Pass (>=50%)

Test method functionality of classes

Player (2), province (2), units (category) (2), troop production buildings (1), AI (3), GloriaRomanus, Game (3)

* Player must be able to invade (attempting to conquer) enemy provinces (3)
* Player must be able to move troops between adjacent regions 1 turn at a time (do not need to implement movement of troops across multiple provinces for a pass) (3)
* Player must be able to build troop production buildings (should take the appropriate number of turns) (1)
* Player must be able to recruit soldiers of any category (not necessary to consider money in pass-level, but it should take the appropriate number of turns) (3)
* Trivial but sensible implementation of battle resolver and campaign AI. For example (you may implement another sensible implementation, but you must be able to explain why it is suitable during your iteration demos):
  + Battle resolver: army strength calculated as the sum of *number of soldiers in unit x attack x defense* for all units in the army. Each army then has a uniformly random chance of winning calculated as: *army strength/(army strength + enemy army strength)*. The winning army eliminates a uniformly random proportion of the losing army from: *((winning army strength)/(winning army strength+losing army strength) x 100%)* to 100% of the losing army. The losing army eliminates a uniformly random proportion of the winning army from: 0% to *((losing army strength)/(winning army strength+losing army strength) x 100%)* to 100% of the winning army. A successful invading army should destroy the enemy army as per the spec, and move its soldiers into the captured province. (3)
  + Campaign AI: random enemy choice of building construction, random enemy choice of unit recruitment, random invasions/movement of units by enemy (5)
* Player is able to save/load games (1)
* Player is able to end the turn, progressing to the next player turn after the Campaign AI implements all enemy moves for their turn (2)

GETTER SETTER STYLE

thisIsAnAttribute: getter -> getThisIsAnAttribute. Setter -> setThisIsAnAttribute

Int unitHealth: getter = getUnitHealth setter = setUnitHealth(arguments)

Credit (>=65%)

* All requirements for a pass, and all of the following:
* Player is able to move troops between multiple provinces at a time. All rules regarding movement of troops implemented (e.g. not being able to move units moved into a province conquered in the current turn), except implementing roads
* Player is able to build (and thus obtain the effects of) troop production buildings, farms, ports, markets, mines, basic walls (do not need ballista or archer towers for a credit)
* Costs for buildings/soldiers are implemented, and money (gold) is implemented
* Wealth and taxes mechanic is implemented
* Can win the game by conquering all provinces, or lose the game by losing all provinces, and see an appropriate victory/defeat message

Distinction (>=75%)

* All requirements for a credit, and all of the following:
* Campaign victory and loss implemented fully, including conjunction/disjunctions of subgoals, and random selection of a campaign goal when starting a campaign
* Added all types of roads (including effects on movement points of roads), archery/ballista towers (and some upgrade to defense for towers)
* Implement main menu, including ability to choose a faction from the main menu
* Mercenaries implemented correctly

High Distinction (>=85%)

* All requirements for a distinction, and all of the following:
* Smiths, and effects of smith buildings implemented
* Soldier special abilities implemented
* Full implementation of battle resolver and campaign AI (including all aspects within these such as inflicting of casualties, routing, etc...)

Full marks

* All functionality in the "preliminary client requirements" implemented correctly