Sources:

CISC190 Java Programming

First, make sure you review chapter 14 section 2 part 1 about TranslateTransition animation techniques.

If you want to create a simple dropdown button that will help a dropdown menu appear by sliding down or up when you click the dropdown button, here are some steps to consider:

* You can use StackPane container, put dropdown button on top of the dropdown menu.
* Make sure that you don’t see the dropdown menu on the scene’s output.
* Create a method to attach to the dropdown button’s setOnAction event.
* Make sure this method helps with the TranslateTransition animation techniques to help the dropdown menu slide down when you click the dropdown button, then slide up when you click that button again.

Ex:

Syntax in **Main.java**

**package** application;

**import** javafx.application.Application;

**import** javafx.scene.Scene;

**import** javafx.stage.Stage;

**import** javafx.scene.control.Button;

**import** javafx.scene.control.Label;

**import** javafx.scene.layout.StackPane;

**import** javafx.scene.layout.VBox;

**import** javafx.scene.layout.HBox;

**import** javafx.animation.TranslateTransition;

**import** javafx.util.Duration;

**public** **class** **Main** **extends** Application {

**@Override**

**public** **void** **start**(**Stage** primaryStage) {

**Button** **butt1** = **new** **Button**(**"Click Me!!!"**);

butt1.**setId**(**"butt1"**);

**Label** **label1** = **new** **Label**(**"Content1"**);

label1.**setId**(**"label1"**);

**StackPane** **stackPane1** = **new** **StackPane**(butt1,label1);

label1.**toBack**();

label1.**setTranslateY**(**0**);

butt1.**setOnAction**(**even**->{

**TranslateTransition** **slideTransition** = **new** **TranslateTransition**(**Duration**.***millis***(**800**),label1);

**if**(label1.**getTranslateY**() == **0**) {

//slide label1 down

slideTransition.**setToY**(**50**);

}**else** {

//slide label1 up

slideTransition.**setToY**(**0**);

}

slideTransition.**play**();

});

**Scene** **scene** = **new** **Scene**(stackPane1, **400**, **300**);

scene.**getStylesheets**().**add**(**getClass**().**getResource**(**"application.css"**).**toExternalForm**());

primaryStage.**setTitle**(**"Program 1"**);

primaryStage.**setScene**(scene);

primaryStage.**show**();

}

**public** **static** **void** **main**(**String**[] args) {

***launch***(args);

}

}

Syntax in **application.css**

/\* JavaFX CSS - Leave this comment until you have at least create one rule which uses -fx-Property \*/

#butt1, #label1 {

-fx-font-size:20px;

-fx-text-fill:blue;

-fx-padding:10px;

}

#label1{

-fx-border-style:solid;

-fx-border-width:3px;

-fx-border-color:yellowgreen;

-fx-padding:10px;

}

A screenshot of a computer program

AI-generated content may be incorrect.

If you click “Click Me!!!” button, “Content1” label will slide down

A white rectangle with blue text

AI-generated content may be incorrect.

If you click “Click Me!!!” button again, “Content1” label will slide up

A white rectangle with green border and blue text

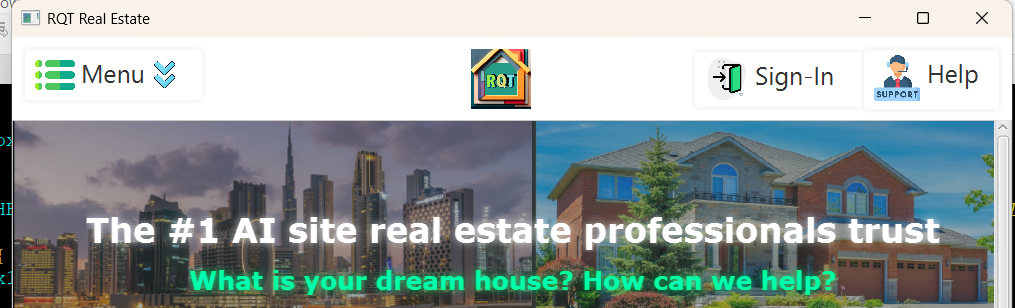
AI-generated content may be incorrect.

**II.More complex dropdown menu with slidedown animation effect**

**\*Scenario 1:**

**The program in this scenario can be found in my GitHub’s Repositories’ RQT\_Real\_Estate**

Let’s assume that you have a top navigation bar with a menu button and some more buttons like this:



When you click the Menu button, you want the Dropdown Menu Tab(or Bar) to appear by sliding down and it should be placed in front of the poster below, or you may want the Dropdown Menu Tab( or Bar) to appear by sliding down and make the poster below to be slided or pushed down

Ex:

A screenshot of a computer

AI-generated content may be incorrect.

**\*Here are two approaches to solve this problem:**

**a)Method1:**

**+When you click the Menu button, make sure that your program make the ScrollPane control appear and expand its height in pixels.**

First, try to attach a method into the Menu button so that it can be triggered and activate some codes when you click this button. Let’s call this method **show\_topNavigationBar1\_HBox1\_Dropdown\_Menu\_Tab()**

You can use VBox that includes one container containing the Top Navigation Bar, and another ScrollPane control containing one HBox container for the Dropdown Menu Tab( or Bar) that contains some buttons inside it and other tools when needed.

The ScrollPane control should be removed from the VBox’s ObservableList so that you can only see the Top Navigation Bar. The ScrollPane control should hold some HBox that hold the buttons like some images above.

Make sure the ScrollPane’s Pref Height is 0 so that you won’t see any content inside it when you make it appear when you click the Menu button.

\*The logic for the **show\_topNavigationBar1\_HBox1\_Dropdown\_Menu\_Tab()** is like this:

You check if the VBox contains the ScrollPane or not.

If the VBox doesn’t contain the ScrollPane, then it should add the ScrollPane into the ObservableList, here you may see the ScrollPane appears on the program output below the Top Navigation Bar and above the Poster (it will push the Poster down). Then, make sure the ScrollPane show vertical scroll bar and horizontal bar when needed. Next, using animation effects (TimeLine, KeyValue, and KeyFrame) to expand the ScrollPane’s prefer height, let’s assume 80 pixels for example (the ScrollPane’s prefer height’s expanding effect will slide the Poster down too) in 0.4 seconds( depends on your needs. The longer the duration, the longer you have to look at the ScrollPane on your program’s output contains a small portion of its inside content).

If the VBox contains the ScrollPane, then using animation effects (TimeLine, KeyValue, and KeyFrame) to shrink the ScrollPane’s prefer height, let’s assume 0 pixels for example (this will also slide the Poster up) in 0.4 seconds( depends on your needs), then it should remove the ScrollPane from the ObservableList when the animation effects are finished so that you can only see the Top Navigation Bar on your program’s output.

**Code Example(not a complete version, only for TopNavigationBarController.java file):**

**package** application.MainPageScene.TopNavigationBar;

**import** java.io.IOException;

**import** javafx.fxml.FXML;

**import** javafx.scene.layout.VBox;

**import** javafx.scene.layout.BorderPane;

**import** javafx.scene.layout.HBox;

**import** javafx.scene.control.ScrollPane;

**import** javafx.scene.control.Label;

**import** javafx.scene.control.Button;

**import** javafx.scene.image.ImageView;

**import** javafx.geometry.Insets;

**import** javafx.animation.KeyFrame;

**import** javafx.animation.KeyValue;

**import** javafx.animation.Timeline;

**import** javafx.util.Duration;

**public** **class** TopNavigationBarController{

**@**FXML

VBox topNavigationBar1\_VBox1;

**@**FXML

BorderPane topNavigationBar1\_BorderPane1;

**@**FXML

HBox topNavigationBar1\_BorderPane1\_Left\_HBox1;

**@**FXML

Button topNavigationBar1\_BorderPane1\_Left\_HBox1\_MenuButt;

**@**FXML

ImageView topNavigationBar1\_BorderPane1\_Center\_Logo;

**@**FXML

HBox topNavigationBar1\_BorderPane1\_Right\_HBox1;

**@**FXML

Button topNavigationBar1\_BorderPane1\_Right\_HBox1\_SignIn\_Butt;

**@**FXML

Button topNavigationBar1\_BorderPane1\_Right\_HBox1\_Help\_Butt;

**@**FXML

ScrollPane topNavigationBar1\_ScrollPane1;

**@**FXML

HBox topNavigationBar1\_HBox1\_Dropdown\_Menu\_Tab;

**@**FXML

HBox topNavigationBar1\_HBox1\_Dropdown\_Menu\_Tab\_HBox1;

**@**FXML

Button topNavigationBar1\_HBox1\_Dropdown\_Menu\_Tab\_HBox1\_Buy\_Butt;

**@**FXML

Button topNavigationBar1\_HBox1\_Dropdown\_Menu\_Tab\_HBox1\_Rent\_Butt;

**@**FXML

Button topNavigationBar1\_HBox1\_Dropdown\_Menu\_Tab\_HBox1\_Sell\_Butt;

**@**FXML

Button topNavigationBar1\_HBox1\_Dropdown\_Menu\_Tab\_HBox1\_Loan\_Butt;

**@**FXML

Button topNavigationBar1\_HBox1\_Dropdown\_Menu\_Tab\_HBox1\_Agents\_Butt;

**@**FXML

Button topNavigationBar1\_HBox1\_Dropdown\_Menu\_Tab\_HBox1\_News\_Butt;

**@**FXML

HBox topNavigationBar1\_HBox1\_Dropdown\_Menu\_Tab\_HBox2;

**@**FXML

Button topNavigationBar1\_HBox1\_Dropdown\_Menu\_Tab\_HBox2\_Close\_Butt;

**public** **void** **initialize**() {

topNavigationBar1\_VBox1.getChildren().remove(topNavigationBar1\_ScrollPane1);

}

**@**FXML

**public** **void** **show\_topNavigationBar1\_HBox1\_Dropdown\_Menu\_Tab**() **throws** IOException{

**if**(!topNavigationBar1\_VBox1.getChildren().contains(topNavigationBar1\_ScrollPane1)) {

// If the tab is not already in the VBox, add it to the VBox

topNavigationBar1\_VBox1.getChildren().add(topNavigationBar1\_ScrollPane1);

// Configure scroll behavior

topNavigationBar1\_ScrollPane1.setVbarPolicy(ScrollPane.ScrollBarPolicy.ALWAYS);

//animation effect that slide the tab down by expanding the height of the ScrollPane

Timeline **timeline** = **new** Timeline();

KeyValue **keyValue** = **new** KeyValue(topNavigationBar1\_ScrollPane1.prefHeightProperty(), **80**);

KeyFrame **keyFrame** = **new** KeyFrame(Duration.seconds(**0.4**), keyValue);

timeline.getKeyFrames().add(keyFrame);

timeline.play();

}**else** {

//animation effect that slide the tab up by shrinking the height of the ScrollPane

Timeline **timeline2** = **new** Timeline();

KeyValue **keyValue2** = **new** KeyValue(topNavigationBar1\_ScrollPane1.prefHeightProperty(), **0**);

KeyFrame **keyFrame2** = **new** KeyFrame(Duration.seconds(**0.4**), keyValue2);

timeline2.getKeyFrames().add(keyFrame2);

timeline2.play();

// Remove the tab from the VBox after the animation is finished

timeline2.setOnFinished(**event** -> {

topNavigationBar1\_VBox1.getChildren().remove(topNavigationBar1\_ScrollPane1);

});

}

}

**@**FXML

**public** **void** **close\_topNavigationBar1\_HBox1\_Dropdown\_Menu\_Tab**() **throws** IOException{

//animation effect that slide the tab up by shrinking the height of the ScrollPane

Timeline **timeline** = **new** Timeline();

KeyValue **keyValue** = **new** KeyValue(topNavigationBar1\_ScrollPane1.prefHeightProperty(), **0**);

KeyFrame **keyFrame** = **new** KeyFrame(Duration.seconds(**0.4**), keyValue);

timeline.getKeyFrames().add(keyFrame);

timeline.play();

// Remove the tab from the VBox after the animation is finished

timeline.setOnFinished(**event** -> {

topNavigationBar1\_VBox1.getChildren().remove(topNavigationBar1\_ScrollPane1);

});

}

}

Notice that I’m not going to specify all the code from FXML file and CSS here in this document to make it short, yet the code above actually worked. Your job now is just create your own project and add the code like above into it and test

**b)Method2: (I haven’t thought about this method yet by this time. I’ll come back to this document to write more about it later)**