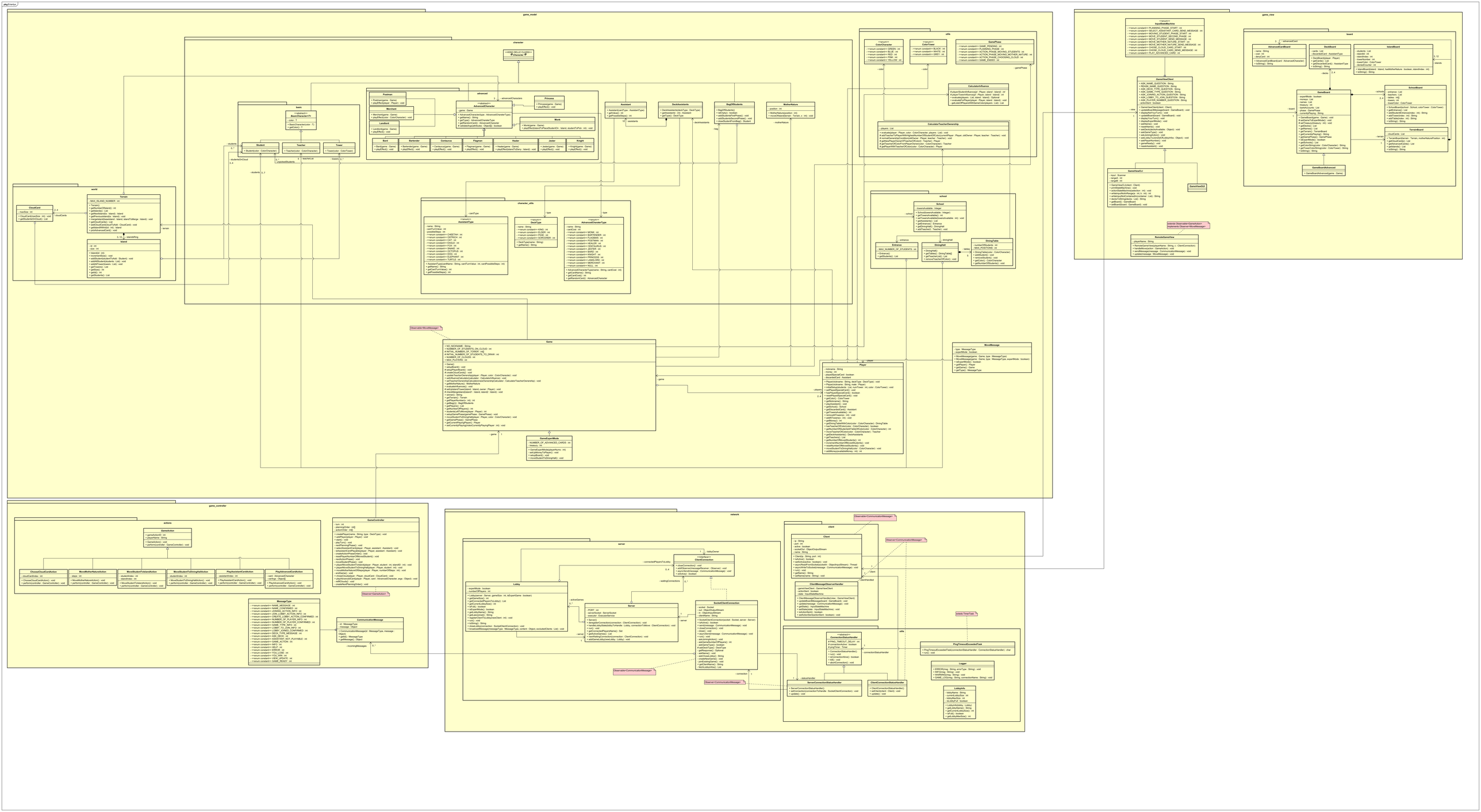
pkg Eriantys
game_model
character
<<UNDO DELLE CLASSI>>
AbstractCharacter
advanced
<<abstract>>
AdvancedCharacter
- game : Game
+ AdvancedCharacter(type : AdvancedCharacterType)
+ getType() : AdvancedCharacterType
+ getRandomCard() : AdvancedCharacter
# validateAnyAttribute : Object[] : boolean
- advancedCharacters
Postman
+ Postman(game : Game)
+ playEffect(player : Player) : void
Merchant
+ Merchant(game : Game)
+ playEffect(player : ColorCharacter) : void
Landlord
+ Landlord(game : Game)
+ playEffect() : void
Princess
+ Princess(game : Game)
+ playEffect() : void
Work
+ Work(game : Game)
+ playEffect(player : Player, island : int) : void
Bard
+ Bard(game : Game)
+ playEffect() : void
Barbender
+ Barbender(game : Game)
+ playEffect() : void
Centaurus
+ Centaurus(game : Game)
+ playEffect() : void
Flagman
+ Flagman(game : Game)
+ playEffect() : void
Healer
+ Healer(game : Game)
+ playEffectSideeffect(every : Island) : void
Jester
+ Jester(game : Game)
+ playEffect() : void
Knight
+ Knight(game : Game)
+ playEffect() : void
beas
<<abstract>>
AbstractCharacterTv
- color : T
+ BasicCharacter(color : T)
+ getColor() : T
Student
+ StudentEnum : ColorCharacter
Teacher
+ Teacher(color : ColorCharacter)
Tower
+ Tower(color : ColorTower)
- students
- studentsOnCloud
- studentsDiningRoom
- teacherLeft
- towers
Assistant
+ Assistant(cardType : AssistantType)
+ getValue() : int
+ getPossibleSteps() : int
- assistants
DeckAssistants
+ DeckAssistants(deckType : DeckType)
+ getAssistant(int : int) : Assistant
+ deck : DeckType
- deckAssistants
BagOfStudents
+ BagOfStudents()
+ isEmpty() : boolean
+ addStudent(student : Student) : void
+ addStudentsAroundPhase() : void
+ drawStudentFromBag() : Student
- bag
MotherNature
- position : int
+ MotherNature(position : int)
+ moveOfNumber(number : int, remain, x : int) : void
- motherNature
utils
<<enum>>
ColorCharacter
+ <<enum constant>> GREEN : int
+ <<enum constant>> BLUE : int
+ <<enum constant>> RED : int
+ <<enum constant>> PINK : int
+ <<enum constant>> YELLOW : int
- color
<<enum>>
ColorTower
+ <<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>> MOVE_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
getperStudentInRoomcol(p : Player, island : Island) : int
getperTowerInIsland(p : Player, island : Island) : int
+ evaluate(player : List, island : Island) : Optional
+ getNumberOfProfessorsOwnedBy(player : Player) : int
- getJustOfPlayersWithSameColor(player, : List) : List
CalculateTeacherOwnership
- players : List
+ evaluate(player : Player, color : ColorCharacter, players : List) : void
+ getNumberOfPlayersWithEqualNumberOfStudentsOfColorLoserPlayer : Player, oldOwner : Player, teacher : Teacher) : void
+ numOfOwnershipConditionsAllOtherPlayers(color : ColorCharacter, teacher : Teacher) : int
# getTeacherOfColorFromPlayerOwnerColor(color : ColorCharacter) : Teacher
# getPlayerWithTeacherOfColor(color : ColorCharacter) : Player
character_utils
<<enum>>
AssistantType
- name : String
- value : 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(namePlane : String, cardTurnValue : int, cardPossibleSteps : int)
+ getName() : String
+ getCardTurnValue() : int
+ getPossibleSteps() : int
- cardType
<<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
- cardCost : 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
world
CloudCard
- mcolour : int
+ CloudCard(mcolor : int) : void
+ getStudentsOnCloud() : List
- cloudCards
Terrain
- MAX_ISLAND_NUMBER : int
+ Terrain()
+ getNumberOfIsland() : int
+ getIslands() : List
+ getIsland(island : Island) : Island
+ getPreviousIsland(is : Island) : int
+ mergeIsland(islandsIsland : Island, islandToMerge : Island) : void
+ getIsland(id : int) : Island
+ addCloudCardsStudentToAll : CloudCard) : void
+ getIslandWithId(id : int) : Island
+ pickAdvancedCard() : void
- terrain
- islandsRing
Island
- id : int
- size : int
+ Island(id : int)
+ increaseSize() : void
+ addStudentsStudentToAll : Student) : void
+ addStudentsOfColor(color : Color) : void
+ addMoreTowers(towers : List) : void
+ getTowers() : List
+ getSize() : int
+ getID() : int
+ getStudents() : List
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
- destination : 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
- controller : int[]
- actionOrder : int[]
+ createPlayer(name : String, type : DeckType) : void
+ getPlayer(player : Player) : void
+ start() : void
+ setup() : void
- newPlanningPhase() : void
+ selectAssistantCard(player : Player, assistant : Assistant) : void
+ getAvailableAssistantCards(player : Player, assistant : Assistant) : void
+ checkActionPhaseOrder() : void
- nextPlayerNumberOfMovedStudent() : void
+ moveStudent() : 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
+ playerMoveCloudCard(player : Player, cloudCard : CloudCard) : void
+ playAdvancedCardy(player : Player, card : AdvancedCharacter, args : Object) : void
- refillCloudS() : void
- checkNextPlanningOrder() : void
Observer<GameAction>
MessageType
+ <<enum constant>> NAME_MESSAGE : int
+ <<enum constant>> NAME_ACCEPTED : int
+ <<enum constant>> JOINING_ACTION_INFO : int
+ <<enum constant>> JOIN_LOBBY_ACTION_INFO : int
+ <<enum constant>> NUMBER_OF_PLAYERS_CONFIRMED : int
+ <<enum constant>> NUMBER_OF_PLAYERS_MESSAGE : int
+ <<enum constant>> NUMBER_OF_PLAYER_CONFIRMED : int
+ <<enum constant>> GAME_TYPE_INFO : int
+ <<enum constant>> LOBBY_TO_JOIN_INFO : int
+ <<enum constant>> NUMBER_OF_PLAYER_CONFIRMED : int
+ <<enum constant>> DECK_TYPE_MESSAGE : int
+ <<enum constant>> ASK_DECK : int
+ <<enum constant>> NOTIFY_NOT_PLAYABLE : int
+ <<enum constant>> GAME_ACTION : int
+ <<enum constant>> INFO : int
+ <<enum constant>> WARN : int
+ <<enum constant>> ERROR : int
+ <<enum constant>> YOU_LOSE : int
+ <<enum constant>> YOU_WIN : int
+ <<enum constant>> VIEW_UPDATE : int
+ <<enum constant>> GAME_READY : int
CommunicationMessage
- id : MessageType
- message : Object
+ CommunicationMessage(id : MessageType, message : Object)
+ getID() : MessageType
+ getMessage() : Object
- incomingMessages
network
server
Lobby
- experiMode : boolean
- numberOfPlayers : int
+ Lobby(server : Server, gameSize : int, isExpertGame : boolean)
+ getID() : int
+ getCorrectNumberPlayersToLobby() : List
+ getCurrentLobbySize() : int
+ isFull() : boolean
+ isExpertMode() : boolean
+ getLobbyName() : String
+ getNumberOfPlayers() : int
+ registerClientToLobby(socketClient : int) : void
+ run() : void
+ stopLobby() : void
+ closeLobby(connection : SocketClientConnection) : void
+ broadcastMessage(messageType : MessageType, content : Object, excludedClients : List) : void
- activeGames
Server
- PORT : int
- serverSocket : ServerSocket
- executor : ExecutorService
+ Server()
+ handleClientConnection(connection : ClientConnection) : void
+ handleLobbyStatelisteningToHandle : Lobby, connectionToMove : ClientConnection) : void
+ run() : void
+ getLobbyByGameNameyName() : Set
+ getAllGames() : List
+ newWaitingConnection(connection : ClientConnection) : void
+ addGameLobbyNewLobby : Lobby) : void
- lobbyOwner
- connectedPlayersToLobby
- waitingConnections
- server
- socket
<<interface>>
ClientConnection
+ closeConnection() : void
+ asyncSend(message : Object, Observer) : void
+ asyncSend(message : CommunicationMessage) : void
+ isAlive() : boolean
SocketClientConnection
- socket : Socket
- out : ObjectOutputStream
- in : ObjectInputStream
- clientName : String
+ SocketClientConnection(socket : Socket, server : Server)
+ getClientName() : String
+ asyncMessage : CommunicationMessage) : void
+ closeConnection() : void
+ isAlive() : boolean
+ asyncSend(message : Communication Message) : void
+ run() : void
+ setup(MsgToSRV) : void
- askGameNumberOfPlayers() : int
- askGameType() : boolean
- askGameNumberOfPlayers() : int
- getRequiredNumber() : Optional
- askName() : void
- askGameLobbyNameyNew() : String
- chooseNewGame() : void
- joinExistingGame() : void
- getClientNameyNew() : String
- listOfLobbiesNew() : List
- connection
Observer<CommunicationMessage>
Observable<CommunicationMessage>
client
Client
- ip : String
- port : int
- active : boolean
- socketOut : ObjectOutputStream
- name : String
+ Client(ip : String, port : int)
+ getIP() : String
+ setAlreadyInactive : boolean) : void
+ asyncReadFromSocket(socketIn : ObjectInputStream) : Thread
+ asyncSend(socketOut : ObjectOutputStream, message : Communication Message) : void
+ run() : void
+ getName() : String
+ setName(name : String) : void
- client
- clientHandled
ClientMessageObserverHandler
- gameViewClient : GameViewClient
- ip : String
- state : InputStateMachine
+ ClientMessageObserverHandler(view : GameViewClient)
+ updateBoardMessage(board : GameBoard) : void
+ update(message : CommunicationMessage) : void
+ getIP() : String
+ setState(state : InputStateMachine) : void
+ isAlreadyIn() : boolean
+ setAlreadyInactive(isAlreadyIn : boolean) : void
utils
<<abstract>>
ConnectionStatusHandler
PING_TIMEOUT_DELAY : int
connectionStatus : boolean
+ ConnectionStatusHandler()
+ ConnectionStatusHandler(int) : void
+ run() : void
+ isConnectedAlive() : boolean
+ ME() : void
+ setLastConnection() : void
- connectionStatusHandler
ServerConnectionStatusHandler
+ ServerConnectionStatusHandler()
+ setConnection(connectionToHandle : SocketClientConnection) : void
+ update() : void
- statusHandler
ClientConnectionStatusHandler
+ ClientConnectionStatusHandler()
+ setClientSocket(client : Client) : void
+ update() : void
Observable<TimerTask>
PingTimeoutExceedeTask
+ PingTimeoutExceedeTask(connectionStatusHandler : ConnectionStatusHandler) : char
+ run() : void
Logger
+ ERROR(msg : String, errorType : String) : void
+ INFO(msg : String) : void
+ WARN(msg : String) : void
+ GAME_LOG(msg : String, connectionName : String) : void
LobbyInfo
- lobbyName : String
- numberOfPlayers : int
- lobbyMaxSize : int
- isLobbyFull : boolean
+ LobbyInfo(lobby : Lobby)
+ getLobbyName() : String
+ getNumberOfPlayers() : int
+ isFull() : boolean
+ getLobbyMaxSize() : int
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
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_NUMBER_OF_PLAYERS_QUESTION : String
+ actionStart : boolean
+ GameViewClient(client : Client)
+ updateBoardMessage(board : GameBoard) : void
+ displayNotYourTurn() : void
+ updateBoard(board : GameBoard) : void
+ displayYourTurn() : void
+ displayExpertMode() : void
+ displayInfo() : void
+ readName() : void
+ askDeckJackAvailable : Object) : void
+ askGameType() : void
+ askJoiningAction() : void
+ askLobbyToJoinOfLobbyInfo : Object) : void
+ askNumberOfPlayers() : void
+ getNetwork() : void
+ readAssistant() : void
- view
- board
GameBoard
- mcurrentMode : boolean
- clouds : List
- names : List
- towers : List
- teacherLeft : int
- islands : List
- phase : GamePhase
- currentMoPlaying : String
+ GameBoard(numberOfPlayers) : void
+ setGameToExpertMode() : void
+ setTreasuryUnlocked(v : int) : void
+ getDecks() : List
+ getClouds() : List
+ getTerrain() : TerrainBoard
+ getCurrentMoPlaying() : String
+ getGamePhase() : GamePhase
+ getNumberOfPlayers() : int
+ getSchools() : List
+ getColorOfingusIsle : ColorCharacter) : String
+ getNumberOfProfessorsRegistrator : ColorTower) : String
+ toString() : String
- schools
- decks
- islands
- terrain
SchoolBoard
- entrance : List
- teachers : List
- tables : int[]
- towers : int
- towerColor : ColorTower
+ SchoolBoard(school : School, colorTower : ColorTower)
+ getEntrance() : List
+ addStudentAtIndexandIndex(index, int) : String
+ getTeachers() : List
+ addTableOrder(v : int) : String
+ toString() : String
TerrainBoard
- cloudCards : List
+ TerrainBoard(terrain : Terrain, motherNaturePosition : int)
+ getCloudCards() : List
+ getAdvancedCards() : List
+ getIslands() : List
+ toString() : String
GameBoardAdvanced
+ GameBoardAdvanced(game : Game)
DeckBoard
- cards : List
- discardedCard : AssistantType
+ DeckBoard(player : Player)
+ getCards() : List
+ getDiscardedCard() : AssistantType
+ toString() : String
AdvancedCardBoard
- name : String
- cost : int
- deryCard : int
+ AdvancedCardBoard(card : AdvancedCharacter)
+ toString() : String
- advancedCard
IslandBoard
- students : List
- islandId : int
- islandIndex : int
- numberOfIsland : int
- towerColor : ColorTower
- demandCounter : int
+ IslandBoard(island : Island, hasMotherNature : boolean, islandIndex : int)
+ toString() : String
- islands
GameViewCLI
- input : Scanner
- rangeA : int
- rangeB : int
+ GameViewCLI(client : Client)
+ getNetwork() : Network
+ askIncludeMoreExpansion() : void
+ whatkindofMotherRange(a : int, b : int) : int
+ askNumberOfPlayersAndGameMode() : String
+ deckCLIStringIsJoin() : List : String
+ getBoard() : GameBoard
+ setBoard(board : GameBoard) : void
GameViewGUI
extends Observable<GameAction>
implements Observer<MoveMessage>
RemoteGameView
- clientName : String
+ RemoteGameView(clientName : String, c : ClientConnection)
+ handleBroadcast(msg : GameAction) : void
+ sendMessage(message : CommunicationMessage) : void
+ update(message : MoveMessage) : void
Observer<MoveMessage>
Game
+ NO_NICKNAME : String
+ NUMBER_OF_STUDENTS_ON_CLOUD : int
# INITIAL_NUMBER_OF_TOWER : int[]
# INITIAL_NUMBER_OF_STUDENTS_TO_DRAWN : int
# MAX_NUMBER_OF_DECKS : int
# MAX_PLAYERS : int
+ Game()
+ Game(int) : void
# setupPlayersBoard() : void
# createCloudCards() : void
# updateTeacherOwnership(player : Player, color : ColorCharacter) : void
# setInfluenceCalculatorCalculator : CalculateInfluence) : void
# setTeacherOwnershipCalculator(teacherOwnershipCalculator : CalculateTeacherOwnership) : void
# getMotherNature() : MotherNature
# evaluateInfluence() : void
# getNumberOfTowersOnIsland(island, owner : Player) : void
# checkMergeIslandsIsland : Island, island2 : Island) : void
+ winner() : String
+ getPlayer() : Player
+ getPlayerAssistant(v : int) : int
+ getBag() : BagOfStudents
+ getIsland() : Island
+ getNumberOfPlayers() : int
+ studentsLeftToMoveplayer : Player) : int
+ moveStudentToIslandy(player : Player, student : int, islandID : int) : void
+ moveStudentToDiningHall(player : Player, color : ColorCharacter) : void
+ getGamePhase() : GamePhase
+ getCurrentPlayingPlayer() : Player
+ setCurrentPlayingPlayer(currentPlayingPlayer : int) : void
- game
- player
- cp
- owner
GameExpertMode
- NUMBER_OF_ADVANCED_CARDS : int
+ GameExpertMode() : void
+ GameExpertMode(gameName : int)
+ setLastMoveToPlayer() : void
+ setup(Boolean) : void
+ moveStudentToDiningHall() : void
Player
- nickname : String
- money : int
- usedAdvancedCard : boolean
- discardedCard : Assistant
+ Player(nickname : String, deckType : DeckType) : void
+ Player(nickname : String, mode : Player)
+ initialSetup(students : List, numTower : int, color : ColorTower) : void
+ getNickname() : String
+ hasPlayedSpecialCard() : boolean
+ resetPlayedSpecialCard() : void
+ getMoney() : int
+ getNickname() : String
+ playAssistant() : void
+ getDeck() : DeckTypes
+ getDiscardedCard() : Assistant
+ getTowersAvailable() : int
+ setTowersAvailable(v : int) : void
+ addMTowers() : void
+ getBlinkyC() : int
+ getNumberOfProfessorsOwnedcolor : ColorCharacter) : String
+ getTeacherOfColorcolor : ColorCharacter) : boolean
+ getNumberOfStudentsAtTableOfColorcolor : ColorCharacter) : int
+ hiveTeacherOfColorcolor : ColorCharacter) : Teacher
+ getDeckAssistants() : DeckAssistants
+ getTower() : int
+ getNumberOfMovedStudents() : int
+ incrementNumberOfMovedStudents() : void
+ resetNumberOfMovedStudents() : void
+ moveStudentToDiningHallcolor : ColorCharacter) : void
+ addMoreLocalsIntoMoney : int) : int
MoveMessage
- type : MessageType
- expertMode : boolean
+ MoveMessage(game : Game, type : MessageType)
+ MoveMessage(game : Game, type : MessageType, expertMode : boolean)
+ toSpecialMessage() : Object
+ getPlayer() : Player
+ getGame() : Game
+ getType() : MessageType