

James Gossling

PlayController

PlayController(inQ : inBoundQueue, rules : Rules)
play(table: Table, player: Player, views: ViewFacade) : Event

InBoundQueue

- serialVersionUID : long
- pushBack : Stack<Event>

InBoundQueue()
add(e : Event) : boolean
take() : Event
pushBack(e : Event)
pushBack(e : Event)
isEmpty() : boolean

«interface»
Rules

eval(nextE: Event, table, Table, player: Player) : Move

«interface»
Event

dispatch(rules : RulesDispatch, table : Table, player : Player)

«interface»
Table

addPile(pile : Pile)
addPlayer(p : Player)
getPile(string : String) : Pile
removeFromPile(string : String, c : Card)
addToPile(string : String, c : Card)
addToScore(p : Player, i : int)
isMatchOver() : boolean
setMatchOver(over : boolean)
getRandom() : Random
getHost() : Party
partiesReady() : boolean
getCurrentPlayer() : Player
setQuorum(quorum: Quorum)
getQuorum() : Quorum
getPlayers() : Collection<Player>
getPlayerMap() : Map<Integer, Player>
createPlayer(pos : Integer, socketId : String)
lookupPlayer(socketId : String) : Player
getPlayer(pos : Integer) : Player

«interface»
Player

addToScore(amount : int) : int
getPlayerNum() : int
getSocketId() : String
getScore() : int

«interface»
Party

«interface»
Move

apply(table : Table)
apply(views : ViewFacade)
isMatchEnd() : boolean

ViewFacade

- factory : ViewFactory
- views : List<View>
- remotIdLookup : Map<String, String>

ViewFacade(factory : ViewFactory)
createView(role : PartyRole, pos : Integer, socketId
getDefaultView() : View
send(cmd : Marshalls)
add(view : View)
register(item : PresentationObject)
getRemotId(selector : String) : String