Deliverables for Iteration 3 – State Machines (8%)

The deliverable for Iteration 3 aims to specify the major flow of the Kingdomino gameplay (including phases such as the creation of first draft, the selection of the first domino, domino placement, domino selection, creation of a new draft, last domino placement, calculation of player score and revealing the final result) using state machines and implement them with the help of automated code generation as provided by Umple. You will be provided a set of scenarios specified Gherkin which serve as an acceptance test suite for your statemachine. The deliverable is due on **Sunday, March 22, 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. Note that you are responsible for this deliverable as a team.

1 Statemachine Specification(s)

You need to implement state-based behavior for the major flow of the ***Kingdomino*** gameplay. A set of Gherkin scenarios are provided to you in a separate branch (iteration-3) which you need to integrate into your source code repository. The draft of an **Umple file** is also provided to you in *Gameplay.ump* as a starting point in the same branch.

All relevant triggers, guards and actions in your statemachine (on transitions as well as in states) need to be defined in (public or private) methods. A state machine itself may only show which methods are called for guards and actions. All new methods need to be documented using JavaDoc.

You are required to use the pre-declared methods in your statemachine "as-is" (i.e. you still need to provide their implementation). Moreover, you are allowed to define extra private methods if you need. You need to use the automated code generation support of Umple to derive a fully functional implementation of the two features from the state machine.

The source code automatically generated from the statemachine specification (as well as from the domain model) has to be committed to the GitHub repository using the predefined naming pattern.

2 Implementation of Controller

You need to define two methods in the Controller interface to interact with the statemachine(s) by triggering relevant events and accepting calls (as part of actions) from the statemachine. Any new methods need to be documented using JavaDoc.

By the end of this Deliverable, one could play an entire Kingdomino game by calling the relevant methods of controller (as if it were a command line application), but no graphical user interface is provided yet.

Each team is required to use the **common Umple domain model** provided to you. 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.

The whole team is also responsible for implementing the statemachine of the **Kingdomino** application. Your application will be assessed as a whole.

3 Mapping of Gherkin Scenarios

The mapping of each step in the Gherkin scenarios need to be assigned to individual team members documented in the JavaDoc header (*@author*) of each step mapping code. Nevertheless, the mapping of Gherkin steps will be graded for the entire team this time (with shared responsibility). As part of this deliverable, you will need to:

- Connect the When clause to trigger a step in your statemachine.
- Implement all missing Given clauses to ensure that your statemachine is in the designated source state before executing a test case.
- Implement all And clauses which are related to relevant guards in the statemachine.
- Implement all Then clauses to check for appropriate actions and target states.
- All of these steps shall use the Umple instance model (Kingdomino object) or an appropriate query method (similarly to Iteration 2).
- 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 Project Report

You need to document the behavior of your state machine on the project wiki of your team. This should include a UML Statechart diagram for each of your statemachine and a brief (1 long sentence or 1 short paragraph) documentation of any extra methods you introduced for guard and actions.

Submission

Your team is required to follow the General Rules explained in the Project Overview document.

Marking Scheme

Deliverables for Iteration 3 of Project	***Marks***
Kingdomino project (team mark)	90
Statechart specification in Umple	25/90
Statechart implementation	20/90
Implementation of Gherkin step mappings	15/90
Execution of acceptance tests	20/90
Documentation in JavaDoc (for all manually implemented methods)	10/90
Statechart Project Report (team mark)	10
Description of extra methods for triggers, guards and actions	05/10
UML Statechart diagram	05/10
Total Marks:	100
The total mark may be adjusted based on the actual contributions of a team member to the deliverables.	