Comp 110-003 - Assignment 7:

Main and Console Input

Date Assigned: Wed Nov 2, 2011

Completion Date: Mon Nov 14, 2011 (midnight)

In this assignment, you will learn about main, console input, if statements, and while statements. At the end of the assignment, you will define the user-interface shown in the Figures, by building on the code you wrote for the previous assignment.

If you are starting the assignment after while loops have been covered and feel confident about your programming skills, then implement part 4 directly after reading all three parts. Otherwise implement the intermediate steps in parts 1, 2, and 3 before attempting part 4.

Your program should reuse as much as possible the code from the classes and interfaces of the previous assignment. Feel free to extend the classes and interfaces defining a vehicle.

This is the hardest assignment given so far in terms of the amount of coding you have to do in the given time. So start on it immediately. The submission date assumes that by the end of the lecture on Nov 7, we will cover the parts of chapters 10, 11, 12, and 13 that are needed for the assignment. If this does not happen, we will postpone the submission by two weekdays.

Part 1 – Modifying AVehicle

Add two editable int properties called OffsetX and OffsetY to your AVehicle/Vehicle class/interface. The values of OffsetX and OffsetY specify the distances from the x-axis to the left-most edge of the Body rectangle is and from the y-axis to the top-most edge of the Cabin rectangle, respectively.

Part 2 – AVehicleDriver

Implement a console-based main class that displays the information about the current size and location of each of the parts making up a vehicle. It takes as input two integers, the width and height of the vehicle (Figure 1). Based on these values, it creates a vehicle and prints the information shown in Figure 1 (left). In addition to printing the information, it also displays an ObjectEditor edit window for the vehicle.

```
AVehicleDriver [Java Application] C:\Program F
Please enter the width:
100
Please enter the height:
30
Car information:
----------------
Cabin information
Upper-left corner:
X: 35 Y: 10
Width: 50 Height: 15
Body information
Upper-left corner:
X: 10 Y: 25
Width: 100 Height: 15
Front tire information
Upper-left corner:
X: 15 Y: 32
Width: 15 Height: 15
Rear tire information
Upper-left corner:
X: 90 Y: 32
Width: 15 Height: 15
```

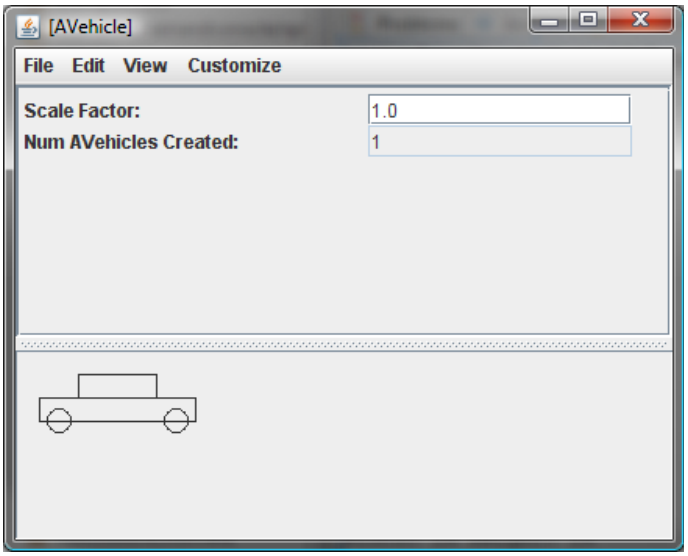

Figure 1

For this part of the assignment, you need to look at chapter 10. You should implement and use the Console class given in chapter 10, that is, make a Console class and paste into it the Console code in the notes.

Part 3 – Scaling and Moving the Vehicle

Extend your AVehicleDriver as follows. After the user enters the initial width and height of the car, the user is prompted for a command to either move or scale the vehicle.

To move the vehicle, the user first types “move” and presses Enter (Figure 2 (top)). Following this, the user is prompted, one at a time, for the new values of the OffsetX and OffsetY properties of your vehicle. After the user enters the offsets, you should adjust the position of the vehicle and print all of the information in Figure 1 (left) again. NOTE: to show the car in the new location in the ObjectEditor window, you must click View/Refresh in the window (Figure 2 (bottom)).

To scale the vehicle, the user first types “scale” and presses Enter. Following this, the user is prompted for the new value of the Scale Factor property of your vehicle. After the user enters the new scale factor, you should scale your vehicle accordingly. Moreover, you should print all of the information in Figure 1 (left) again. NOTE: to show the car in the new location in the ObjectEditor window, you must click View/Refresh in the window.

To quit the program, the user enters “quit” as the command.

```
AVehicleDriver [Java Application] C:\Program Files\Java\jre1.6.0_02\bi
Please enter a command (move, scale, or quit):
move
Enter the new OffsetX value:
100
Enter the new OffsetY value:
50
```

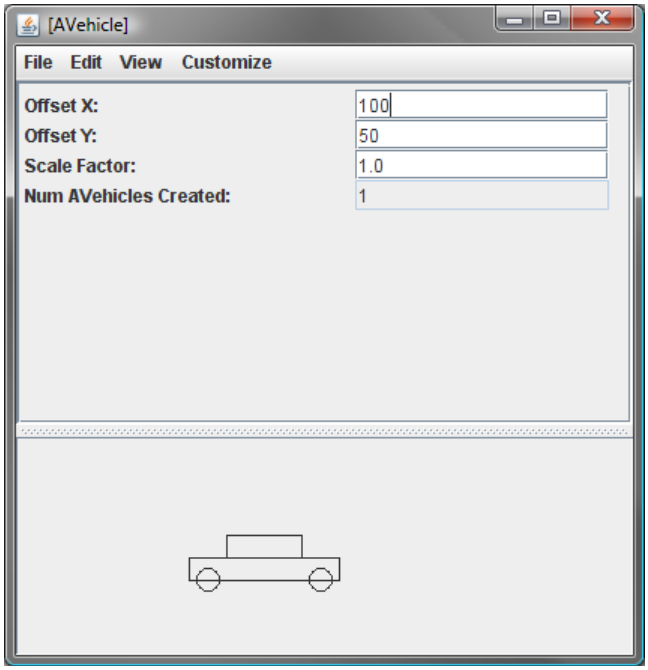

Figure 2

Part 4 – Multiple Scale and Move Commands

Extend the code you have written so far for AVehicleDriver so that a user can enter an arbitrary number of move and scale commands. After each command, you should update your vehicle and print all of the information about the vehicle (as in part 1). Also, after each command, if you invoke the Refresh command from the View menu in the ObjectEditor window, the car should be shown correctly – at the right location and the right size.

Figure 3 below shows what happens when a user first moves the car to (100,50) and then scales it by a factor of 2. In the figure, the initial car had a width of 100 and a height of 30.

Part 5 – Bonus (10%)

Extend your program to accept a second version of the move command that works as follows. Instead of entering "move" and the OffsetX and OffsetY values on separate lines, the user enters them all at once on one line as follows:

```
AVehicleDriver [Java Application] C:\Program Files\Java\jre1.6.0_02\bin
Please enter a command (move, scale, or quit):
move
Enter the new OffsetX value:
100
Enter the new OffsetY value:
50
Please enter a command (move, scale, or quit):
scale
Enter the new scale factor:
2
Please enter a command (move, scale, or quit):
```

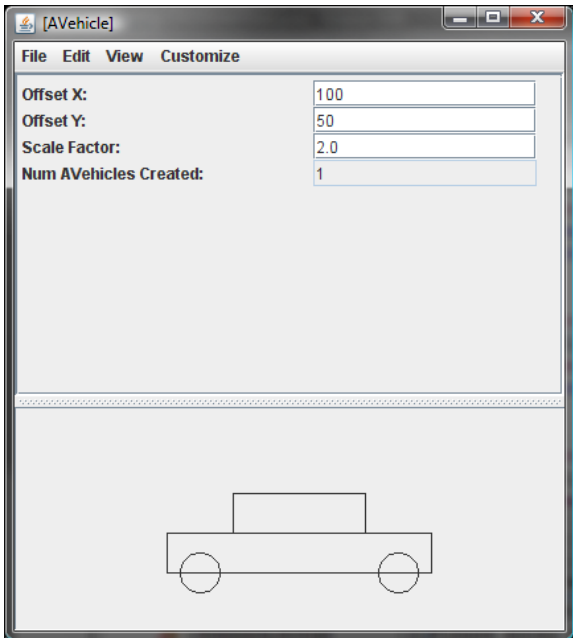

Figure 2

```
m OffsetX OffsetY
```

where OffsetX and OffsetY are integers and a single space separates "m" from OffsetX and another single space separates OffsetX and OffsetY. The result of this command is equivalent to using the original move command and entering "m," the OffsetX value, and the OffsetY value on separate lines. For example, the command

```
m 100 200
```

moves the car to the location at which the left-most edge of the car is 100 pixels away from the X-axis and the top-most edge of the car is 200 pixels away from the Y-axis.

The original move and scale commands should still work.

Part 6 – Bonus (10%)

Modify the above command to work with arbitrary number of additional space characters between “m” and OffsetX and between OffsetX and OffsetY. The following commands should all work:

```
m 100 200
m          100 200
m 100              200
m               100                  200
```

All of them should move the car to the location at which the left-most edge of the car is 100 pixels away from the X-axis and the top-most edge of the car is 200 pixels away from the Y-axis.

The original move and scale commands should still work.