

PV260 Software Quality

Assignment 2 - Code Refactoring

Spring 2022

1 General Information

1.1 Dates

- Assignment start: 31.3.2022
- Dry Run submission (optional): 10.4.2022 23:59
- Dry Run notification: 13.4.2022
- Assignment deadline (**Tasks 1-5**): 18.4.2021 16:00, **Code reviews**: 24.4. 23:59

1.2 Submission

Please submit your solution as a txt file with link to a private Gitlab repository to the Homework vault (Odevzdávárna) called *[Java.Lasaris]Assignment 2: Refactoring*:

https://is.muni.cz/auth/el/fi/jaro2022/PV260/ode/java_lasaris_assignment_2_refactoring/.

- The solution should be submitted as Maven project.
- The tasks should be solved in the groups of two.
- Name of the submitted file should be as follows: *teamnumber-lastname1-lastname2-assignment1.txt*.
- Only one solution per group should be submitted.

1.3 Evaluation

The maximum points for this assignment is 15. The points will be distributed based on both fulfilling the functional requirements, compliance of the code with SOLID and Clean Code principles and proper git usage.

Additional points can be awarded for high quality solutions.

1.4 Project

This assignment works with the following project:

- TRON game: <http://www.mediafire.com/download/ucj53886o8a1164/Tron.zip>

2 Mandatory Tasks

The Tasks 1-4 are not meant to be completed in any particular order.

2.1 Task 1

- Make it possible to add more players to TRON easily (so 3,4,5... people can play at the same time each with their own uniquely coloured bike)
- Addition or removal of players should be as simple as possible, take into account the collision detection, controls configuration, rendering etc.
- Addition of player will be done offline, that is, to change number of players the source of the game will be altered and the whole code recompiled. There is no need to create any kind of game menu.

2.2 Task 2

- Adapt current code to enable using as much of it as possible as a generic engine for implementing other games.
- Provide a way for these new implementations to bind with the engine, e.g. interfaces that need to be implemented.
- Make the current TRON game such implementation of the engine.

2.3 Task 3

- Add the possibility to control any player by mouse (e.g. turn left/counter-clockwise or right/clockwise using left and right mouse buttons respectively)

2.4 Task 4

- Separate the state of the game from its graphical representation, that is have one class or package represent the Model and other the Presentation.
- The model must contain and provide access for the player bikes, events such as collision between two players happened and so forth.
- There is no dependency from Model to Presentation, only from Presentation to Model
- The model itself should not handle frame timing (`Thread.sleep()`), that is, the responsibility of calling some update method on the model is outside of the model, preferably in the extracted engine code from previous tasks.

2.5 Code review

- After the coding tasks deadline (not before!), create a merge request from your working branch to the master branch in the Gitlab. Assign members from the reviewer team as reviewers (you have to add them to your project first).
- The reviewer teams should do a code review for assigned merge requests
- The reviewer teams for each team are listed in Table 1.
- Additionally, assign seminar tutors - Stanislav Chren (xchren1) and Xuan Linh Phamová (xphamov2) to the merge request, as well). We will review your code, too and will also review the comments from other teams.

Table 1: Review assignments

Team	Reviews for	Team	Reviews for	Team	Reviews for
1	11	6	5	11	10
2	1	7	6		
3	2	8	7		
4	3	9	8		
5	4	10	9		

3 Optional Tasks

3.1 Task 5

- Use the extracted engine to implement some other game (e.g. pong, snake,...)