Concordia University
Department of Computer Science
and Software Engineering

Advanced Programming Practices
SOEN 6441 --- Winter 2024

Project Description

Deadline:	Intermediate delivery 1:	February 18 2024
	Intermediate delivery 2:	March 17 2024
	Final delivery:	April 9 2024
Evaluation:	Intermediate delivery 1:	15% of final mark
	Intermediate delivery 2:	15% of final mark
	Final delivery:	20% of final mark
Late submission:	not accepted	
Teams:	the project is a team assignment	

General description

The project is to be undertaken teams of 5 members and consists of the building of a challengingly large Java program. The completion of the project is divided into three separate components: (1) the *First and Second Intermediate Project Delivery* are intermediate operational build of the software, effectively demonstrating the full implementation of some important software features; (2) the *Final Project Delivery* is the demonstration of the finalized version of your software. During the project deliveries, you also have to demonstrate that your code includes many of the Java features presented in the lectures, and that you effectively use the tools presented in the lectures in your project. All project deliveries are to be undertaken in a zoom session with a marker where the team presents the implemented features to the markers following a pre-circulated grading sheet.

It is important to realize that the project description is purposely incomplete, and that it is one of your duties in this project to: 1) elicit and formulate all the missing details before you start the implementation, 2) limit the scope of the project according to the time that is available, 3) determine what design decisions will be made, as well as 4) what tools will be used for the implementation. These activities require some investigations and discussions that are important aspects of real-life software development and this project.

Problem statement

The specific project for this semester consists in building a simple “Risk” computer game. The developed program will have to be compatible with the rules and map files and the command-line play of the “Warzone” version of Risk, which can be found at: https://www.warzone.com/. A Warzone game setup consists of a connected graph map representing a world map, where each node is a country and each edge represents adjacency between countries. Two or more players can play by placing armies on countries they own, from which they can attack adjacent countries to conquer them. The objective of the game is to conquer all countries on the map.

Map

The game map is a connected graph where each node represents a territory owned by one of the players. Edges between the nodes represent adjacency between territories. The map is divided into subgraphs that represent continents. A continent is a connected subgraph of the map graph, and every territory belongs to one and only one continent. Each continent is given a control value (a “bonus” in Warzone terminology) that determines the number of armies per turn that is given to a player that controls all of it. During game play, every territory belongs to one and only one player and contains one or more armies that belong to the player owning the territory. In your implementation, it will be expected that the game can be played on any connected graph that is defined by the user before play, saved as a text file representation, and loaded by the game during play.

Game

The game starts by the startup phase, where the number of players is determined, then all the territories are randomly assigned to the players. Then the turn-based main play phase begins, in which all players are creating a set of battle orders. After all the orders have been given for all players for a turn, they are executed by the game engine: The orders are 1) deployment orders, 2) advance orders, 3) special orders resulting from using cards:

deploy: place some armies on one of the current player's territories.
advance: move some armies from one of the current player's territories (source) to an adjacent territory (target). If the target territory belongs to the current player, the armies are moved to the target territory. If the target territory belongs to another player, an attack happens between the two territories.
bomb: destroy half of the armies located on an opponent's territory that is adjacent to one of the current player's territories.
blockade: triple the number of armies on one of the current player's territories and make it a neutral territory.
airlift: advance some armies from one of the current player's territories to any another territory.
negotiate: prevent attacks between the current player and another player until the end of the turn.

Issuing Orders: Each player, in round-robin turn order, give one of their orders. Once all the players have signified that they don't have any more orders for this turn, the game engine executes all the orders. At the beginning of every turn, the players are given a number of armies that depends on the number of territories they own (# of territories owned divided by 3, rounded down. If the player owns all the territories of an entire continent, the player is given a number of armies corresponding to the continent's control bonus value. In any case, the minimal number of reinforcement armies for any player is 3. Once the total number of reinforcements is determined for the player's turn, the player may start giving deployment orders, i.e. placing their reinforcement armies on some of the territories they own. The players can also give advance orders. To do so, the player may choose one of the territories they own that contains one or more armies and order them to advance to another adjacent territory. If the target territory belongs to the same player, army units are moved to the target territory. If the target territory belongs to another player, an attack is simulated when the order is executed. Once all orders have been issued by a player, they signify so to the game engine. Once all the players have signified that they are finished issuing orders, the orders are executed.

Executing Orders: When executing a deploy order, a number of army units is added to a territory. A move is simply moving army units from one territory to the other. An attack is simulated by the following battle simulation mechanism: First, the attacking player decides how many armies are involved. Then, each attacking army unit involved has 60% chances of killing one defending army. At the same time, each defending army unit has 70% chances of killing one attacking army unit. If all the defender's armies are eliminated, the attacker captures the territory. The attacking army units that survived the battle then occupy the conquered territory. A player receives a card at the end of his turn if they successfully conquered at least one territory during their turn. Cards can be played as the players are giving orders. Each card has a different effect that influence the course of the game:

bomb: the target country loses half of their army units.
reinforcement: the player receives 5 reinforcement army units.
blockade: the target territory's army units count is tripled, and the territory becomes neutral.
airlift: move any number of army units from one of your territories to another territory, even if they are not adjacent.
diplomacy: until the end of the turn, you and the target player cannot attack each other.

Any player than does not control at least one territory is removed from the game. The game ends at any time one of the players owns all the territories in the map.

References

Warzone
Domination game maps