Assignment 1

Exercise 1

Task 1

Where are getters/setters or public variables used3 in your source code? Refactor nine4 of these cases, so that they are not necessary anymore and describe your refactoring/redesign.

Description of the refactorings

- [x] Board: remove method getBoard

This getter is unused, thus it can safely be refactored. Was done in ea632f1, before copying the checkers game.

- [x] Board: remove method getAllPieces

This getter is unused, thus it can safely be refactored. Was done in ea632f1, before copying the checkers game.

- [x] Board: move check if it's the last row to Row enum

The row enum can determine itself if it's the last row. This way we can hide that the coordinates are implemented with an enum.

- [x] Board: move check if it's the firstrow to Row enum

The row enum can determine itself if it's the first row. This way we can hide that the coordinates are implemented with an enum.

- [ ] Move:getCoordinatesBetween: RowIndex between in Row selber berechnen
- [ ] Move:getCoordinatesBetween: ColIndex between in Column selber berechnen
- [ ] Move:isJumpMove: rowDiff in Row berechnen
- [ ] Move:isJumpMove: colDiff in Col berechnen
- [ ] MoveLength: methoden von Row udn Col verwenden für die differenz

Exercise 2

Task 1

Google (used to?) ask their employees to spend 20% of their time at Google on a project that their job description does not cover. As a result of the 20% Project, Google now has services such as Gmail and AdSense. This is your occasion to have similar freedom. You can decide what to do next to your game:5 It can be an extension/improvement from any perspective, such as improved code quality

or novel features. Define your own requirements and get them approved by your tutor (especially in terms of load). Afterwards you must implement the requirements.

Description of the requirements

Task 2

During the analysis and design phases of this extension use responsibility driven design and UML (push to the repository the single PDF file including all the produced documents)

Responsibility Driven Design

CRC Cards

UML

Exercise 3

Task 1

Write a natural language description of why and how the pattern is implemented in your code.

Description of why and how the pattern is implemented

Task 2

Make a sequence diagram of how the pattern works dynamically in your code

Sequence diagram

Task 3

Make a class diagram of how the pattern is structured statically in your code

Class diagram