Milestone 1

We will be refactoring and adding features to Risk, a strategy board game based on conquering your enemies. This project was built for a CSSE376 project and because this was developed with test driven development. This means the tests are already built, however the test incorporates mocking in some parts, meaning that some larger refactoring changes to class could result in failing tests because of failed mocks. Each week we will be performing an analysis, implementing our listed refactors, and adding features; these updates will then be shared with our project advisor. Advice from our project advisor will then be the first priority for the next sprint.

Management

Project Repository (Github)

Ticket Management (Trello)

Features

Milestone 2

- Add Names to each user and let them select their color.
- Ability to have different maps.
- Updated User Interface
  - Provide an explanation of the game at the beginning.
  - Provide explanation dialog during the first round of each stage.
  - Provide dialog to indicate switching to the next player's turn.

Milestone 3

- Add “Wild Cards”
- Add “Secret Mission Mode”
  - Will require adding mission cards
- Update User Interface
  - Reorganization play screen to better inform users.
  - Provide information for attack, indicate selected territories, draw a line between.

Refactoring Changes

Milestone 1

- Divergent Change (The Change Preventers) - the game and user interface are together in a single class, we must redesign the game class to be a separate game loop without the coupling to user interface. Then make a new main class that incorporates the user interface and the game loop (Game, Main).
- Duplicated Code (The Dispensable) - Duplicated code between methods occupyTerritory and addArmiesToTerritory, have one call the other. (Player.occupyTerritory)
- Shotgun Surgery (The Change Preverters) - Add Continent class for holding name, number of each territory, and bonus. Allow this to be flexible enough to allow other maps (Territory).
- Unclear Parameter Names - Rename parameters for loseArmies, addAdjanctTerritory, and addArmies (Territory).
- Separate Methods - for attackTerritory it's unclear what the return is for without documentation, so add a separate method for checking if all troops are lost (Territory).
- Unrelated Method - The role dice method has no associated to the player and could be placed in another location. Additionally, no reason to have a throw (Player.rollDice).
- Overloaded Methods - Methods such as attacking are huge and could be broken up into helper methods (Game)
- Hardcoded Value - fix logic with hard coded constant of (Player.hasWon).
- Optional Graph Model: could convert territory grid to graph for a more flexible design if you wanted to drastically change the map. This isn’t needed since Risk is an already defined game but its an idea.
- Magic Numbers - Various magic numbers are strewn about in the code and should be replaced with descriptively named constants

Milestone 2

- Unclear Method Name - Is there any reason it is called pictured territory and not just territory (Card).
- Data Class (The Dispensables) - Simple data class with just getters/setters (Card).
- Switch Statement (Object Oriented Abusers) - Remove switch statement and make more abstract (TerritoryButton.setPlayer)
- Switch Statement (Object Oriented Abusers) - Remove switch statements and make them more abstract (CardTrader.generateNewCard).
- Generalize win conditions (Player.hasWon) so that multiple game modes are supported (players automatically win if they complete their secret mission)

General

- Dead Code (The Dispensables) - Over Complicated logic, why is a throw error needed. Also possible better name for the function (CardTrader.getCurrentSetValue).
- Switch Statement (Object Oriented Abusers) - Remove switch statement and make more abstract enum symbol (CardTrader.tradeInCardSet).
- *Alot of methods can be made protected because they are not public methods, but they are needed to verify data in testing.*