

## **Assignment 4 -Java Concurrency Report**

### **Introduction**

The task was to create multithreaded Java game while ensuring thread safety and enough concurrency for it to function well. When the user presses start a specified number of words should fall from the top of the panel at different speeds. When the words reach the bottom red zone, they disappear. The user is supposed to type a word correctly before it reaches the red zone, upon which the word disappears and a new one starts falling. The game continues for a specified number of times. To restart the game the user presses the “End” button, and to end the game the user presses the quit button.

### **Methods and Classes**

Made changes to the WordApp class and the WordPanel.

#### **Modifications**

##### **1.Threadd class**

I created an inner Threadd class which implements Runnable in the WorPanel class. This class is responsible for making the words fall at different speeds. Inside the WordPanel, I created a thread for each word that falls.To ensure thread safety and proper concurrency I made use of synchronized run method in the Threadd inner class.For this problem synchronized seemed to be enough to ensure thread safety amongst shared variables.

##### **2.Added Updater method**

Updater method takes in the parameters score ot type Score, play of type Boolean, missed of type JLabel and totalWords of type int. The purpose of this method is to update the count of missed words (ie words that reach the red zone ),the caught words and the score the user gets.

### **Ensuring thread safety**

Java swing is thread safe, but its safety could have been affected by calls from it. I ensured that the buttons pressed and the text input do not affect the thread safety of swing .I used synchronized shared variables. I also used synchronized run() method in the Threadd class.

### **System Validation**

Tested the code using different inputs and took note of the behavior of the game.

### **Code Architecture**

In modifying the code and implementing some functions I made sure to follow the Model-View - Controller pattern for user interfaces in my code architecture. The model comprises the WordDictionary class, the array of WordRecords and the Score class. The controller are the threads, the inner Thread class and Updater methods that I used to alter the view, the animation code I implemented in the run method in WordApp. The view is the GUI.

### **Additional Features**

When end button is pressed, there is a pop up telling the user to press ok to restart the game

When user presses Quit button, a popup asks the user to press ok to quit.