UML CLIENT SIDE NETWORK

PeriodicUpdater

- messageSerializer: MessageSerializer
- gameID: String
- serverAdapter: ServerAdapter

+ PeriodicUpdater(gId: String,s: ServerAdapter)
+ run(): void

ServerAdapter

- nextCommands: Commands
- requestContent: String
- messageDeserialize: MessageDeserializer
- server: Socker
- outputStm ObjectOutputStream
- inputStm ObjectInputStream
- observers List<ServerObservers>

+ ServerAdapter(server: Socket)
+ addObserver(observer ServerObserver): void
+ removeObserver(observer ServerObserver): void
+ stop(): void
+ requestJoinGame(input: String): void
+ requestCheckModel(input: String): void
+ requestSendDivinities(input: String): void
+ requestSendDivinity(input: String): void
+ requestSendStartingPosition(input: String): void
+ requestSendChosenPawn(input: String): void
+ requestSendChosenMove(input: String): void
+ run(): void
+ handleServerConnection(): void
+ doJoinGame(): void
+ doSendDivinities(): void
+ doSendDivinity(): void
+ doSendStartingPosition(): void
+ doSendChosenPawn(): void
+ doSendChosenMove(): void
+ receiveEndGame(): void
+ doCheckModel(): void

ServerObserver

+ receiveUsernameTaken(response: String): void
+ receiveNewPlayerConnected(player, :Player,gameID: String): void
+ receiveDivinities(divinities: String): void
+ receivePossibleDivinities(response: String): void
+ receivePawn(pawn: String): void
+ receiveMoves(moves: String): void
+ receiveEndGame(endGame: String): void
+ receiveModelUpdate(game: Game): void

0..* — 1