

## Deliverables for Iteration 4 – Implementation (11%)

The deliverable for Iteration 4 is to complete the implementation of Kingdomino application with a graphical user interface (GUI) following the Model-View-Controller pattern. You must use the Controller and Model developed in the previous deliverables. If you failed to implement any features from previous iterations, you must also provide an implementation those missing features. You are also required to state who worked on which features. The deliverables are due on **Tuesday, April 14, 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 Implementation of the Graphical User Interface of Kingdomino

In this deliverable, your main goal is to develop the graphical user interface (i.e. the View) of the **Kingdomino** application and integrate it with the existing Model and Controller. By default, you are expected to use GUI technologies (Swing, Java2D) we cover during the tutorial, but you may ask for permission to use other modern Java GUI frameworks (provided that they can be incorporated into the automated Gradle build). As such, your Kingdomino application is expected to be fully playable by the end of this deliverable. In fact, the grading of this deliverable will be dominated by the functional testing of your application by human users.

Evenly assign the **Kingdomino** features of Phase 2 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. 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 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 **Umple domain model** provided to you and the **controller methods developed** by your team in Deliverable 2 and 3. 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. Your **GUI is required to properly use the existing controller methods** (without reimplementing them in the view). The existing Gherkin scenarios (implemented for Deliverable 2 and 3) should still execute correctly, but you do not need to extend those with GUI related steps.

The source code of the GUI should clearly be documented in the JavaDoc header surely including the author (*@author*) and the identification of the respective feature.

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 4).

### 2 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 50/80, while the team mark is 30/80. 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 50/50 for their

individual mark. Team member X who did not implement the two features assigned to her/him will receive 0/50 for the individual mark. The whole team will receive 27.5/30 for the team mark ( $= 30 \cdot (n-m)/n$  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 87.5/90, while team member X will receive 27.5/80.

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 30/80 and the other team members will all receive 80/80.

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

### **3 Documentation of Bonus Features on Project Wiki**

Bonus features implemented by your team need to be explicitly listed and briefly documented (1 short paragraph description) on the project wiki. In addition to the announce major features, you may state what you consider as extra delivery of your team (incl. small enhancements, use of some technologies). Please also document the complexity of each bonus feature (e.g. number of lines of code).

It is unlikely that you receive bonus marks for something that you do not state as a bonus. But all bonus marks are based on the individual judgement of the course instructor (i.e. no upfront grading scheme).

### **4 Statement of Work Distribution**

Fill out the attached Excel sheet "2020-01 ECSE223 Project - Iteration 4 - Statement of Work Distribution.xlsx" and submit it on MyCourses as your deliverable. The Statement of Work Distribution will trigger the grading mechanism explained 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.

The Statement of Work Distribution should also document the responsibilities related to Bonus Features.

## **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><b>Deliverables for Iteration 4 of Project</b></i>	<i><b>Marks</b></i>
<b>Kingdomino</b> project (Iteration 4)	95
Implementation of features (individual mark)	50/95
Implementation of features (team mark)	30/95
Uniform look and feel of the application (team mark)	5/95
Usability of the application (team mark)	5/95
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.	