We chose the MVC architectural style for our gameplay (in game) design because it is convenient way to integrate our gameplay design. The MVC style consists of three layers as it is designated in its name, Model, View, and Controller.

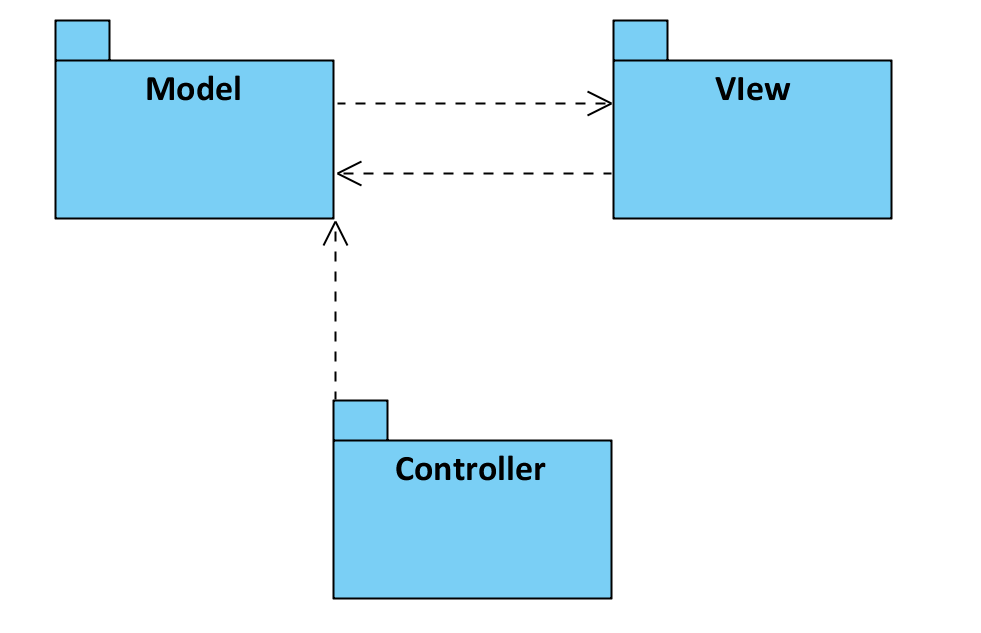
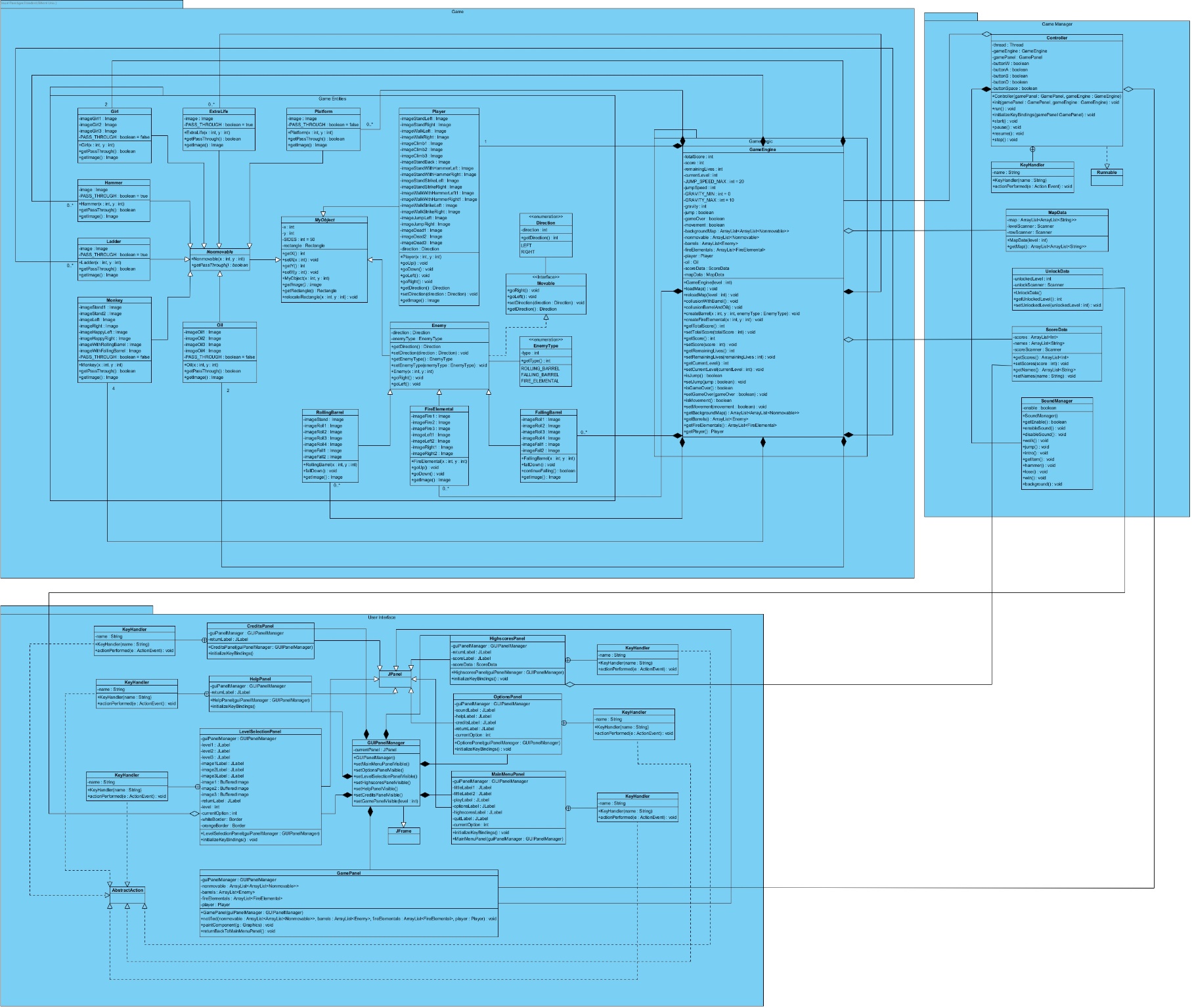


Figure 2.1.1: High-Level Representation of Subsystem Decomposition

Firstly, controller layer accepts inputs from the user and commands model (sometimes view) to execute according to inputs. Mainly, controller sends updates to model, which by this way model updates model’s state. Secondly, model which is also central and most important component of the layer. Additionally, model gets data from controller and sends to view which layer view displays. Additionally, View-Controller is optional for different systems.

In the following page class diagram (Figure 2.1.2) is placed which it is separated to 3 parts according to MVC architectural style. As it is seen from 3 layers, every layer has main classes for each layer. Main class of the model part is “GameEngine.java” class which object classes are connected to “GameEngine.java”. Besides, for controller layer “Controller.java” class is main class for controller layer. “Controller.java”, “MapData.java”, and “ScoreData.java” classes have relationship with “GameEngine.java” which demonstrates relationship between model layer (“GameEngine.java”) and Controller layer (“Controller.java”, “MapData.java”, “ScoreData.java”). Finally, after these relationship between View layer displays the program (game).

**2.1.2 Class Diagram according to MVC**