**CHAPTER 2**

**BACKGROUND THEORY**

In this chapter, the overview discussion of the JavaFX and the detail structure explanations of their components.

**2.1. History of JavaFX**

Sun Microsystems created the Java Programming Language and presented JDK 1.0 in 1995/96. To support GUI programming, Java introduced AWT (Abstract

Windowing Toolkit) in JDK 1.1 (1997), and Swing in JDK 1.2 (1998). But many developers felt Swing was over-complex and Java on the desktop never really took off as it did on the server.

Sun Microsystems tried several ways to make it easier to create Java GUI applications. One of these was a scripting language called JavaFX Script 1.0 (2008) which allows developers to build much more complex user Interfaces. But JavaFX Script was not Java. It is a totally new language and never really caught on with Java developers.

When Oracle acquired Sun Microsystems, they killed off JavaFX as a scripting language but added its functionality into the Java Language as JavaFX 2.0 (2011). They enhanced it as the new way to develop user interfaces, intended to replace Swing. Starting from JDK 8 (2014), JavaFX was part of JDK (as JavaFX 8). Oracle will continue to maintain the Swing library but will not enhance it. Swing and JavaFX can be used together. But for writing new Java applications, JavaFX is recommended as it offers a much simpler way to create desktop applications, and you can write more powerful applications with much less code.

A new graphics engine - Prism, a hardware accelerated pipeline, is coupled with Glass, as new windowing tookit, to produce high quality graphics for JavaFX applications. A new declarative markup language called FXML. It is based on XML and enables developers to define a user interface for a JavaFX application.

There are also a number of sample Java applications that come with the SDK to show developers how to build different kinds of JavaFX applications.