Group 1
Design Documentation

For GameState, I created it and made methods in an attempt to give other classes all of the data they would need to provide all functionality, as a sort of “bottleneck” of information. In practice, the current implementation likely broke every principle imaginable in an effort to preserve functionality with smelly code.

The GameState constructor needs to be the way it is so every parameter can be accessed in view at the start of the game, so each type of data stored in GameState must be initialized. Helper functions are part of the constructor process, and perform necessary initializations for card functions and the like. In the future, card functions will likely be defined outside of GameState, as otherwise that's one more on to the list of SRP violations.

The two place$[X]Token functions are necessary as ways to change each planet’s token count that is stored within GameState, and are to be called whenever a function that changes tokens needs to place one. In our current implementation however, it is more than likely there are some methods that work around it, and will need to be updated for the next milestone.

For Card, cards now all have a class variable of interface CardAction, and a method that calls CardAction.action(). This is necessary in order to give each and every defined card a unique action, which is defined upon the initialization of a new type of card. Furthermore, the 3 different types of cards (ResistCard, AbilityCard, GalaxyNewsCard) all extend Card, with small differences between them to accommodate for their individual quirks. For example, this is necessary for when a playCard method is called on ResistCard to put said ResistCard back into the used pile of cards.

View

Player input will be handled by view. This includes the DisplayUpdater, ViewElements, and Buttons class. These communicate with GameContoller to update the game pieces using the controllers methods. The controller updates the model which updates view to retrieve the needed information.