# A History of Java Graphical User Interfaces

Modern computing relies on Graphical User Interfaces, the now familiar stylized windows, buttons, emojis and so on that makes computing open to the masses. While operating systems implement these at a broad level, they are independent of one another, and one operating system’s GUI stylization differs from another operating systems. Programs that are then wanting to be run over each OS needed to code for that particular OS’s GUI. Java aimed to fix this when it was released in 1995 with the Abstract Window Toolkit (AWT). AWT utilized the native OS’s subroutines to implement GUI elements from the OS itself thus ensuring that a Java based program run on a Windows OS looks like a Windows based program. However, if that same Java program is then run on a Linux OS it will appear as a most other Linux OS programs do. AWT allowed early developers to code in a more generic sense and Java would then ensure that the GUI style and actions performed by the program were run by the native OS. The result is a bit more resource intensive as the program itself is asking the OS to create the various implementations of items coded into the program.

To solve this along with modernizing Java in a quickly changing early internet world, Java teamed up with Netscape who had developed what at the time was called the Internet Foundation Classes. Together in 1997 Java and Netscape developed this into what was called the Java Foundation Classes, which again had a name change to what today we know as Swing. Swing solved the heavy OS dependencies by coding the classes and components within the Swing package itself. Though that’s not to say Swing couldn’t utilize the OS’s components. This is because Swing extends AWT’s functionality allowing Swing to piggyback off the advances of AWT and improve on them where it could. Often in fact both AWT and Swing are implemented within the same GUI window.

Swing for its part ushered in the early days of the web and helped create the thriving environment that let the web flourish. The next broad technological development would however stand on the back of Swing’s successor, JavaFX. JavaFX was released in late 2008 shortly after the release of the first iPhone. With the success of the iPhone and later competitors in the Android ecosystem Java found itself again in demand from developers searching for a platform agnostic means of creating the latest programs. JavaFX helped bridge that early transition phase that saw many people moving away from a desktop environment to live their lives through a mobile device, able to consume media anywhere at moment’s notice. JavaFX sees continued development to this day in a more stable computing environment.

References

Israr, F. (2020, May 15). *Introduction to abstract window toolkit (AWT) in Java*. OpenGenus IQ: Computing Expertise & Legacy. Retrieved November 6, 2022, from <https://iq.opengenus.org/basics-awt-java/>

Wikipedia contributors. (2022, January 16). Abstract Window Toolkit. In *Wikipedia, The Free Encyclopedia*. Retrieved 20:34, November 6, 2022, from <https://en.wikipedia.org/w/index.php?title=Abstract_Window_Toolkit&oldid=1066085313>

Wikipedia contributors. (2022, June 9). Swing (Java). In *Wikipedia, The Free Encyclopedia*. Retrieved 20:34, November 6, 2022, from <https://en.wikipedia.org/w/index.php?title=Swing_(Java)&oldid=1092233091>

Wikipedia contributors. (2022, October 5). JavaFX. In *Wikipedia, The Free Encyclopedia*. Retrieved 21:09, November 6, 2022, from <https://en.wikipedia.org/w/index.php?title=JavaFX&oldid=1114283412>

*Difference between AWT and swing in java*. GeeksforGeeks. (2022, January 10). Retrieved November 6, 2022, from https://www.geeksforgeeks.org/difference-between-awt-and-swing-in-java/