

**Department of Engineering and Computer Science**

## **Refactoring**

**Advanced Programming Practices**

**Fall 2018 Team 3**

Dmitry Kryukov

Ksenia Popova

Rodolfo Mateus Mota Miranda

Nikitha Papani

**Concordia University**

**Montreal, Quebec, Canada**

**November 2018**

For the build 3 we did several refactor operations.

We chose the refactor targets (described below), so that the code would be more understandable, readable and maintainable (easier to make changes, fixes and add new functionality).

Some of the applied refactoring operations (renaming, empty lines) are done so that the code follows the chosen code convention.

- Implemented the Observer pattern. All Views are implemented as an Observer and the main game logic (game.java) is implemented as an Observable.
- Separated the information panel from the main game file to the separate View.
- Moved the phase logic methods from the main game class to the Player model.
- Renamed some of the methods so that their names follow the chosen code convention (e.g. `init_data_structures` renamed to `initDataStructures`).
- Renamed some of the methods and variables so that their names are more understandable and describe the purpose of these variables and methods.
- Added empty lines to wrap methods and important part of the code, so it is easier to read and understand.
- Created the Test Suite to run Unit tests (instead of running them separately as it was done for the build 1 and 2).
- Deleted the blocks of code that were not used (commented lines or excessive code).
- Added the comments to the blocks of code to describe the processes in these blocks.
- Every class should have the comments at the top which clearly describes the purpose of the creation of class, author information & date on which the class is created.
- All the functions should be commented properly with details of parameters and return type.
- Javadoc API will be used for documenting the classes and functions in the

project.

- There should not be any unused function, variable or constants in the project.
- Inline comments should be included in the code for complex loops and/or functionalities to make it easier to be comprehended.