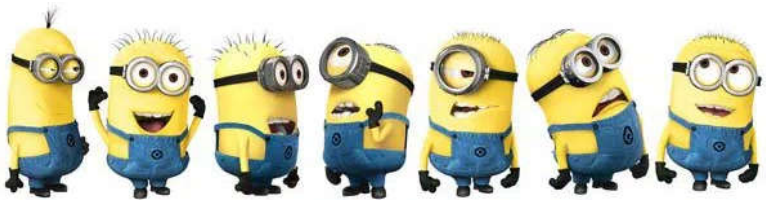
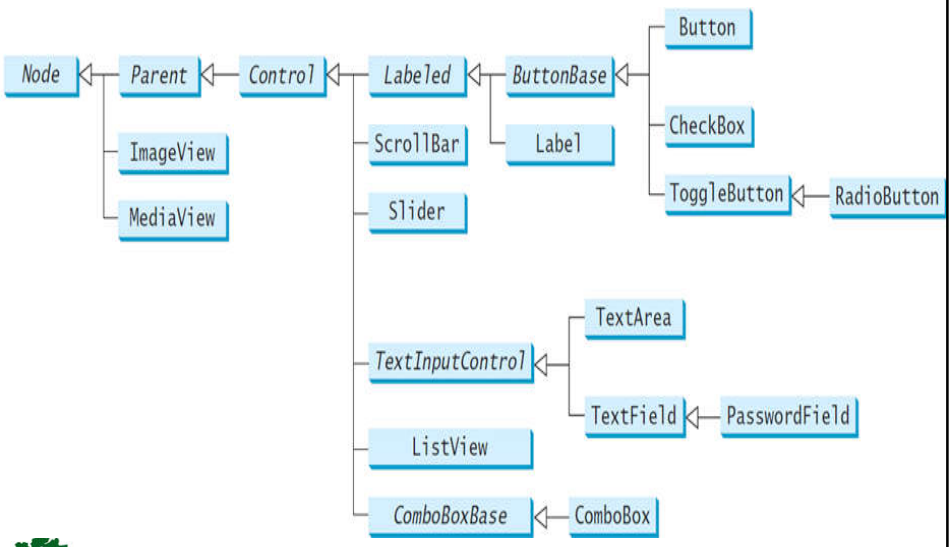
BIRZEIT UNIVERSITY

JavaFX UI Controls



By: Mamoun Nawahdah (Ph.D.)
2018

Frequently Used UI Controls



2

Labeled

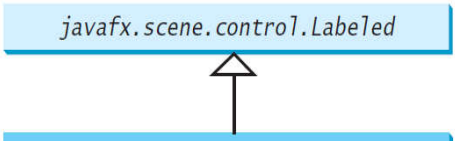
- A ***label*** is a display area for a short text, a node, or both.
- It is often used to label other controls (usually text fields).
- **Label**s and **Button**s share many common properties. These common properties are defined in the **Labeled** class.

javafx.scene.control.Labeled	
-alignment: ObjectProperty<Pos>	Specifies the alignment of the text and node in the labeled.
-contentDisplay: ObjectProperty<ContentDisplay>	Specifies the position of the node relative to the text using the constants TOP, BOTTOM, LEFT, and RIGHT defined in ContentDisplay.
-graphic: ObjectProperty<Node>	A graphic for the labeled.
-graphicTextGap: DoubleProperty	The gap between the graphic and the text.
-textFill: ObjectProperty<Paint>	The paint used to fill the text.
-text: StringProperty	A text for the labeled.
-underline: BooleanProperty	Whether text should be underlined.
-wrapText: BooleanProperty	Whether text should be wrapped if the text exceeds the width.

3

Label

- The **Label** class defines labels.

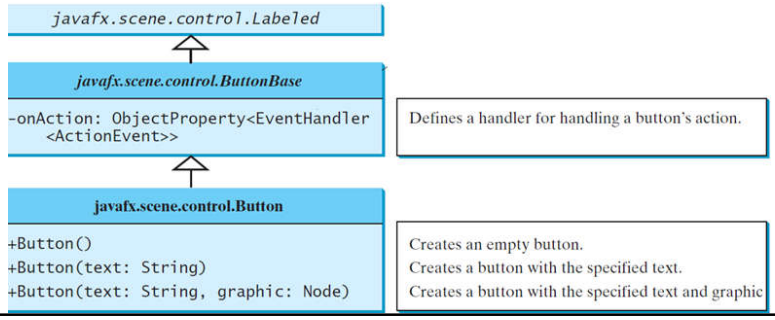


javafx.scene.control.Label	
+Label()	Creates an empty label.
+Label(text: String)	Creates a label with the specified text.
+Label(text: String, graphic: Node)	Creates a label with the specified text and graphic.

4

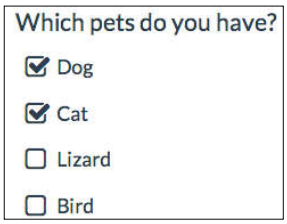
ButtonBase and Button

- A ***Button*** is a UI component that triggers an action event when clicked.
- JavaFX provides regular buttons, toggle buttons, check box buttons, and radio buttons.
- The common features of these buttons are defined in **ButtonBase** and **Labeled** classes.

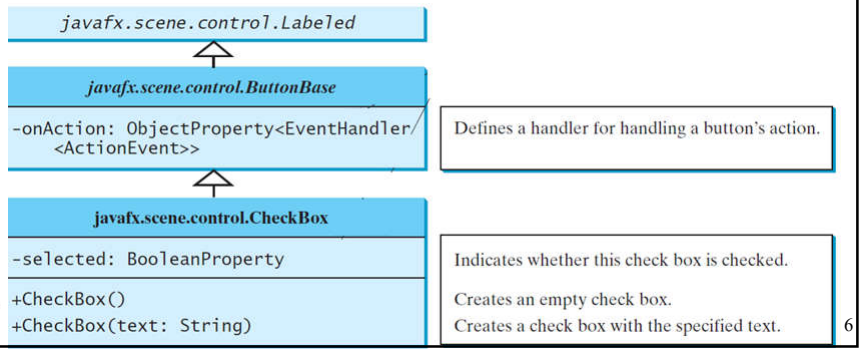


5

CheckBox

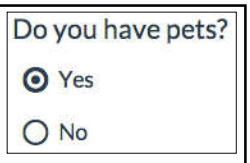


- A **CheckBox** is used for the user to make a selection (**0 or more**).
- **CheckBox** inherits all the properties such as **onAction**, **text**, **graphic**, **alignment**, **graphicTextGap**, **textFill**, **contentDisplay** from **ButtonBase** and **Labeled**.

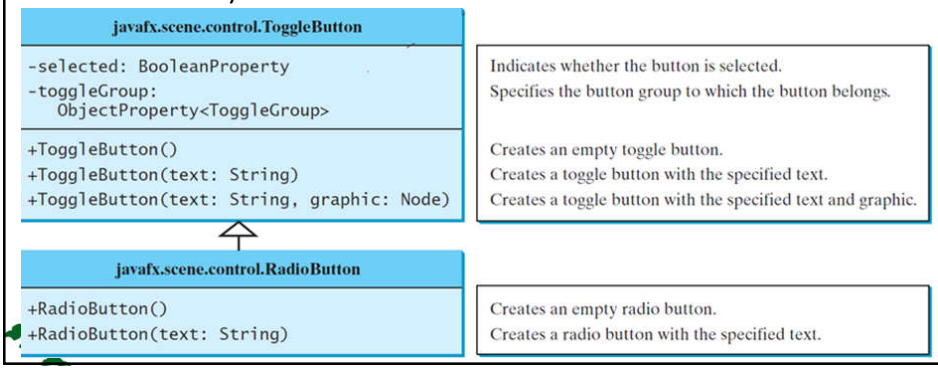


6

RadioButton



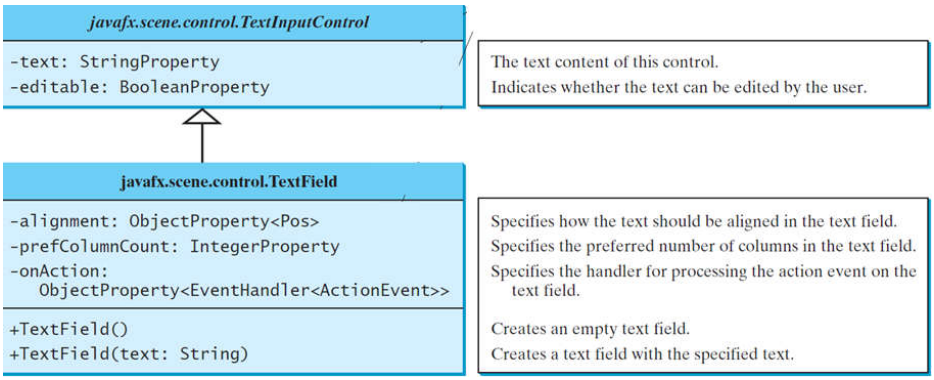
- Radio buttons, also known as *option buttons*, enable you to choose a **single** item from a group of choices.
- In appearance radio buttons resemble check boxes, but check boxes display a square that is either checked or blank, whereas radio buttons display a circle that is either filled (if selected) or blank (if not selected).



7

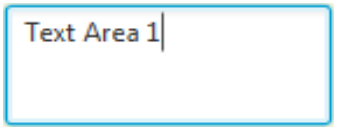
TextField

- A text field can be used to enter or display a string.
- **TextField** is a subclass of **TextInputControl**.

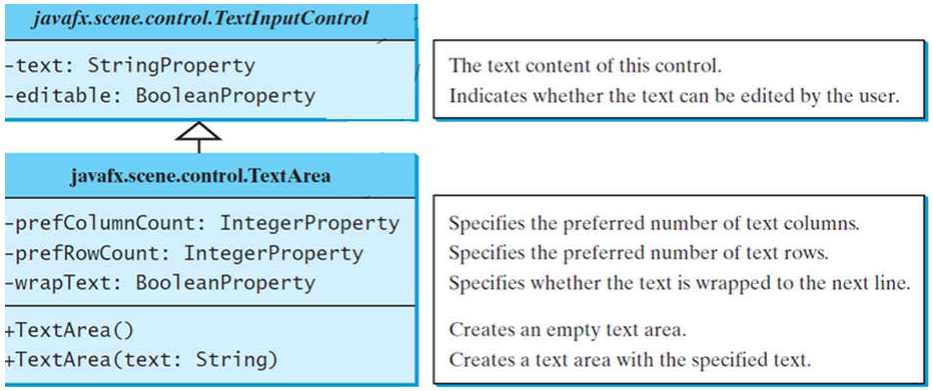


8

TextArea

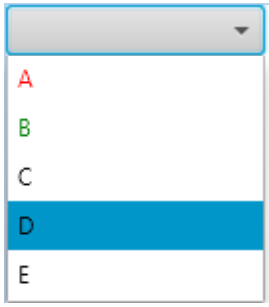


❖ A **TextArea** enables the user to enter *multiple* lines of text.

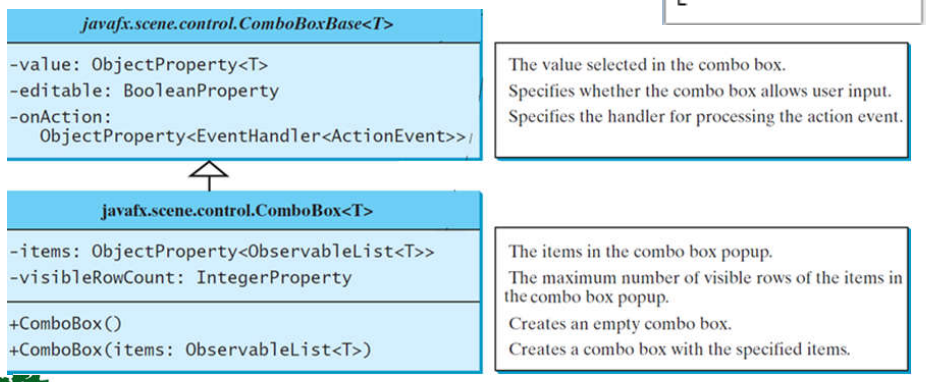


9

ComboBox



❖ A combo box, also known as a choice list or drop-down list, contains a list of items from which the user can choose.



10

ListView

- A **list view** is a component that performs basically the same function as a combo box, but it enables the user to choose a single value or multiple values.

(a) Single selection

(b) Multiple selection

(c) Multiple selection

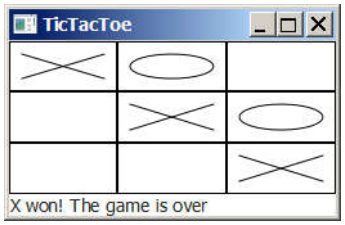
11

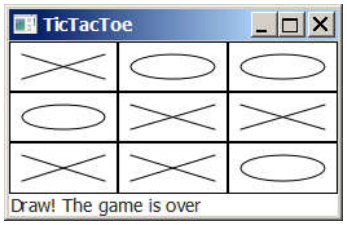
ListView

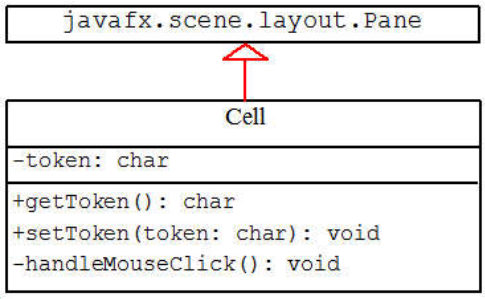
javafx.scene.control.ListView<T>	
-items: ObjectProperty<ObservableList<T>>	The items in the list view.
-orientation: BooleanProperty	Indicates whether the items are displayed horizontally or vertically in the list view.
-selectionModel: ObjectProperty<MultipleSelectionModel<T>>	Specifies how items are selected. The `SelectionModel` is also used to obtain the selected items.
+ListView()	Creates an empty list view.
+ListView(items: ObservableList<T>)	Creates a list view with the specified items.

12

Case Study: TicTacToe







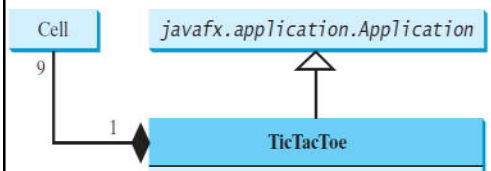
Token used in the cell (default: ' ').

Returns the token in the cell.

Sets a new token in the cell.

Handles a mouse click event.

Case Study: TicTacToe cont.



Indicates which player has the turn, initially X.

A 3 × 3, two-dimensional array for cells.

A label to display game status.

Constructs the TicTacToe user interface.

Returns true if all cells are filled.

Returns true if a player with the specified token has won.

Building a Microwave Interface