HW 10 Question 1

Java is one of many programming languages that allows a user to support and create a graphical user interface, otherwise known as a GUI. Graphical user interfaces are extremely common, as we interact with them all the time unknowingly, such as going about using windows in general. Instead of executing and doing everything through command line, you have a convenient easy to use visual display to navigate through instead. These displays consist of various different components. These graphical components can be buttons, labels, windows, etc. All these elements of a graphical user interface allow a user to interact with different applications and programs without difficulty. The typical elements of a graphical user interface may vary depending on the application, but it can include but are not limited to the following: input controls, informational elements, and navigational elements. Input controls include things such as buttons, dropdowns and lists, check boxes, and text fields. Informational elements consist of items such as icons, labels, banners, or notifications. Lastly, navigational elements compose of sidebars, menus, and breadcrumbs.

The programming language of java is no stranger to graphical user interfaces. It has had graphical user interfaces capabilities since the mid 1990’s, starting its original graphical user interface library, Abstract Window Toolkit (AWT). AWT is a collection of Graphical user interface widgets, which is now a part of the Java Foundation Classes, also known as JFC, the standard API for providing a graphical user interface to a program. It was an initial effort to create a set of cross platform GUI classes. However it was too clunky to use. It has now been superseded by Swing, which was launched in 2007. Since Swing came about, it has become the primary graphical user interface technology for Java. It is a newer graphical user interface technology, which has been written from ground up and allows for more versatile graphics better graphical user interface construction. Compared to AWT, Swing has better features, better compatibility, and better design. Besides being able to bring a better set of widgets, it draws its own widgets rather that relying on the OS’s high level user interface module, which AWT did. Swing allows you to either use the native platforms look and feel, or a cross platform look that mirrors on various different platforms. Both technologies still exist in Java now, so at times it can be easy for one to mix them up. Swing however, is being superseded as well, eventually by JavaFx. JavaFX has also been around for a similar period of time, but began to gain popularity once Oracle purchased Java from Sun Microsystems. JavaFX differs from Swing in the sense that it has an entirely different set of components and terminology from Swing, as well as some key features. Such features include support for CSS, or cascading style sheets, which is a web component for embedding a web page inside an FX application, and the ability to play web multimedia content.

Regardless of which framework is newer or more recent, it is still in best practice to get familiar with both JavaFX and Swing, as Swing is still a widely used framework for many developers around the world. Also, there are many applications that incorporate Swing as well. Between all the frameworks, there are also similar components and usage in some areas, so its best to have an understanding of both frameworks for that regard.

Works Cited

*Developing a Java GUI*. (2019b, February 20). ThoughtCo. <https://www.thoughtco.com/gui-2034108>

Deitel, P. (n.d.). *Javaâ¢ How To Program (Early Objects), Tenth Edition*. OâReilly Online Learning. <https://www.oreilly.com/library/view/javatm-how-to/9780133813036/ch25lev2sec1.html>

Techopedia. (2011, November 25). *Abstract Window Toolkit (AWT)*. Techopedia.com. <https://www.techopedia.com/definition/3735/abstract-window-toolkit-awt>

<https://courses.cs.washington.edu/courses/cse331/11au/lectures/lect12-basicGUI.pdf>

LRS

1. I successfully got the assignment done. I initially had a deprecation error but that was resolved
2. It took a few hours to complete
3. It was easy in the aspect that everything is already there. Challenging in a sense of what kind of GUI to create
4. I wrote the program myself with help from your lectures and youtube tutorials
5. When I encountered issues, I came back to your videos and went on youtube
6. I learned about the different GUI frameworks in Java, and how to implement them and create my own

Program Output

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated