**Auguste Kiendrebeogo**

CMSC 411 SUMMER 2022

SCOREBOARDING PROJECT

**USER MANUAL**

1. **OS and compiler used**

The program was designed on **Windows 10** using the **Java programming language**. The program was coded using **Eclipse IDE for Java Developers Version: 2021-06 (4.20.0) Build id: 20210612-2011**.

I downloaded [Eclipse IDE](https://www.eclipse.org/downloads/packages/release/oxygen/3a/eclipse-ide-java-developers) and installed it as a normal program on my Windows 10 computer. For the GUI, the JTable for Java works fine on my side without any specific updates. However, in case there are issues, the Eclipse Marketplace found in the Help section of the Eclipse IDE can be used to **freely** download the necessary libraries. It is free for all the libraries I used in this project.

Graphical user interface, text, application

Description automatically generated

1. **Running the files on Eclipse**

After installing Eclipse, it is possible to run the files. To run the files on Eclipse, you should create a Java project first. This is done by going to **File 🡪 New 🡪 Project** as you can see in the screenshot below.

Graphical user interface, application

Description automatically generated

Then, you select **Java Project** and click **Next**.

Graphical user interface, text, application, email

Description automatically generated

We then give the project a name. I must also mention that by default my Eclipse IDE uses the **JavaSE-16 Execution environment JRE**. We unselect the **Create module-info.java file** then we don’t have to deal with creating a Java module. We click the **Finish** button, and the project will be created.

Graphical user interface, text, application, email

Description automatically generated

Once the project is created, we go to the new created project that will appear on the **Package Explorer** on the left side of the IDE. We then need to create a Java class by making a right-click on the **src** folder, selecting **New**, and then **Class**. **We name the class the same way the file is named. We can then copy the content of the file to the corresponding class. We do that for all the files.**

Graphical user interface, text, application, email

Description automatically generated

**In this project, the SimulatorDriver JAVA file contains the main function.** To run the program**,** click on the **Run** button (see screenshot below).

**Graphical user interface, application

Description automatically generated**

1. **How the program works**

The program has two (2) parts:

* **One part with no GUI**. Here, the user enters the values on the console of the compiler as you can see below.

Graphical user interface, text, application, email

Description automatically generated

* **The second part comprises a GUI** and you can see a sample screenshot below.

Graphical user interface, application, table

Description automatically generated

On the console part, the user is asked to enter the number of Functional Units needed for the program. The user is not allowed to enter more than 32 integer registers and/or 32 floating-point registers. The program will ask the user to re-enter the data if such a violation occurs.

Once the user finishes entering the inputs, the GUI will appear. Here, the user will need to click multiple times on the button **> Increment Clock** to see the different steps of the scoreboarding. The user can pause and see the different results that appear on the screen.

Note that the program reads a text file containing the MIPS code in the **SimulatorDriver JAVA file**. Therefore, to read the corresponding file, you should **manually** modify the file String in the program that is actually presented in the program as “**static String file = "data.txt";**” the text file should be put in the corresponding folder for the program in **eclipse-workspace** in the program files of the computer. The text should be put in the outside part of the project folder as you can see in the following screenshot.

Graphical user interface, table

Description automatically generated

**DISCLAIMER**:

The code is based on previous work done by “[thePairedElectron](https://github.com/thePairedElectron/ScoreBoardingViz)” (<https://github.com/thePairedElectron/ScoreBoardingViz> ) cited in the project files. As such, it contains different copied parts but most of the work was personally done to adapt it to this project.

With more time, I can make this program flow better and smoother. Hopefully, you will find it interesting.