Deliverables for Iteration 2 – UI Mock-up, Controller Interfaces, Cucumber mappings (6%)

The deliverables for Iteration 2 are (a) UI mock-ups, (b) the specification of the Controller interface, and (c) mappings of Gherkin scenarios to your ***Quoridor*** application for each feature of Phase 1. You are also required to state who worked on which features. The deliverables are due on **Friday, October 11, 2019, at 23:30**.

See the Project Overview document for a general description of the ***Quoridor*** application, an overview of all deliverables, technical constraints, and general rules regarding project reports, submission of deliverables, and member contributions.

1 Features Required for Iterations 2 and 3

1. Start a new game,
2. Provide or select user name,
3. Set total thinking time,
4. Initialize board
5. Rotate wall,
6. Grab wall,
7. Move wall,
8. Drop wall,
9. Save position,
10. Load position,
11. Validate position,
12. Switch player (aka. Update board)

Assign the development of these ***Quoridor*** features to your team members, i.e., each team member is individually responsible for two features. If there are fewer than six team members in your team, the remaining features have to be implemented by the whole team. Each team member will be graded individually based on the quality of the UI mock-ups, controller interface, and mapping of Gherkin scenarios to Java code for the assigned features. **The project report should clearly state individual roles**.

As further technical input for Deliverable 2, you will receive (1) a reference Umple model, (2) an initial source code base, and (3) the detailed specification of these features in the form of Gherkin scenarios. From now on, these Gherkin scenarios will serve as the main specification of what you need to do.
You must follow the instructions on how to integrate this initial content with your repository at https://github.com/McGill-ECSE223-Fall2019/ecse223-tutorial-notes/wiki/Project-Iteration-2.

2 UI Mock-ups and Project Report

As a team, create a UI mock-up for all twelve features. Each team member will be responsible for the UI of her/his assigned features in Deliverable 3, but since the features need to be **integrated into a common UI**, this is a shared responsibility for Deliverable 2. As a team, you also need to ensure that the look and feel of the application is uniform across all features. You may choose your favorite drawing tool for the UI mock-up or a UI prototyping tool such as Pencil (https://pencil.evolus.vn/) and then import the mock-up into your project report. You may also draw the UI mock-up by hand, scan it (or take a photo), and include it in your report as long as the mock-up is clearly legible. Your **project report** shall be documented as a **new wiki page** (referring to Project Deliverable 2) with a uniform look and formatting.

3 Specification of Controller Interface and Default Implementation

You need to individually specify all operations for your assigned features in the Controller interface (which interface needs to be placed in the *ca.mcgill.ecse223.quoridor.controller* package). The interface consists of the **full method declaration** (incl. parameters, return type, etc.) relevant for your assigned features. In addition to all **modifier methods**, do not forget to include all **query methods** required for the features. The same query method may be used for several features. As a team, ensure the consistent use of query methods across features.

Each controller method needs to be briefly **documented using JavaDoc** (see public tutorials, e.g.: https://www.baeldung.com/javadoc), but you are not required to generate standalone HTML files. The JavaDoc specification should clearly state the name of the corresponding Gherkin feature and the name of the team member who is responsible for it.

Moreover, you need to create a **default implementation** of the operations present in the Control interface where the method body consists of throwing an *java.lang.UnsupportedOperationException* (i.e. the method body is almost empty).

Furthermore, you may find yourself adding private **helper methods** to the Controller or public helper methods to Model classes, because they are used by the public Controller methods. A private Controller helper method does not need to be shown in the interface specification, but a public Model helper method needs to be shown in the interface specification for the Model class.

Each team is required to use the **common Umple domain model** provided to you to ensure compatibility for later deliverables. Note that any public Model helper method will have to be added to the common Umple domain model. As such, the Umple model can be extended, but existing definitions in the Umple model cannot be changed!

4 Mapping of Gherkin Scenarios

Each team member is individually responsible for providing a mapping of each step in the Gherkin scenarios of the features they are responsible to the actual Java code of your group. Individual responsibilities of Gherkin step mappings should clearly be documented in the JavaDoc header (*@author*) of each step mapping code. Some step mappings are provided in the initial content of the repository.

By default, each step defined in Gherkin throws the default *PendingException* indicating that the actual mapping is missing. Nevertheless, it enables Cucumber to execute the scenarios (as acceptance tests) even when they are not fully implemented. As part of Deliverable 2, your task is to

- Map all actions (When clause) to calls to your Controller method.
- Implement the mapping ***all non-GUI related steps*** in the preconditions (Given clause) and postconditions (Then clause). The mapping of these steps shall use the Umple instance model (Quoridor object).
- No further action is required for GUI-related steps, thus they can still throw *PendingException*.
- Ensure that your project successfully compiles (no compile errors).
- Ensure that the execution of Gherkin steps can be successfully initiated by Cucumber (using the Gradle build file provided to you in the initial content of the source code base). Note that there will be steps which are not yet implemented, which is perfectly acceptable for Deliverable 2.

Note that the given acceptance tests (Gherkin scenarios) are only required to pass for Deliverable 3.

Submission

In general, your team is required to follow the General Rules explained in the Project Overview document. In addition to that, you are required to submit the Statement of Work Distribution document (as discussed above)

Marking Scheme

Deliverables for Iteration 2 of Project	***Marks***
UI Mock-ups of features (team mark)	20
Uniform look and feel of the GUI (team mark)	10
Specification of Controller interface for features (individual mark)	15
Javadoc documentation of methods in Controller interface (individual mark)	15
Mapping of Gherkin scenario steps (individual mark)	30
Uniform look and formatting of the project report (team mark)	5
Clear statement of team responsibilities in project report (team mark)	5
Total Marks:	100
The total mark may be adjusted based on the actual contributions of a team member to the deliverables.	