Deliverables for Iteration 2 – Controller Implementation (10%)

The deliverables for Iteration 2 are the implementation of the features of the ***Kingdomino*** application for the controller and model layers. including (a) the Controller methods, and (b) Gherkin scenarios, but excluding the user interface (UI). You are also required to state who worked on which features. The deliverables are due on **Sunday, March 01, 2020, at 23:30**.

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

1 Specification of Controller Interface and Documentation in JavaDoc

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.kingdomino.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.

2 Implementation of Kingdomino Features (incl. Model and Controller)

In this deliverable, your main goal is to implement the controller code for the required 24 features of the ***Kingdomino*** application. The application you need to develop includes both the model, and the controller for the 24 features but it excludes the view (i.e. the GUI code).

Evenly assign the ***Kingdomino*** features to your team members, i.e., each team member is **individually** responsible for four features. If there are fewer than six team members in your team, the remaining features have to be implemented by the whole team or designated members of your team (e.g. volunteers in your team). Note that you need to clearly record the assignment of features in the Statement of Work document (see Section 5). 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 **Kingdomino** application (e.g. completeness, integration and build issues), thus you application will also be assessed as a whole.

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! You are required to use the code generated by Umple.

Note that the ***Kingdomino*** 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 5).

Note that future deliverables depend on the features required for this deliverable. If a feature is not

implemented for this deliverable, it will have to be implemented later for a different deliverable anyway. In any case, all features are expected to be implemented by the end of Iteration 4.

3 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:

- Implement the Given clauses to bring you application into a designated initial state.
- Map all actions (When clause) to calls to your Controller method.
- Implement all Then clauses by using the Umple instance model (Kingdomino object) or an appropriate query method.
- 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.

4 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 5 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.

5 Statement of Work Distribution

Fill out the attached Excel sheet "2020-01 ECSE223 Project - Iteration 2 - 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 4 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

Deliverables for Iteration 3 of Project	***Marks***
Kingdomino project	95
Implementation of features (individual mark)	60/95
Implementation of features (team mark)	30/95
Correctness of project build and setup (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.	