

Second project report

Candidati:

Giovanni Liboni

Matricola VR363021

Enrico Giordano

Matricola VR359169

Alberto Marini

Matricola VR359129

Alessandro Falda

Matricola VR359333

Contents

I	Introduction	2
II	Graphical interface	3
1	Design pattern for graphical interface	4
2	Swing bug	5

Part I

Introduction

This project implements an asynchronous system consists in three principal parts:

1. a station, that controls velocity of cars;
2. some automatic cars, that set their velocity randomly during the ride and set the ideal velocity thanks to the station;
3. some manual cars, that set their velocity randomly during the ride and receive “break” message (but they are not obliged to slow down).

During the execution, is instantiated 50 manual cars and 40 automatic cars and change the speed of each car randomly during the runtime. After a simple ride, the cars decide randomly if they will be exit or not.

Part II

Graphical interface

The graphical interface is built with *swing* Java library and consists in two JFrame called “WallGraphic” and “DebugInterface”. The first interface (WallGraphic) is a representation of the situation, composed by a station, some automatic cars and some manual cars. The second interface (DebugInterface) is a textual console that show the program flow, cars display and state and station state. This interface has three option:

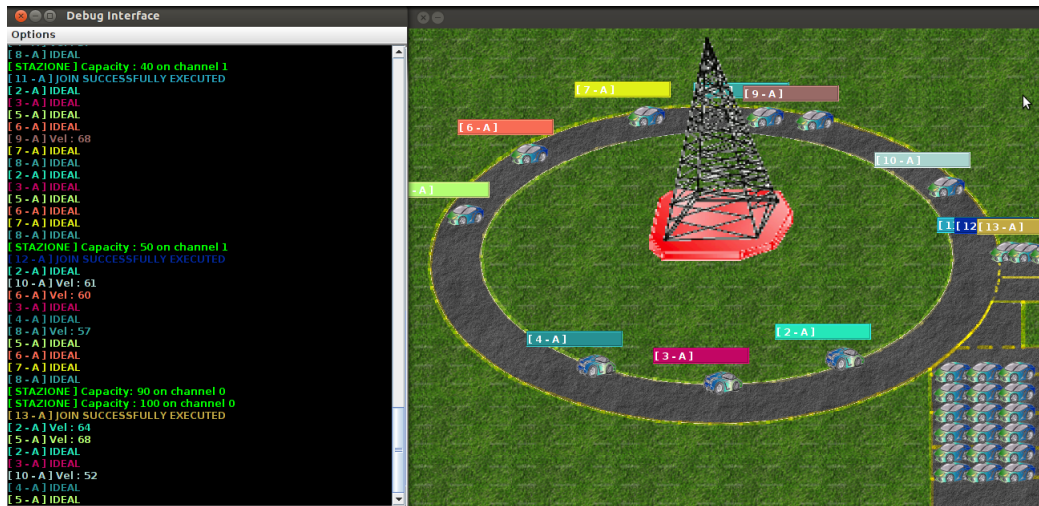


Figure 1: ScreenShot

1. “pause”, that stops the program flow acquisition;
2. “resume”, that resume the program flow acquisition (and enable autoscroll);
3. “watch”, that disable or enable the autoscroll of the scrollbar.

Only 200 rows are shown in the screen; after 200 rows, DebugInterface clean itself and delete old rows. This was made because after 200 setText in the same JLabel probably will crash the program.

This is composed by a JFrame that contains a JPanel that contains a JLabel with a black background and a text that was updated by station and cars display.

WallGraphic is composed by a JFrame that contains a JPanel with a “wallpaper” (the circuit). This JPanel contains in different levels (Z ordered) a station (that is a JLabel with an image) and some cars (that are a JLabels with an image). The cars move theirselves in asynchronous way into the ride in 10 different direction (in order to approximate the elliptical path).

The car label have a superior JLabel that contains its id and the car type: the “M” letter represent the “Manual car” and the “A” letter represent the “Automatic car”. When a car change its direction (X orientation), it changes its image.



Figure 2: car label

Every car is a graphical thread that execute the move() method in order to move itself during the ride; when a car exit to the circuit, the thread dead. The velocity is graphically represented by a thread sleep during the movement.



Figure 3: car label (different orientation)



Figure 4: car label

This is the Z order of the JLabel:

-1: background image;

0: station;

-1: cars;

In this way, the cars will pass graphically behind the station and on the background image.
The graphical class are contained in the “*graphics*” package.

1 Design pattern for graphical interface

Every class of this project must use these graphical interface, so it was implement a **Façade pattern**. In fact there is a general class, “ScenarioGraphic”, that include in itself every graphical instance; in this way, when a class must use a graphical instance, simply call the ScenarioGraphic methods and ScenarioGraphic can modify the graphical instances.

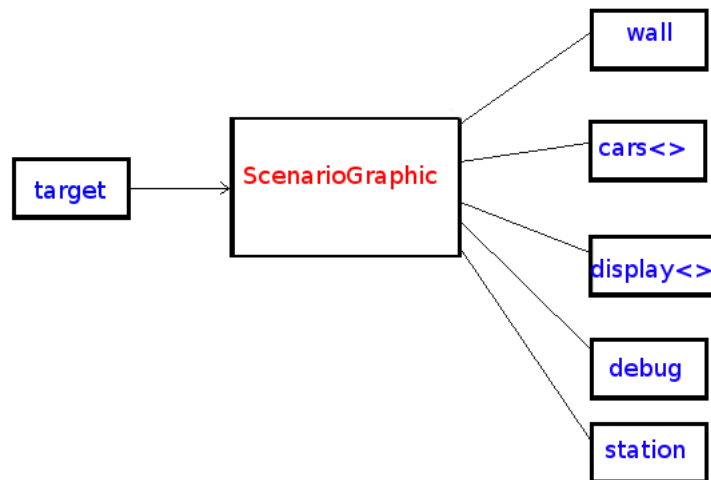


Figure 5: Façade pattern

2 Swing bug

```
Exception in thread "Thread-1" java.lang.NullPointerException
at javax.swing.BufferStrategyPaintManager.flushAccumulatedRegion(BufferStrategyPaintManager.java:4
at javax.swing.BufferStrategyPaintManager.copyArea(BufferStrategyPaintManager.java:351)
at javax.swing.RepaintManager.copyArea(RepaintManager.java:1232)
at javax.swing.JViewport.blitDoubleBuffered(JViewport.java:1621)
at javax.swing.JViewport.windowBlitPaint(JViewport.java:1590)
at javax.swing.JViewport.setViewPosition(JViewport.java:1135)
at javax.swing.plaf.basic.BasicScrollPaneUI$Handler.vsbStateChanged(BasicScrollPaneUI.java:1044)
at javax.swing.plaf.basic.BasicScrollPaneUI$Handler.stateChanged(BasicScrollPaneUI.java:1033)
at javax.swing.DefaultBoundedRangeModel.fireStateChanged(DefaultBoundedRangeModel.java:365)
at javax.swing.DefaultBoundedRangeModel.setRangeProperties(DefaultBoundedRangeModel.java:302)
at javax.swing.DefaultBoundedRangeModel.setValue(DefaultBoundedRangeModel.java:168)
at javax.swing.JScrollBar.setValue(JScrollBar.java:463)
at graphics.DebugInterface.run(DebugInterface.java:50)
at java.lang.Thread.run(Thread.java:701)
```