Architectural Design

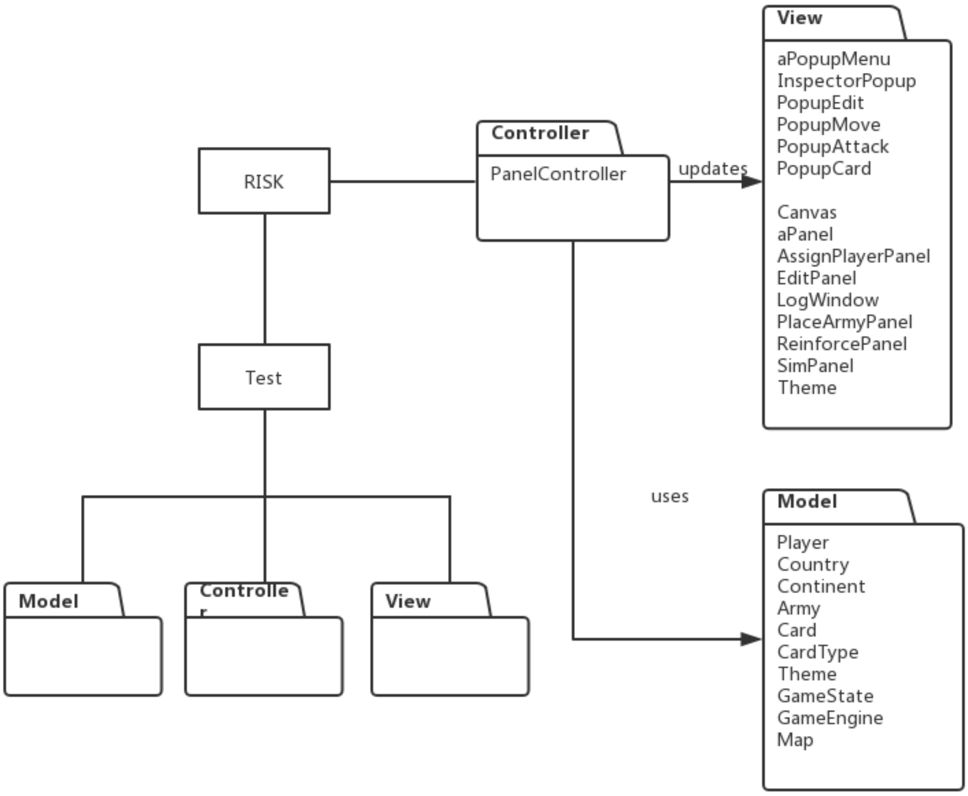
The game uses the MVC software architecture to divide the software system into three basic parts: Model, View, and Controller.

Model: Used to encapsulate data related to the business logic of the application and how to process the data. "Model" has direct access to data.

Controller: Used to control the application, organizing different levels. It processes and responds to user behavior and changes in the data model.

View: It access to the data model that it monitors (Model), and responsible for formatting data and presenting it to the user.

The following is a package diagram of Build1 and a description of each class.



Controller

Class Name	Definition
PanelController	The control class decides which specific Panel to use based on the game phase.

View

Class Name	Definition
aPopupMenu	defines a base class for a popup
InspectorPopup	Used to display popup properties
PopupEdit	creates a pop-up menu to edit continents in the map
PopupMove	defines the movement of the popup
PopupCard	creates a pop-up to achieve exchange cards for armies function
PopupAttack	defines the attack of the popup

Class Name	Definition
Canvas	defines a canvas and functions that perform a series of operations on the map. It is an observer object

aPanel	defines the base class for the Panel
AssignPlayerPanel	defines the assigning player panel It extends aPanel and implements ActionListener
EditPanel	Edit the Panel. It extends aPanel
LogWindow	creates an easy to use and beautiful log window user only needs to use add() function to append very well-formatted text to the log window
PlaceArmyPanel	defines the place player panel It extends aPanel
ReinforcePanel	defines the reinforcepanel. It extends aPanel
SimPanel	Supplemental Panel for reinforce stage
Theme	Define the color of the interface

Model

Class Name	Definition
Player	defines a player and its actions
Country	Define a new country, and define methods for adding, deleting, and querying
Continent	Define a new continent, and define

	methods for adding, deleting, and querying
Army	defines an army and define methods for adding, deleting, and querying
Card	defines a card and define methods for adding, deleting, and querying
CardType	Is an enumeration class，defines the category of the card
Theme	Define the color of the interface
GameState	Is an enumeration class ,indicates different game states and status（EDITMAP, CHOOSEPLAYER, STARTUP, REINFORCE, ATTACK, FORTIFY）
Map	defines a map and all methods related to the map. It is an observable object
GameEngine	controls the flow of the game and defines the function of the game initialization