**SOEN 6441: RISKGAME BUILD 3 DOCUMENTATION:**

**ARCHITECTURE DESIGN**

**GROUP MEMBERS:**  
**40087621 An Nguyen**  
**40104630 Naga Satish**

**Architecture Design – Model, View and Controller architecture (MVC)**

Triggers model’s method

Updates View

User’s action

Interacts

**Controller**

+RiskGameController

+RiskPlayerController

+RiskAddMapController

+RiskStrategyController

+RiskTournamentController

+RiskBoardController

**View**

+RiskGameView

+RiskPlayerView

+RiskAddMapView

+RiskStrategyView

+RiskTournamentView

+RiskBoardView

**Model**

+RiskGameModel

+RiskPlayerModel

+RiskAddMapModel

+RiskStrategyModel

+RiskTournamentModel

+RiskBoardModel

**Description:**

The application adopted the MVC architecture.

View files: There are 6 view files for each frame showing the respective frame of a specific action the user can perform. The main frame on which the user will play the game is the RiskBoardView file.

Model files: 6 view files correspondent to 6 model files containing the logic for each frame and after a specific action from the user, if any components on the view changed, the model file will update the view accordingly. For RiskBoardView frame, the RiskBoardModel has a method called updateTheBoardScreenData. If after an action has been performed and the data has been changed, that method will be called to update the latest data to the view.

Controller files: 6 controller files to act as a bridge between model and view files. In particular, they contain actionListeners when a specific action is performed.

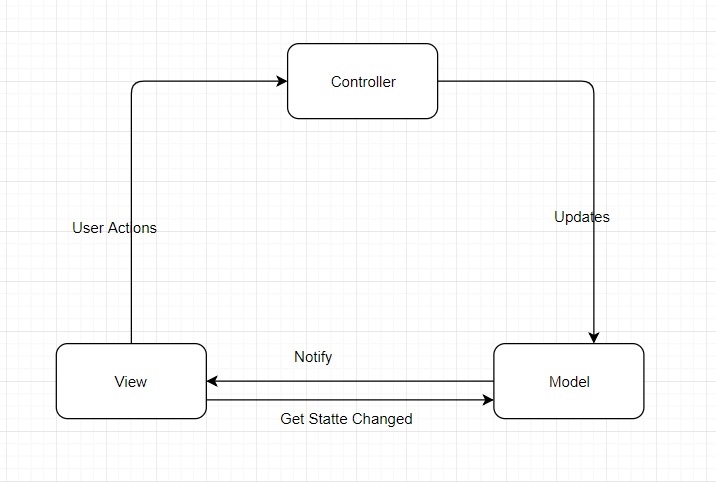
**Observer Pattern:**

The Observer defines a one to many relationship so that when one object changes state, the others are notified and updated automatically.

There are two type of observer model.

1. Push Model
2. Pull Model.

**How the observer Pattern works in Project:**



1. The User interacts with the view, if the user interacts with the any view component, then view informs the controller about the actions.
2. It’s controller duty how the actions should be manipulated by calling the method in the Model.
3. Model performs the actions based on the controller data. When something state has changed in the model then the model updates the state in the view.