Typing Game Project Description

As someone eager to improve my typing skills while simultaneously enhancing my programming knowledge, I embarked on creating a typing game for my class assignment. This project not only served as a learning tool for Python programming but also offered a practical benefit in honing my typing abilities.

Initially, my vision was to design a game akin to the popular Google Dinosaur Run. However, I encountered several challenges in replicating its design, leading me to pivot towards a different, yet equally engaging concept. The final product is a dynamic game developed in Pygame, a Python library known for its flexibility in game development.

In this game, words cascade from the top of the screen, and the player's objective is to type them accurately before they reach the bottom. It's a thrilling test of speed and accuracy, where each correctly typed word increases the player's score. The game features a responsive interface, tracking the player’s progress and allowing the opportunity to restart by pressing 'R' upon losing.

The project primarily utilizes Python, particularly for its game loop and graphical user interface (GUI), ensuring a seamless and interactive gaming experience. Additionally, I incorporated C++ for backend functionalities, such as processing a string of words and exporting them into a text file. This file then feeds into the Python-based game, creating an ever-changing array of words for the player to tackle.

Overall, this typing game stands as a testament to the power of practical application in learning new skills, both in programming and typing.

Objectives

Objective I: Proposal

Programming Language Selection:

- Python: Chosen for its simplicity and extensive libraries, particularly Pygame, which facilitates game development. Python is ideal for developing the game's core loop and GUI.
- C++: Used for its efficiency in handling file operations. It generates the text file containing the list of words used in the game.

Project Implementation Story:

- Conceptualization: Inspired by the need to improve typing skills and explore Python's capabilities, the idea was to create an engaging typing game.
- Evolution: Initially modeled after Google's Dinosaur Run, the project evolved into a unique game where words rain down from the top of the screen in Pygame, challenging players to type them correctly before they reach the bottom.

Objective II: Design Document

GUI Design:

- A straightforward and interactive interface where words dynamically appear and descend from the top of the screen.
- Display elements include the current score, typed word, and game over screen.

Classes and Structures:

- The Python script encapsulates game mechanics, including word movement, scoring, and player input.
- The C++ program focuses on file operations, creating a source file for the words used in the game.

Flow Chart: In text(i was having trouble with the websites making a flowchart and uploading it

Start
Initialize Game:

- Initialize Pygame.
- Set up the screen dimensions.
- Load words from the text file.
- Initialize game variables (like score, word speed, fonts).

Main Game Loop: This is the central part of your flowchart.

- Process Input: Handle keypress events. Include sub-steps for:
  - Quitting the game.
  - Handling backspace for typing.
  - Entering a word (check if typed word matches any on-screen word).
  - Restarting the game after a game over.
- Update Game State:
  - Move words down the screen.
  - Check for collision with the bottom of the screen.
  - Add new words at random intervals.

- Render Graphics:
    - Clear screen.
    - Draw words, dinosaur, and score.
- Check Game Over Condition: If any word collides with the dinosaur, set the game to game over state.

Game Over:

- Display "Game Over" message.
- Option to restart by pressing 'R'.

Restart Game: If 'R' is pressed, reset the game variables and return to the "Initialize Game" step.

Database Design:

- Not applicable as the game does not require a database. Word data is managed through a text file generated by the C++ code.