Risk Game Milestone 4- UML Class Diagram

Team Group- Sarah Abdallah 2020-12-08

Please note: Certain classes and interfaces
(such as EventObject, ActionListener, JFrame,
and EventListener are depicted in this diagram,
but are not elaborated because they are
included from Java.

XML

- - MAP_TAG:String
- - CONTINENT_TAG:String
- - CONTINENT_NAME_TAG:String
- - CONTINENT_REINFORCEMENTS_TAG:String
- - COUNTRY_TAG:String
- - COUNTRY_NAME_TAG:String
- - TROOPS_TAG :String
- - X_TAG :String
- - Y_TAG :String
- - NEIGHBORS_TAG :String
- - NEIGHBOR_TAG :String
- - GAME_TAG :String
- - PLAYERS_TAG :String
- - AIPLAYER_TAG :String
- - PLAYER_TAG :String
- - PLAYER_NAME_TAG :String
- - PLAYER_OWNED_TAG :String
- - CURRENT_PLAYER_TAG :String
- - HISTORY_TAG :String
- - CONTINENT_COLOR_TAG:String
- - DEFAULT_MAP_FILENAME:String

- +getDefaultMap():Map
- +mapFromXML(inputFile:File):Map
- + saveGame(model:GameModel):void
- + loadGame(filename:String):GameModel
- -makeMapElement(map:Map, doc:Document) Element
- -mapMaker(continentNodes:NodeList):Map

GameFrame

- -PLAYERSINFO:ArrayList<JTextArea>
- -UPDATELINE:JLabel
- -playerTurnInfo:String
- -BUTTONS:ArrayList<JButton>
- -board:mxGraphComponent
- -GAMECONTROLLER:GameController
- -BOARDPANEL:JPanel
- -graph:mxGraph
- -showHistory:JMenuItem
- -gameMenu:JMenu
- + VERTEX_STYLE:String
- + VERTEX_STYLE_ONE_WORD:String
- +WIDTH:int
- +HEIGHT:int
- +EDGE_STYLE:String
- +PLAYER1:String

- +GameFrame(name:String)
- +GameFrame(name:String, gameModel:GameModel)
- -makeFrame(gameModel:GameModel):void
- -setPlaceholder():void
- +reset():void
- -makeMenuBar():void
- -createBoard(map:Map):mxGraphComponent
- -insertVertex(name:String, xPosition:int, yPosition:int):Object
- -insertEdge(vertex1:Object ,vertex2:Object, color:String):void
- -getVertexStyle(name:String):String
- -getColorForPlayerIndex(playerIndex:int):String
- +handleGameStart(gameModel:GameStartEvent):void
- +updateColors(players:ArrayList<Player>):void
- +getConstraints(gridX:int, gridY:int, gridWidth:int, gridHeight:int, fill:int):GridBagConstraints
- +handlePlayerTurnUpdate(playerTurn: PlayerTurnEvent):void
- +handleTurnStateChange(turnState:TurnStateEvent):void
- +handleReset():void
- +handlePlayerElimination(eliminatedEvent:PlayerEliminatedEvent):void
- +handleGameOver(gameOverEvent: GameOverEvent):void
- +handleStateUpdate(playerState:PlayerStateEvent):void
- +handleOwnerChange(ownerChange:OwnerChangeEvent):void
- +main(args:String[]):void

JFrame

GameController

- +State:enum
- -HELP_COMMAND:String
- -NEW_COMMAND:String
- -ATTACK_COMMAND:String
- -MOVE_COMMAND:String
- -END_COMMAND:String
- -EMPTY:String
- -HISTORY_COMMAND:String
- -GAMEMODEL:GameModel
- -from:String
- -to:String
- -state:State

- +GameController(gm:GameModel)
- +addGameBoard(gameBoard:mxGraphComponent):void
- -toAttackPhase():void
- +actionPerformed(e:ActionEvent):void
- -createNewGame():void
- +getCurrentReinforcements():int
- + reinforcementState(clickedCountry: String):void
- + attackState(clickedCountry: String):void
- + movementState(clickedCountry: String):void
- + helpCommand():void
- + newCommand():void
- + attackCommand():void
- + moveCommand():void
- + endCommand():void
- + historyCommand():void
- + setEmpty():void
- + checkCountry(clickedCountry: String) Boolean
- + checkSuccessfulMove():void
- - checkReinforcements():void
- - loadCommand():void
- - saveCommand():void

<<enumeration>> State
- UNDECLARED
- REINFORCEMENT
- ATTACK
- MOVEMENT

<<interface>> ActionListener

<<interface>> EventListener

<<interface>> GameView

- + handleGameStart(gameModel:GameStartEvent):void
- + handleStateUpdate(playerState:PlayerStateEvent):void
- + handlePlayerTurnUpdate(playerTurn:PlayerTurnEvent):void
- + handleOwnerChange(ownerChange: OwnerChangeEvent):void
- + handleTurnStateChange(turnState:TurnStateEvent):void
- + handleResetView():void
- + handlePlayerElimination(eliminatedEvent:PlayerEliminatedEvent):void
- + handleGameOver(gameOverEvent:GameOverEvent):void
- +reset():void

GameModel

- -players:ArrayList<Player>
- -currentPlayer:Player
- -currentPlayerReinforcements:int
- -map:Map
- -GAMEVIEWS: ArrayList<GameView>
- -history:StringBuilder
- -gameStarted:boolean
- -BEGINNING_TROOPS:int[]

- +GameModel()
- +GameModel(players:ArrayList<Player>)
- +resetModel():void
- +getPlayers():ArrayList<Player>
- +getMap():Map
- +addGameView(gameView:GameView):void
- -resetView():void
- -updateGameViewsStart():void
- -updateGameViewsTurnState(newState:State):void
- -updateGameViewsState(stateInfo:String, playerOrder:int):void
- -updateGameViewsOwnerChange(countryName:String, playerOrder:int):void
- -updatePlayerTurn(playerName:String):void
- +userCreateGame():boolean
- +loadGame():boolean
- -generateGame():void
- -getReinforcements():int
- +getCurrentPlayerReinforcements():int
- +placeAllReinforcements(country:Country):void
- +placeAllReinforcements(country:Country):void
- +playAttack(attackingCountry:Country, defendingCountry:Country, blitzAttack:boolean):void
- +playAttack(attackingCountry:Country, defendingCountry:Country, blitzAttack:boolean):void
- -checkAttackValid(attacking:Country, defending:Country):void
- -performBlitzAttack(attack:Country, defend:Country):void
- -performAttack(attack:Country, defend:Country):void
- -ownerChange(defend:Country, attack:Country, minimumMove:int):void
- -handlePlayerElimination(p:Player):void
- -handleGameOver():void
- +addPlayer(player:Player):void
- +nextPlayer():void
- +showCurrentPlayer():void
- +getCurrentPlayer():Player
- +getRemainingPlayers():int
- +moveTroops(origin:Country , destination:Country, toMove:int):boolean
- +moveTroops(origin:String, destination:String):boolean
- +updateState():void
- +showErrorPopUp(e:Exception):void
- +showMessage(message:String)
- -getPlayerNames():LinkedList<Player>
- +playerOwner(country:String):boolean
- +troopSelect(minimum:int, maximum:int):int
- -makeProperCountryNames(name:String):String
- +getHistory():String
- +getHelp():String
- +setMap(map:Map):void
- +setPlayers(players:ArrayList<Player>):void
- +setCurrentPlayer(currentPlayer:Player):void
- +setHistory(history:StringBuilder):void

<<interface>> Serializable

EventObject

AIPlayer

- -model:GameModel

- +AIPlayer(name:String, gameModel:GameModel)
- +setModel(model:GameModel):void
- +handleError(e:Exception):void
- +handleMessage(message:String)
- +playTurn():void
- -autoFullReinforcements(reinforcements:int):void
- +autoAttack():void
- -autoMove():void
- +troopSelect(minimum:int, maximum:int):int

GameOverEvent

- -WINNER:Player

- +GameOverEvent(gameModel:GameModel, winner:Player)
- +getWinner():Player

PlayerEliminatedEvent

- -ELIMINATED:Player
- -PLAYERNUMBER:int

- +PlayerEliminatedEvent(gameModel:GameModel , eliminated:Player,playerNumber:int)
- +getEliminated():Player
- +getPlayerNumber():int

GameStartEvent

- -MAP:Map
- -PLAYERS:ArrayList<Player>

- +GameStartEvent(gameModel:GameModel, map:Map, players:ArrayList<player>)
- +getPlayers():ArrayList<Player>
- +getMap():Map

OwnerChangeEvent

- -COUNTRYNAME:String
- -PLAYERORDER:int

- +OwnerChangeEvent(gameModel:GameModel , countryName:String, playerOrder: int)
- +getCountryName():String
- +getOrder():int

PlayerTurnEvent

- -PLAYERNAME:String
- -ORDER:int

- +PlayerTurnEvent(gameModel:GameModel, name:String,order:int)
- +getName():String
- +getOrder():int

TurnStateEvent

- -NEWSTATE:State

- +TurnStateEvent(gameModel:GameModel , newState:State)
- +getNewState():State

PlayerStateEvent

- -INFO:String
- -PLAYERORDER:int

- +PlayerStateEvent(gameModel:GameModel , info:String,order:int)
- +getInfo():String
- +getOrder():int

Player

- -COUNTRIES:LinkedList<Country>
- -NAME:String

- +Player(name:String)
- +addCountry(country:Country):void
- +assignBeginningTroops(beginningTroops:int):void
- +getName():String
- +isEliminated():boolean
- +numberOfCountries():int
- +lost(c:Country):void
- +hasCountry(country:Country):boolean
- +hasCountry(country:String):boolean
- +hasCountries(countries:ArrayList<Country>):boolean
- +getInfo():String
- +sortCountries():void
- -pathExists(start:Country, finish:Country):boolean
- -getAccessibleCountries(country:Country, goal:Country, accessibleCountries:ArrayList<Country>):void
- +getPerimeterCountries():ArrayList<Country>
- +getInnerCountries():ArrayList<Country>
- +playTurn():void
- +getCountries():LinkedList<Country>
- +getNumberOfTroops():int
- +troopSelect(minimum:int, maximum:int):int
- +handleError(e:Exception):void
- +handleMessage(message:String):void

Map

- -COUNTRIES:HashMap<String, Country>
- -CONTINENTS:HashMap<String,Continent>
- -POSITIONSCOUNTRIES:HashMap<Country,Point>
- +BLACK:String

- +Map(countries:HashMap<String, Country>, continents: HashMap<String, Continent>, positionsCountries:HashMap<Country, Point>)
- +getCountry(name:String):Country
- +getContinents():ArrayList<Continent>
- +getReinforcements():int
- +getShuffledKeys():ArrayList<String>
- +getAllCountries():ArrayList<Country>
- +getPosition(country:Country):Point
- +checkMapValidity():boolean
- +getAccessibleCountries(accessibleCountries:HashMap<String, Boolean>,currentCountry:Country):void

Continent

- -NAME:String
- -countries:ArrayList<Country>
- -REINFORCEMENTS:int
- -color:String

- +Continent(name:String, reinforcements:int, color:String)
- +getName():String
- +getCountries():ArrayList<Country>
- +getReinforcements():int
- +addCountries(countries:ArrayList<Country>):void
- +addCountry(country:Country):void
- +getColor():String

Country

- -NAME:String
- -NEIGHBORS:ArrayList<Country>
- -troops:int

- +Country(name:String)
- +Country(name:String, troops:int)
- +getName():String
- +addNeighbor(neighbor:Country):void
- +hasNeighbor(c:Country):boolean
- +getNeighbors():ArrayList<Country>
- +addTroop(num:int):void
- +getTroops():int
- +removeTroops(num:int):void
- +toString():String
- +equals(e:Object):boolean