⚡

Operation Contracts

Operations: *pauseGame()*
Cross-References: Use-Cases: Pause/Resume Game
Pre-Conditions:

- Game must be already set up.
- The previous playing game has been saved.

Post-Conditions:

- No instance was created.
- No association were formed
- continueSaved() method has been called.

Operations: *resumeGame()*
Cross-References: Use-Cases: Pause/Resume Game
Pre-Conditions:
-Game must be already set up.
Post-Conditions:

- A game instance has been created.
- Previous saved game was loaded.

Operations: *setPlayerCount()*
Cross-References: Use-Cases:(Set Game) Game Setup
Pre-Conditions:

- Game should have just started.
- playerList has been completed.
- Players has been added to list

Post-Conditions:

- Number of players should be fixed
- setPlayerCount(PlayerList playerList)
- A PlayerList instance was created as playerList (Instance creation)
- Number of players has been counted by an enhanced for loop. If the number of players was not valid, the playerNumberError instance was created as countError. (Instance creation)
- Count error was given to reportError(Error e) method.

Operations: *createGame()*
Cross-References: Use-Cases:(Set Game) Game Setup
Pre-Conditions:

- Game should not have started.

Post-Conditions:

- Screen displays game.

- createGame()
- A Game instance was created (Instance creation.)
- addPlayer() method was called.
- A playerList was taken from addPlayer() method.
- game.playerList became playerList from addPlayer(). (Attribute modification)
- game.history became current history. (Attribute modification)

Operations: saveGameSeesion(Game game**)**
Cross-References: Use-Cases: Pause/Resume Game, (Set Game)Game Setup

Pre-Conditions:
- Game is paused.

Post-Conditions:
- Current game progress is stored.
- A Game instance was created as game. (Instance creation)
- game was associated with current game instance(Association formed)
- game.History is assigned to history(Attribute   modification)

Operations: *reportError(Error e)*
Cross-References: Use-Cases: All use cases
Pre-Conditions:
- Player tries invalid action.
- The corresponding error instance has been created and provided.
Post-Conditions:
- System displays an error message accordingly.
- An Error instance was created as an error. (instance creation) **-** No attribute modification.
- No association formed. - @toString() method of error was returned to the user.

Operations: showHelp((optional) str helpCode)
Cross-References: Use-Cases: All use cases
Pre-Conditions:
- Player tries invalid action.
- The corresponding error instance has been created and provided.
Post-Conditions:
- System displays help options.
- General information expression/attribute was created as genInfo.
- General information was assigned to genInfo. **-** genInfo printed/displayed.
- A specific request exp./attribute is created as specReq.
- A click or command was taken from the user and assigned to specReq. (This could be quit from help such as “q”)
- The specific request was displayed to the user.
- Process continue until the user quit from help.
- (optional) Additional or more specific requests accepted.

Operations: addPlayer(str name)
Cross-References: Use-Cases: (Set Game) Game Setup

Pre-Conditions:

- Game should have just started.
- Name of the player has been decided and provided.

Post-Conditions:

- A list of players was called a playerList.
- Player instance was created.(Instance Creation)
- Name is assşgned to Player instance. (Attribute modification)
- Player instance has been added to the playerList.
- A boolean or a message returned.(F.e: “Player has been added”)

Operations: rollAllPlayers()

Cross-References: Use-Cases: Decide player order

Pre-Conditions:

- A round should be just starting

Post-Conditions:

- Dice instances were created. (Instance Creation)
- Dice instances are associated with players (Association Formed)
- Dice instances are rolled (Attribute Modification)

Operations: continueSaved(Game game)

Cross-References: Use-Cases: Set Up Game

Pre-Conditions:

- Previous playing game was paused.
- Previous playing game has been saved.
- Game was launched again by the Player.

Post-Conditions:

- A Game instance was created as a game. (instance creation)
- game was associated with saved Game data. (Association formed)
- This game instance attributes were assigned the previous data(attribute modification)

Operations: gameNext()

Cross-References: Use-Cases: Set Up Game

Pre-Conditions:

- Game was launched
- Game instance was created
- Players has been added and initialized

Post-Conditions:

- Country instances has been created (instance creation)
- Country instances were associated with desired countries. (Association formed)
- Country names were became assigned country names(attribute modification)
- Country positions were became assigned country names(attribute modification)

Operations: toggleCountry()

Cross-References: Use-Cases: Enable/Disable Countries

Pre-Conditions:

- Game is in running mode
- A player has the turn

Post-Conditions:

- Clicked territory is marked as toggled (Association Formed)

Operations: startGame(Game game)
Cross-References: Use-Cases: Enable/Disable Countries
Pre-Conditions:

- a Game instance is created
- Game instance is not in running mode

Post-Conditions:

- Game mode is marked as running (Attribute modification)
- Game is in running mode

Operations: decideFirstPlayer(PlayerList playerList)
Cross-References: Use-Cases: Decide Player Order
Pre-Conditions:

- playerList is created
- A round should be just starting

Post-Conditions:

- rollAllPlayers() method have been called
- rolls associated with players get compared **-** player with the best roll gets first place (association created)
- other players are shuffled (association created)

Operations: startRounds()
Cross-References: Use-Cases: Decide Player Order
Pre-Conditions:

- first player is decided

Post-Conditions:

- first player’s turn

Operations: playRound()
Cross-References: Use-Cases: Decide Player Order
Pre-Conditions:

- players’ order is decided

Post-Conditions:

- every player played their turn

Operations: shareTerritories()
Cross-References: Use-Cases: Share Territories and Place Armies
Pre-Conditions:

- game just started- players’ order is decided

Post-Conditions:

- selectTerritory() method is called for players in loop

Operations: selectTerritory()
Cross-References: Use-Cases: Share Territories and Place Armies
Pre-Conditions:
- territory is toggled and approved
Post-Conditions:
- territory’s owner attribute is modified (Attribute modification)
- player’s territories list is modified (Attribute modification)
- one Army instance is created (Instance Creation)
- exactly one army is appended to player’s armies list (Attribute modification) - exactly one army is appended to player’s armies list (Attribute modification)

Operations: chooseActionExchange(Army userArmy)
Cross-References: Use-Cases: Exchange Armies
Pre-Conditions:
- A round has been started
- Player has been ready for choosing the territory
Post-Conditions:
- Army instance army was created. (Instance creation)
- userArmy was associated with army instance.
- Based on the userArmy.quantity and exchangeRate attributes, appropriate exchange options were displayed to use

Operations: chooseOwnTerritory(Territory territory)
Cross-References: Use-Cases: Exchange Armies
Pre-Conditions:
- A round has been started
- Player has been ready for choosing the territory
Post-Conditions:
- A Territory instance has been created.(Instance created)
- Given territory instance has been associated with created territory. (Association formed)
- The territory's availability attribute has been controlled.(Attribute modification/checking)
- If the territory is available, the territory's availability has been changed to taken.(Attribute modification)
- Return a boolean assigned and conditions of exchange.
- If it is not available, an Error instance has been created as terriotoryError.(Instance creation)
- The error has been thrown to the user.

Operations: exchangeArmies(Army givenArmy, Army takenArmy)
Cross-References: Use-Cases: Exchange Armies
Pre-Conditions:
- Armies has been stayed with user.
- Proper armies should be sufficient for exchanging.
- Army object’s instances has been created for users(There are many three types of armies)
Post-Conditions:
- Two Army instances have been created. (ınstance creation)
- Armies associated with given and taken armies. (Association formed)

- Exchange conditions of takenArmies are checked with givenArmies.
- If conditions are satisfied, change occurs. givenArmy.quantity and takenArmy.quantity has been changed according to exchange conditions. (Attribute modification)
- Boolean returned for a successful exchange.
- Otherwise, for not sufficient conditions, an Error instance exchangeError has been created, and thrown to the user. (Instance creation)

Operations: doAnotherExchange()
Cross-References: Use-Cases: Exchange Armies
Pre-Conditions:
- player chose Not Enough? option
Post-Conditions:
- exchange army phase is restarted via chooseDesiredArmy() method

Operations: stationOnCaptured()
Cross-References: Use-Cases: Conquer Territory
Pre-Conditions:
- a territory was just captured
Post-Conditions:
- player chooses number of armies to cross to conquered territory from attacker territory
- chosen armies are added to conquered territory’s armies list (Attribute modification)
- chosen armies are removed from attacker territory’s armies list (Attribute modification)
- getTerritoryOrArmyCard method is called

Operations: getTerritoryOrArmyCard()
Cross-References: Use-Cases: Conquer Territory
Pre-Conditions:
- armies were stationed to conquered territory
Post-Conditions:
- conqueror gets a random territory or army card from the deck (Association Formed)
- territory cards are added to player’s territoryCards list (Attribute Modification)
- army cards are added to player’s armyCards list (Attribute Modification)