Practical Sheet nº8

Content

- JavaFX – widgets, 3D widgets and dealing with several Stages and Scenes
- UI Patterns

In this class, we will explore GUI's additional widgets and apply UI patterns.

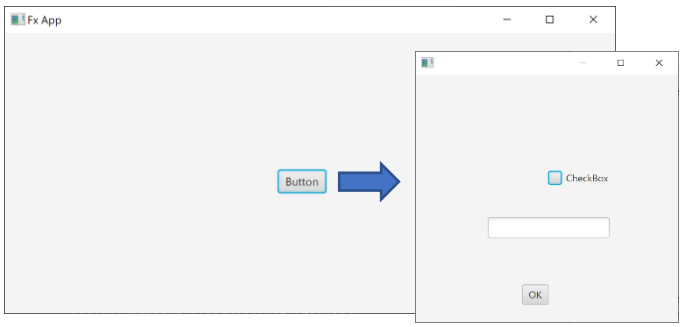
8 – Additional Widgets and UI Patterns

8.1 – Dealing with several Stages and Scenes

The following event handler code enables the opening of a new window (without FXML) when the button is clicked.

```
def onButtonClicked(): Unit = {
  val secondStage: Stage = new Stage()
  secondStage.setScene(new Scene(new HBox(4, new Label("Second window"))))
  secondStage.show()
}
```

a) Create a very simple JavaFX application with a Button using FXML (see Figure below) that when clicked opens a new window.
Hint: use `FXMLLoader`

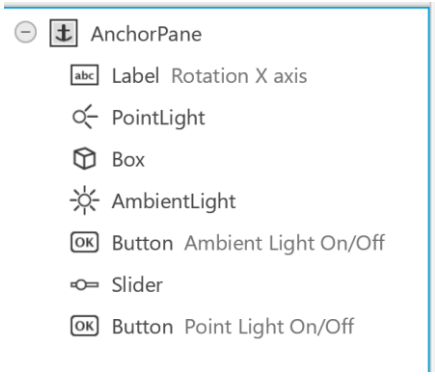


b) Transform the new opened window into a modal window (a modal window is any type of window that is a child of a parent window and usurps the parent's control). While the modal window in open it is not possible to interact with the parent window. Pressing the OK button should close the modal window.
Hints: to close the modal window use *`widget`*`.getScene().getWindow.hide()` where *`widget`* is a widget present in the current window (e.g. `button2.getScene().getWindow()`. The hide call is equivalent to close.
While creating the new Stage use the associated `initModality` and `initOwner` methods to create the modal window.

c) Create a JavaFX application with a Button using FXML that when clicked substitutes the content of the current window by the one present in another FXML file. In other words, change the scene associated to the existing stage.
Hint: to change the current scene use `widget.getScene().setRoot(mainViewRoot)` where *widget* is a widget present in the current window and *mainViewRoot* is the root of the other scene.

8.2 – JavaFX 3D

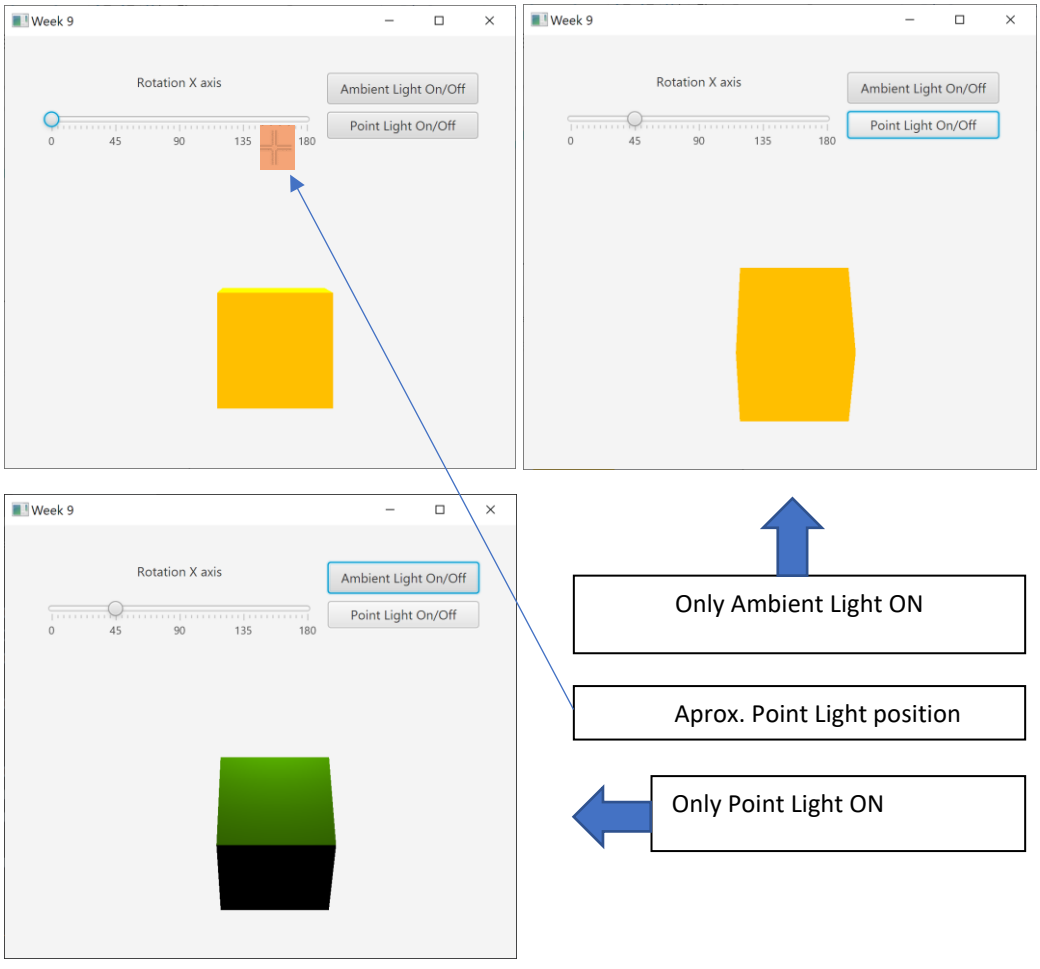
Create a JavaFX 3D application like the one presented in the figures below. Use the following Scene Graph and properties:



Some properties used:

- Ambient Light Color: `#ffbf00`
- Point Light Color `#7fff00`
- Event to deal with the slider: `On Mouse Dragged`
- Setting of the camera in the `start` method:

```
scene.setCamera(new PerspectiveCamera(false))
```



8.3 – UI Patterns (Movable Panels)

When you want users to feel a sense of ownership of the software or at least have fun with it, the *Movable Panels UI Pattern* tells to put modules of content into a box that can be opened and closed independently of each other.

Put the slider, the label and the button into a part of a movable panel and the 3D box into another part (see the figure below).