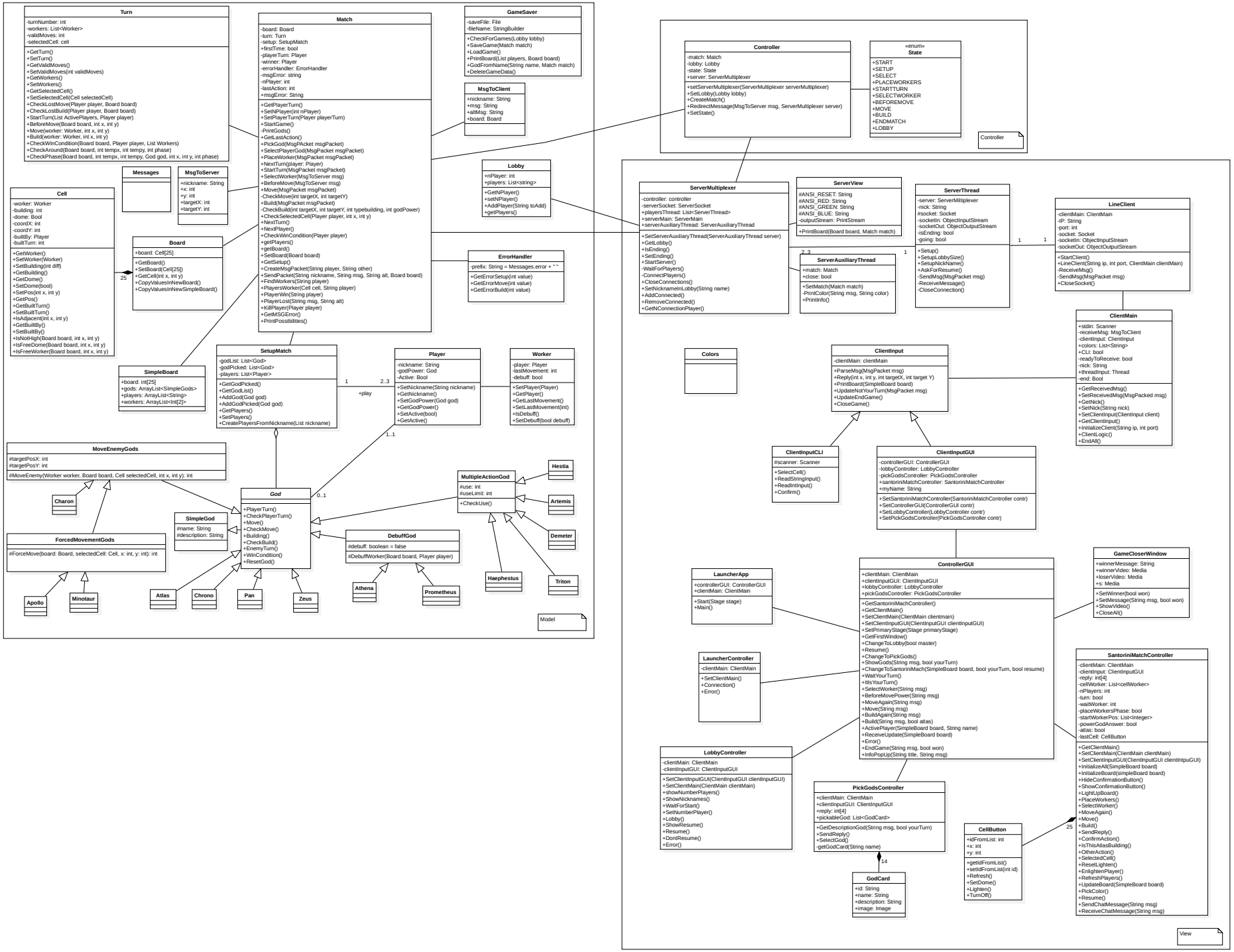
Model::Main


Turn
-turnNumber: int
-workers: List<Worker>
-validMoves: int
-selectedCell: cell
+GetTurn()
+SetTurn()
+GetValidMoves()
+SetValidMoves(int validMoves)
+GetWorkers()
+SetWorkers()
+GetSelectedCell()
+SetSelectedCell(Cell selectedCell)
+CheckLostMove(Player player, Board board)
+CheckLostBuild(Player player, Board board)
+StartTurn(List ActivePlayers, Player player)
+BeforeMove(Board board, int x, int y)
+Move(worker: Worker, int x, int y)
+Build(worker: Worker, int x, int y)
+CheckWinCondition(Board board, Player player, List Workers)
+CheckAround(Board board, int tempx, int tempy, int phase)
+CheckPhase(Board board, int tempx, int tempy, God god, int x, int y, int phase)
Match
-board: Board
-turn: Turn
-setup: SetupMatch
+firstTime: bool
-playerTurn: Player
-winner: Player
-errorHandler: ErrorHandler
-msgError: string
-nPlayer: int
-lastAction: int
+msgError: String
+GetPlayerTurn()
+SetNPlayer(int nPlayer)
+SetPlayerTurn(Player playerTurn)
+StartGame()
-PrintGods()
+GetLastAction()
+PickGod(MsgPacket msgPacket)
+SelectPlayerGod(MsgPacket msgPacket)
+PlaceWorker(MsgPacket msgPacket)
+NextTurn(player: Player)
+StartTurn(MsgPacket msgPacket)
+SelectWorker(MsgToServer msg)
+BeforeMove(MsgToServer msg)
+Move(MsgPacket msgPacket)
-CheckMove(int targetX, int targetY)
+Build(MsgPacket msgPacket)
-CheckBuild(int targetX, int targetY, int typebuilding, int godPower)
+CheckSelectedCell(Player player, int x, int y)
+NextTurn()
+NextPlayer()
+CheckWinCondition(Player player)
+getPlayers()
+getBoard()
+SetBoard(Board board)
+GetSetup()
+CreateMsgPacket(String player, String other)
+SendPacket(String nickname, String msg, String alt, Board board)
+FindWorkers(String player)
+PlayersWorker(Cell cell, String player)
+PlayerWin(String player)
+PlayerLost(String msg, String alt)
+KillPlayer(Player player)
+GetMSGError()
+PrintPossibilities()
GameSaver
-saveFile: File
-fileName: StringBuilder
+CheckForGames(Lobby lobby)
+SaveGame(Match match)
+LoadGame()
+PrintBoard(List players, Board board)
+GodFromName(String name, Match match)
+DeleteGameData()
MsgToClient
+nickname: String
+msg: String
+allMsg: String
+board: Board
Messages
MsgToServer
+nickname: String
+x: int
+y: int
+targetX: int
+targetY: int
Lobby
+nPlayer: int
+players: List<string>
+GetNPlayer()
+setNPlayer()
+AddPlayer(String toAdd)
+getPlayers()
Cell
-worker: Worker
-building: int
-dome: Bool
-coordX: int
-coordY: int
-builtBy: Player
-builtTurn: int
+GetWorker()
+SetWorker(Worker)
+SetBuilding(int diff)
+GetBuilding()
+GetDome()
+SetDome(bool)
+SetPos(int x, int y)
+GetPos()
+GetBuiltTurn()
+SetBuiltTurn()
+IsAdjacent(int x, int y)
+GetBuiltBy()
+SetBuiltBy()
+IsNotHigh(Board board, int x, int y)
+IsFreeDome(Board board, int x, int y)
+IsFreeWorker(Board board, int x, int y)
25
Board
+board: Cell[25]
+GetBoard()
+SetBoard(Cell[25])
+GetCell(int x, int y)
+CopyValuesInNewBoard()
+CopyValuesInNewSimpleBoard()
ErrorHandler
-prefix: String = Messages.error + " "
+GetErrorSetup(int value)
+GetErrorMove(int value)
+GetErrorBuild(int value)
SimpleBoard
+board: int[25]
+gods: ArrayList<SImpleGods>
+players: ArrayList<String>
+workers: ArrayList<Int[2]>
SetupMatch
-godList: List<God>
-godPicked: List<God>
-players: List<Player>
+GetGodPicked()
+GetGodList()
+AddGod(God god)
+AddGodPicked(God god)
+GetPlayers()
+SetPlayers()
+CreatePlayersFromNickname(List nickname)
1
2..3
+play
Player
-nickname: String
-godPower: God
-Active: Bool
+SetNickname(String nickname)
+GetNickname()
+SetGodPower(God god)
+GetGodPower()
+SetActive(bool)
+GetActive()
Worker
-player: Player
-lastMovement: int
-debuff: bool
+SetPlayer(Player)
+GetPlayer()
+GetLastMovement()
+SetLastMovement(int)
+IsDebuff()
+SetDebuff(bool debuff)
1..1
MoveEnemyGods
#targetPosX: int
#targetPosY: int
#MoveEnemy(Worker worker, Board board, Cell selectedCell, int x, int y): int
Charon
God
+PlayerTurn()
+CheckPlayerTurn()
+Move()
+CheckMove()
+Building()
+CheckBuild()
+EnemyTurn()
+WinCondition()
+ResetGod()
0..1
MultipleActionGod
#use: int
#useLimit: int
+CheckUse()
Hestia
Artemis
Demeter
Triton
Haephestus
SImpleGod
#name: String
#description: String
ForcedMovementGods
#ForceMove(board: Board, selectedCell: Cell, x: int, y: int): int
DebuffGod
#debuff: boolean = false
#DebuffWorker(Board board, Player player)
Apollo
Minotaur
Atlas
Chrono
Pan
Zeus
Athena
Prometheus
Model
Controller
-match: Match
-lobby: Lobby
-state: State
+server: ServerMultiplexer
+setServerMultiplexer(ServerMultiplexer serverMultiplexer)
+SetLobby(Lobby lobby)
+CreateMatch()
+RedirectMessage(MsgToServer msg, ServerMultiplexer server)
+SetState()
«enum»
State
+START
+SETUP
+SELECT
+PLACEWORKERS
+STARTTURN
+SELECTWORKER
+BEFOREMOVE
+MOVE
+BUILD
+ENDMATCH
+LOBBY
Controller
ServerMultiplexer
-controller: controller
-serverSocket: ServerSocket
+playersThread: List<ServerThread>
+serverMain: ServerMain
+serverAuxiliaryThread: ServerAuxiliaryThread
+SetServerAuxiliaryThread(ServerAuxiliaryThread server)
+GetLobby()
+IsEnding()
+SetEnding()
+StartServer()
-WaitForPlayers()
-ConnectPlayers()
+CloseConnections()
+SetNicknameInLobby(String name)
+AddConnected()
+RemoveConnected()
+GetNConnectionPlayer()
ServerView
#ANSI_RESET: String
#ANSI_RED: String
#ANSI_GREEN: String
#ANSI_BLUE: String
-outputStream: PrintStream
+PrintBoard(Board board, Match match)
ServerThread
-server: ServerMiltiplexer
-nick: String
#socket: Socket
-socketIn: ObjectInputStream
-socketOut: ObjectOutputStream
-isEnding: bool
-going: bool
+Setup()
+SetupLobbySize()
+SetupNickName()
+AskForResume()
-SendMsg(MsgPacket msg)
-ReceiveMessage()
-CloseConnection()
1
1
LineClient
-clientMain: ClientMain
-IP: String
-port: int
-socket: Socket
-socketIn: ObjectInputStream
-socketOut: ObjectOutputStream
+StartClient()
+LineClient(String ip, int port, ClientMain clientMain)
-ReceiveMsg()
-SendMsg(MsgPacket msg)
+CloseSocket()
2..3
1
ServerAuxiliaryThread
+match: Match
+close: bool
+SetMatch(Match match)
-PrintColor(String msg, String color)
+PrintInfo()
ClientMain
+stdin: Scanner
-receiveMsg: MsgToClient
-clientInput: ClientInput
+colors: List<String>
+CLI: bool
-readyToReceive: bool
-nick: String
+threadInput: Thread
-end: Bool
+GetReceivedMsg()
+SetReceivedMsg(MsgPacked msg)
+GetNick()
+SetNick(String nick)
+SetClientInput(ClientInput client)
+GetClientInput()
+InitializeClient(String ip, int port)
+ClientLogic()
+EndAll()
Colors
ClientInput
-clientMain: clientMain
+ParseMsg(MsgPacket msg)
+Reply(int x, int y, int targetX, int target Y)
+PrintBoard(SimpleBoard board)
+UpdateNotYourTurn(MsgPacket msg)
+UpdateEndGame()
+CloseGame()
ClientInputCLI
#scanner: Scanner
+SelectCell()
+ReadStringInput()
+ReadIntInput()
+Confirm()
ClientInputGUI
-controllerGUI: ControllerGUI
-lobbyController: LobbyController
-pickGodsController: PickGodsController
+santoriniMatchController: SantoriniMatchController
+myName: String
+SetSantoriniMatchController(SantoriniMatchController contr)
+SetControllerGUI(ControllerGUI contr)
+SetLobbyController(LobbyController contr)
+SetPickGodsController(PickGodsController contr)
LauncherApp
+controllerGUI: ControllerGUI
+clientMain: ClientMain
+Start(Stage stage)
+Main()
ControllerGUI
+clientMain: ClientMain
+clientInputGUI: ClientInputGUI
+lobbyController: LobbyController
+pickGodsController: PickGodsController
+GetSantoriniMachController()
+GetClientMain()
+SetClientMain(ClientMain clientmain)
+SetClientInputGUI(ClientInputGUI clientInputGUI)
+SetPrimaryStage(Stage primaryStage)
+GetFirstWindow()
+ChangeToLobby(bool master)
+Resume()
+ChangeToPickGods()
+ShowGods(String msg, bool yourTurn)
+ChangeToSantoriniMach(SimpleBoard board, bool yourTurn, bool resume)
+WaitYourTurn()
+ItsYourTurn()
+SelectWorker(String msg)
+BeforeMovePower(String msg)
+MoveAgain(String msg)
+Move(String msg)
+BuildAgain(String msg)
+Build(String msg, bool atlas)
+ActivePlayer(SimpleBoard board, String name)
+ReceiveUpdate(SimpleBoard board)
+Error()
+EndGame(String msg, bool won)
+InfoPopUp(String title, String msg)
GameCloserWindow
+winnerMessage: String
+winnerVideo: Media
+loserVideo: Media
+s: Media
+SetWinner(bool won)
+SetMessage(String msg, bool won)
+ShowVideo()
+CloseAll()
LauncherController
-clientMain: ClientMain
+SetClientMain()
+Connection()
+Error()
SantoriniMatchController
-clientMain: ClientMain
-clientInput: ClientInputGUI
-reply: int[4]
-cellWorker: List<cellWorker>
-nPlayers: int
-turn: bool
-waitWorker: int
-placeWorkersPhase: bool
-startWorkerPos: List<Integer>
-powerGodAnswer: bool
-atlas: bool
-lastCell: CellButton
+GetClientMain()
+SetClientMain(ClientMain clientMain)
+SetClientInputGUI(ClientInputGUI clientIntpuGUI)
+InitializeAll(SimpleBoard board)
+InitializeBoard(simpleBoard board)
+HideConfirmationButton()
+ShowConfirmationButton()
+LightUpBoard()
+PlaceWorkers()
+SelectWorker()
+MoveAgain()
+Move()
+Build()
+SendReply()
+ConfirmAction()
+IsThisAtlasBuilding()
+OtherAction()
+SelectedCell()
+ResetLighten()
+EnlightenPlayer()
+RefreshPlayers()
+UpdateBoard(SimpleBoard board)
+PickColor()
+Resume()
+SendChatMessage(String msg)
+ReceiveChatMessage(String msg)
LobbyController
-clientMain: ClientMain
-clientInputGUI: ClientInputGUI
+SetClientInputGUI(ClientInputGUI clientInputGUI)
+SetClientMain(ClientMain clientMain)
+ShowNumberPlayers()
+ShowNicknames()
+WaitForStart()
+SetNumberPlayer()
+Lobby()
+ShowResume()
+Resume()
+DontResume()
+Error()
PickGodsController
+clientMain: ClientMain
+clientInputGUI: ClientInputGUI
+reply: int[4]
+pickableGod: List<GodCard>
+GetDescriptionGod(String msg, bool yourTurn)
+SendReply()
+SelectGod()
-getGodCard(String name)
14
GodCard
+id: String
+name: String
+description: String
+image: Image
CellButton
+idFromList: int
+x: int
+y: int
+getIdFromList()
+setIdFromList(int id)
+Refresh()
+SetDome()
+Lighten()
+TurnOff()
25
View