Software Implementation

Sergio van Amerongen
0952200

Stefan Cloudt
0940775

Daan de Graaf
0956112

Robert van Lente
0953343

Tom Peters
0948730

Berrie Trippe
0948147

Responsible:
Berrie Trippe
b.trippe@student.tue.nl

March 30, 2016

1 Introduction

Having designed our software it is time to actually implement it. This document will describe the implementation of the design described in the document Software Design. It will however, not contain the actual code itself, nor any information of how specific methods and classes are implemented, but only describe the general conventions of the designed software and a user manual of our machine in relation to the software. It will also briefly mention how we have implemented timers in our program.

2 Naming/Java conventions

Since we implement our software with Java as our programming language it is important to use the standard Java conventions for our code. We will therefore explain the most important conventions we stick to:

2.1 Proper indentation and brackets.

To keep the code clean and readable it is important to use proper indentation when nesting code.

Example:

```
if (...) {
    while (...) {
        indentation
    }
    indentation
}
```

As is visible we also use the standard conventions for the use of curly brackets, making loops and methods structurized.

2.2 Naming conventions

To clearly distinct variables, constants, methods and classes we use the Java conventions for naming them. Constants are written in uppercase letters with an underscore between the words.

Example:
public static final int $MOTOR\_TURN\_SPEED = 120$;

Variables are written according to the lowerCamelCase convention, where the first word has a first lowercase letter and the other words are written with first upper case letter.

Example:
public final GyroSensor gyroSensor;

Methods are also written in lowerCamelCase convention.

Example:
```
private void run() {
    indentation;
}
```

Classes are written according to the UpperCamelCase convention.

Example:
```
public class MotorLeftState extends MotorState {
    indentation;
}
```

3 User Manual

3.1 Machine preparation

Before using the machine and running any programs on it, the machine should be sufficiently prepared to ensure proper operation. The machine should be positioned on a level surface, in a not too disruptive lit room (e.g. no very powerful lights pointing directly or indirectly at the machine).

3.2 Starting the machine and program

After the machine has been prepared properly as described in the previous section, the machine can be booted up. This is done by holding down the center button on the main brick until the display shows the booting screen of LeJos. After the machine has been booted, the display will show a main menu. To start the sorting program, the user should navigate to the Programs item by using the left and right button on brick and then pressing the center button. Then the user should navigate to the Main program and press the center button to start it. The machine will now start the actual program.

3.3 Selecting a mode

After the program has been started, the user will be prompted with a mode selection screen. The user can choose a mode with the left, center and right buttons. The modes are explained in the machine design document. After a mode has been chosen, sensors and motors will be calibrated, so parts will start to move during this process.

3.4 Filling the input tube

After program initialization is done, the user is allowed to open the input tube by rotating it horizontally and pulling the yellow part on the top side of the tube. The user may need to keep the tube in place with his other hand to prevent it from moving back vertically while opening it the tube. The grey bar attached to the yellow pull can now be rotated vertically to move it out of the way. Ensure that the black/grey pin is inserted in the bottom part of the tube. The tube can now be filled with black and white discs. After the tube has been filled, the tube must be closed by moving the yellow pull back to the top of the tube and attaching the black connectors to the side of the tube. Then the tube must be rotated back vertically to its position above the moving arm. The black/grey pin on the bottom of the tube can be removed. The pin can optionally be inserted in the small blue bar on the left side of the machine to store it.

3.5 Start the sorting operation

The center button must be pressed to start the sorting operation

3.6 Pausing and resuming the sorting operation

While the machine is sorting, the sorting operation can be paused by pressing the center button.While the machine is paused, the sorting operation can be resumed its sorting operation by pressing the center button again.

3.7 Stopping the sorting operation

While the machine is sorting, the sorting operation can be stopped by pressing the down button. The operation will be stopped completely and the user can start the operation again if desired.

3.8 Shutting down the machine

In order to shut down the machine, one must go back the mode selection menu, and afterwards press reset.

3.9 Troubleshooting

The machine supports detection of different kinds of errors that may occur during the sorting operation. When such an error occurs, a light will start to flash red or orange and the machine will show a message on the display describing the error that occurred. When the error is fatal (The led blinks red) has occurred, the user can return to the mode selection screen by pressing the down button. The user is expected to have resolved the problem before returning to the mode selection screen.

4 Timer

Some warnings depend on time. The errors and warnings like gyro does not stabilize, disc did not reach basket and disc took longer than average to reach basket depend on time. In code this is implemented by a Timer class which gives the time between two method calls. However we need to set the thresholds in order to be able to decide whether or not an error or warning should occur. To do this we made a simple app, TimerApp which sorts the discs and measures the time between the moment that the motor started to turn and the time the disc fell onto the lever. The simple app then calculates the average, minimal time and maximum time.

5 Conclusion

this chapter has explained the naming conventions important for our software implementation phase. Since the Java conventions are known we have used and explained these in this file. We have also discussed the use of timers in our software implementation since these were not mentioned before. Lastly we have included a user manual that describes the user interaction in relation to the software. The next document will describe how we test our implementation and validate correctness.