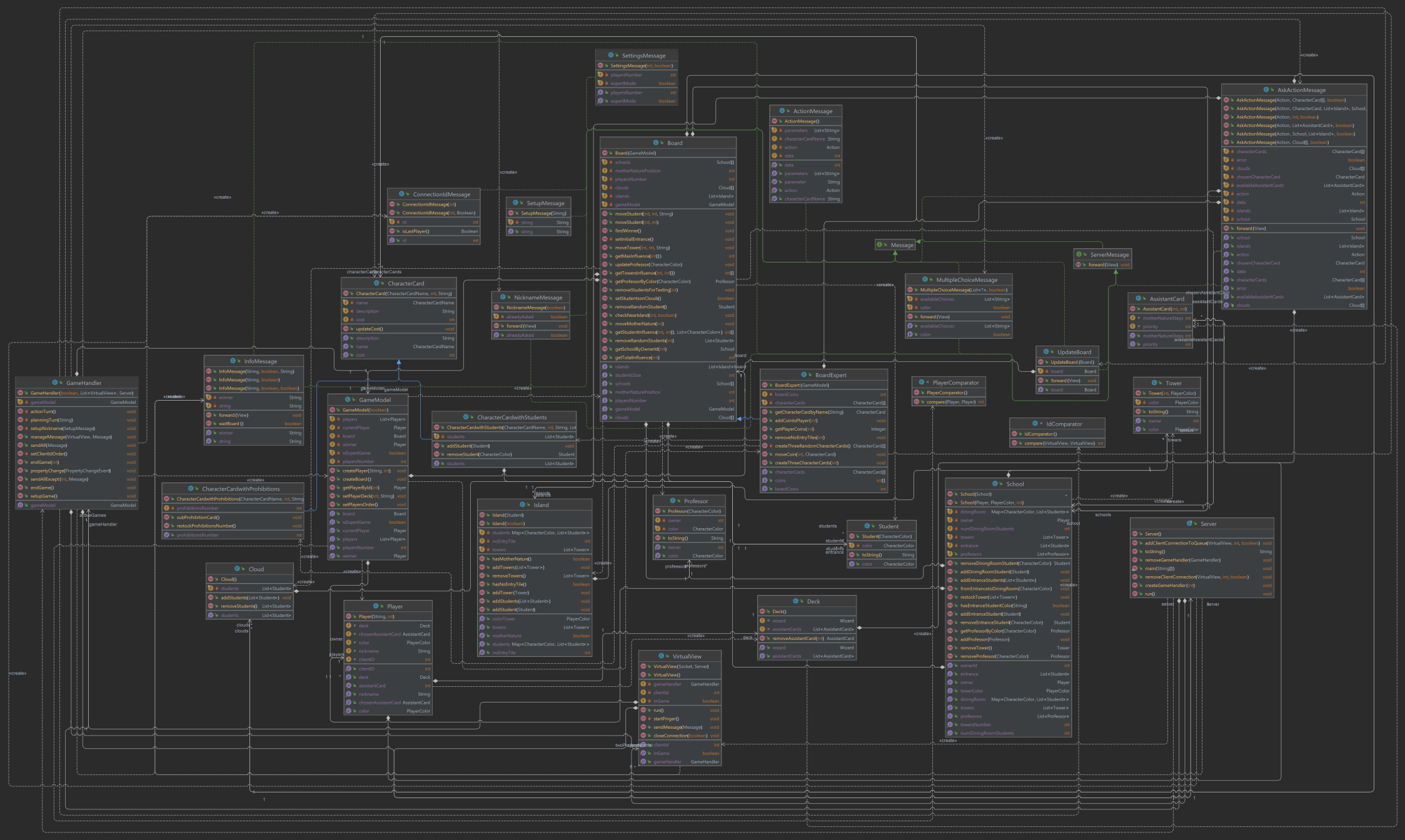
SettingsMessage
SettingsMessage(int, boolean)
playersNumber int
expertMode boolean
playersNumber int
expertMode boolean
ActionMessage
ActionMessage()
parameters List<String>
characterCardName String
action Action
data int
data int
parameters List<String>
parameter String
action Action
characterCardName String
AskActionMessage
AskActionMessage(Action, CharacterCard[], boolean)
AskActionMessage(Action, CharacterCard[], List<Island>, School,
AskActionMessage(Action, int, boolean)
AskActionMessage(Action, List<AssistantCard>, boolean)
AskActionMessage(Action, School, List<Island>, boolean)
AskActionMessage(Action, Cloud[], boolean)
characterCards CharacterCard[]
error boolean
clouds Cloud[]
chosenCharacterCard CharacterCard
availableAssistantCards List<AssistantCard>
action Action
data int
islands List<Island>
school School
forward(View) void
school School
islands List<Island>
action Action
chosenCharacterCard CharacterCard
data int
characterCards CharacterCard[]
error boolean
availableAssistantCards List<AssistantCard>
clouds Cloud[]
ConnectionIdMessage
ConnectionIdMessage(int)
ConnectionIdMessage(int, Boolean)
id int
isLastPlayer() Boolean
id int
SetupMessage
SetupMessage(String)
string String
string String
Board
Board(GameModel)
schools School[]
motherNaturePosition int
playersNumber int
clouds Cloud[]
islands List<Island>
gameModel GameModel
moveStudent(int, int, String) void
moveStudent(int, int) void
findWinner() void
setInitialEntrance() void
moveTower(int, int, String) void
getMaxInfluence(int[]) int
updateProfessor(CharacterColor) void
getTowersInfluence(int, int[]) int[]
getProfessorByColor(CharacterColor) Professor
removeStudentsForTesting(int) void
setStudentsonCloud() boolean
removeRandomStudent() Student
checkNearIsland(int, boolean) void
moveMotherNature(int) void
getStudentInfluence(int, int[], List<CharacterColor>) int[]
removeRandomStudent(int) List<Student>
getSchoolByOwnerId(int) School
getTotalInfluence(int) int
islands List<Island>
studentsSize int
schools School[]
motherNaturePosition int
playersNumber int
gameModel GameModel
clouds Cloud[]
Message
ServerMessage
forward(View) void
MultipleChoiceMessage
MultipleChoiceMessage(List<?>, boolean)
availableChoices List<String>
color boolean
forward(View) void
availableChoices List<String>
color boolean
AssistantCard
AssistantCard(int, int)
motherNatureSteps int
priority int
motherNatureSteps int
priority int
CharacterCard
CharacterCard(CharacterCardName, int, String)
name CharacterCardName
description String
cost int
updateCost() void
description String
name CharacterCardName
cost int
NicknameMessage
NicknameMessage(boolean)
alreadyAsked boolean
forward(View) void
alreadyAsked boolean
UpdateBoard
UpdateBoard(Board)
board Board
forward(View) void
board Board
Tower
Tower(int, PlayerColor)
color PlayerColor
toString() String
owner int
color PlayerColor
InfoMessage
InfoMessage(String, boolean, String)
InfoMessage(String, boolean)
InfoMessage(String, boolean, boolean)
winner String
string String
forward(View) void
waitBoard() boolean
winner String
string String
GameHandler
GameHandler(boolean, List<VirtualView>, Server)
gameModel GameModel
actionTurn() void
planningTurn(String) void
setupNickname(SetupMessage) void
manageMessage(VirtualView, Message) void
sendAll(Message) void
setClientIdOrder() void
endGame(int) void
propertyChange(PropertyChangeEvent) void
sendAllExcept(int, Message) void
endGame() void
setupGame() void
gameModel GameModel
GameModel
GameModel(boolean)
players List<Player>
currentPlayer Player
board Board
winner Player
isExpertGame boolean
playersNumber int
createPlayer(String, int) void
createBoard() void
getPlayerById(int) Player
setPlayerDeck(int, String) void
setPlayersOrder() void
board Board
isExpertGame boolean
currentPlayer Player
players List<Player>
playersNumber int
winner Player
CharacterCardwithStudents
CharacterCardwithStudents(CharacterCardName, int, String, List
students List<Student>
addStudent(Student) void
removeStudent(CharacterColor) Student
students List<Student>
BoardExpert
BoardExpert(GameModel)
boardCoins int
characterCards CharacterCard[]
getCharacterCardbyName(String) CharacterCard
addCointoPlayer(int) void
getPlayerCoins(int) Integer
removeNoEntryTiles(int) void
createThreeRandomCharacterCards() CharacterCard[]
moveCoin(int, CharacterCard) void
createThreeCharacterCards(int) void
characterCards CharacterCard[]
coins int[]
boardCoins int
PlayerComparator
PlayerComparator()
compare(Player, Player) int
IdComparator
IdComparator()
compare(VirtualView, VirtualView) int
CharacterCardwithProhibitions
CharacterCardwithProhibitions(CharacterCardName, int, String
prohibitionsNumber int
subProhibitionCard() void
restockProhibitionsNumber() void
prohibitionsNumber int
Island
Island(Student)
Island(boolean)
students Map<CharacterColor, List<Student>>
noEntryTile int
towers List<Tower>
hasMotherNature() boolean
addTowers(List<Tower>) void
removeTowers() List<Towers>
hasNoEntryTile() boolean
addTower(Tower) void
addStudents(List<Student>) void
addStudent(Student) void
colorTower PlayerColor
towers List<Tower>
motherNature boolean
students Map<CharacterColor, List<Student>>
noEntryTile int
Professor
Professor(CharacterColor)
owner int
color CharacterColor
toString() String
owner int
color CharacterColor
Student
Student(CharacterColor)
color CharacterColor
toString() String
color CharacterColor
School
School(School)
School(Player, PlayerColor, int)
diningRoom Map<CharacterColor, List<Student>>
owner Player
numDiningRoomStudents int
towers List<Tower>
entrance List<Student>
professors List<Professor>
removeDiningRoomStudent(CharacterColor) Student
addDiningRoomStudent(Student) void
addEntranceStudents(List<Student>) void
fromEntrancetoDiningRoom(CharacterColor) void
restockTower(List<Tower>) void
hasEntranceStudentColor(String) boolean
addEntranceStudent(Student) void
removeEntranceStudent(CharacterColor) Student
getProfessorByColor(CharacterColor) Professor
addProfessor(Professor) void
removeTower() Tower
removeProfessor(CharacterColor) Professor
ownerId int
entrance List<Student>
owner Player
towerColor PlayerColor
diningRoom Map<CharacterColor, List<Student>>
towers List<Tower>
professors List<Professor>
towersNumber int
numDiningRoomStudents int
Server
Server()
addClientConnectionToQueue(VirtualView, int, boolean) void
toString() String
removeGameHandler(GameHandler) void
main(String[]) void
removeClientConnection(VirtualView, int, boolean) void
createGameHandler(int) void
run() void
Cloud
Cloud()
students List<Student>
addStudents(List<Student>) void
removeStudents() List<Student>
students List<Student>
Player
Player(String, int)
deck Deck
chosenAssistantCard AssistantCard
color PlayerColor
nickname String
clientID int
clientID int
deck Deck
assistantCard int
nickname String
chosenAssistantCard AssistantCard
color PlayerColor
Deck
Deck()
wizard Wizard
assistantCards List<AssistantCard>
removeAssistantCard(int) AssistantCard
wizard Wizard
assistantCards List<AssistantCard>
VirtualView
VirtualView(Socket, Server)
VirtualView()
gameHandler GameHandler
clientId int
inGame boolean
run() void
startPinger() void
sendMessage(Message) void
closeConnection(boolean) void
clientId int
inGame boolean
gameHandler GameHandler
«create»
«create»
«create»