SYSC 3110 Final Project

Team Group

Author: Baillie Noell

Data Structures Explanation Milestone 2:

Changes from Milestone 1:

General Changes:

- Command, CommandWords, and Parser classes were removed, as they are unused with the GUI
- Game class has been renamed to GameModel
- GameController, GameFrame, and Event classes (GameOverEvent, GameStartEvent, OwnerChangeEvent, PlayerEliminatedEvent, PlayerStateEvent, and TurnStateEvent) were added

Changes in Map Class:

- Countries and continents are no longer stored in Lists but are now in a HashMap. This facilitates fast addition and lookup, with the keys being the country names, and returning the Country object.
- The printMap method was removed as it is no longer used.
- loadContinents is now done with an array of country names in one call, instead of using indices and for loops due to now using a HashMap.

Changes in Continent Class:

- All countries are added to a continent in one call using an array of country names, which is stored in a HashSet and then loaded into the country ArrayList in the Continent object.

Changes in Country Class:

- addNeighbor method now adds the neighbors to each others neighbor list in one call, cutting the number of method calls in half.
- Added a hasNeighbor method to reduce the repeated chaining of method calls to determine if a country is the neighbor of another

Changes to Player Class:

- Renamed the removeCountry method to lost, as it is called when an attack is lost by the defender.
- The getCountrySize method was removed
- The Player class was modified to remove the need for the eliminated field. The isEliminated method now returns true if the player no longer has any countries. This also removed the need for checkEliminated and setEliminated methods.
- Added a method hasCountries to make reinforcement calculation clearer and to reduce coupling of our classes. Now, instead of currentPlayer.getCountries().containsAll(continent.getCountries()) which gets an list of countries from Player and Continent we get currentPlayer.hasCountries(continent.getCountries()) which passes the list of continents of the Continent into Player meaning we don't need to know how countries is stored in Player.)
- Added a getInfo method that returns a string containing the Player status

- Added a sortCountries method
- added a method to find the perimeter Countries allowing for better automatic troop placement which will be used with the AI player, but was also used while testing.

Changes to GameModel (formerly Game) Class:

- The unused constructor, and the parser field were removed
- currentPlayerReinforcements field was add to remove having to check for the number of reinforcements multiple times per turn
- gameViews ArrayList was added to store the viewers of the game model
- Methods addGameView, userCreateGame, and playerOwns were added for use with the MVC model
- - getReinforcements method was updated so that players now manually choose where their reinforcements are placed, instead of the game auto placing them
- playAttack was modified to accommodate the MVC model
- A blitz attack option was added, which allows a country to repeatedly attack another until the defender is taken, or the attacker can no longer attack
- nextPlayer now has a flag to check if the game has been started, to suppress unnecessary messages
- getCurrentPlayer method, and showErrorPopUp, showMessage, getIntInput, and getPlayerNames events were added in order to run tests and supress JOptionPane popups during testing.
- gameModel tests create a dummyView, and send the above events to the dummy view to simulate game behavior without the tester having to manually click throughout the tests

New Additions for Milestone 2:

GameView interface:

- Methods updateGameViewsStart, resetView, updateGameViewsTurnState, updateGameViewsState, updateGameViewsOwnerChange, updatePlayerTurn, handlePlayerElimination, and handleGameOver were added
- These methods are used for event handling in various scenarios throughout the game, and are present in GameModel viewers

Game Controller Class:

- Implements a finite state machine. The state machine has three states, with each corresponding to a phase of a turn in risk.
    1. Reinforcement phase: Player places calculated number of reinforcements in countries of their choice.
    2. Attack phase: a player can attack other countries. They may attack as many times as they wish.
    3. Movement phase: A player at the end of their turn can move their troops from one owned country to another, as long as there is a path of other owned countries between the two.
- The controller gets the mouse clicks from the player on the map, and retrieves the country that was clicked on (if there is one)

- The controller then chooses and sends the appropriate command to game model based on the current state and other conditions.
- The controller also handles menu events. Currently start game is the only menu option, but more can be added.

Game Frame Class:

The game frame contains:

- A JMenuBar with a JMenu game and JMenuItem Start Game
- JButtons for placing reinforcements, making an attack, entering the move phase, and ending a turn
- A JScrollPane for each player, listing all of their countries and the number of troops in each
- The map, which can be clicked on to interacted with, and is made of the JGraphX package

The Map:

- The map is made using the JGraphX package
- This package allows a graph to be made out of cells, which can be placed in any location, and edges between the cells
- The cells contain a value, which we have chosen to just be a String which is the name of the country the cell is representing, as well as location and style information
- The edges are automatically created between two vertices, or cells, using the insertEdge method
- An event listener is added to the graph, and when the graph is clicked it returns the value (country name) of the cell at the coordinates of the click event
- The event handlers then use the country name that was returned on mouse click