
49     2">
50         <children>
51             <Label minHeight="20.0" prefHeight="17.0" prefWidth="108.0"
52             text="Size:" />
53             <TextField fx:id="tfSize" editable="false" />
54             <ListView fx:id="lvSize" prefHeight="200.0" prefWidth="200.
55             0" />
56         </children>
57     </VBox>
58     <VBox alignment="TOP_CENTER" prefHeight="200.0" prefWidth="100.0"
59     GridPane.columnIndex="3">
60         <children>
61             <Button fx:id="btnOk" alignment="CENTER" contentDisplay="
62             CENTER" mnemonicParsing="false" onAction="#close" prefHeight="25.0"
63             prefWidth="109.0" text="OK" />
64         </children>
65         <padding>
66             <Insets top="20.0" />
67         </padding>
68     </VBox>
69     <StackPane prefHeight="150.0" prefWidth="200.0" GridPane.rowIndex
70     ="1">
71         <children>
72             <VBox alignment="CENTER_LEFT" prefHeight="200.0" prefWidth
73             ="100.0" spacing="2.0" style="-fx-border-color: #d3d3d3; -fx-border-
74             radius: 5;">
75                 <children>
76                     <CheckBox fx:id="underline" mnemonicParsing="false"
77                     prefHeight="23.0" prefWidth="168.0" text="Underline" />
78                     <Label alignment="BOTTOM_LEFT" prefHeight="74.0"
79                     prefWidth="168.0" text="Color:" />
80                     <HBox alignment="CENTER_LEFT" prefHeight="25.0"
81                     prefWidth="168.0" spacing="10.0">
82                         <children>
83                             <ComboBox fx:id="chooseColor" prefHeight="25.0"
84                             prefWidth="149.0" />
85                             <VBox alignment="CENTER" prefHeight="25.0"
86                             prefWidth="16.0" style="-fx-border-color: #d3d3d3; -fx-border-radius: 3

```

```

69 ;">
70         <children>
71             <Label alignment="CENTER" minHeight="15.
0" minWidth="20.0" prefHeight="15.0" prefWidth="11.0" text="A"
textAlignment="CENTER">
72                 <font>
73                     <Font name="System Bold" size="12.
0" />
74                 </font>
75             </Label>
76             <Rectangle fx:id="colorIdent" arcHeight
="5.0" arcWidth="5.0" fill="#ff451f" height="5.0" stroke="BLACK"
strokeType="INSIDE" strokeWidth="0.0" width="20.0" />
77         </children>
78         <padding>
79             <Insets bottom="2.0" left="2.0" right="2
.0" top="2.0" />
80         </padding>
81     </VBox>
82 </children>
83 </HBox>
84 </children>
85 <padding>
86     <Insets bottom="15.0" left="15.0" right="15.0" top="
15.0" />
87 </padding>
88 </VBox>
89 <Label alignment="CENTER" prefHeight="17.0" prefWidth="41.
0" style="-fx-background-color: #fff;" text="Effects" textAlignment="
CENTER" textFill="#0a11e8" translateX="-66.0" translateY="-56.0" />
90 </children>
91 </StackPane>
92 <StackPane prefHeight="150.0" prefWidth="200.0" GridPane.
columnIndex="1" GridPane.columnSpan="2" GridPane.rowIndex="1">
93     <children>
94         <VBox alignment="CENTER" prefHeight="200.0" prefWidth="100
.0" style="-fx-border-color: #d3d3d3; -fx-border-radius: 5;">
95             <children>
96                 <Label fx:id="sampleTxt" alignment="CENTER"
prefHeight="201.0" prefWidth="355.0" style="-fx-border-color: #d3d3d3
;" text="AaBbYyZz" textAlignment="CENTER" />
97             </children>
98             <padding>
99                 <Insets bottom="15.0" left="15.0" right="15.0" top="
15.0" />
100             </padding>
101         </VBox>
102         <Label alignment="CENTER" prefHeight="17.0" prefWidth="47.
0" style="-fx-background-color: #fff;" text="Sample" textAlignment="

```

```
102 CENTER" textFill="#0a11e8" translateX="-69.0" translateY="-56.0" />
103     </children>
104 </StackPane>
105 </children>
106 <padding>
107     <Insets bottom="10.0" left="10.0" right="10.0" top="10.0" />
108 </padding>
109 </GridPane>
110
```

```
1 package com.example.vphw05;
2
3 import javafx.application.Platform;
4 import javafx.beans.property.*;
5 import javafx.beans.value.ChangeListener;
6 import javafx.beans.value.ObservableValue;
7 import javafx.collections.FXCollections;
8 import javafx.collections.ObservableList;
9 import javafx.event.ActionEvent;
10 import javafx.fxml.FXML;
11 import javafx.scene.control.*;
12 import javafx.scene.paint.Color;
13 import javafx.scene.shape.Rectangle;
14
15 import javafx.scene.text.Font;
16 import javafx.scene.text.FontPosture;
17 import javafx.scene.text.FontWeight;
18
19 public class HelloController {
20
21     @FXML
22     private Button btnOk;
23
24     @FXML
25     private ComboBox<String> chooseColor;
26
27     @FXML
28     private Rectangle colorIdent;
29
30     @FXML
31     private ListView<String> lvFont;
32
33     @FXML
34     private ListView<Integer> lvSize;
35
36     @FXML
37     private ListView<String> lvStyle;
38
39     @FXML
40     private TextField tfFont;
41
42     @FXML
43     private TextField tfSize;
44
45     @FXML
46     private TextField tfStyle;
47
48     @FXML
49     private CheckBox underline;
```

```

50
51     @FXML
52     private Label sampleTxt;
53
54     ObservableList<String> fonts = FXCollections.observableArrayList(
Font.getFamilies());
55     ObservableList<String> styles = FXCollections.observableArrayList("
Regular", "Italic", "Bold", "Bold Italic");
56     ObservableList<Integer> sizes = FXCollections.observableArrayList(8
, 10, 12, 14, 18, 24, 36);
57     ObservableList<String> colors = FXCollections.observableArrayList("
Black", "Green", "Blue", "Yellow", "Magenta", "Cyan", "White", "Gray",
"Light Gray", "Dark Gray", "Pink", "Orange");
58     ObservableList<Color> colorList = FXCollections.observableArrayList
(Color.BLACK, Color.GREEN, Color.BLUE, Color.YELLOW, Color.MAGENTA,
Color.CYAN, Color.WHITE, Color.GRAY, Color.LIGHTGRAY, Color.DARKGRAY,
Color.PINK, Color.ORANGE);
59
60     @FXML
61     void initialize(){
62         lvFont.getSelectionModel().setSelectionMode(SelectionMode.
SINGLE);
63         lvFont.getItems().addAll(fonts);
64         lvFont.getSelectionModel().select(0);
65         tfFont.setText(fonts.get(0));
66         lvFont.getSelectionModel().selectedItemProperty().addListener(
new ChangeListener<String>() {
67             @Override
68             public void changed(ObservableValue<? extends String>
observableValue, String s, String t1) {
69                 tfFont.setText(t1.trim());
70             }
71         });
72
73         lvStyle.getSelectionModel().setSelectionMode(SelectionMode.
SINGLE);
74         lvStyle.getItems().addAll(styles);
75         lvStyle.getSelectionModel().select(0);
76         tfStyle.setText(styles.get(0));
77         lvStyle.getSelectionModel().selectedItemProperty().addListener(
new ChangeListener<String>() {
78             @Override
79             public void changed(ObservableValue<? extends String>
observableValue, String s, String t1) {
80                 tfStyle.setText(t1);
81             }
82         });
83
84         lvSize.getSelectionModel().setSelectionMode(SelectionMode.

```

```

84 SINGLE);
85         lvSize.getItems().addAll(sizes);
86         lvSize.getSelectionModel().select(2);
87         tfSize.setText(sizes.get(2).toString());
88         lvSize.getSelectionModel().selectedItemProperty().addListener(
new ChangeListener<Integer>() {
89             @Override
90             public void changed(ObservableValue<? extends Integer>
observableValue, Integer s, Integer t1) {
91                 tfSize.setText(t1.toString());
92             }
93         });
94
95         chooseColor.getItems().addAll(colors);
96         chooseColor.getSelectionModel().select(0);
97         colorIdent.setFill(Color.BLACK);
98         chooseColor.getSelectionModel().selectedItemProperty().
addListener(new ChangeListener<String>() {
99             @Override
100            public void changed(ObservableValue<? extends String>
observableValue, String s, String t1) {
101                IntegerProperty idx = new SimpleIntegerProperty(colors
.indexOf(t1));
102                colorIdent.setFill(colorList.get(idx.getValue()));
103                toChange();
104            }
105        });
106
107        tfFont.textProperty().addListener(new ChangeListener<String
>() {
108            @Override
109            public void changed(ObservableValue<? extends String>
observableValue, String s, String t1) {
110                toChange();
111            }
112        });
113
114        tfStyle.textProperty().addListener(new ChangeListener<String
>() {
115            @Override
116            public void changed(ObservableValue<? extends String>
observableValue, String s, String t1) {
117                toChange();
118            }
119        });
120
121        tfSize.textProperty().addListener(new ChangeListener<String
>() {
122            @Override

```

```

123         public void changed(ObservableValue<? extends String>
observableValue, String s, String t1) {
124             toChange();
125         }
126     });
127
128     underline.selectedProperty().addListener(new ChangeListener<
Boolean>() {
129         @Override
130         public void changed(ObservableValue<? extends Boolean>
observableValue, Boolean aBoolean, Boolean t1) {
131             sampleTxt.setUnderline(t1);
132         }
133     });
134 }
135
136 void toChange() {
137     sampleTxt.setTextFill(colorIdent.getFill());
138
139     StringProperty fontName = new SimpleStringProperty(tfFont.
getText());
140     StringProperty fontStyle = new SimpleStringProperty(tfStyle.
getText());
141     DoubleProperty fontSize = new SimpleDoubleProperty(Double.
parseDouble(tfSize.getText()));
142     FontWeight fontWeight;
143     FontPosture fontPosture;
144
145     switch (fontStyle.getValue().toUpperCase()){
146         case "BOLD":{
147             fontWeight = FontWeight.BOLD;
148             fontPosture = FontPosture.REGULAR;
149             break;
150         }
151         case "ITALIC":{
152             fontWeight = FontWeight.NORMAL;
153             fontPosture = FontPosture.ITALIC;
154             break;
155         }
156         case "BOLD ITALIC":{
157             fontWeight = FontWeight.BOLD;
158             fontPosture = FontPosture.ITALIC;
159             break;
160         }
161         default:{
162             fontWeight = FontWeight.NORMAL;
163             fontPosture = FontPosture.REGULAR;
164             break;
165         }

```



```
166         }
167
168         sampleTxt.setFont(Font.font(fontName.getValue(), fontWeight,
fontPosture, fontSize.getValue()));
169     }
170
171     @FXML
172     void close(ActionEvent event){
173         Platform.exit();
174     }
175 }
176
```

```
private Label sampleTxt;
```

```
ObservableList<String> fonts = FXCollections.observableArrayList();  
ObservableList<String> styles = FXCollections.observableArrayList();  
ObservableList<Integer> sizes = FXCollections.observableArrayList();  
ObservableList<String> colors = FXCollections.observableArrayList();  
ObservableList<Color> colorList = FXCollections.observableArrayList();
```

```
@FXML
```

```
void initialize(){
```

```
    lvFont.getSelectionModel().setSelectionMode(SelectionMode.SINGLE);  
    lvFont.getItems().addAll(fonts);  
    lvFont.getSelectionModel().select(0);  
    tfFont.setText(fonts.get(0));  
    lvFont.getSelectionModel().selectedItemProperty().addListener((obs, oldVal, newVal) -> {  
        if (newVal != null) {  
            tfFont.setText(newVal);  
        }  
    });  
}
```

```
@Override
```

```
public void changed(ObservableValue<String> obs, String oldValue, String newValue) {  
    tfFont.setText(t1.trim());  
}
```

