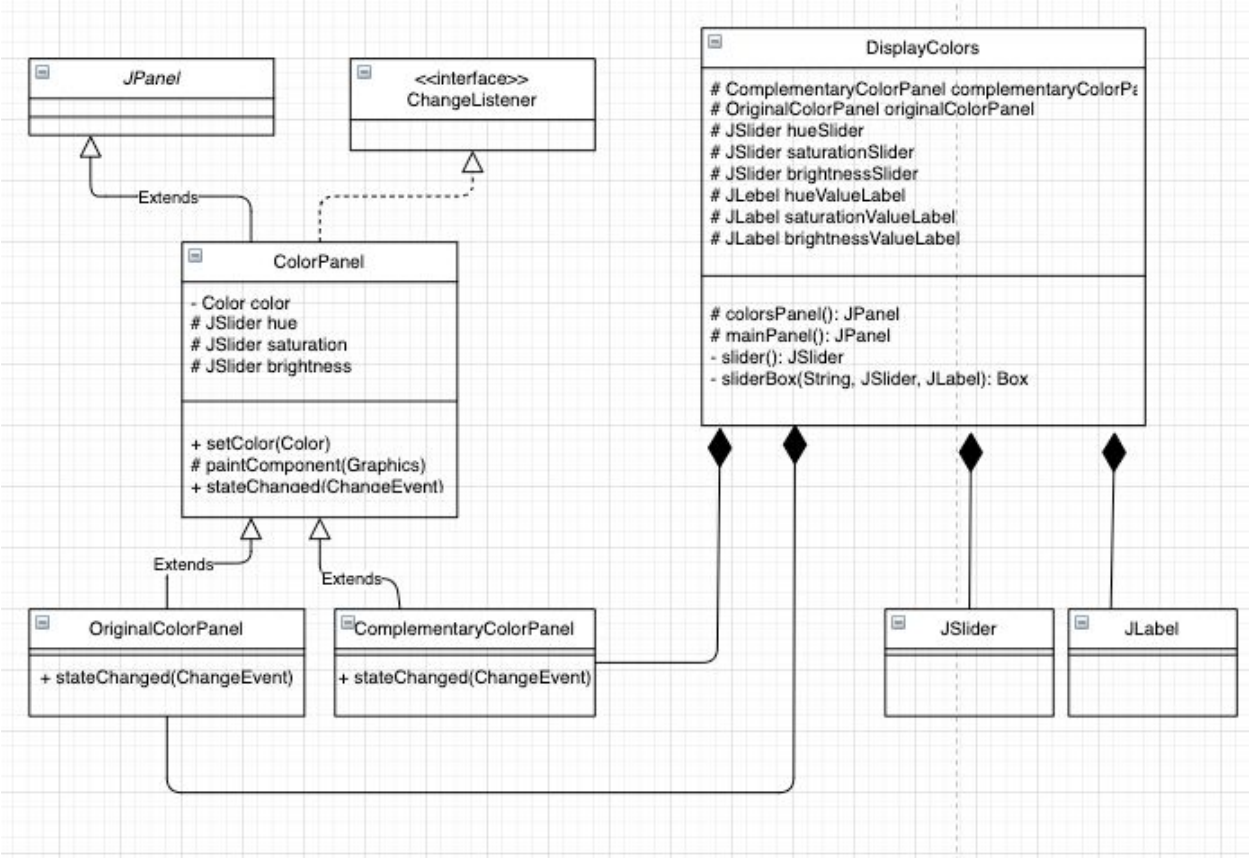
Amit Verma
EE 461L

Tutorial 10

Task 2 UML and Questions

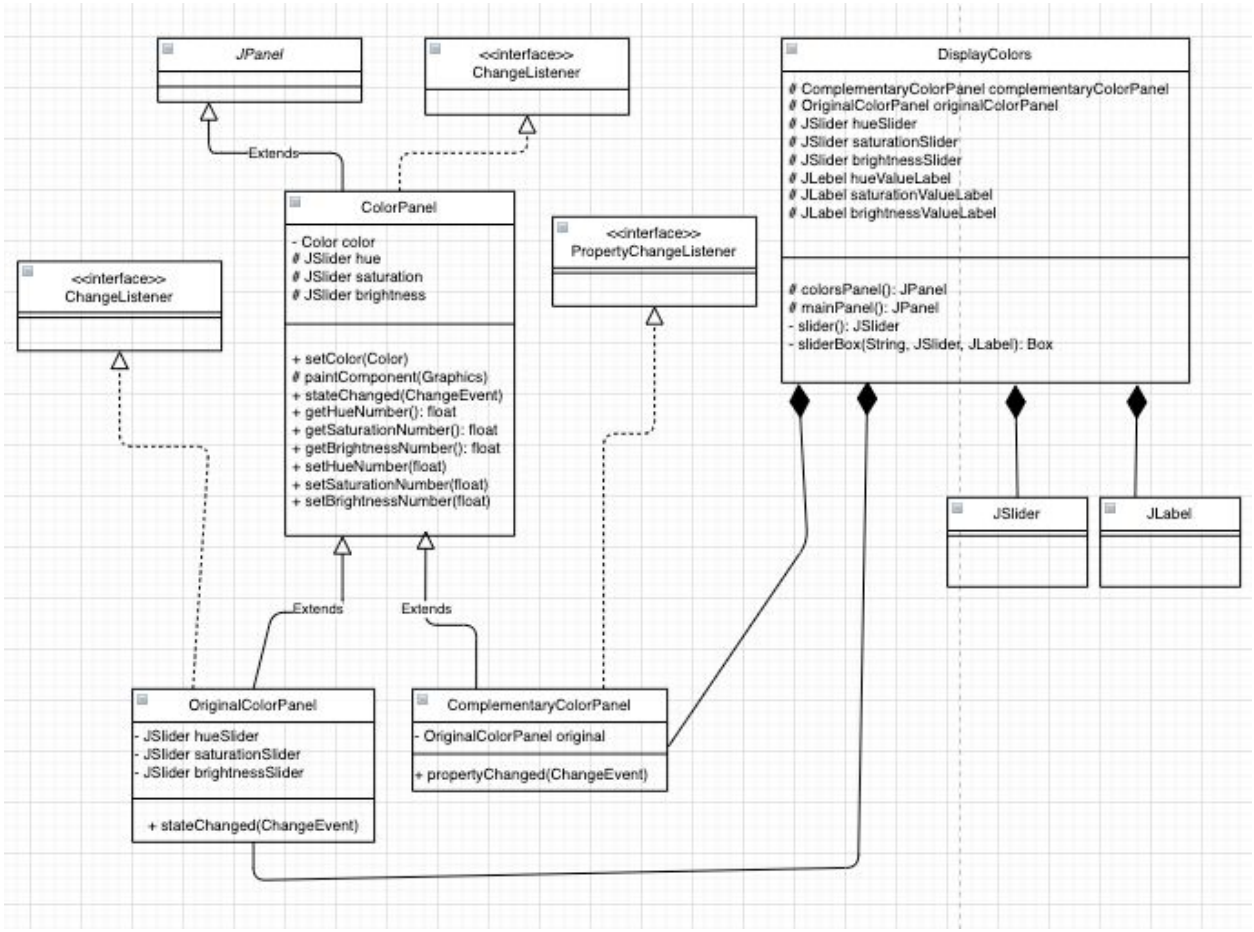


The logic for updating the 'original' color was put in OriginalColorPanel.java, while the logic for computing the 'complementary' color was put in ComplementaryColorPanel.java.

Task 3 UML and Questions

Amit Verma
EE 461L

Tutorial 10



ComplementaryColorPanel is observing OriginalColorPanel

Task 4 UML and Questions

We see that the MVC pattern helps us better understand the layout of our code for design and implementation. The MVC pattern consists of a Model, View, and a Controller, which all act with each other in order to provide structured code. Within this tutorial, we see how the Model, View, and Controller apply here as there are aspects of our code from Task 3 that interfaces with the user, observers other modules, and such. That said, after reading MVC documentation we see that Color represents the Model as it is the data of this application. The View, which is the data that is presented to the user, is DisplayColors as it is what provides the GUI for the user to experience and interface with. This Color class, which is our Model, will update the DisplayColors, which is our View and observables, as the GUI contains color panels that display the color data to the user. Now, lastly, the Controller, existing between the Model and View, of our design pattern is what will contain the logic or brains to compute the colors to display to the user. Upon interacting with the View (the GUI shown from DisplayColors), the

Tutorial 10

user is changing the input values that will change the Model (which is the Color). In our case, the Controller is composed of the ColorPanel, OriginalColorPanel, and ComplementaryColorPanel classes. Note, our classes in the Controller are the observers as they observe the sliders that a user is changing in View, which are our observables, and then changes the Model, which is the new color (Color class). In doing so, the Color class updates the View (color panels shown on the GUI) to reflect what the user should see. In its entirety, this whole process represents the MVC pattern in practice. On the following page is a UML diagram of this design pattern applied to this tutorial:

Source:

https://www.codecademy.com/articles/mvc

Amit Verma
EE 461L

Tutorial 10