

Deliverables for Iteration 3 – Implementation (10%)

The deliverables for Iteration 3 are the complete implementation of the features of the **Quoridor** application as specified in Iteration 2 including (a) the Controller methods, (b) Gherkin scenarios, and (c) the user interface (UI). You are also required to state who worked on which features. The deliverables are due on **Sunday, November 03, 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 source code, and member contributions.

1 Implementation of Quoridor features (incl. Model, View and Controller)

In this deliverable, your main goal is to implement the required 12 features (of Stage 1) of the **Quoridor** application. The application you need to develop includes both the model, the view and the controller for the designated 12 features (as well as the mapping of Gherkin scenarios as acceptance tests, see below).

Evenly assign the **Quoridor** features as specified in Iteration 2 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. Note that you may wish to change the assignment of some features compared to Iteration 2, but you need to clearly record this change in the project report by providing a brief justification. Each team member will be graded individually based on the quality of the implementation of the assigned features. However, the **whole team is also responsible** for the **Quoridor** application including its uniform look and feel across all features and the completeness of the application. Your application will be assessed as a whole.

Each team is required to use the **common Umlle 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 Umlle domain model. As such, the Umlle model can be extended, but existing definitions in the Umlle model cannot be changed! You are required to use the code generated by Umlle.

Note that the **Quoridor** features are not independent but rather depend on each other in various ways. It is not a valid argument to claim that your feature does not work because another feature is not working and that your feature is working perfectly by itself. If your feature is not working for whatever reason, you will lose marks. If a team member does not manage to implement an assigned feature, then the team members of dependent features need to step up. Furthermore, this needs to be stated clearly in the Statement of Work Distribution (see Section 4).

Keep in mind that future deliverables depend on the features required for this deliverable. If a feature is not implemented for this deliverable, then it will have to be implemented later for a different deliverable anyway. In any case, all features are expected to be implemented for later iterations towards the end of the semester.

2 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. As part of this deliverable:

- Map all actions (When clause) to calls to your Controller method.
- Implement all Given, Then and And clauses (if you have not done in Iteration 2). These steps shall use the Umlle instance model (Quoridor object) or an appropriate query method.
- Implement **all GUI related steps** by acquiring data from your View.

- 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).
- Ensure that all Gherkin scenarios pass successfully as acceptance tests.

3 Grading

Each feature contributes to the individual mark of the team member who was assigned to the feature and to a team mark. The individual mark for the features is 60/90, while the team mark is 30/90. For example, assume that all team members except for team member X implement all their features successfully. The team members who implemented their assigned features will receive 60/60 for their individual mark. Team member X who did not implement the two features assigned to her/him will receive 0/60 for the individual mark. The whole team will receive 25/30 for the team mark ($= 30 * (n-m)/n$ where n is the total number of features and m is the total number of features that have not been implemented). Therefore, the team members who implemented their assigned features will receive 85/90, while team member X will receive 25/90.

If someone else in the team decides to implement a feature instead of the team member assigned to the feature and this is reported in the Statement of Work Distribution (see Section 4 below), then the team mark increases, but the individual mark stays the same. For example, if another team member implements the two assigned features of team member X, then team member X will receive 30/90 and the other team members will all receive 90/90.

The look and feel of the application is a shared responsibility of the whole team.

4 Statement of Work Distribution

Fill out the attached Excel sheet "2019-01 ECSE223 Project - Iteration 3 - Statement of Work Distribution.xlsx" and submit it on MyCourses as your deliverable. The Statement of Work Distribution will trigger the grading mechanism explained in Section 1 above. The final grade for this submission will also take into account whether the work was distributed evenly as indicated in the Statement of Work Distribution and as evidenced by the contributions of team members to your team's GitHub repository.

Submission

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 (see above).

Marking Scheme

<i>Deliverables for Iteration 3 of Project</i>	<i>Marks</i>
Quoridor project	95
Implementation of features (individual mark)	60/95
Implementation of features (team mark)	30/95
Uniform look and feel of the application (team mark)	5/95
Clear Statement of Work Distribution (team mark)	5/5
Total Marks:	100
The total mark may be adjusted based on the actual contributions of a team member to the deliverables.	