Assignment 7

Arizona State University - CSE205 - Assignment #7

Due Date Friday, March 6th, 5:30pm

Important: This is an individual assignment. Please do not collaborate.

No late assignment will be accepted.

Make sure that you write every line of your code. Using code written by someone else will be considered a violation of the academic integrity and will result in a report to the Dean's office.

You must submit your files on Gradescope, which is provided on course website.

Minimal Submitted Files

You are required, but not limited, to turn in the following source file:

Assignment7.java (https://canvas.asu.edu/courses/44324/files/13127513/download?wrap=1) **(https://canvas.asu.edu/courses/44324/files/13127513/download?wrap=1)** (Download this file and use it as your driver program for this assignment.)

PaneWithRectangles.java (https://canvas.asu.edu/courses/44324/files/13127516/download?wrap=1) **(https://canvas.asu.edu/courses/44324/files/13127516/download?wrap=1)** (Download this file. You need to add more code to complete it.)

Requirements to get full credits in Documentation

1. The assignment number, your name, StudentID, Lecture day/time, and a class description need to be included at the top of each file/class.
2. A description of each method is also needed.
3. Some additional comments inside of methods (especially for a "main" method) to explain code that are hard to follow should be written.

New Skills to be Applied

In addition to what has been covered in previous assignments, the use of the following items, discussed in class, will probably be needed:

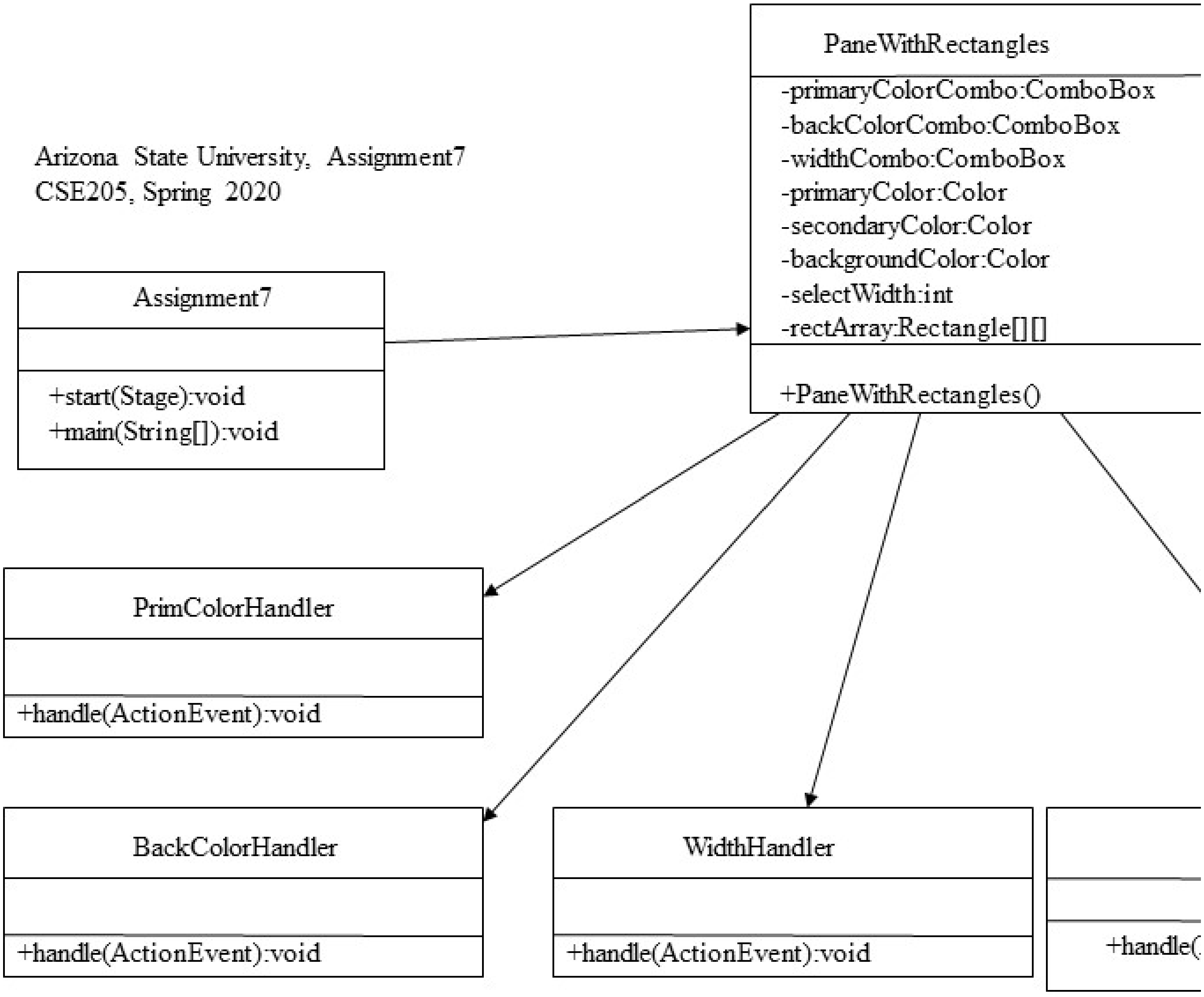
JavaFX

Classes may be needed:
Button, ComboBox, Color, Graphics, Line, EventHandler, ActionEvent, MouseEvent. You may use

other classes.

Program Description

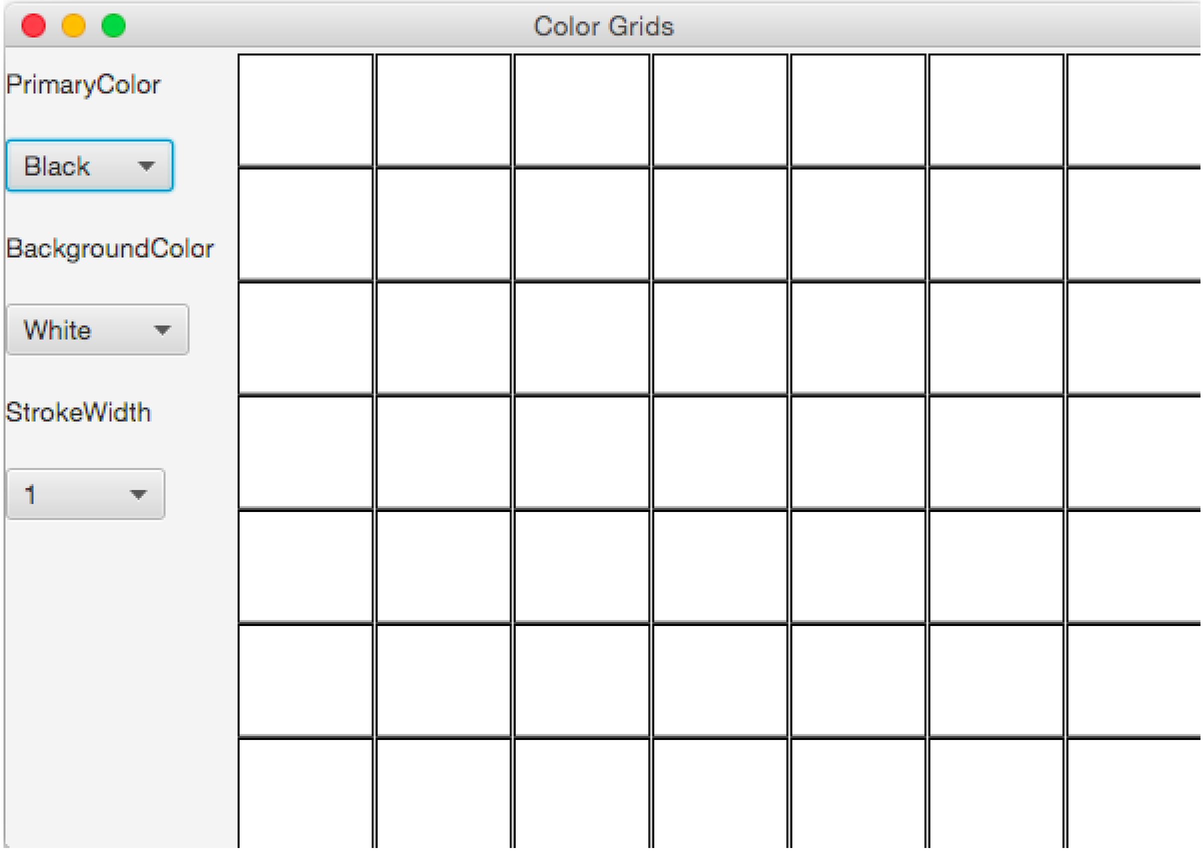
Class Diagram:



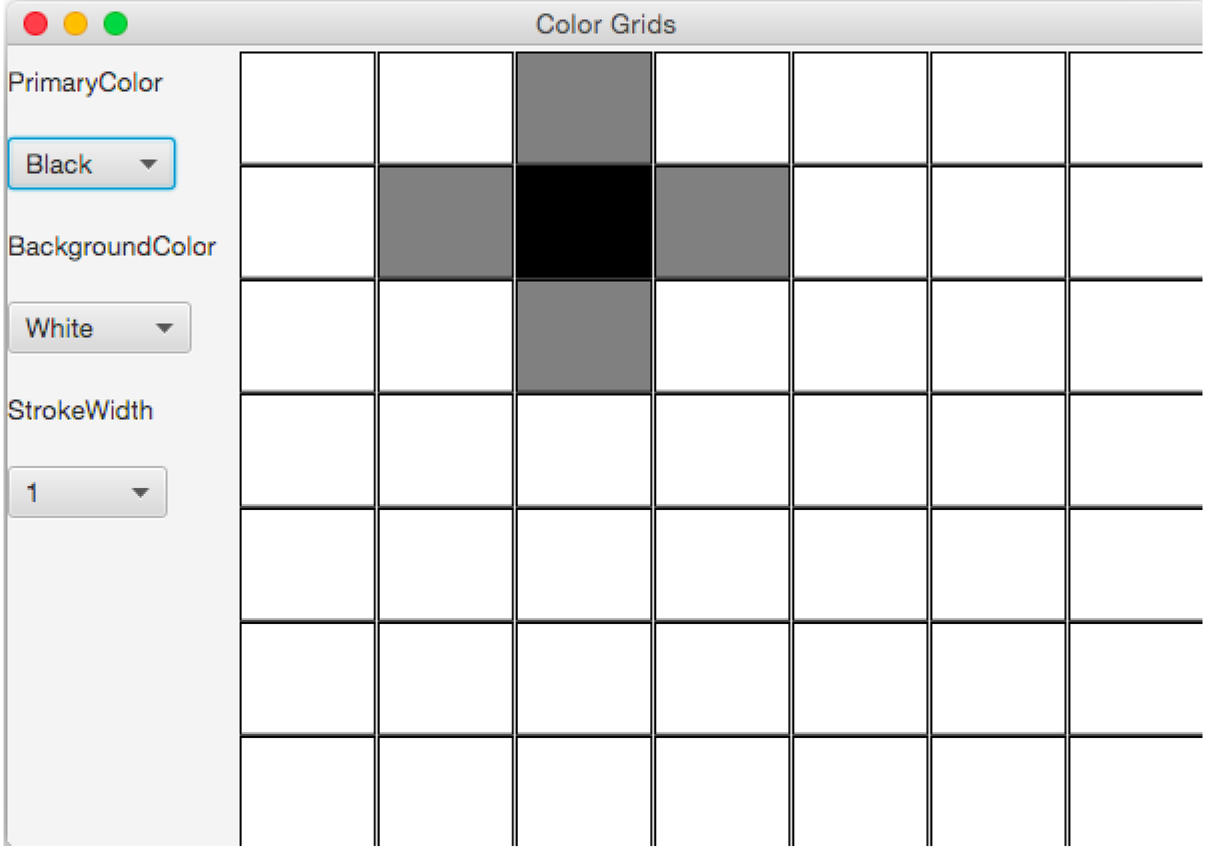
Write a Java program that constructs a Graphical User Interface using JavaFX.

The GUI should contain two panes. The left pane should contain three Combo Boxes, one to select a primary color (black, blue, red, or green), one to select a background color (white, yellow, or orange), and one to select a stroke width of lines (1, 3, 5, or 7). The initial primary color is black, the initial background color is white, and the initial stroke width is 1. The right pane should contain a grid consisting of 29 rectangles whose outline stroke is black. The inside of these rectangles should be initialized to white. It is recommended that you create a two-dimensional array of Rectangle objects with 7 rows and 7 columns, and add them to the right hand side pane using a Grid Pane. It is highly recommended to create nested loops to initialize this array of Rectangle objects. The pane with rectangles can have its width to be 470 and its height to be 390, thus the width of rectangles will be 470/7, and the height of rectangles will be 390/7.

(The size of the applet here is approximately 600 X 400).



A user can drag a mouse into the right hand side pane. The rectangle where the pointer of a mouse that is being dragged is located should become the chosen primary color, in this case, black, and the rectangles that are located above, below, left and right of the rectangle where a mouse pointer is should become its secondary color that is gray.



PrimaryColor

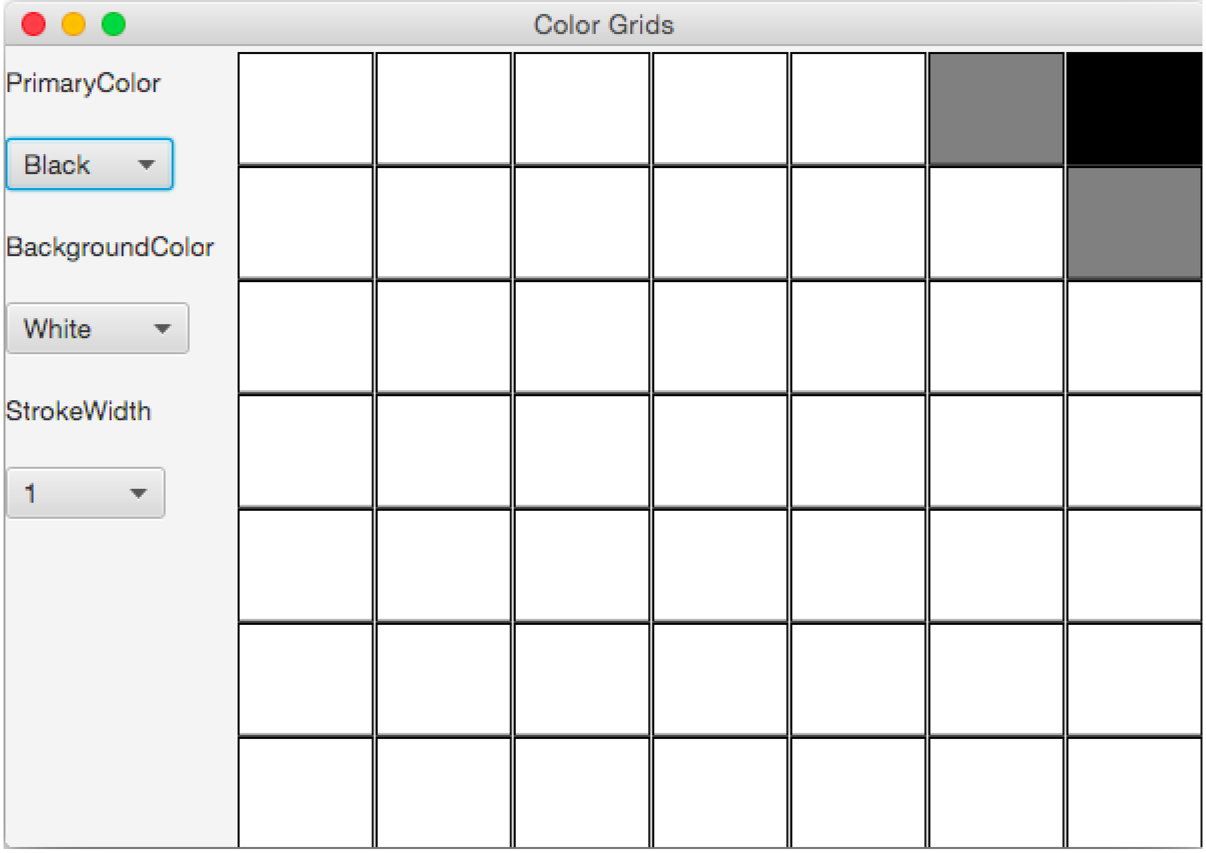
Black

BackgroundColor

White

StrokeWidth

1



PrimaryColor

Black

BackgroundColor

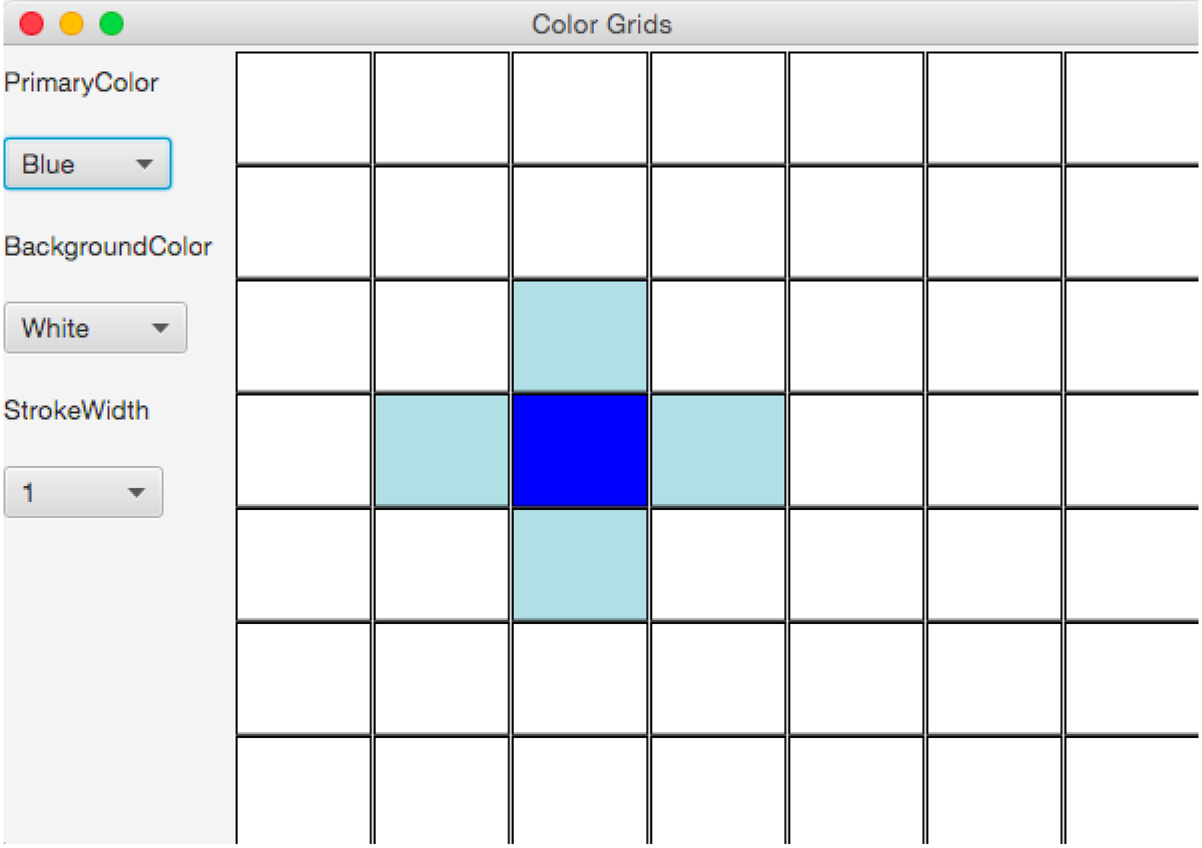
White

StrokeWidth

1

The following is the list of primary colors with their secondary colors that should work with the combo box:

Primary Color	Secondary Color
Color.BLACK	Color.GRAY
Color.BLUE	Color.POWDERBLUE
Color.RED	Color.PINK
Color.GREEN	Color.LIGHTGREEN


Color Grids

PrimaryColor

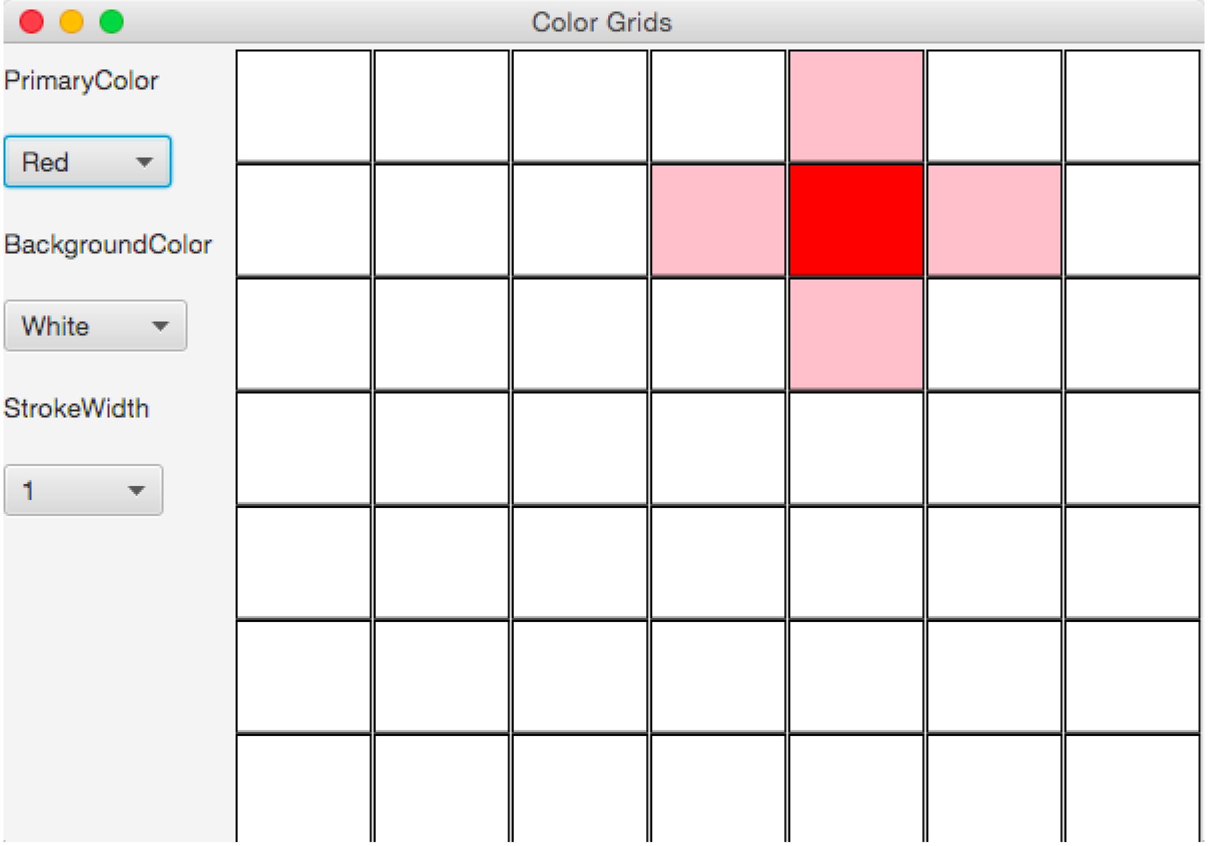
Blue

BackgroundColor

White

StrokeWidth

1



PrimaryColor

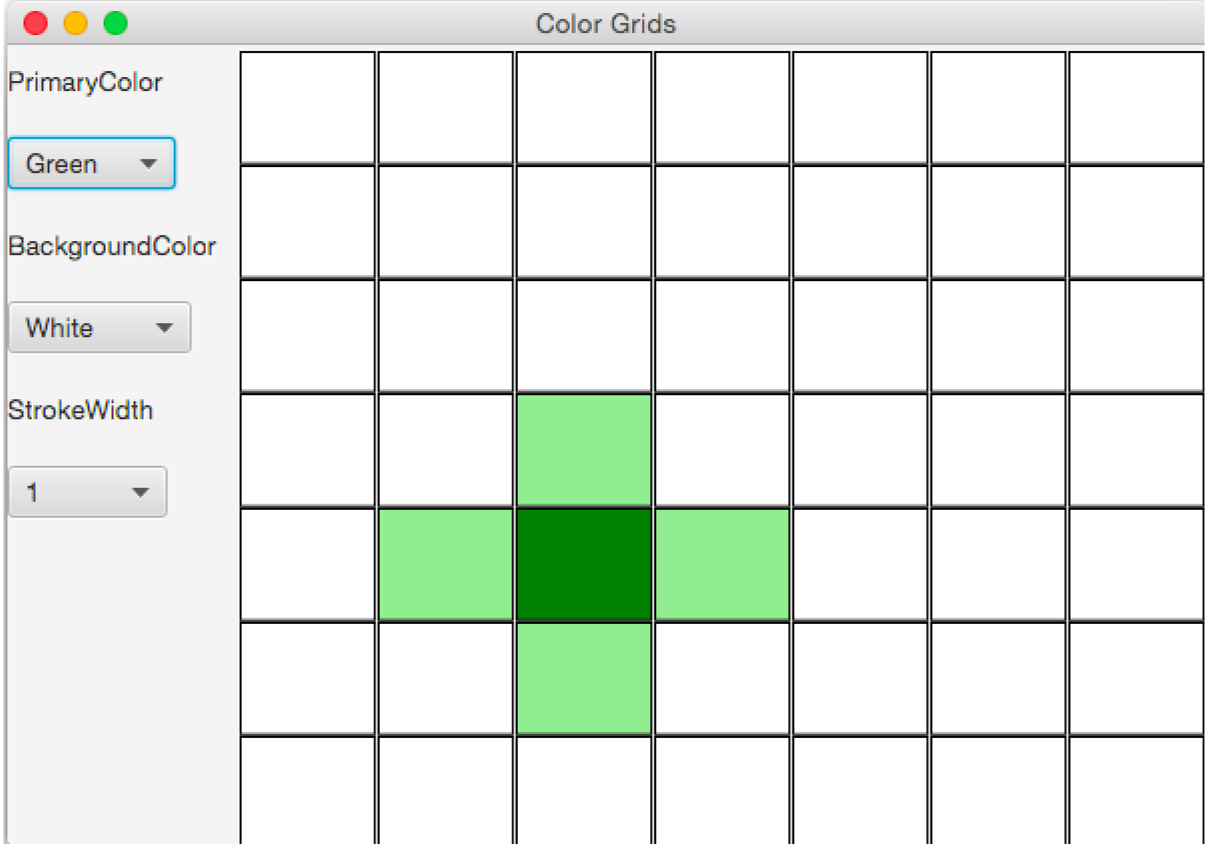
Red

BackgroundColor

White

StrokeWidth

1



PrimaryColor

Green

BackgroundColor

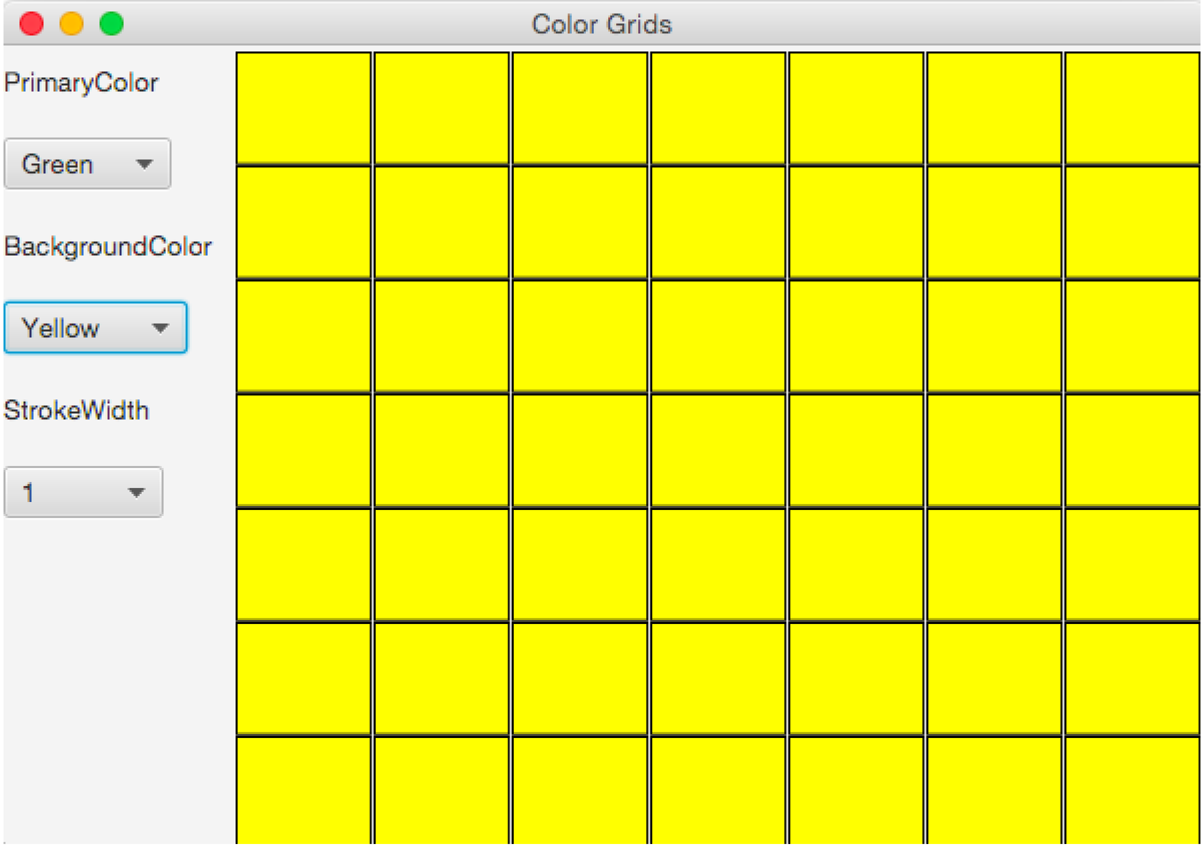
White

StrokeWidth

1

A user should also be able change the right pane's background color to Color.WHITE, Color.YELLOW, or Color.ORANGE. When a user changes the background color using the combo box, it should change the color of all rectangles to that color.

For instance, if a user chooses yellow background color, then it should look like:



PrimaryColor

Green

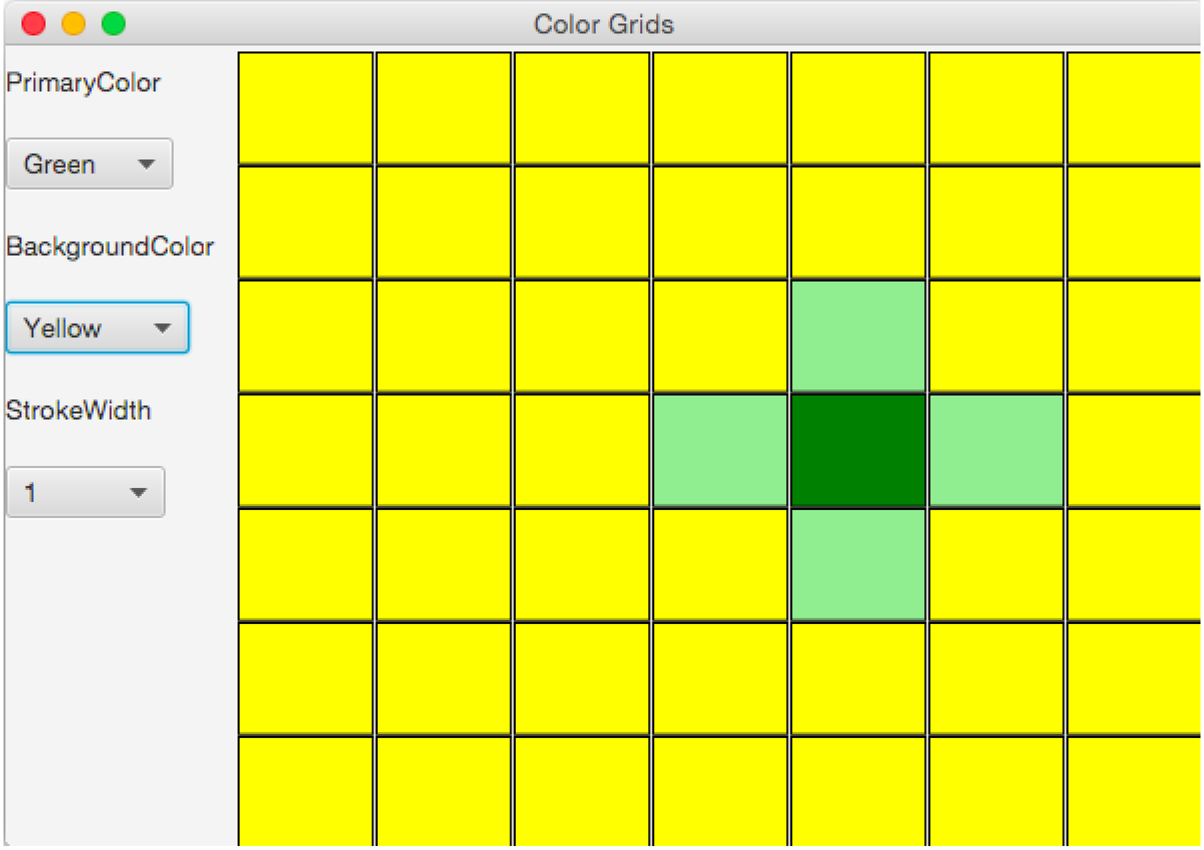
BackgroundColor

Yellow

StrokeWidth

1

Then a user can drag a mouse again to choose one of rectangles:



PrimaryColor

Green

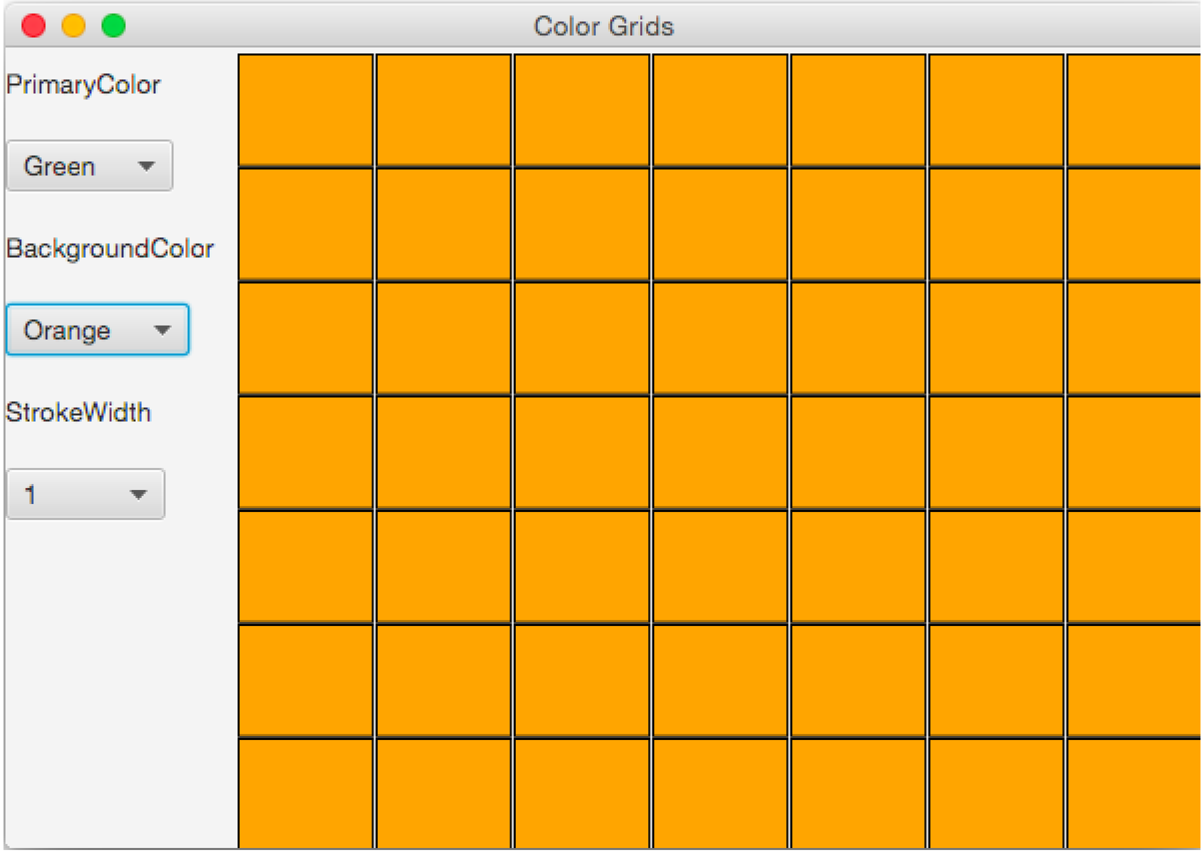
BackgroundColor

Yellow

StrokeWidth

1

If a user chooses orange background color, then it should look like:



PrimaryColor

Green

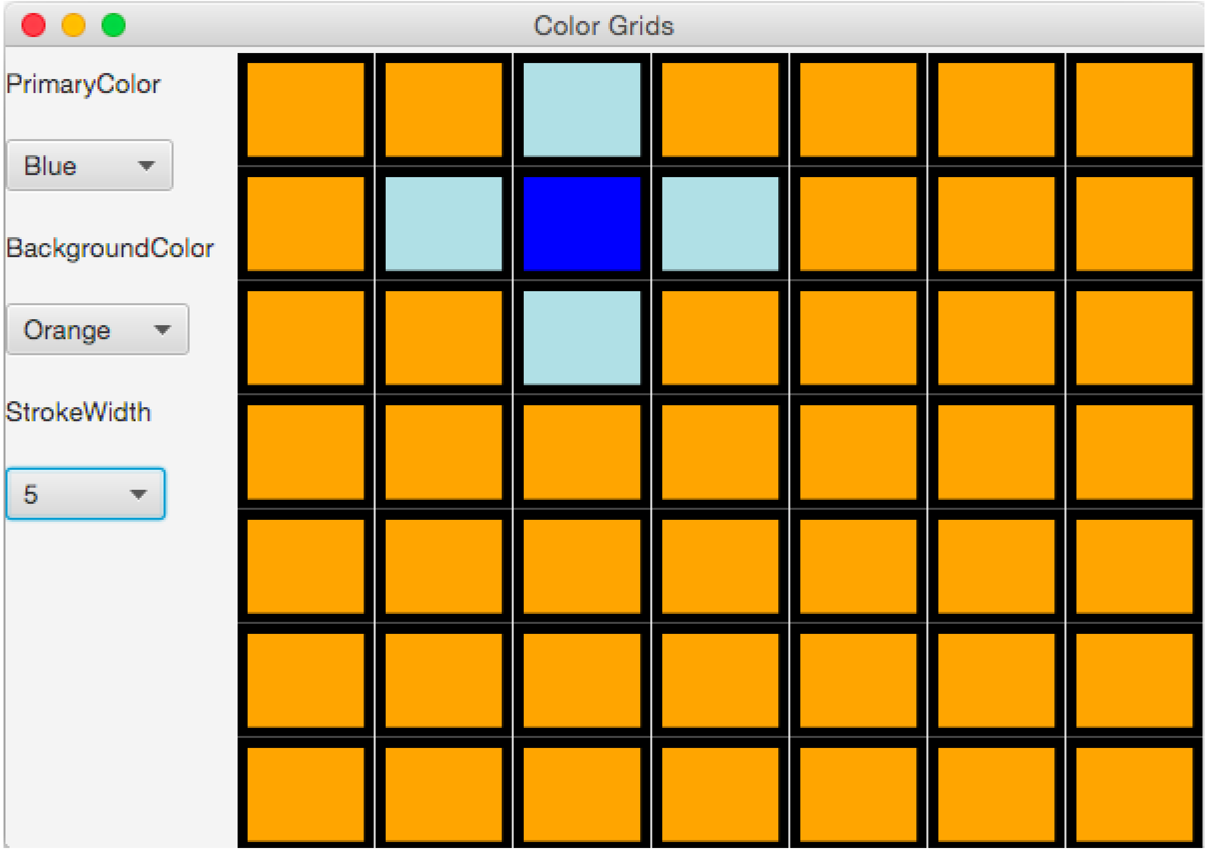
BackgroundColor

Orange

StrokeWidth

1

A user should also be able to change the width of outline (stroke) of rectangles using its combo box:



PrimaryColor

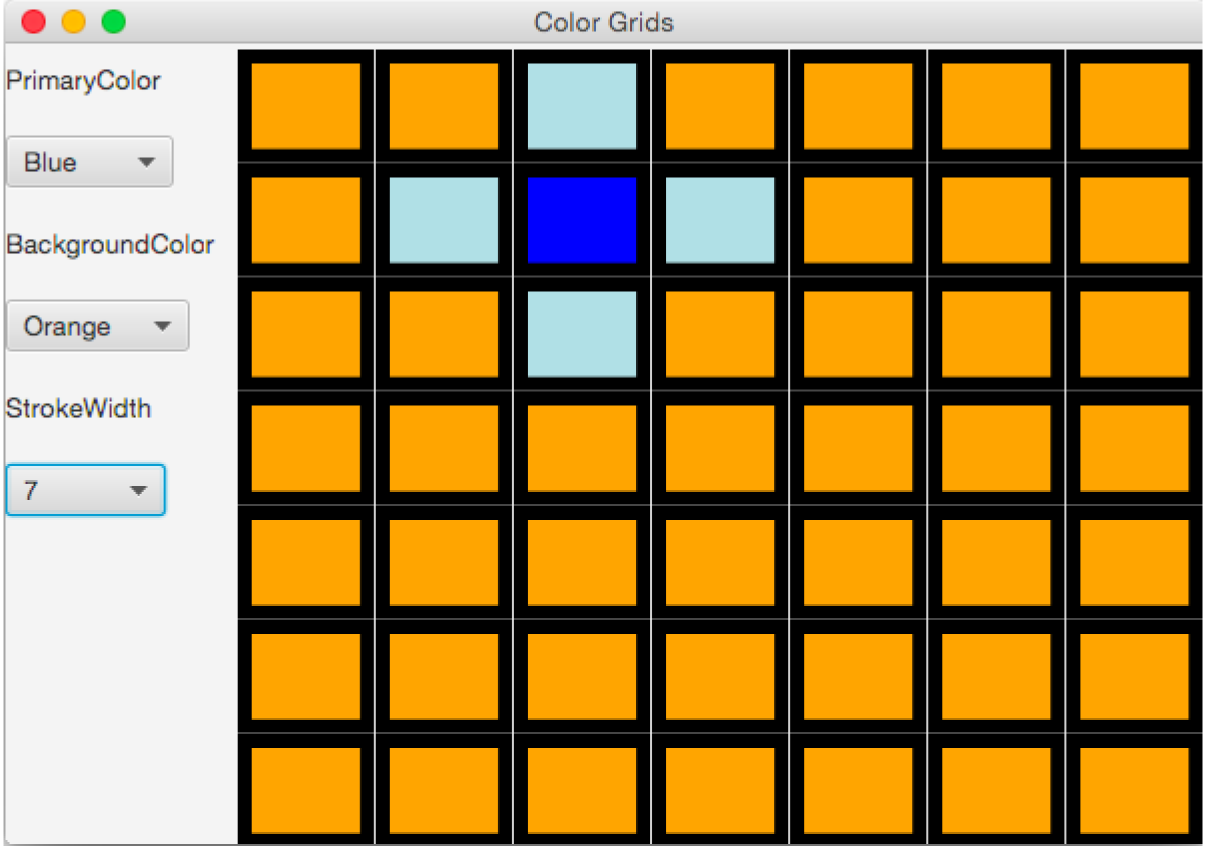
Blue

BackgroundColor

Orange

StrokeWidth

5



Here is how it should work:

Class description

PaneWithRectangles class

The PaneWithRectangles class organizes all nodes in the GUI. It should contain a pane at the left hand side with three combo boxes and three labels, and another pane at the right hand side containing 49 objects of Rectangle class. Your pane needs to have the same appearance as the shown picture.

Attribute name	Attribute type	Description
primaryColorCombo	ComboBox	A combo box to choose a primary color from black, blue, red, and green. Its default is black.
backgroundCombo	ComboBox	A combo box to choose a

		background color from white, yellow, and orange. Its default is white
widthCombo	ComboBox	A combo box to choose a stroke width for rectangles from 1, 3, 5, or 7. Its default is 1
primaryColor	Color	A color used for the rectangle pointed by a mouse pointer. Its initial color is black.
secondaryColor	Color	A color used for the rectangles above, below, left, and right of the rectangle with its primary color. Its initial color is gray.
backgroundColor	Color	A default background color of rectangles that do not have its primary or secondary color. Its initial color is white.
selectWidth	int	A width of the strokes of rectangles. Its initial width is 1.0
rectArray	Rectangle[][]	A two-dimensional array of Rectangle objects. Its number of rows should 7 and its number of columns should be 7.

This class should have a constructor:

public PaneWithRectangles()

This is where all nodes are arranged. Add as many instance variables as you like to this class, and initialize them in this constructor. Then they need to be arranged using layout panes so that the GUI will have the required appearance in this constructor. Objects of the Handlers are also instantiated here and are used to set for their corresponding node(s) such as a combo box.

MouseHandler class (defined as a nested class of PaneWithRectangles class)

The MouseHandler class implements EventHandler<MouseEvent> interface. It must implement the following method:

public void handle(MouseEvent event)

This method should work with the pane containing 49 rectangles. If the mouse is being dragged on the pane with 49 rectangles, then the rectangle where the mouse pointer is pointing should change its color to the currently chosen primary color (by the combo box), and the rectangles above, below, left and right of the rectangle with the primary color should have the corresponding secondary color. For instance, if a pointed rectangle is black, then rectangles above, below, left and right should become gray.

PrimColorHandler class (defined as a nested class of PaneWithRectangles class)

The PrimColorHandler class implements EventHandler<ActionEvent> interface. It must implement the following method:

public void handle(ActionEvent event)

This method should work with the combo box to choose a primary color. It should set the Color variables primaryColor and secondaryColor based on the one is chosen in the combo box.

BackColorHandler class (defined as a nested class of PaneWithRectangles class)

The BackColorHandler class implements EventHandler<ActionEvent> interface. It must implement the following method:

public void handle(ActionEvent event)

This method should work with the combo box to choose a background color. It should set the Color variable backgroundColor based on the one is chosen in the combo box.

WidthHandler class (defined as a nested class of PaneWithRectangles class)

The WidthHandler class implements EventHandler<ActionEvent> interface. It must implement the following method:

public void handle(ActionEvent event)

This method is executed in case the width combo box is used to select a width of strokes for rectangles. It should set the variable selectWidth based on the one that is chosen in the combo box. For this combo box, as soon as another width is selected, the stroke width of rectangles should change its width to the selected one. When the stroke width increases, the width and the height of the 49 rectangles need to be reduced by (selectWidth-1).

How to get started:

Download the following files and use them as a base of your program:
Assignment7.java (hw7/Assignment7.java)
PaneWithRectangles.java (hw7/PaneWithRectangles.java)

Step 1: Complete the constructor or PaneWithRectangles by instantiating/initializing nodes such as combo boxes, labels, and rectangles and arranging their layouts by following their descriptions.

Step 2: Create each handler class described above by defining their handle( ) method.

Step 3: Make sure to set an object of the handler class to their corresponding component in the constructor of PaneWithRectangles (after they are instantiated).

Error Handling

While your GUI is running, it should not generate exceptions. For instance, ArrayIndexOutOfBoundsException should not occur when a user is dragging a mouse on the pane. Your code should check if it is not going out of bounds for the two dimensional array of Rectangles objects.

What to turn in:

-Submit your Assignment7.java and PaneWithRectangles.java files

using Gradescope-> Assignment7 on canvas.asu.edu. Make sure that your files are compiling. You can submit multiple times until the assignment deadline.

Grading Policy:

- submit assignment on time
- indicate assignment number, name, lecture time, and description of each class clearly in each submitted java file
- 5 pts - Documentation (header with your name, your information, and program description at the top of each file and comments within your code and methods, each method needs a description)

1 pt - Indentation and spacing (easy to read)

6 pts - Required classes/methods and functionalities implemented

8 pts - Your program minimally need to have the following functionalities:

1. 2 points: Appropriate components such as combo boxes, labels, and rectangles, etc. are shown using the layout described.
2. 3 points: Each rectangle that is pointed by the mouse pointer while dragging changes its color to the selected primary color, and its above, below, left and right rectangles change to its corresponding secondary color. It should not generate ArrayIndexOutOfBoundsException.
3. 1 point: The primary color combo box is change the primary color and its corresponding secondary color of the selected rectangles by mouse dragging as described.
4. 1 point: The background color combo box is changing the color of the rectangles as described.
5. 1 point: The stroke width combo box is changing the stroke width of the rectangles as described.

Total points: 20

ASU disclaimer (http://www.asu.edu/asuweb/disclaimer/)