Developing applications using JavaFX

Cheshta, Dr. Deepti Sharma

M.Tech Scholar, Dept. of CSE, Advanced Institute of Technology & Management, Palwal, Haryana, India

HOD, Dept. of CSE, Advanced Institute of Technology & Management, Palwal, Haryana, India

ABSTRACT: Developing a game using Java or any other technology may look haunting at first but then you realize that programming language   made things easier for the developers. Few decades back , Games were developed either using C or C++. However, things started changing when mobile gadgets took over as the most popular platform for gaming. Games for Android and IOS are still being developed even today using Java. Award-winning games like Angry Birds was developed using Java. In this paper we will be mainly focusing on developing games using JAVA FX which is a library of Java that is used widely to build rich Internet applications. The quality of a game is dependent on the technology which we are using for the game development and the technology is directly proportional to the availability of libraries and engines in the language.. JavaFX has various good libraries for developing games .The GUI applications that are developed using this JAVA library can run across several platforms such as Desktops, Mobile Devices, TVs, Tablets, etc.

KEYWORDS: JavaFX, Rich Internet Applications , GUI, game development

I. INTRODUCTION

JavaFX basically is library of Java that is used to develop Rich Internet Applications. The GUI applications that are developed using this JAVA library can run across several platforms such as Desktops, Mobile Devices, TVs, Tablets, etc.Decades Back, for creating these types of GUI Applications using the language Java , the programmers were dependent on Libraries such as AWT and Swings. After the rise of JavaFX, Java based programmers can easily build GUI applications that has rich content.

Rich Internet Applications are those applications that has same look and feel as the Desktop applications. These applications provides user with the feeling as if they are working on the Desktop applications[1][2]These application provides users with nice visual experience than the normal web applications. For running Rich Internet Application we do not require any additional softwares.

For the game development , you'll need two things

- **WINDOW**

You will need a window to hold everything together.

First and foremost thing you need is to control the window of your game. You should be able to minimize, maximize, and change the size settings according to your preference. Apart from binding the things together, the window will be serving as a center of command for the code that will

be managing the user interface of the application'.

- **ANIMATION**

Certainly ,   a game without any animation[3] and effects would be boring. Animation is the heart of any game. The more the animation , the more eye catching it will be. The animation is one of the most versatile , and one of the most powerful tool the developer can use to communicate with the application.

II. RELATED WORK

Just like all the other successful languages , Java also is evloving so as to survive with the current market trends. This also applies to its libraries. The most impacting evolution was found in its GUI frameworks. Because of several limitations of its GUI Framework AWT were replaced by swings and then by JavaFx because it offers flexibilty and is not platform dependent. Now -a- days JavaFx is used to create various applications that can run on various mobile devices including the Android's and IOS. This worldwide acceptability is causing an urge to work on it more and more.

III.ADVANTAGES OF USING JAVAFX

JavaFX was developed to feed applications with nice GUI features like smooth animation, audio and video playback, and styles using the Cascading Style Sheets (CSS).

For decades, application developers were using Swing tas highly efficient toolkit for developing graphical user interfaces (GUIs) and providing interactivity to Java applications. Although, few of today's most popular GUI features could not be easily implemented by using Swing. So there arises the need of a java component that can cater the need of today's popular Graphical User Interface.
Then came JavaFx into picture.[4][5][6]

- **FXML**
FXML is nothing but an XML-based markup language that helps the developers to develop a user interface (UI) in a JavaFX application. The UI is developed separately from the application logic . Swing have never offered such a approach for building a user interface. FXML enables the UI developers to easily maintain the complex user interfaces.
- **JavaFX Scene Builder**
To enable developers to create the layout of the applications, there is a design tool in JavaFX named as JavaFX Scene Builder that will help the developers to build their designs.
Developers can drag and drop UI components to the Content pane, and the corresponding tool will generate the FXML code that in turn will be used by the IDE's such as NetBeans and Eclipse.

- **CSS Support**
Cascading style sheets contains style definitions that depicts the look and feel of UI elements. The usage of CSS in applications created by JavaFX is just similar to the usage of CSS in Hyper text markup language. With CSS, the developers can easily choose , customize and create themes for JavaFX and scene graph objects.
Using CSS enables the developers to separate the application logic from setting the Look and feel of the application.

- **JavaFX Media Support**
JavaFX is famous for the media support it provides to its users. With this the developers can enhance their web applications by providing support for audio and video files
Media functionality is supported on all the platforms where JavaFX is supported.

- **Animation**
Certainly , a game without any animation and effects would be boring. Animation is the heart of any game. The animation is one of the most versatile , and one of the most powerful tool the developer can use to communicate with the application
Animation brings dynamism and a spectacular look to the User interface of the applications. Animation in a Swing is not a straight forward approach. In contrast to this , JavaFX helps developers to animate things such as graphical objects in their applications with more ease because of the presence of scene graph underlying the platform and the APIs that are built to server the purpose of animation.

- HTML Content

For long decades, Swing developers wanted to render the HTML content in their JAVA applications using swings but JavaFX brought this feature into JAVA by providing a graphical user interface component that has a web view and full browsing functionality that is supported in HTML.

IV. EVOLUTION OF JAVAFX

To answer how JavaFX is evolved we have to go back to the time when AWT was used for the purpose of UI designing .These were depreciated because of several reasons. The Abstract Window Tool Kit have Limited graphical Interfaces and UI controls. The Major drawback using AWT was that the application generated using these components were highly platform dependent[7]. They display different look and feel when targeted to different operating systems .Moreover the look and feel was also constant and can not be changed by the developer. One more major drawbacks of the AWT components is that they are heavy weighted components which consumes much memory. Hence these are used less.

To overcome the limitations of AWT swings came into picture. Swings were built on AWT. These are the lightweight components. These have a pluggable look feel[8][9]. The application generated using these components are highly platform independent. They display same look and feel when targeted to different operating systems. These swings possess richer set of components than AWT . Since they are light in weight , Processing of the swing components is very fast . These support only 2-Dimensional graphics with no or less support for animated objects. These are only compatible with Desktop applications not for the mobility support. Hence their use is reducing with time.
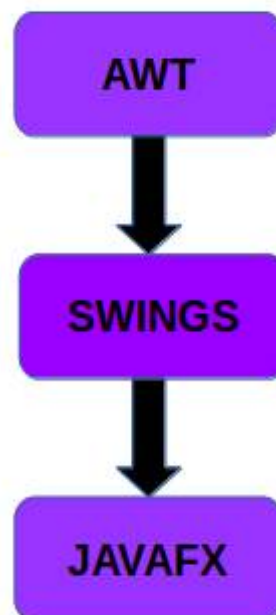

Figure 1 depicts how JavaFX emerged

Now, to have more features in user Interface that can cater to the needs of present world , there was an advent of JavaFX that has endless components that are rich in graphics and animations. Since these are pure Java based components they possess platform independence with higher mobility. They have constant and consistent look and feel across various operating systems. Moreover they support the facility of CSS that can help the developer to customize the look and feel of the components they are going to use. The Presence of CSS for the components can make it look far better , presentable and appealing to the end user. Java Fx provides support for animation that helps in making the graphical user interface interactive . These further supports 3-Dimensional graphics and view.These are best for creating some interactive applications and are adopted world-wide by many developers.

V. APPLICATION STRUCTURE OF JAVAFX

Basically a JavaFx application have three components in general [10]. These are : Stage Scene and Scene Graph

- **Stage**

A stage is a window that contains all the necessary objects of JavaFX application. This is represented by Stage class in the package named as javafx.stage. The primary stage is generally created by the platform itself. The created stage object is then passed as an argument to the start() method defined in the Application Class stage has two deciding parameters that helps in determining its position : Width and Height. Show() method is called for displaying the content of the stage.

- **Scene**

A scene helps in the representation of the physical contents of JavaFX application. It consists of the contents of a scene graph. The scene object is represented by the class Scene of package javafx.scene .At a moment, the scene object is added to a single stage only.
Developer can create a scene by instantiating the object of Scene Class. The size of the Scene object can be set by passing the values of parameters width and height along with the value of root node to the constructor.

- **Scene Graph and Nodes**

A scene graph is a hierachical data structure (tree-like) that represents the contents of a scene. While, a node is generally a graphical or visual object of a scene graph.
A node may include −Geometrical objects (2D and 3D) , UI Controls, Containers , Media elements
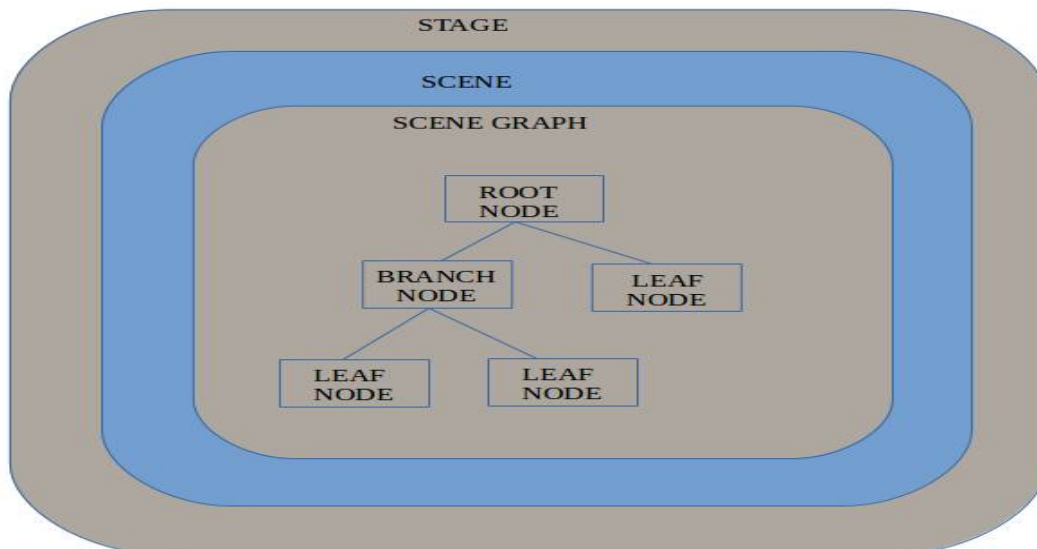Now a node can be of three types: -

Figure 2 depicting the application structure of JavaFX

◆ **Root Node** − The very first Scene Graph is known as the Root node.
◆ **Branch Node/Parent Node** − The node which can have child nodes are called as branch/parent nodes.
◆ **Leaf Node** − The node which can not have child nodes is known as leaf node. For example, Rectangle, Ellipse, Box are examples of leaf nodes.

VI. LIFECYCLE

Developing a JavaFX application[11][12] begins with the Application class, from where all JavaFX applications are extended. The main class will call the launch() method, which in turn will then invoke the init() method and then the start() method, and finally it will wait for the application to finish, and hence will call the stop() method. Out of these methods, the start() method is only abstract method that must be overridden.

VII. CONCLUSIONS

JavaFX is known to be the standard tool for GUI development JavaFX can be used to develop your own interfaces for enterprise apps. JavaFX connects to other Java APIs, and uses FXML to separate the UI design from the application logic. To make your GUIs more interactive, JavaFX provides many effects that can be applied. Different images , charts, and animation can be to your JavaFX projects. Although CSS can be integrated to customize your JavaFX interface designs. metrics in future with some modifications in design considerations the performance of the proposed algorithm can be compared with other energy efficient algorithm. We have used very small network of 5 nodes, as number of nodes increases the complexity will increase. We can increase the number of nodes and analyze the performance.

REFERENCES

[1]"JavaFX Overview (Release 8)". Oracle Corporation Docs.oracle.com.
[2]Cheshta, Dr.Deepti Sharma "JavaFX framework & Comparative Analysis" IJCSC Volume-8 Issue-2 March 2017 - Sept 2017
[3]Safrata, Pavel (2012-12-18). "More of JavaFX open-sourced". Oracle Corporation.
[4] "JavaFX Roadmap". Oracle Corporation. 2012-12-01.
[5]Meyer, David (2011-10-06). "JavaFX 2.0 arrives and heads for open source". ZDNet
[6] Marinacci, Joshua (2009-06-09). "Top 5 Most Important Features in JavaFX 1.2
[7]The Complete Reference ,Mcgraw Hill Education Pvt. Ltd , Ninth EditionY
[8]Kaul, Jeet (2008-12-16). "JavaFX — the road ahead"
[9]Yang, Chien. "3D Features Planned for Version 8". Oracle Corporation.
[10]"JavaFX FAQ". Oracle Corporation Oracle.com.
[11]OpenJFX Project". Oracle Corporation Docs.oracle.com
[12] JavaFX Supported Configurations". Oracle.com.

BIOGRAPHY

Ms. Cheshta is an M. Tech Scholar in the department of Computer Science & Engineering at Advanced Institute of Technology & Management, Palwal, Haryana, India

Dr. Deepti Sharma is H.O.D in the department of Computer Science and Engineering at Advanced Institute of Technology and Management ,Palwal, Haryana,India