**QUBIX PROJECT REPORT**

**Hangman Havoc**

**Introduction:**

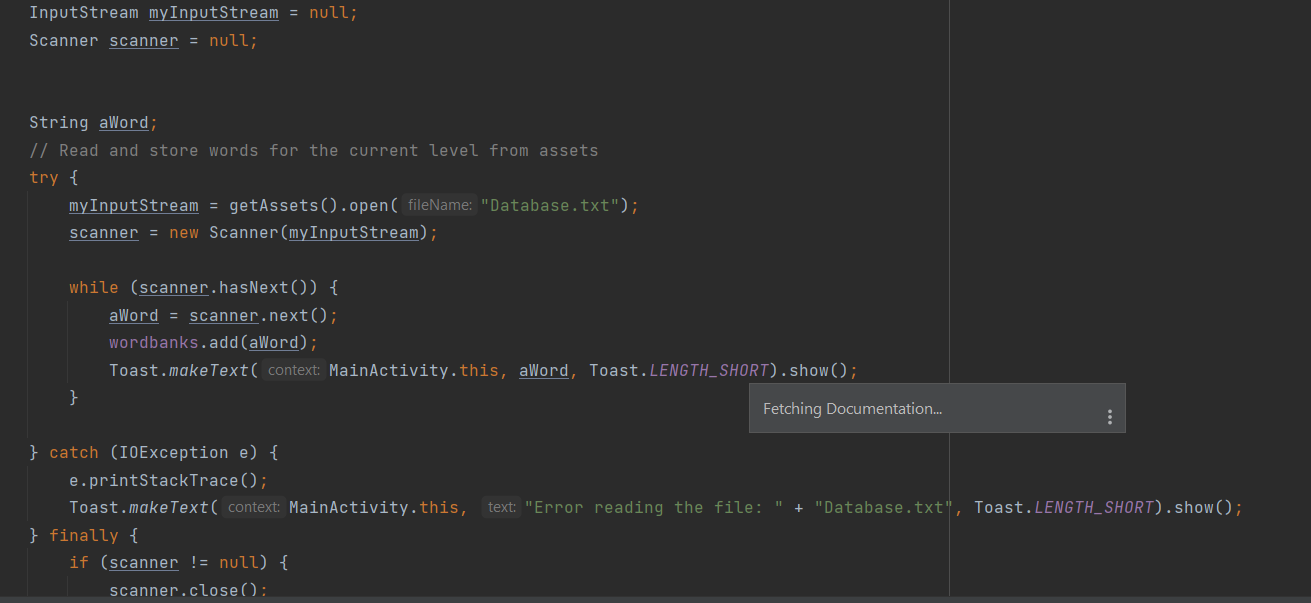
Hangman Havoc is a classic word-guessing game where players attempt to guess a hidden word by suggesting letters one at a time. The project involves developing the Hangman game using a GUI based software development like android studios.

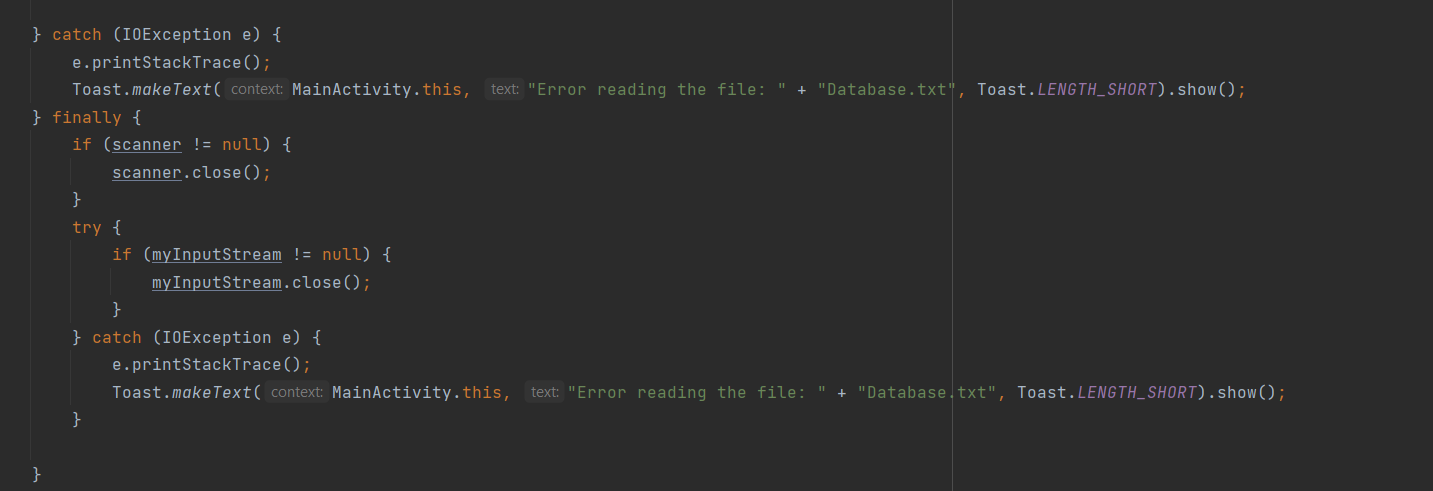
**Game Overview:**

The Hangman game's code encompasses various components and functionalities that make up the game's structure. The code initializes the user interface for the game, including TextViews for displaying the word and an EditText for input. It also lays the groundwork for animations, although some animation-related code is currently commented out. The game reads a list of words from a file named *Database.txt* and stores them in an ArrayList called *wordbanks*. These words are randomly selected for players to guess, ensuring a dynamic and engaging experience.



The code includes functions for managing various aspects of the game, such as initializing a new game, checking if a letter is in the word, displaying the letters guessed by the player, and resetting the game when needed.The code incorporates animation objects that can be triggered during specific game events, such as correct letter guesses, game loss, or game reset. It's essential to uncomment and integrate these animation elements for a visually appealing game.  
  
A snippet of some of the code:   
  



**Team Roles:**

The development of Hangman Havoc involved collaborative efforts from team members with distinct roles:

Melusi: Responsible for crafting the core game logic and defining gameplay rules. This team member intricately designed the game, including word selection, word display, input handling, and win/loss conditions, while ensuring adherence to Hangman's fundamental rules.

Nompilo: Managed the game's word database, curating an extensive list of words suitable for gameplay. They organized and stored these words in the *Database.txt* file and wrote the code to access and utilize this word bank.

Mthabisi: Oversaw the creation of the game's graphical interface and functionality using Android Studio. This included designing interface components, implementing visual elements, and ensuring a seamless user experience. Additionally, they played a vital role in project management, ensuring team timelines and milestones were met.

**Requirements for Running the Game:**

* Several essential components must be in place to successfully run Hangman Havoc on your Android device or emulator. You will need to have Android Studio, the Android SDK, and a Java Development Kit (JDK) installed on your development environment. Ensure that you have configured either an emulator or a physical Android device for testing. It is crucial to have the game's assets, including the "Database.txt" word list file, properly stored in the project's assets folder. Additionally, make certain that the layout resources for UI components precisely match the code references within the project. If the game requires specific permissions, consider specifying them in the AndroidManifest.xml file. Pay attention to the API level compatibility, ensuring that it aligns with the version requirements set for the game. With all these prerequisites met, you will be ready to compile and build the game code within Android Studio and subsequently run the game by clicking the "Run" or "Debug" button, allowing you to enjoy the Hangman Havoc game seamlessly on your Android platform.  
  The path to the mainActivity: "Myapp2\app\src\main\java\com\example\myapp2\MainActivity.java"

**Limitations:**

The limitations we faced as first-time android studios users was trying to understand the android ecosystem and how it functioned which was a bit challenging. Trying to design a user-friendly and visual appealing interface since we were not well versed in the UI design principles. Thoroughly testing the app on various Android devices and screen sizes to ensure compatibility was one of our major issues. We had trouble running the app and we could not handle errors properly.  
  
We suggest that with a better-updated version of Android Studio, it should function as long as the files are in order  
  
Challenges: We wanted to add Levels, with each level having unique themes and a countdown timer, but we could not find a way to integrate our code to Android Studio, so we had to remove them and make the game a little bit simpler.

**References:**

Google. (n.d.). Retrieved July 26, 2023, from the Android Developer Website: <https://developer.android.com/studio>.

Florence, M. (n.d). *History to history class: The origin of classroom games.* The Chipper. <https://chschiper.com/2019/12/from-history-tohistory-class-the-origin-of-classroom-games/>.

Pirple. (2019, April 25). Java Project – Hangman App (Android). <https://youtube.com/playlist?list=PLgTkNlNsy9gVCkeaoudJJWr4P3Gc-AV7f&si=BsU3MF3gGiGoR43x>.