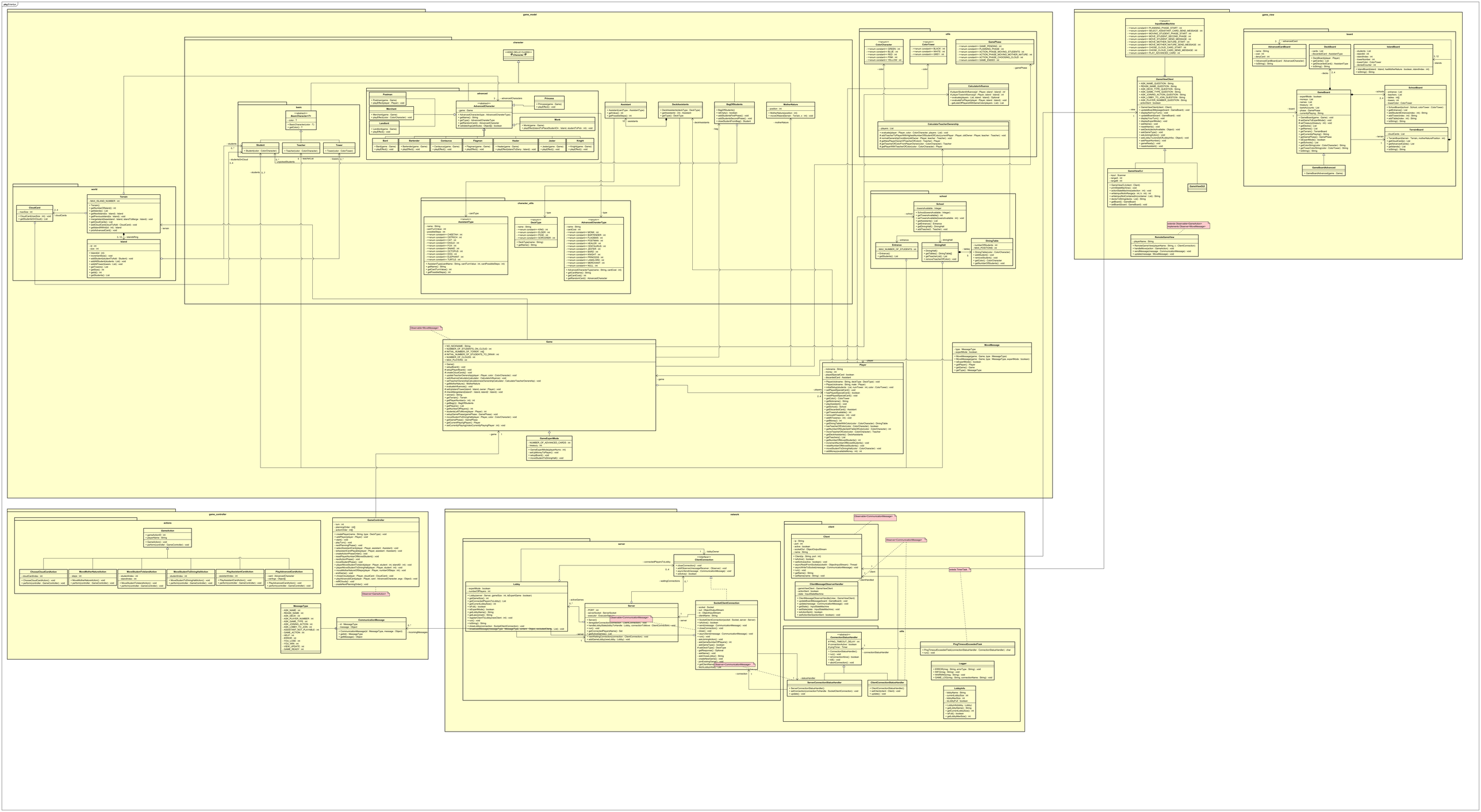
pkgEriantys
game_model
character
<<static>>
CHIAVI DELLE CLASSI
Character
advanced
Postman
+ PostmanGame : Game
+ playRing(player : Player) : void
Merchant
+ MerchantGame : Game
+ playRes(color : ColoreCharacter) : void
Landlord
+ LandlordGame : Game
+ playRes(c1, c2) : void
Princess
+ PrincessGame : Game
+ playRes(c) : void
<<abstract>>
AdvancedCharacter
- game : Game
+ AdvancedCharacter(type : AdvancedCharacterType)
+ getCost() : int
+ getType() : AdvancedCharacterType
+ getRandomCard() : AdvancedCharacter
# validateArgsAttribute : Object[] : boolean
- advancedCharacters
3..3
Bard
+ BardGame : Game
+ playRes(c) : void
Barbender
+ BarbenderGame : Game
+ playRes(c) : void
Centaurus
+ CentaurusGame : Game
+ playRes(c) : void
Flagman
+ FlagmanGame : Game
+ playRes(c) : void
Healer
+ HealerGame : Game
+ playRes(StudentToGrey : Island) : void
Jester
+ JesterGame : Game
+ playRes(c) : void
Knight
+ KnightGame : Game
+ playRes(c) : void
Work
+ MonkGame : Game
+ playRes(StudentToPlaceStudentOn : Island, studentToPick : int) : void
Assistant
+ AssistantLastType : AssistantType[]
+ getValue() : int
+ getPossibleSteps() : int
10..1
- assistants
DeckAssistants
+ DeckAssistantsJokeType : DeckType[]
+ getHasJoke() : int[] : Assistant
+ getDeck() : DeckType
- deckAssistants
BagOfStudents
+ BagOfStudents()
+ isEmpty() : boolean
+ addStudent(pickedFrom) : void
+ addStudentsAroundPhase() : void
+ drawStudentFromBag() : Student
- bag
MotherNature
- position : int
+ MotherNature(position : int)
+ moveOfNaturalnumber(numero_n : int) : void
- motherNature
beas
<<abstract>>
AbstractBeanTP
- color : T
+ BeanCharacter(color : T)
+ getColor() : T
Student
+ StudentEnter : ColoreCharacter
Teacher
+ TeacherEnter : ColoreCharacter
Tower
+ TowerEnter : ColoreTower
0..7
- students
0..n
- studentsOnCloud
1..4
0..n
- students
0..7
0..n
- teacherLeft
1
- teacherLeft
0..n
- towers
0..n
- towers
0..n
UnplayedStudents
world
CloudCard
- mnEnter : int
+ CloudCardInitSize : int[] : void
+ getStudentsOnCloud() : List
1..4
- cloudCards
Terrain
- MAX_ISLAND_NUMBER : int
+ Terrain()
+ getNumberOfIsland() : int
+ getIslands() : List
+ getIslandWithId(id : Island) : Island
+ getPreviousIsland(is : Island) : int
+ mergeIslands(firstIsland : Island, islandToMerge : Island) : void
+ getIsNoEntryCard() : int
+ addCloudCardsStartToAll() : CloudCard[] : void
+ getIslandWithId(id : int) : Island
+ pickAdvancedCard() : void
3..12
- islandsRing
- terrain
Island
- id : int
- size : int
+ Island(id : int)
+ increaseSize() : void
+ addStudentsStudentToAll() : Student[] : void
+ addStudentsOnIsland(s : Student) : void
+ addMoveTowers(towers : List) : void
+ getTowers() : List
+ getSize() : int
+ getID() : int
+ getStudents() : List
character_utils
<<enum>>
AssistantType
- name : String
- cardType : int
- possibleSteps : int
+ <<enum constant>> CHEETAH : int
+ <<enum constant>> OWL : int
+ <<enum constant>> CAT : int
+ <<enum constant>> EAGLE : int
+ <<enum constant>> FOX : int
+ <<enum constant>> SNAKE : int
+ <<enum constant>> OCTOPUS : int
+ <<enum constant>> DOG : int
+ <<enum constant>> ELEPHANT : int
+ <<enum constant>> TURTLE : int
+ AssistantType(cardName : String, cardTurnValue : int, cardPossibleSteps : int)
+ getName() : String
+ getCardTurnValue() : int
+ getPossibleSteps() : int
- cardType
1
<<enum>>
DeckType
- name : String
+ <<enum constant>> KING : int
+ <<enum constant>> ELDER : int
+ <<enum constant>> PIXIE : int
+ <<enum constant>> SORCERER : int
+ DeckType(name : String)
+ getName() : String
- type
<<enum>>
AdvancedCharacterType
- name : String
- cost : int
+ <<enum constant>> MONK : int
+ <<enum constant>> BARTENDER : int
+ <<enum constant>> CENTAURUS : int
+ <<enum constant>> POSTMAN : int
+ <<enum constant>> HEALER : int
+ <<enum constant>> MERCHANT : int
+ <<enum constant>> JESTER : int
+ <<enum constant>> BARD : int
+ <<enum constant>> KNIGHT : int
+ <<enum constant>> PRINCESS : int
+ <<enum constant>> LANDLORD : int
+ <<enum constant>> MERCENARY : int
+ <<enum constant>> NULL : int
+ AdvancedCharacterType(name : String, cardCost : int)
+ getCardName() : String
+ getCardCost() : int
+ getRandomCard() : AdvancedCharacter
- type
utils
<<enum>>
ColoreCharacter
- <<enum constant>> GREEN : int
- <<enum constant>> BLUE : int
- <<enum constant>> RED : int
- <<enum constant>> PINK : int
- <<enum constant>> YELLOW : int
- color
<<enum>>
ColoreTower
- <<enum constant>> BLACK : int
- <<enum constant>> WHITE : int
- <<enum constant>> GREY : int
- color
GamePhase
- <<enum constant>> GAME_PENDING : int
- <<enum constant>> PLANNING_PHASE : int
- <<enum constant>> MOVING_STUDENT_PHASE_STUDENTS : int
- <<enum constant>> ACTION_PHASE_MOVING_MOTHER_NATURE : int
- <<enum constant>> ACTION_PHASE_CHOOSING_CLOUD : int
- <<enum constant>> GAME_ENDED : int
- gamePhase
CalculateInfluence
pUpdateStudentInRoomsIsl : Player, island : Island : int
pUpdateTowerInIsl : Player, island : Island : int
evaluateIsland : List, island : Island : Optional
+ evaluateIslandWithCharacterEffect(island : Island) : void
- getLastOfPlayerWithHighestSetOfColor(players : List) : List
CalculateTeacherOwnership
- players : List
+ evaluate(player : Player, color : ColoreCharacter, players : List) : void
+ getMaxOfPlayerWithHighestNumOfStudentsOfColor(currentPlayer : Player, oldOwner : Player, teacher : Teacher) : void
+ numOfOwnershipConditionsAllOwner : Player, teacher : Teacher) : void
+ getNumOfStudentsOfColor(p : Player, c : ColoreCharacter) : int
# getTeacherOfColorFromPlayerOwnership(color : ColoreCharacter) : Teacher
# getPlayerWithTeacherOfColor(color : ColoreCharacter) : Player
game_view
<<enum>>
InputStateMachine
+ <<enum constant>> PLANNING_PHASE_START : int
+ <<enum constant>> SELECT_ASSISTANT_CARD_SEND_MESSAGE : int
+ <<enum constant>> MOVING_STUDENT_PHASE_START : int
+ <<enum constant>> MOVE_STUDENT_SEND_MESSAGE : int
+ <<enum constant>> MOVE_STUDENT_SEND_MESSAGE : int
+ <<enum constant>> MOVE_MOTHER_NATURE_START : int
+ <<enum constant>> MOVE_MOTHER_NATURE_SEND_MESSAGE : int
+ <<enum constant>> CHOSE_CLOUD_CARD_START : int
+ <<enum constant>> CHOSE_CLOUD_CARD_SEND_MESSAGE : int
+ <<enum constant>> PLAY_ADVANCED_CARD : int
GameVueClient
+ ASK_NAME_QUESTION : String
+ REASK_NAME_QUESTION : String
+ ASK_DECK_TYPE_QUESTION : String
+ ASK_GAME_MODE_QUESTION : String
+ ASK_JOINING_ACTION_QUESTION : String
+ ASK_LOBBY_TO_JOIN_QUESTION : String
+ ASK_NUMBER_OF_PLAYERS_QUESTION : String
- actionSent : boolean
+ GameViewClient(client : Client)
+ updateBeanMessage(board : GameBoard) : void
+ displayNotYourTurn() : void
+ updateGameBoard(board : GameBoard) : void
+ displayYourTurn() : void
+ displayExpertMode() : void
+ setActionSent() : void
+ reaskName() : void
+ askDeckJokeAvailable : Object) : void
+ askGameMode() : void
+ askJoiningAction() : void
+ askLobbyToJoinOfLobbyInfos : Object) : void
+ askNumberOfPlayers() : void
+ getNetwork() : int
+ reaskAssistant() : void
- view
1..1
GameViewCLI
- input : Scanner
- rangeA : int
- rangeB : int
+ GameViewCLI(client : Client)
+ getNetwork(port : int) : void
+ askIncludeMoreVerbose(text : int) : void
+ whileInputNotRangeOk(a : int, b : int) : int
+ askNumberOfPlayersAndGameMode() : String
+ deckToStringIsIcts() : List : String
+ getBoard() : GameBoard
+ updateBoard(board : GameBoard) : void
GameViewGUI
extends Observable<GameAction>
implements Observer<MoveMessage>
RemoteGameView
- clientName : String
+ RemoteGameView(clientName : String, c : ClientConnection)
+ handleBeanAction(bm : GameAction) : void
+ sendMessage(message : CommunicationMessage) : void
+ updateMessage(m : MoveMessage) : void
board
AdvancedCardBoard
- name : String
- cost : int
- deryCard : int
+ AdvancedCardBoard(card : AdvancedCharacter)
+ toString() : String
- advancedCard
3..3
DeckBoard
- cards : List
- discardedCard : AssistantType
+ DeckBoard(player : Player)
+ getCards() : List
+ getDiscardedCard() : AssistantType
+ toString() : String
- decks
2..4
IslandBoard
- students : List
- islandId : int
- islandIndex : int
- numberOfIsland : int
- towerColor : ColoreTower
- deniedCounter : int
+ IslandBoard(island : Island, hasMotherNature : boolean, islandIndex : int)
+ toString() : String
- islands
1..12
GameBoard
- importMode : boolean
- players : List
- names : List
- towers : int
- numberOfIslands : List
- phase : GamePhase
- currentPlaying : String
+ GameBoard(numberOfPlayers : int)
+ setGameToExpertMode() : void
+ setTreasuryTreasury : int) : void
+ getDecks() : List
+ getPlayers() : List
+ getTerrain() : TerrainBoard
+ getCurrentPlaying() : String
+ getGamePhase() : GamePhase
+ getNumberOfPlayers() : int
+ getSchools() : List
+ getColorOfPlayer(color : ColoreCharacter) : String
+ getNumberOfTowersOfColor(color : ColoreTower) : String
+ toString() : String
- schools
2..4
- board
1
SchoolBoard
- entrance : List
- teachers : List
- tables : int[]
- towers : int
- towerColor : ColoreTower
+ SchoolBoard(school : School, colorTower : ColoreTower)
+ getEntrance() : List
+ addStudentToEntrance(student : int) : String
+ removeStudentFromEntrance() : void
+ setTableOrder(n : int) : String
+ toString() : String
- terrain
1..1
TerrainBoard
- cloudCards : List
+ TerrainBoard(terrain : Terrain, motherNaturePosition : int)
+ getIslandsNumber() : int
+ getAdvancedCards() : List
+ getIslands() : List
+ toString() : String
GameBoardAdvanced
+ GameBoardAdvanced(game : Game)
game_controller
actions
GameAction
+ gameActionID : int
+ playerName : 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
- islandIndex : 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
- cardIndex : int
+ PlayAdvancedCardAction() : void
+ performUpdate(controller : GameController) : void
GameController
- turn : int
- gameController : int[]
- actionOrder : int[]
+ createPlayerGame(name : String, type : DeckType) : void
+ getPlayer(player : Player) : void
+ start() : void
+ getTurn() : int
- nextPlanningPhase() : void
+ selectAssistantCard(player : Player, assistant : Assistant) : void
+ getNumberOfPlayersPlayedAssistant(player : Player, assistant : Assistant) : void
- createActionPhaseOrder() : void
- nextPlayerNumberOfMovedStudent() : void
- nextMoveStudentPhase() : void
- moveStudentPhase() : void
+ playerMoveStudentToIslandy(player : Player, student : int, islandID : int) : void
+ playerMoveStudentToDiningHall(player : Player, student : int) : void
+ moveMotherNatureOfStepsy(player : Player, numberOfSteps : int) : void
- endGame() : void
+ chooseCloudCard(player : Player, cloudCard : CloudCard) : void
+ playAdvancedCard(player : Player, card : AdvancedCharacter, args : Object) : void
- refillCloudS() : void
- createNextPlanningOrder() : void
Observer<GameAction>
MessageType
- ASK_NAME : int
- REASK_NAME : int
- ASK_GAME : int
- ASK_PLAYER_NUMBER : int
- ASK_GAME_TYPE : int
- ASK_JOINING_ACTION : int
- ASK_LOBBY_TO_JOIN : int
- ASSISTANT_NOT_PLAYABLE : int
- ASK_ACTION : int
- OKAY : int
- ERROR : int
- YOU_LOSE : int
- YOU_WIN : int
- VIEW_UPDATE : int
- GAME_READY : int
CommunicationMessage
- id : MessageType
- message : Object
+ CommunicationMessage(id : MessageType, message : Object)
+ getID() : MessageType
+ getMessage() : Object
0..*
- incomingMessages
network
server
Lobby
- experiMode : boolean
- numberOfPlayers : int
+ Lobby(server : Server, gameSize : int, isExperGame : boolean)
+ getNumber() : int
+ getConnectedPlayersToLobby() : List
+ getCurrentLobbySize() : int
+ getID() : int
+ isExperMode() : boolean
+ getLobbyName() : String
+ setLobbyName() : void
+ registerClientToLobby(clientNet : int) : void
+ run() : void
+ getLobbyInfo() : String
+ closeLobby(connection : SocketClientConnection) : void
+ broadcastMessage(messageType : MessageType, content : Object, excludeOtherMsg : List) : void
- activeGames
0..*
1
- lobbyOwner
- connectedPlayersToLobby
0..4
- waitingConnections
0..*
<<interface>>
ClientConnection
+ closeConnection() : void
+ asyncSendMessage(messageToSender : Observer) : void
+ asyncSendMessage(communicationMessage) : void
+ isAlive() : boolean
Server
- PORT : int
- serverSocket : ServerSocket
- executor : ExecutorService
- lobbies : List<Lobby>
- waitingConnection : List<ClientConnection>
+ Server()
+ handleClientConnection(connection : ClientConnection) : void
+ handleLobbyStatesToNetwork(lobby : Lobby, connectionToNotify : ClientConnection) : void
+ run() : void
+ getNetworkGetPlayersName() : Set
+ getNextGameID() : int
+ removeWaitingConnection(connection : ClientConnection) : void
+ addGameLobbyToNet(lobby : Lobby) : void
- server
1..*
- server
SocketClientConnection
- socket : Socket
- out : ObjectOutputStream
- in : ObjectInputStream
- clientName : String
+ SocketClientConnection(socket : Socket, server : Server)
+ getClientName() : String
+ asyncMessage : CommunicationMessage) : void
+ closeConnection() : void
+ isAlive() : boolean
+ asyncSendMessage : CommunicationMessage) : void
+ run() : void
+ getLoginMsg() : int
- askGameNumberOfPlayers() : int
+ setGameType() : boolean
+ getGameNumber() : int
- getPlayerNumber() : Optional
- setName() : void
- getNameFromClient() : String
- createNewGame() : void
- joinExistingGame() : void
- getClientNetwork() : int
- lobbyLeaveMsg() : void
- connection
1..1
Observable<CommunicationMessage>
Observer<CommunicationMessage>
client
Client
- ip : String
- port : int
- active : boolean
- socketOut : ObjectOutputStream
- name : String
+ Client(ip : String, port : int)
+ getIsActive() : boolean
+ setIsInactive : boolean) : void
+ asyncNewProcSocket(socketIn : ObjectInputStream) : Thread
+ asyncSendToSocket(message : CommunicationMessage) : void
+ run() : void
+ getName() : String
+ setName(name : String) : void
- client
1..1
- clientHandled
ClientMessageObserverHandler
- gameViewClient : GameViewClient
- client : Client
- state : InputStateMachine
+ ClientMessageObserverHandler(view : GameViewClient)
+ updateBeanMessage(board : GameBoard) : void
+ update(message : CommunicationMessage) : void
+ setGameToExpertMode() : void
+ setStateIsIcts : InputStateMachine) : void
+ isAlreadyGetIcts() : boolean
+ setAskJoinGameSentTrue() : boolean) : void
Observer<CommunicationMessage>
extends TimeTask
utils
<<enum>>
ConnectionStatusHandler
PING_TIMEOUT_DELAY : int
connectionActive : boolean
+ ping() : void
+ ConnectionStatusHandler() : void
+ run() : void
+ isConnectionAlive() : boolean
+ kill() : void
+ updateConnection() : void
- connectionStatusHandler
1..1
PingTimeoutExceededTask
+ PingTimeoutExceededTask(connectionStatusHandler : ConnectionStatusHandler) : char
+ run() : void
Logger
+ ERROR(msg : String, errorType : String) : void
+ INFO(msg : String) : void
+ WARNING(msg : String) : void
+ GAME_LOG(msg : String, connectionName : String) : void
- statusHandler
1..1
ServerConnectionStatusHandler
+ ServerConnectionStatusHandler()
+ setConnection(connectionToHandle : SocketClientConnection) : void
+ update() : void
ClientConnectionStatusHandler
+ ClientConnectionStatusHandler()
+ setClientUpdate() : Client) : void
+ update() : void
LobbyInfo
- lobbyName : String
- numberOfPlayers : int
- lobbyMaxSize : int
- isLobbyFull : boolean
+ LobbyInfo(lobby : Lobby)
+ getLobbyName() : String
+ getNumberOfPlayers() : int
+ isFull() : boolean
+ getLobbyMaxSize() : int