

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	4

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.

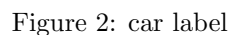
Graphical interface

The figure shows a NetLogo simulation environment. On the left, the command monitor displays a list of commands for agents 5 through 26, all set to 'my velocity'. The status bar at the bottom left indicates '5 - A | Vel - 65'. The main view on the right shows a circular road with several cars. Each car is labeled with its agent ID and 'A' (e.g., [5-A], [20-A]). A central tree structure is visible in the middle of the road. The cars are colored in various shades of blue and green, and the road is a dark grey color.

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.

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.

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.



3



Figure 3: car label (different orientation)



Figure 4: station 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.

In the park, there is a lot of “dead car” that simply occupies the park. This was made because is a graphical strategy to fill a space that may seem empty or messy.

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.

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)
```

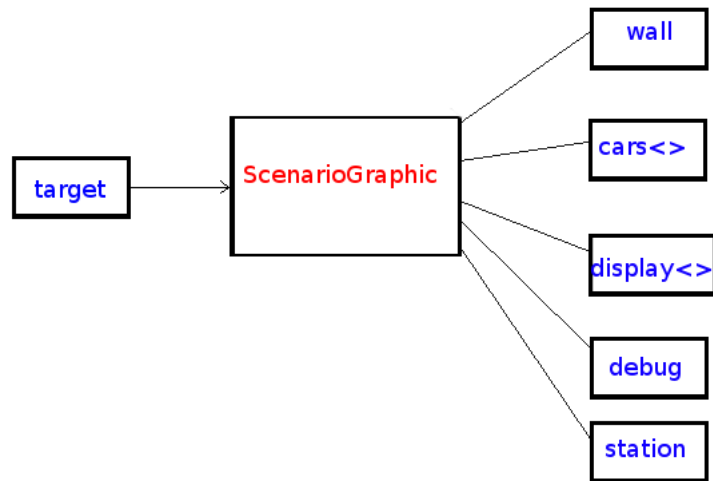


Figure 5: Façade pattern

at java.lang.Thread.run(Thread.java:701)