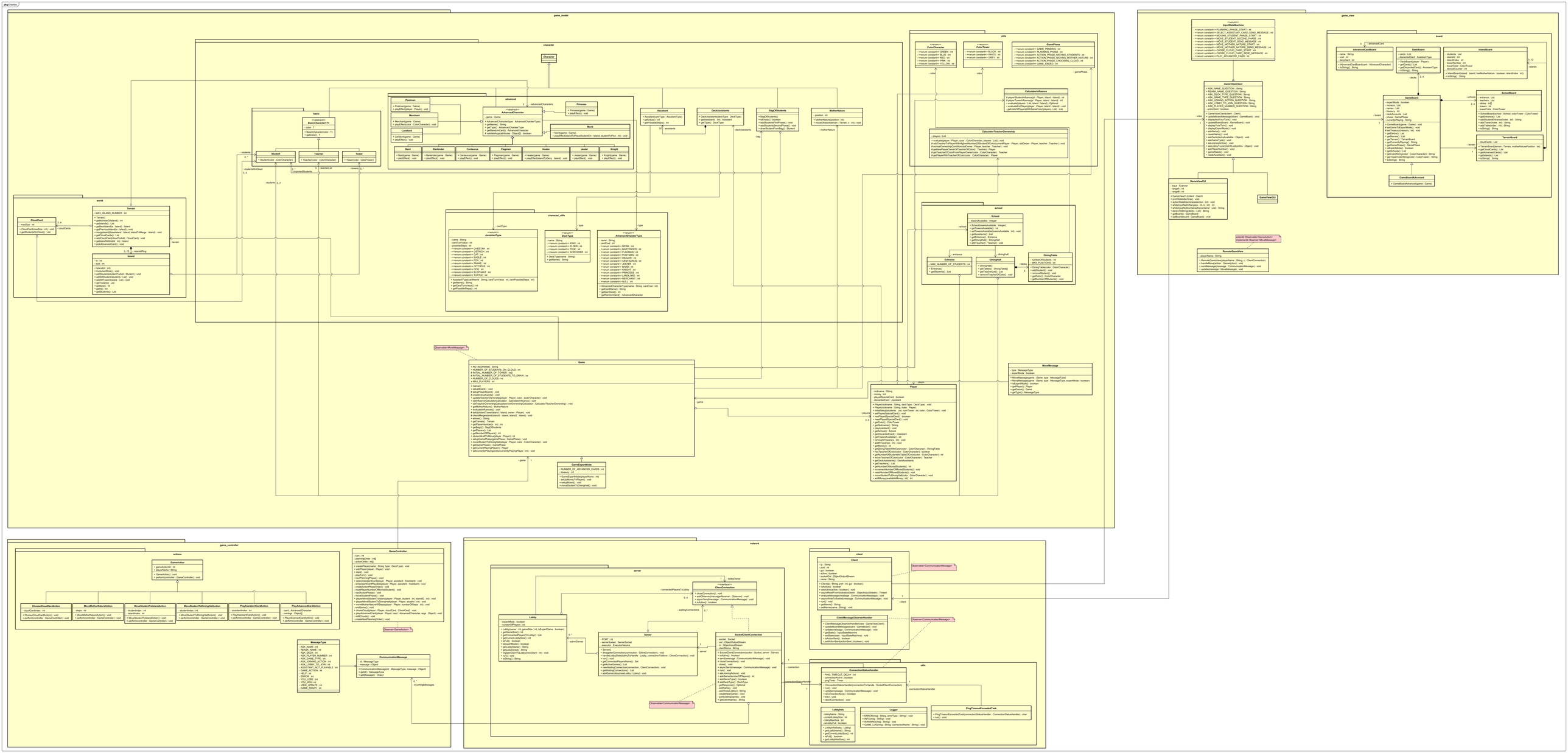
phpDiagram
game_model
character
Character
advanced
Postman
+ Postman(game : Game)
+ play(Recipe(pair : Player) : void
Merchant
+ Merchant(game : Game)
+ play(Recipe(color : CoinCharacter) : void
Landlord
+ Landlord(game : Game)
+ play(Recipe() : void
AdvancedCharacter
game : Game
+ AdvancedCharacter(type : AdvancedCharacterType)
+ getType() : AdvancedCharacterType
+ getRandomCard() : AdvancedCharacter
# validateApplyAttribute() : DeckS) : boolean
Princess
+ Princess(game : Game)
+ play(Red() : void
Bard
+ Bard(game : Game)
+ play(Red() : void
Barbender
+ Barbender(game : Game)
+ play(Red() : void
Centaurus
+ Centaurus(game : Game)
+ play(Red() : void
Flagman
+ Flagman(game : Game)
+ play(Red() : void
Healer
+ Healer(game : Game)
+ play(RedStudentFrOwn) : Island) : void
Jester
+ Jester(game : Game)
+ play(Red() : void
Knight
+ Knight(game : Game)
+ play(Red() : void
Monk
+ Monk(game : Game)
+ play(RedStudentFrPlaceStudentOn : Island, studentToPick : int) : void
basic
AbstractTP
AbstractCharacterTP
- color : T
+ BasicCharacter(color : T)
+ getColor() : T
Student
+ StudentEnum : CoinCharacter
Teacher
+ TeacherEnum : CoinCharacter
Tower
+ TowerColor : ColorTower
Assistant
+ AssistantUserType : AssistantType)
+ getValue() : int
+ getPossibleSteps() : int
DeckAssistants
+ DeckAssistants(deckType : DeckType)
+ getAssistants() : Int) : Assistant
+ getDeck() : DeckType
BagOfStudents
+ BagOf(Students)
+ isEmpty() : boolean
+ getNumberOfStudents() : int
+ addStudentsAccordPlayer() : void
+ drawStudentFromBag() : Student
MotherNature
- position : int
+ MotherNature(position : int)
+ getPositionIsland(remain_n : int) : int
world
CloudCard
- nsize : int
+ CloudCard(nsize [n]) : void
+ getStudentsOnCloud() : List
Terrain
- MAX_ISLAND_NUMBER : int
+ Terrain()
+ getNumberOfIslands() : int
+ getIsland(n) : List
+ getPreviousIsland(n : Island) : Island
+ getNextIsland(n : Island) : Island
+ mergeIslandSameIsland : Island, islandToMerge : Island) : void
+ getIsland(n : int) : Island
+ addCloudCardsStudentToAdd : CloudCard) : void
+ getIslandWithStud : int) : Island
+ pickAdvancedCard() : void
Island
- id : int
- size : int
+ Island(id : int)
+ increaseSize() : void
+ addStudentsStudentToAdd : Student) : void
+ addTowerTowerToAdd : Tower) : void
+ addNoEntryTowerToAdd : Tower) : void
+ getTowers() : List
+ getSize() : int
+ getID() : int
+ getStudents() : List
character_utils
AssistantType
- name : String
- value : int
- possibleSteps : int
+ mnum constant= CHEETAH: int
+ mnum constant= DOG: int
+ mnum constant= CAT: int
+ mnum constant= EAGLE : int
+ mnum constant= FOX: int
+ mnum constant= SNAKE : int
+ mnum constant= OCTOPUS: int
+ mnum constant= LION: int
+ mnum constant= ELEPHANT : int
+ mnum constant= TURTLE : int
+ AssistantType(pName : String, cardTurnValue : int, cardPossibleSteps : int)
+ getName() : String
+ getValue() : int
+ getPossibleSteps() : int
DeckType
- name : String
+ mnum constant= KING: int
+ mnum constant= ELDER : int
+ mnum constant= POSE : int
+ mnum constant= SORCERER : int
+ DeckType(name : String)
+ getName() : String
AdvancedCharacterType
- name : String
- cost : int
+ mnum constant= MONK : int
+ mnum constant= BARTENDER : int
+ mnum constant= CENTAUR : int
+ mnum constant= POSTMAN : int
+ mnum constant= HEALER : int
+ mnum constant= FLAGMAN : int
+ mnum constant= JESTER : int
+ mnum constant= BARD : int
+ mnum constant= KNIGHT : int
+ mnum constant= LANDLORD : int
+ mnum constant= CHANCELLOR : int
+ mnum constant= MERCHANT : int
+ mnum constant= NULL : int
+ AdvancedCharacterType(name : String, cardCost : int)
+ getCardName() : String
+ getCardCost() : int
+ getRandomCard() : AdvancedCharacter
utils
<enum>
ColorCharacter
- mnum constant= GREEN : int
- mnum constant= BLUE : int
- mnum constant= RED : int
- mnum constant= PINK : int
- mnum constant= YELLOW : int
<enum>
ColorTower
- mnum constant= BLACK : int
- mnum constant= WHITE : int
- mnum constant= GREY : int
GamePhase
- mnum constant= GAME_PENDING : int
- mnum constant= PLANNING_PHASE : int
- mnum constant= ACTION_PHASE_MOVING_STUDENTS : int
- mnum constant= ACTION_PHASE_MOVING_MOTHER_NATURE : int
- mnum constant= ACTION_PHASE_CHOOSING_CLOUD : int
- mnum constant= GAME_ENDED : int
CalculateInfluence
pUpdateStudentInRoomcp() : Player, island : Island) : void
pUpdateTowerInRoomcp() : Player, island : Island) : void
evaluate(pIsles : List, island : Island) : Optional
pUpdateTeacherInRoomcp() : Player, pOldOwner : Player) : void
- getLastOfPlayerWithSameColorIsland() : List : List
CalculateTeacherOwnership
- pIsleIs : List
+ evaluate(pIsler : Player, color : ColorCharacter, pIsleIs : List) : void
# addTeacherToPlayerWithHigherCoinsLowerPlayer : Player, pIsleOwner : Player, teacher : Teacher) : void
# noneOfOwnershipConditionsIsOwner : Player, teacher : Teacher) : void
# addTeacherToPlayerWithSameCoins : Player, teacher : Teacher) : void
# getTeacherOfColorFromPlayerOwnerIsIn : ColorCharacter) : Teacher
# getPlayerWithTeacherOfColorIsIn : ColorCharacter) : Player
school
School
- towersAvailable : Integer
+ SchoolJ(towersAvailable : Integer)
+ getTowersAvailable() : int
+ setTowersAvailable(towersAvailable : int) : void
+ getDiningHall() : DiningHall
+ getEntrance() : Entrance
+ getDiningHall() : DiningHall
+ addTeacher(t : Teacher) : void
Entrance
- MAX_NUMBER_OF_STUDENTS : int
+ Entrance()
+ getStudents() : List
DiningHall
+ DiningHall()
+ getDiningTable() : DiningTable[]
+ getTeacher(ct) : List
+ removeTeacherOf(ct) : void
DiningTable
- numberOfStudents : int
- MAX_POSITIONS : int
+ DiningTable(color : ColorCharacter)
+ addStudent() : void
+ removeStudent() : void
+ getNumberOfStudents() : int
+ getNumberOfStudents() : int
Game
+ NO_NOWNME : String
+ MAX_NUMBER_OF_STUDENTS_ON_CLOUD : int
# INITIAL_NUMBER_OF_TOWER : int[]
# INITIAL_NUMBER_OF_STUDENTS_TO_DRAW : int
# MAX_NUMBER_OF_CLOUDS : int
# MAX_PLAYERS : int
+ Game()
+ setupBoard() : void
+ getNumberOfBoard() : int
# initializeCloudCards() : void
+ updateTeacherOwnership(player : Player, color : ColorCharacter) : void
+ setInfluenceCalculatorIsCalculator : CalculateInfluence) : void
+ setTeacherOwnershipCalculatorIsCalculator : CalculateTeacherOwnership) : void
+ getMotherNature() : MotherNature
+ evaluateIfGameIsOver() : void
+ pUpdateStudentInRoomcp(island, owner : Player) : void
# checkAssignpossibleplayer() : Island, island2 : Island) : void
+ winner() : String
+ getPlayers() : List
+ getPlayerAssistant() : int) : int
+ getBag() : BagOfStudents
+ getBag() : int
+ getNumberOfPlayers() : int
+ studentLeftToWinnerplayer : Player) : int
+ moveStudentToDiningHall(pIsler : Player, color : ColorCharacter) : void
+ moveStudentToDiningHallplayer : Player, color : ColorCharacter) : void
+ getGamePhase() : GamePhase
+ getCurrentPlayingPlayer() : Player
+ setCurrentPlayingIndexCurrentPlayingPlayer : int) : void
GameExpertMode
- NUMBER_OF_ADVANCED_CARDS : int
+ GameExpertMode()
+ GameExpertMode(playerNums : int)
+ setUpBoard() : void
+ setupBoard() : void
+ moveStudentToDiningHall() : void
Player
- nickname : String
- money : int
- hasPlayedCard : boolean
- discardedCard : Assistant
+ Player(nickname : String, deckType : DeckType) : void
+ Player(nickname : String, mate : Player)
+ initialSetup(students : List, numTower : int, color : ColorTower) : void
+ getNickname() : String
+ hasPlayedSpecialCard() : boolean
+ resetPlayedSpecialCard() : void
+ getMoney() : int
+ getNickname() : String
+ playAssistant() : void
+ getDeck() : DeckType
+ getDiscardedCard() : Assistant
+ getTowersAvailable() : int
+ setTowersAvailable() : void
+ addNTowers(n : int) : void
+ getMinus1() : int
+ getDiningTableWithColor(color : ColorCharacter) : DiningTable
+ hasTeacherOfColor(color : ColorCharacter) : boolean
+ getNumberOfStudentsOfColorInDiningHallOfPlayer(color : ColorCharacter) : int
+ moveTeacherOfColor(color : ColorCharacter) : Teacher
+ getDeckAssistants() : DeckAssistants
+ getBag() : BagOfStudents
+ getNumberOfMovedStudents() : int
+ incrementNumberOfMovedStudents() : void
+ resetNumberOfMovedStudents() : void
+ moveStudentToDiningHallplayer : ColorCharacter) : void
+ addMoney(availableMoney : int) : int
MoveMessage
- type : MessageType
- expertMode : boolean
+ MoveMessage(game : Game, type : MessageType)
+ MoveMessage(game : Game, type : MessageType, expertMode : boolean)
+ isExpertMode() : boolean
+ getMove() : int
+ getGame() : Game
+ getType() : MessageType
game_view
<enum>
InputStateMachine
+ mnum constant= PLANNING_PHASE_START : int
+ mnum constant= SELECT_ASSISTANT_CARD_SEND_MESSAGE : int
+ mnum constant= MOVING_STUDENT_PHASE_START : int
+ mnum constant= SELECT_STUDENT_SEND_MESSAGE : int
+ mnum constant= MOVE_STUDENT_SEND_MESSAGE : int
+ mnum constant= MOVE_MOTHER_NATURE_START : int
+ mnum constant= MOVE_MOTHER_NATURE_SEND_MESSAGE : int
+ mnum constant= CHOSE_CLOUD_CARD_START : int
+ mnum constant= CHOSE_CLOUD_CARD_SEND_MESSAGE : int
+ mnum constant= PLAY_ADVANCED_CARD : int
GameViewClient
+ ASK_NAME_QUESTION : String
+ REASK_NAME_QUESTION : String
+ ASK_DECK_TYPE_QUESTION : String
+ ASK_GAME_TYPE_QUESTION : String
+ ASK_JOINING_ACTION_QUESTION : String
+ ASK_LOBBY_TO_JOIN_QUESTION : String
+ ASK_LOBBY_NAME_QUESTION : String
+ actionStart : boolean
+ GameViewClient(client : Client)
+ updateGameMessageInstant : GameBoard) : void
+ displayNotifyToTurn() : void
+ displayGameBoard(board : GameBoard) : void
+ displayYourTurn() : void
+ displayExpertMode() : void
+ getClient() : Client
+ getTerminal() : Int
+ getGameBoard() : GameBoard
+ getCurrentPlaying() : String
+ getGamePhase() : GamePhase
+ isCurrentPlaying() : boolean
+ getDeckIs() : List
+ getPlayers() : List
+ getCurrentPlayer() : Player
+ setCurrentPlayer(name : int) : void
+ getDeckIs(n : int) : Deck
+ getPlayer() : Int
+ getCurrentPlayer() : Int
+ askLoginAction() : void
+ askLobbyToJoinOfLobbyInfos : Object) : void
+ askLobbyName() : void
+ getNickIs() : int
+ getLoginStringColor : ColorCharacter) : String
+ getLoginStringColor : ColorTower) : String
+ toString() : String
GameViewCLI
- input : Scanner
- rangeA : int
- rangeB : int
+ GameViewCLI(client : Client)
+ getNickname() : String
+ askforBattleAchievement(action : int) : void
+ whichIsyourNotInRange(a : int, b : int) : int
+ displayLobbyToJoin(lobbies : List) : String
+ deleteTrailingIsle(s : List) : String
+ getBoard() : GameBoard
+ setBoard(board : GameBoard) : void
GameViewGUI
RemoteGameView
- clientName : String
+ RemoteGameView(gameName : String, c : ClientConnection)
+ handleBroadcast(m : GameAction) : void
+ sendMessage(message : CommunicationMessage) : void
+ updateMessage : MoveMessage) : void
extends Observable<GameAction>
implements Observer<MoveMessage>
board
AdvancedCardBoard
- name : String
- cost : int
- deckCard : int
+ AdvancedCardBoard(card : AdvancedCharacter)
+ toString() : String
DeckBoard
- cards : List
- discardedCard : AssistantType
+ DeckBoard(player : Player)
+ getCards() : List
+ getDiscardedCard() : AssistantType
+ toString() : String
IslandBoard
- students : List
- islandID : int
- islandSize : int
- hasMotherNature : boolean
- towerColor : ColorTower
- noteaCounter : int
+ IslandBoard(island : Island, hasMotherNature : boolean, islandSize : int)
+ getIsland() : Island
+ toString() : String
GameBoard
- expertMode : boolean
- numPlayers : int
- names : List
- history : int
- advancedCards : List
- phase : GamePhase
- currentPlaying : String
+ GameBoard(numPlayers : int)
+ setGamePhase(phase : int) : void
# setTreasure(treasure : int) : void
+ getDeck() : List
+ getPlayers() : List
+ getCurrentBoard : GameBoard
+ getCurrentPlaying() : String
+ getGamePhase() : GamePhase
+ isExpertMode() : boolean
+ getDeckIs() : List
+ getColorStringColor : ColorCharacter) : String
+ getColorStringColor : ColorTower) : String
+ toString() : String
SchoolBoard
- entrance : List
- teachers : List
- tables : int[]
- towers : int
- towerColor : ColorTower
+ SchoolBoard(school : School, colorTower : ColorTower)
+ getEntrance() : List
+ addStudentInEntrance(student : int) : String
+ getTeachers() : List
+ addTeacher(teacher : int) : String
+ toString() : String
TerrainBoard
- islandCards : List
+ TerrainBoard(terrain : Terrain, motherNaturePosition : int)
+ getIsland() : Island
+ getAdvancedCards() : List
+ getIslandsIs() : List
+ toString() : String
GameBoardAdvanced
+ GameBoardAdvanced(game : Game)
game_controller
actions
GameAction
+ gameActionID : int
+ clientName : String
+ GameAction() : void
+ performUpdate(controller : GameController) : void
ChooseCloudCardAction
- cloudCardIndex : int
+ ChooseCloudCardAction() : void
+ performUpdate(controller : GameController) : void
MoveMotherNatureAction
- steps : int
+ MoveMotherNatureAction() : void
+ performUpdate(controller : GameController) : void
MoveStudentToIslandAction
- studentIndex : int
+ MoveStudentToIslandAction() : void
+ performUpdate(controller : GameController) : void
MoveStudentToDiningHallAction
- studentIndex : int
+ MoveStudentToDiningHallAction() : void
+ performUpdate(controller : GameController) : void
PlayAssistantCardAction
- assistantIndex : int
+ PlayAssistantCardAction() : void
+ performUpdate(controller : GameController) : void
PlayAdvancedCardAction
- card : AdvancedCharacter
+ PlayAdvancedCardAction() : void
+ performUpdate(controller : GameController) : void
GameController
- turn : int
- gameController : int[]
- actionOrder : int[]
+ createPlayerGame : String, type : DeckType) : void
+ addPlayer(player : Player) : void
+ start() : void
+ setup() : void
- newPlanningPhase() : void
+ selectAssistantCardplayer : Player, assistant : Assistant) : void
+ checkIfAllPlayersHaveSelectedAssistant(player : Assistant) : void
- resetAllAssistantPlayedCard() : void
- resetPlayerNumberOfMovedStudents() : void
- newActionPhase() : void
- moveStudentPhase() : void
+ playerMoveStudentToIslandplayer : Player, student : int, islandID : int) : void
+ moveStudentToDiningHallplayer : Player, student : int) : void
+ moveMotherNatureOfStepsplayer : Player, numberOfSteps : int) : void
- endGame() : void
+ chooseCloudCardplayer : Player, cloudCard : CloudCard) : void
+ playAdvancedCardplayer : Player, card : AdvancedCharacter, args : Object) : void
- endActionPhase() : void
- newMotherNaturePlacingOrder() : void
Observer<GameAction>
MessageType
- ASK_NAME : int
- REASK_NAME : int
- ASK_GAME : int
- ASK_PLAYER_NUMBER : int
- ASK_GAME_TYPE : int
- ASK_DECK_TYPE : int
- ASK_LOBBY_TO_JOIN : int
- ASSISTANT_NOT_PLAYABLE : int
- GAME_ACTION : int
- HELP : int
- YOU_WIN : int
- YOU_LOSE : int
- YOU_WIN : int
- GAME_ENDED : int
- GAME_READY : int
CommunicationMessage
- id : MessageType
- message : Object
+ CommunicationMessage(id : MessageType, message : Object)
+ getID() : MessageType
+ getMessage() : Object
network
server
Lobby
- expertMode : boolean
- numberOfPlayers : int
+ Lobby(server : int, gameSize : int, isExpertGame : boolean)
+ getGameSize() : int
+ getAvailableSlots(pLobby) : List
+ getCurrentLobbyNow() : int
+ isFull() : boolean
+ isExpertMode() : boolean
+ getLobbyName() : String
+ getLastJoined() : String
+ registerClientToLobby(newClient : int) : void
+ getID() : int
+ toString() : String
ClientConnection
- socket : Socket
- out : ObjectOutputStream
- in : ObjectInputStream
- clientName : String
+ ClientConnection(socketIsSocket : Socket, server : Server)
+ isAlive() : boolean
+ sendMessage : CommunicationMessage) : void
+ getClientName() : void
+ close() : void
+ asyncSendMessage : CommunicationMessage) : void
+ run() : void
+ waitUntilLogin() : void
+ getNickIsNewClientOfName() : int
+ setGameType() : boolean
+ setDeckType() : DeckType
+ getNickIsNewClient() : int
- askName() : void
- askChoseLobby() : String
- askGameSize() : int
- joinExistingGame() : void
- getUserName() : String
Server
- START : int
- serverSocket : ServerSocket
- executor : ExecutorService
+ Server()
+ addNewClientConnectionconnection : ClientConnection) : void
+ handleLobyDeleteIsToHandle : Lobby, connectionToBeve : ClientConnection) : void
+ run() : void
+ getNextFreePlayerName() : Sel
+ getAllAvailableLobbies() : List
+ resetWaitingConnection(connection : ClientConnection) : void
+ getWaitingConnections() : List
+ addGameLobbyUnwaitlisty : Lobby) : void
ClientMessageObserverHandler
+ ClientMessageObserverHandler(view : GameViewClient)
+ setGameViewClient(view : GameViewClient) : void
+ update(message : CommunicationMessage) : void
+ getStep() : InputStateMachine
+ setStep(step : InputStateMachine) : void
+ isActionSent() : boolean
+ setIsActionSent(actionSent : boolean) : void
client
Client
- ip : String
- port : int
- gui : boolean
- active : boolean
- out : ObjectOutputStream
- name : String
+ Client(ip : String, port : int, gui : boolean)
+ isActive() : boolean
+ setActive(active : boolean) : void
+ startMessageHandlerThreadin : ObjectInputStream) : Thread
+ sendUnwaitingMessage(message : CommunicationMessage) : void
+ asyncWriteToSocket(message : CommunicationMessage) : void
+ getIP() : int
+ getName() : String
+ setName(name : String) : void
Observable<CommunicationMessage>
Observer<CommunicationMessage>
utils
ConnectionStatusHandler
- PING_TIMEOUT_DELAY : int
- connectionIsAlive : boolean
- pingTime : Time
+ ConnectionStatusHandler(connectionToHandle : SocketClientConnection)
+ run() : void
+ updateMessage : CommunicationMessage) : void
+ setConnectionIsAlive() : boolean
+ kill() : void
+ abortConnection() : void
LobbyInfo
- lobbyName : String
- currentLobbySize : int
- isFull : boolean
- isLobbyFull : boolean
+ LobbyInfo(lobby : Lobby)
+ getLobbyName() : String
+ getCurrentLobbySize() : int
+ isFull() : boolean
+ getLobbyMaxSize() : int
Logger
+ ERROR(msg : String, errorType : String) : void
+ INFO(msg : String) : void
+ WARNING(msg : String) : void
+ GAME_LOG(msg : String, connectionName : String) : void
PingTimeoutExceededTask
+ PingTimeoutExceededTask(connectionStatusHandler : ConnectionStatusHandler) : char
+ run() : void