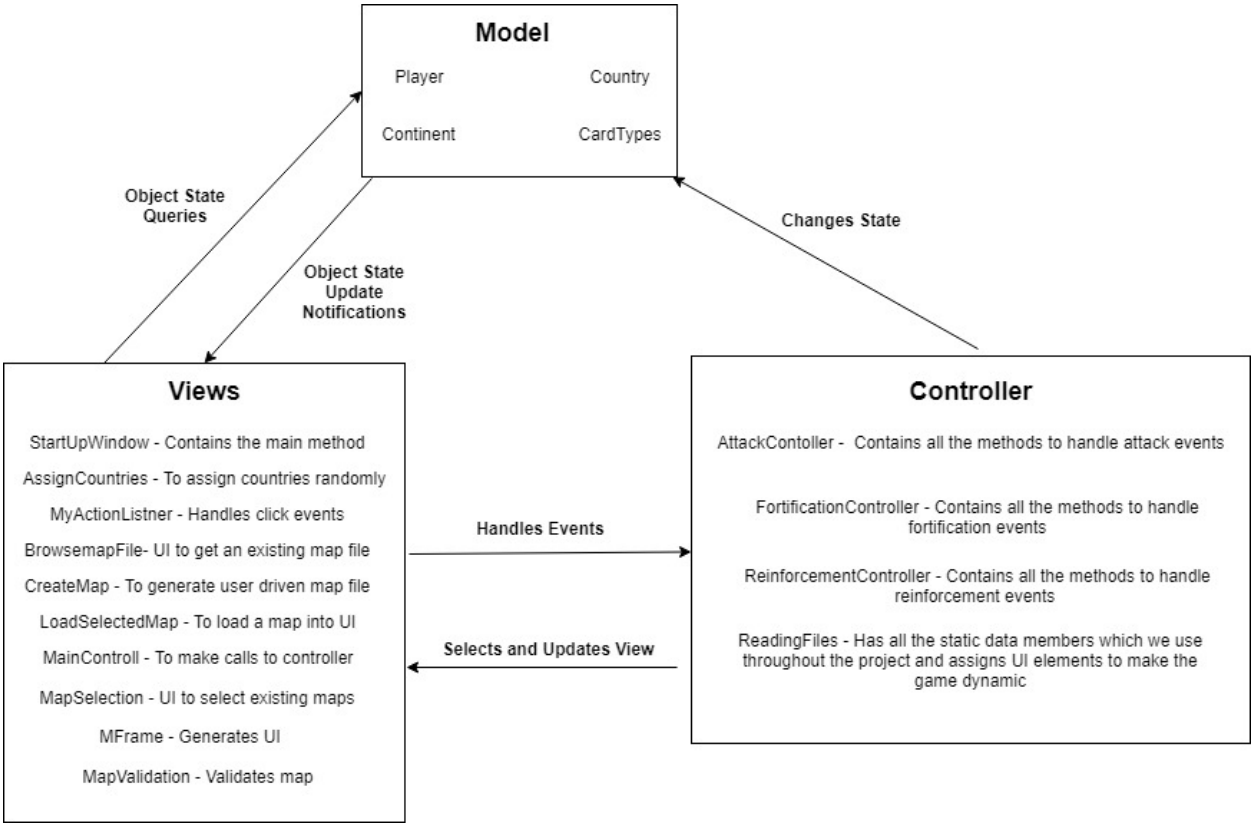
RISK - Architectural Design

The Architecture followed was MVC with Controllers as event handlers, Views as observers and to interact with the user and Models, which are observable, for maintaining of data.



Model – Model has observable classes namely Player, Country and Continent which represents a player that's playing, a territory and a continent in the map respectively. There is an Enum, CardTypes to represent cards in the future builds of the project.

View – Views were done using the JavaSwing GUI. The classes in this are used to generate UI and interact with the user. The user-driven generation of map, loading an existing map, uploading a new map file, assigning countries to players randomly and placing armies in those countries initially etc., has been implemented. ActionListeners were used to make controller calls and update UI accordingly.

Controller – Contains all the methods needed in each phase of the game. Methods in here take model objects as input and update or output the objects or values according to risk rules.

Coding Standards

User-defined types, Classes – First letter of each word is capitalized and is without any special characters

Local variables – all small letters without any special characters

Attributes (data members) – All small letters separated by underscore

Functions/Methods – Method names start with a small letter and after the first word each word's first letter is capitalized (camelCase)

Properties(get/set methods for model class private data members) - starts with a small letter and after the first word each word's first letter is capitalized (camelCase)

Constants – There are no constants used in this project

Static Variables- All words start with capital letter