Topic: For this module’s discussion board assignment, select at least two of the topics below. In your writing, be sure to explain the what, how, and why of the selected topics. If necessary, provide code examples to further illustrate your thoughts.

1. JavaFX Color
2. JavaFX Gradient Color

JavaFX Color is used to encapsulate colors in the default sRGB color space. It provides multiple constructors that allow you to create custom or predefined colors. Colors are defined using double values to determine the combination of red, green, and blue. Additionally, you can modify the opacity, saturation, brightness, and other color variables which combine to make nearly endless possibilities. The color class is implemented in many other classes for convenient handling of the GUI.

The color class provides some interesting methods not discussed above. Here are a few I found interesting:

* invert() – Creates a new color that is inverted from objects current color.
* toString() – Returns a string representing the color.
* Web(String colorString) – returns the color as an HTML/CSS attribute.

Gradient Colors, or color progression, is a technique used to produce smooth color transitions over a region/object. JavaFX implements this thru two classes, RadialGradient and LinearGradient. Radial gradient is used for circular gradient patterns, while linear gradient is used to create linear patterns. Both have two constructors:

* RadialGradient(double focusAngle, double focusDistance, double centerX, double centerY, double radius, boolean proportional, CycleMethod cycleMethod, List<Stop> stops)
* RadialGradient(double focusAngle, double focusDistance, double centerX, double centerY, double radius, boolean proportional, CycleMethod cycleMethod, Stop... stops)
* LinearGradient(double startX, double startY, double endX, double endY, boolean proportional, CycleMethod cycleMethod, List<Stop> stops)
* LinearGradient(double startX, double startY, double endX, double endY, boolean proportional, CycleMethod cycleMethod, Stop... stops)

As you can see, RadialGradients relies on defining a CenterPoint coordinate, while LinearGradient can simply define a start and end point. Which one you will use is largely based on what you want your GUI to look like, which is often not decided by the software developer.

Documentation Links:

Color: <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/paint/Color.html>  
RadialGradient: <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/paint/RadialGradient.html>   
LinearGradient: <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/paint/LinearGradient.html>

Response 1:

Anitha,

Looking at how applications are designed, many use rectangles for nearly everything. The trend over the last few decades has been to have rounded edges. You can achieve that by modifying the Arc height and width (setArcHeight & setArcWidth). Modifying the arc will help you achieve a more modern look.

Christopher,

Text manipulation is one of the most basic features any good GUI framework needs. It may sound simple, but making sure to give developers enough control to create a unique platform is critical. JavaFX provides a variety of methods to give this kind of control, such as:

* setFont()
* setLineSpacing()
* strikethroughProperty()
* underlineProperty()