Object-Oriented Programming Project

< Fuzzy Typers >

01286121 Computer Programming

Software Engineering Program,

Department of Computer Engineering,

School of Engineering, KMITL

By

65011365 Miki Ajiki

65011367 Music Auyeung

65011528 Sirapop Tuntithanakij

Overview

Typing is an important skill that is relevant in the lives of many people in modern society where technological development is more prominent than ever. Ranging from sending messages to writing computer programs. Typing is present in most modes of digital communication. Therefore learning and improving typing skills through practice is essential to effectiveness.

Objective

The objective of the program is to create an aesthetically pleasing way to practice typing skills. We aim to increase enjoyment and motivation by approaching the task through the lens of gaming. With a score-based reward system that fuels players' senses of achievement and a scoreboard which encourages healthy competition. This program will help users familiarize themselves with aspects of their own typing and pave the way for growth. Whether it is for improving accuracy or speed of typing, you can achieve your goals without boredom.

Gameplay

Mode 1: Typing Trials

This game mode is inspired from classic gameplay of typing games. A random word is generated from our existing dictionary for the user to type while a timer runs for 60 seconds. While the time is up, the program records the score (amount of words the user is able to type) and the average speed (words per minute)

Mode 2: Ticking Time Bomb

This game mode is a more challenging twist. The user is to type the random word within five seconds. If the task is accomplished, the timer resets and the game continues. However, if the task is failed and time runs out, the game is over.

Practice Mode:

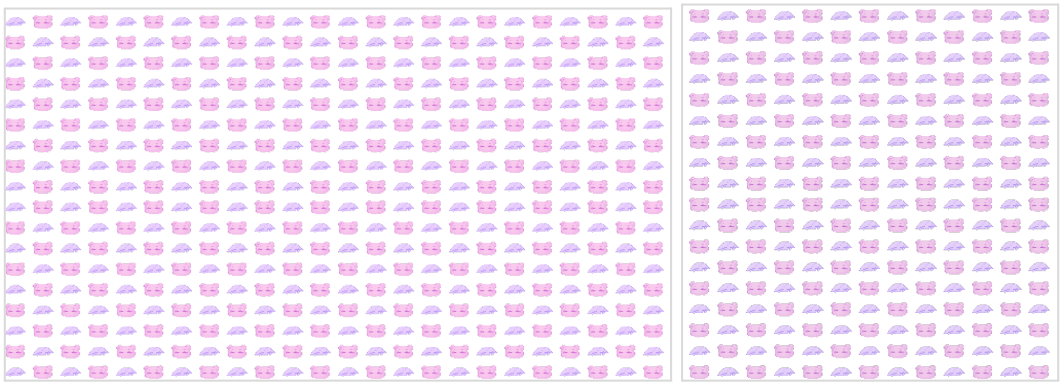
This mode is for pressureless practice with no timer and the user can choose when to exit.

Features

1. **Dictionary**: Generating and randomizing words for typing
    a. The words are stored in a json file
    b. Function for reading json
2. **Compare**: User's input vs. given word
    a. Collects the number of mistakes user makes
    b. Collects number of completed words to **calculate score** and **WPM**
3. **Timer**: runs for certain period of time
    a. The game stops when time is up
    b. Varies for different game modes
4. **Leaderboard**:
    a. Saves scores by generating a json file
    b. Collects usernames and displays
5. **GUI**:
    a. Character Sprites and Animation

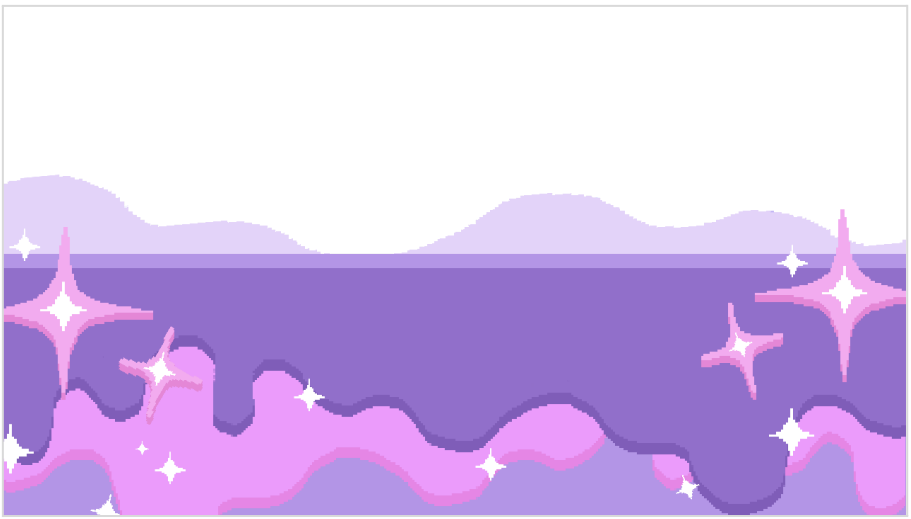
Character Spritesheets

b. Background Images



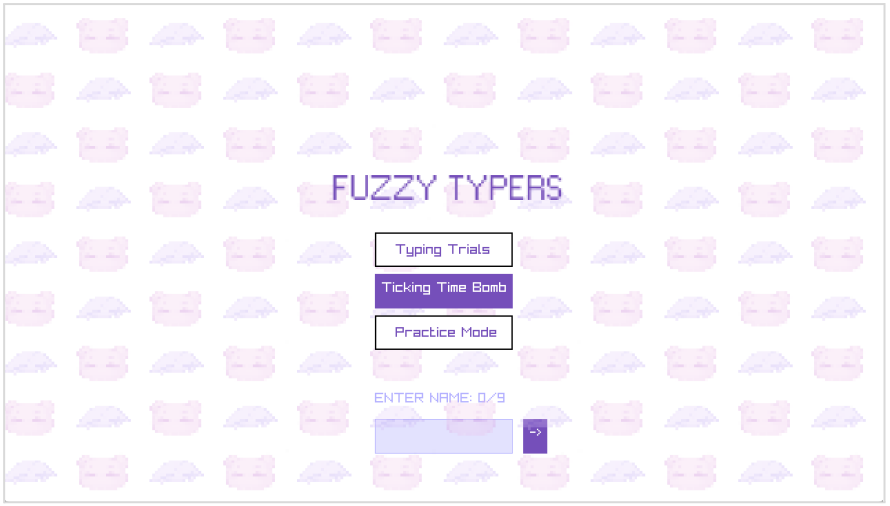
c. Game Graphics and Animation

Window Icons

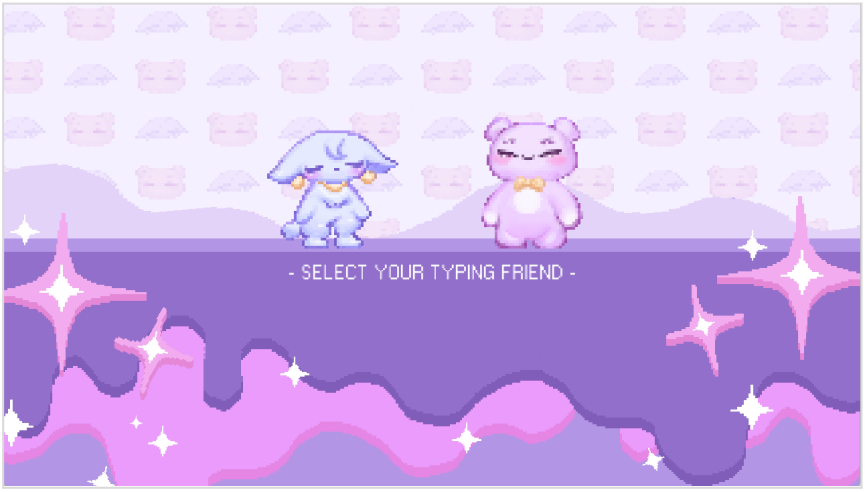


Gameplay background

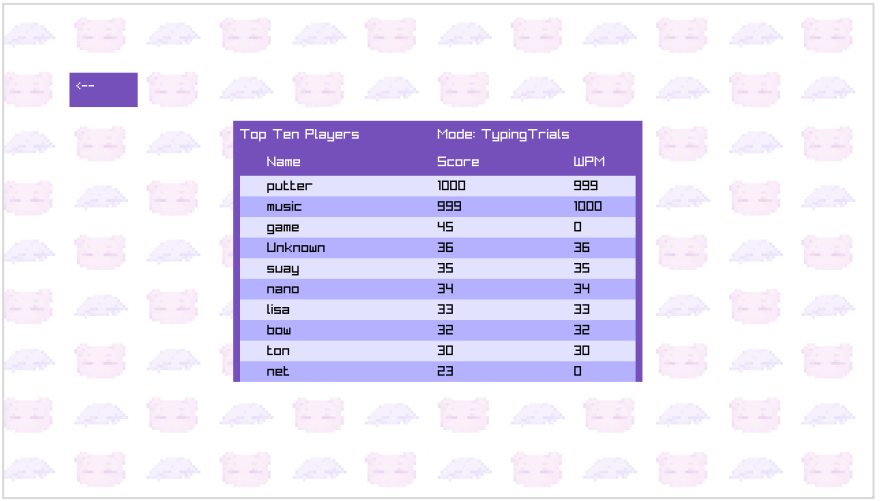
d. Main Menu

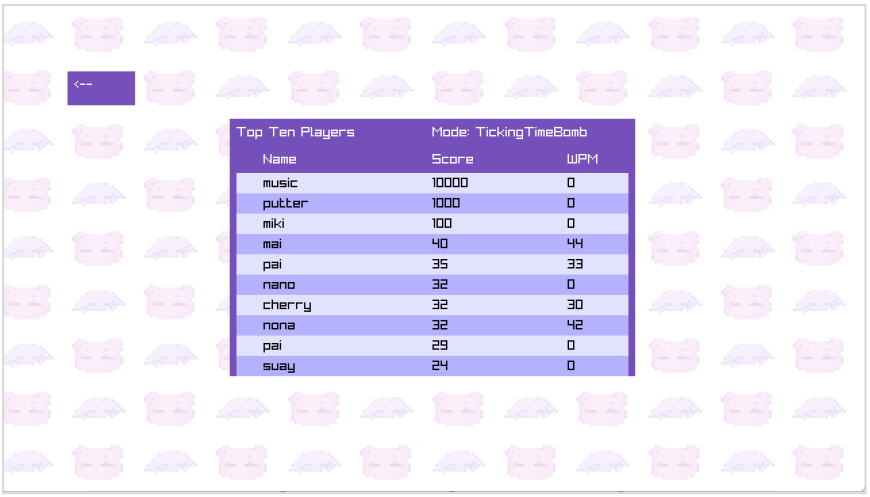


e. Character Selection



f. Leaderboard





g. Audio: Background Music and SFX

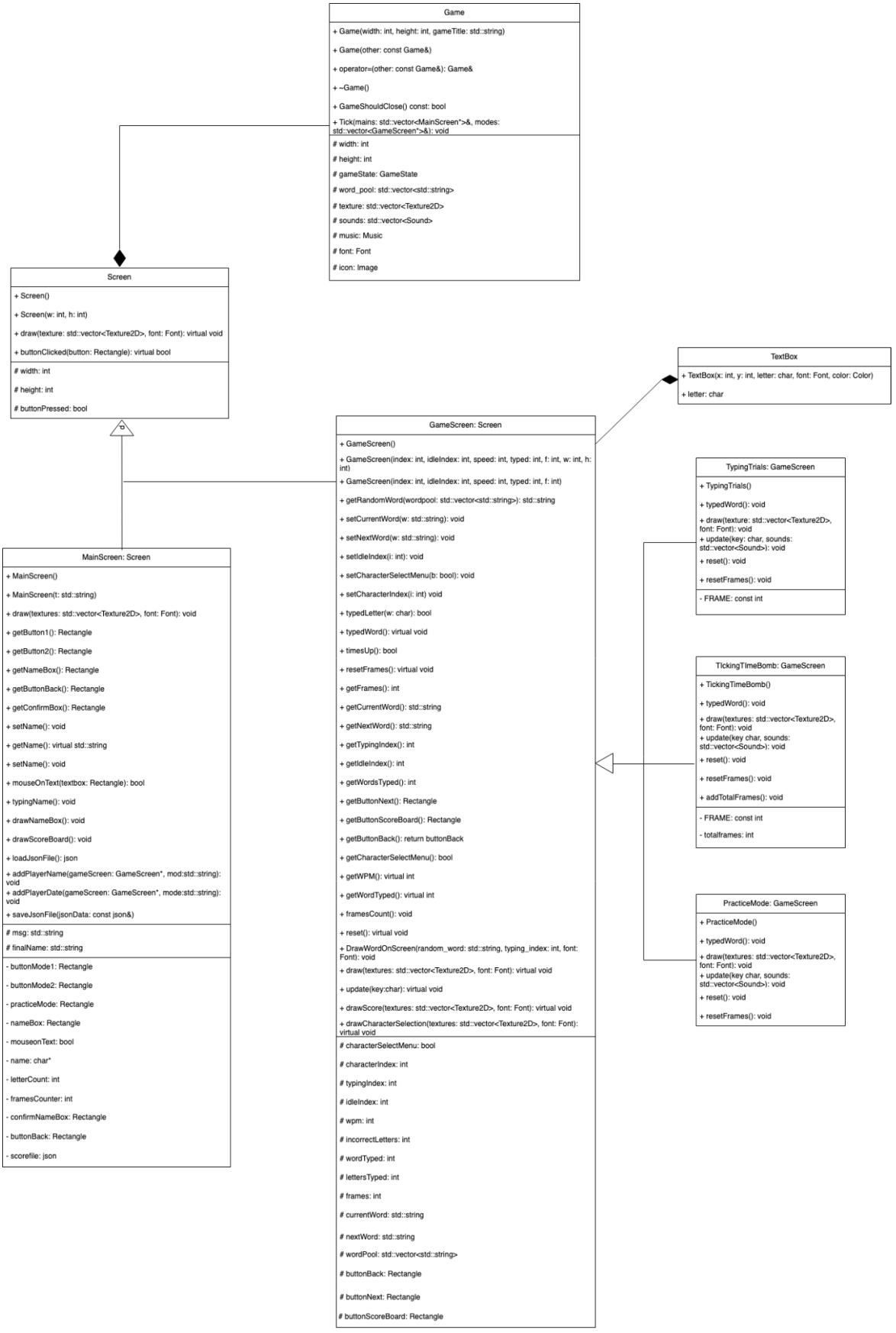
Background Music: Chillpeach - Listen Closely

SFX: https://mixkit.co/free-sound-effects/

External Resources

1. Raylib
2. nlohmann/json

Class Diagram



Individual Contributions

All of our team members participated in the brainstorming process and came up with a brief list of tasks to complete. Each day, we each focused on individual tasks. However, we collaborated heavily and weren't solely responsible for the tasks in the following lists, which is why they may appear more than once.

65011365 Miki Ajiki

- Gameplay: Ticking Time Bomb, Practice
- Leaderboard: Reading Username, Generating Scores, Displaying Scores
- Dictionary: Json file reader

65011367 Music Auyeung

- Gameplay: Typing Trials
- Leaderboard: Reading Username
- Screens: Main Menu, Scores
- GUI: Color Themes, Game screens design
- Class Diagram

65011528 Sirapop Tuntithanakij

- Gameplay: Typing Trials, Practice
- Design: Enums for Program Navigation
- Dictionary: Json File Reader
- GUI: Character Design and Animation, Gameplay Graphics Design, Audio