REPORT:

KHLPAL002

PALESA KHOALI

CSC2002S

ASSIGNMENT 2

Introduction

This report explores concurrency programming by exploring a game of falling words, The game was catching words by typing them correctly before they reached the green bar if the word is typed correctly it disappeared and the player is awarded a score. The score is based on the number of letters in the words spelled correctly, if the word reaches the given limit, the game stops.

Additional classes

The only additional class I added was TheGame, this is where the entire game is happening. The class checks the words as they are falling and also keeps tabs on the state of the game "paused, stopped or inplay". This class also makes use of all the threads in my code.

Changed classes

1. Wordpanel was updated to make sure the panels get updated every time and repainted, background was changed to pink.

2. WordApp was changed to include the main, this is where any listeners to my code was placed

3. WordRecord was kept the same

A description of concurrency features

I used the synchronized methods to prevent thread interference, this method uses intrinsic locks to provide synchronization so that the words could appear more than once on the same screen and not break my code.

Ensuring thread safety

I used synchronized methods to prevent thread interference, and avoid memory consistency errors, deadlock should also be covered by synchronized methods

Conforming to Model-View-Controller pattern

The user of the game interacts with the controller by entering words and in return the controller manipulates the model by allowing the word to disappear and thus updating the view of the game which the user then sees. This is a clear indication that the MVC is conformed