**Tools and technologies & explanations:**

* Source control (git)
  + Over 30 commits to master. Github repository reference in project. Different branches for different features merged into master when feature is completed. Only ever two features at a time. Git command line was used with the commands available to do all the source control features
  + Source control was used at the start of the project where the initial commit was start, to minimize project set-backs by backing up the project in source control
  + Later on in the project Source tree was used to manage source control
* Java
  + Written in Java, a high-level object orientated programming language
* JavaFX
  + A software platform which was used in this project to deliver a desktop based application that can run across a wide variety of devices
* FXML Views
  + FXML views with code in them to display appropriate content that’s formatted in an elegant way. Set by scene builder. The FXML variables in the view are passed to the controller.
* Scene Builder
  + A visual layout tool, the API used, that lets users display interfaces without hard-coding in many lines of code, more easily
  + Variables were named in scene builder then taken across to the controller, using controller skeletons offered in the scene builder modules
* FXML Controllers
  + The back-end of the code that deals with the user interaction of the interfaces. FXML variables are passed to this and local variables are passed around by use of controller factories. Examples of this is programming in a button press taking you to the next “scene” of the same “stage” or starting and displaying a game timer to see how long it takes for the logos to be guessed
  + Included a initialize function, basically inherits initializable, to allow the interface to be dynamic rather than static, allowing buttons to be pressed and actions to occur in real-time
* Netbeans
  + The Java IDE used for this project. One of the most popular IDEs for java which listed is (not exhaustive) Eclipse, Netbeans and IntelliJ IDEA. It is completely written in Java, with many modules available, such is refactoring, debugging and compiling and building programs.
* Github
  + The open source repository used for source control in this project