Rush 01 – Piscine Java

Java FX & Sockets

Summary: Today you will implement an actual client/server game with a full-scale interface

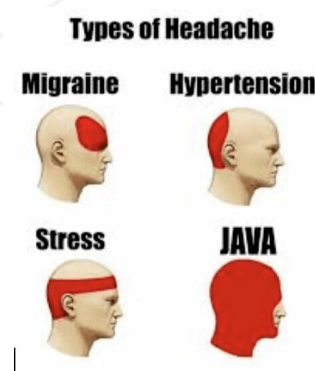# Contents

Chapter I

Foreword

Minimum requirements for Java junior developers:

1. Java Core (type system, OOP, Collections, IO/NIO, Reflection, Generics, Exceptions)
2. Maven
3. JUnit
4. Mockito
5. SQL
6. JDBC
7. Spring Framework (Context, JDBC, MVC, AOP, Security, Data Jpa)
8. HTML/CSS/JS (JQuery)
9. Tomcat
10. Servlets/JSP
11. Freemarker
12. REST API
13. Postman, curl, Swagger
14. English
...

Chapter II

Instructions

- Use this page as the only reference. Do not listen to any rumors and speculations about how to prepare your solution.

- Now there is only one Java version for you, 1.8. Make sure that compiler and interpreter of this version are installed on your machine.

- You can use IDE to write and debug the source code.

- The code is read more often than written. Read carefully the document where code formatting rules are given. When performing each task, make sure you follow the generally accepted Oracle standards

- Comments are not allowed in the source code of your solution. They make it difficult to read the code.

- Pay attention to the permissions of your files and directories.

- To be assessed, your solution must be in your GIT repository.

- Your solutions will be evaluated by your piscine mates.

- You should not leave in your directory any other file than those explicitly specified by the exercise instructions. It is recommended that you modify your .gitignore to avoid accidents.

- When you need to get precise output in your programs, it is forbidden to display a precalculated output instead of performing the exercise correctly.

- Have a question? Ask your neighbor on the right. Otherwise, try with your neighbor on the left.

- Your reference manual: mates / Internet / Google. And one more thing. There's an answer to any question you may have on Stackoverflow. Learn how to ask questions correctly.

- Read the examples carefully. They may require things that are not otherwise specified in the subject.

- Use "System.out" for output

- And may the Force be with you!

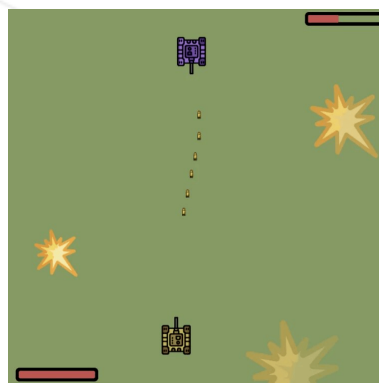- Never leave that till tomorrow which you can do today ;)

Chapter III

Exercise 00 : Tanks!

	Exercise 00
	Tanks!
Turn-in directory : *ex*00/	
Files to turn in : TanksClient-folder, TanksServer-folder	
Allowed functions : All	

JavaFX allows to create high-quality Desktop applications. Despite the fact that JavaFX is not very popular in corporate development, this technology is applied in a wide range of "private" solutions. JavaFX is also a good tool for exploring mechanisms of Java programming language.

Your objective is to implement the client/server game where the Socket server will allow two Java FX clients to play a tank game with each other. An example of the client-side appearance is provided below:

Thus, the game should enable two users to "drive" their tank and decrease the enemy's HP by shooting.

Game mechanics:
1. A tank is only able to move horizontally by pressing left and right arrow keys. Holding down the respective key results in a continuous movement in the respective direction.

2. A tank cannot move beyond the field edge.

3. A single space key press results in one shot. Makin a series of shots by holding down the key is impossible.

4. Hitting the target deducts 5 HPs from the enemy.

5. In the beginning, both players have 100 HPs.

6. The player is always at the bottom of the screen, whereas the enemy is at the top.

7. Tanks can only move if both players are connected to the server.

Additional requirements:

- Interface shall allow to connect to a specific server.

- When either player wins, a modal box shall appear with the statistics of shots, hits, and misses.

- These stats shall be stored in DBMS on the server.

- JavaFX client shall have an executable jar archive that can be launched like a normal application (by clicking on the file).

- README.md file shall be prepared with a set of instructions for application assembly and startup.

You can find all assets in the project page.