# Refactoring List-Group 7

(We write tests for reinforce, startup and fortify phase before refactoring)

1. remove duplicated showCommands  
    At build 1, showCommands such as showPlayer(), showAllPlayer(), showMap() exist in controllers including gameController, startUpGameController, reinforceGameController and fortifyGameController.

* A screenshot of a cell phone

  Description automatically generated
* A screenshot of a cell phone

  Description automatically generated
* At build 2, we extract all showCommand methods and make them a static method in the MapDisplayUtils.class to avoid duplication.

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

1. move player state and corresponding logics (set and get current player, add and remove player, switch players) from the gameController to model named PlayerService.class.

* At build 1, player state and corresponding logic in gameController
* A screenshot of a cell phone

  Description automatically generated
* At build 2, we move player state and corresponding logic in model named PlayerService
* A screenshot of a cell phone

  Description automatically generatedA screenshot of a cell phone

  Description automatically generatedA screenshot of a cell phone

  Description automatically generatedA screenshot of a cell phone

  Description automatically generated

1. remove game controller, unnecessary boolean values and unnecessary dependency of other controllers on gameController. This improves the readability and extensibility of the code.

* At build 1, game controller has references of other controllers. It also has a lot of boolean values to determine which controller to call based on these boolean values.

A screenshot of a cell phone

Description automatically generated

* At build 2, we remove game controller, its dependency on other controllers and remove unnecessary Boolean values.
* A screenshot of a cell phone

  Description automatically generated

1. remove logic relating to reinforce(), exchangeCards() from reinforceController to model Player.class. The reinforceController is used to check the validity of the command

* At build 1, logic of reinforce and exchangeCards and command validation are all in ReinforceGameControllerA screenshot of a cell phone

  Description automatically generated
* At build 2, we move logic of reinforce and exchangeCards to model named Player while keeping command validation in ReinforceGameController
* A close up of a logo

  Description automatically generatedA screenshot of a cell phone

  Description automatically generatedA screenshot of a cell phone

  Description automatically generatedA screenshot of a cell phone

  Description automatically generatedA screenshot of a cell phone

  Description automatically generated

1. remove logic relating fortify() from fortifyController to model Player.class. The fortifyGameController is used to check the validity of the command

At build 1, logic of fortify and command validation are all in FortifyGameController

* A screenshot of a cell phone

  Description automatically generated
* At build 2, we move logic of fortify to model named Player while keeping command validation in FortifyGameController
* A close up of a logo

  Description automatically generated

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

1. add GameView and Controller as the interface.

At build 1, we have only CommandPromptView to receive users’ commands, and send it to different controllers according to the game state.

A screenshot of a cell phone

Description automatically generated

* At build 2, we have three different views, phaseView, dominationView and exchangeCardView. We make these three views implement GameViewInterface. This would make the system more flexible as we could replace these views in the future. The same case for the controllers, we would be able to replace these controllers easily in the future based on changing requirements.
* A screenshot of a cell phone

  Description automatically generatedA screenshot of a cell phone

  Description automatically generatedA screenshot of a cell phone

  Description automatically generated
* A screenshot of a cell phone

  Description automatically generated
* A screenshot of a cell phone

  Description automatically generated

1. refactor mapLoaderTests including extracting logics and parameters in different methods to make tests more readable and understandable.

* At build 1, the mapLoaderTest is hard to read, understand and maintain. A screenshot of a cell phone

  Description automatically generated

1. At build 2, we extract logics and parameters in different methods to make tests more readable and understandable.

* A screenshot of a cell phone

  Description automatically generatedA screenshot of a cell phone

  Description automatically generatedA screenshot of a cell phone

  Description automatically generated