

Name: Prosper Mambambo

Student Number: MMBPRO002

# CSC2002 Report

## **A description of the HungryWordMover class**

The code consists of Hungry Word Mover and Hungry Word. Hungry Word Mover is an exact copy of the Word Mover class but accepts Words of type Hungry Word. this is responsible for creating the thread that causes the word to move horizontally. Hungry Word is a copy of falling Word were I changed MaxY to be MaxX(to set the limit of the X value to check if the Hungry Word doesn't go out of Bound) and changed the SetY method into a setX method and vice versa. the drop in hungry word calls the set x method and increment x to make the word move horizontally along the screen. The word moves horizontally in Hungry Word Mover when drop is called. Hungry Word Mover works with Hungry Word to make the word move across the screen. Hungry Word Mover increments the x value of the Word causing it to move and also is responsible to check if the word is out of bounds if so, it causes the word to move back to the start generating a new word. Within the hungry word it's also where collision testing is performed. The collisions are checked in a way that it loops through the falling words array checking if the hungry word collides with any of the words. If so, it reset the word and increments missed counter.

## **A description of any other classes you added (and a justification of why they were necessary)**

I added the hungry word class which works together with the hungry word mover class. The hungry word class is a class that has drop incrementing x causing the word to move horizontally. Its also the class that's responsible in checking if the word is not going out of the screen. If it is, It generates a new word to manage

that. The Hungry word class is what's used to generate the Hungry word that is then assigned its thread by hungry word mover. This class also contains additional attributes which are width and height which are used in the hungry word mover class to check for collisions by calling getters for width and height and are used in the game panel to call setters for width and height setting the value of the height and width of the word in hungry word.

**A broad description of modifications you made to the existing classes and how you ensured thread safety.**

Some of the modifications I made to existing classes was mainly adding a function in the catchword to detect the hungry word when it's typed to ensure the hungry word is removed and its score is caught and caught is incremented. I had to override the set words function to obtain the hungry word so I can access it in the catch word class and check if it matches with target.

Create a function in typing tutor to allow it to create the threads for hungry word. It creates hungry word. Creates a thread for hungry word and ends of by starting the thread. Added the method to the start button so when the button is clicked the method executes. Create hungry words array and hungry words mover array which are created as in the main method and initialized in the create hungry word threads. Added a functionality to disable the textbox entry when pause is clicked to prevent cheating and enable when start is created.

Added an extra argument to the game panel constructor to allow it to receive hungry word so its capable of drawing hungry word to the string. In game panel that's where I changed the color of the hungry word. Set the width and height of the falling words and hungry words (to use them later to check for collisions in the hungry words mover class).

**Any race conditions that you identified in the original code (for extra credit).**

All the methods in the catch word were not synchronized as well as 2 methods in the falling word methods which are increase speed and reset speed. This methods

are shared and were contributing to major race conditions within the code. I solved the race conditions by synchronizing the methods

## Gitlog

```
willcommits@DESKTOP-7PFOM2K:/mnt/c/Users/pmamb/Videos/typingTutor$ git log | (ln=0; while read l; do echo $ln\: $l; ln=$((ln+1)); done) | (head -10; echo ...; tail -10)
0: commit d0f8f99e63c2213f90008ef3ae43979b36640144
1: Author: willcommits <pmambambo2@gmail.com>
2: Date: Sun Aug 28 15:32:46 2022 +0200
3:
4: working Hungry Word and collisions,duplicate checking working
5:
6: commit 22c92663b7fa2fdbf2e6782e6b2a8f1467d9eaa7
7: Author: willcommits <pmambambo2@gmail.com>
8: Date: Sat Aug 27 23:03:55 2022 +0200
9:
...
31: Author: willcommits <pmambambo2@gmail.com>
32: Date: Mon Aug 22 22:17:35 2022 +0200
33:
34: Working Hungry Word Motion
35:
36: commit 7a2bf36231429bb997b9618c27848e9befabb18f
37: Author: willcommits <pmambambo2@gmail.com>
38: Date: Sat Aug 20 21:57:56 2022 +0200
39:
40: original Version
```