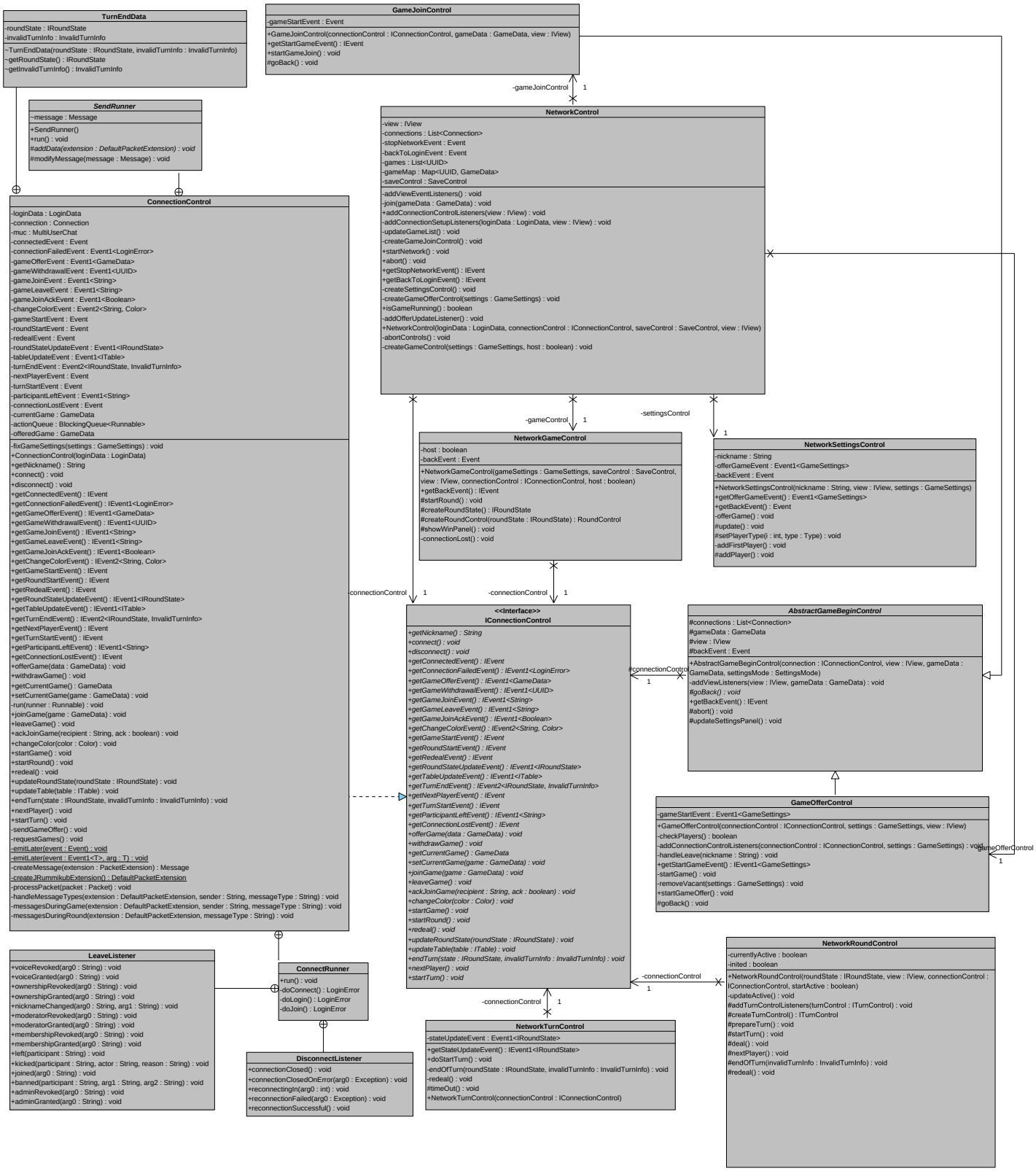
TurnEndData
-roundState : IRoundState
-invalidTurnInfo : InvalidTurnInfo
~TurnEndData(roundState : IRoundState, invalidTurnInfo : InvalidTurnInfo)
~getRoundState() : IRoundState
~getInvalidTurnInfo() : InvalidTurnInfo
GameJoinControl
-gameStartEvent : Event
+GameJoinControl(connectionControl : IConnectionControl, gameData : GameData, view : IView)
+getStartGameEvent() : IEvent
+startGameJoin() : void
#goBack() : void
-gameJoinControl
1
SendRunner
~message : Message
+SendRunner()
+run() : void
#addData(extension : DefaultPacketExtension) : void
#modifyMessage(message : Message) : void
NetworkControl
-view : IView
-connections : List<Connection>
-stopNetworkEvent : Event
-backToLoginEvent : Event
-games : List<UUID>
-gameMap : Map<UUID, GameData>
-saveControl : SaveControl
-addViewEventListeners() : void
-join(gameData : GameData) : void
+addConnectionControlListeners(view : IView) : void
+addConnectionSetupListeners(loginData : LoginData, view : IView) : void
-updateGameList() : void
-createGameJoinControl() : void
+startNetwork() : void
+abort() : void
+getStopNetworkEvent() : IEvent
+getBackToLoginEvent() : IEvent
-createSettingsControl() : void
-createGameOfferControl(settings : GameSettings) : void
+isGameRunning() : boolean
-addOfferUpdateListener() : void
+NetworkControl(loginData : LoginData, connectionControl : IConnectionControl, saveControl : SaveControl, view : IView)
-abortControls() : void
-createGameControl(settings : GameSettings, host : boolean) : void
ConnectionControl
-loginData : LoginData
-connection : Connection
-muc : MultiUserChat
-connectedEvent : Event
-connectionFailedEvent : Event1<LoginError>
-gameOfferEvent : Event1<GameData>
-gameWithdrawalEvent : Event1<UUID>
-gameJoinEvent : Event1<String>
-gameLeaveEvent : Event1<String>
-gameJoinAckEvent : Event1<Boolean>
-changeColorEvent : Event2<String, Color>
-gameStartEvent : Event
-roundStartEvent : Event
-redealEvent : Event
-roundStateUpdateEvent : Event1<IRoundState>
-tableUpdateEvent : Event1<ITable>
-turnEndEvent : Event2<IRoundState, InvalidTurnInfo>
-nextPlayerEvent : Event
-turnStartEvent : Event
-participantLeftEvent : Event1<String>
-connectionLostEvent : Event
-currentGame : GameData
-actionQueue : BlockingQueue<Runnable>
-offeredGame : GameData
-fixGameSettings(settings : GameSettings) : void
+ConnectionControl(loginData : LoginData)
+getNickname() : String
+connect() : void
+disconnect() : void
+getConnectedEvent() : IEvent
+getConnectionFailedEvent() : IEvent1<LoginError>
+getGameOfferEvent() : IEvent1<GameData>
+getGameWithdrawalEvent() : IEvent1<UUID>
+getGameJoinEvent() : IEvent1<String>
+getGameLeaveEvent() : IEvent1<String>
+getGameJoinAckEvent() : IEvent1<Boolean>
+getChangeColorEvent() : IEvent2<String, Color>
+getGameStartEvent() : IEvent
+getRoundStartEvent() : IEvent
+getRedealEvent() : IEvent
+getRoundStateUpdateEvent() : IEvent1<IRoundState>
+getTableUpdateEvent() : IEvent1<ITable>
+getTurnEndEvent() : IEvent2<IRoundState, InvalidTurnInfo>
+getNextPlayerEvent() : IEvent
+getTurnStartEvent() : IEvent
+getParticipantLeftEvent() : IEvent1<String>
+getConnectionLostEvent() : IEvent
+offerGame(data : GameData) : void
+withdrawGame() : void
+getCurrentGame() : GameData
+setCurrentGame(game : GameData) : void
-run(runner : Runnable) : void
+joinGame(game : GameData) : void
+leaveGame() : void
+ackJoinGame(recipient : String, ack : boolean) : void
+changeColor(color : Color) : void
+startGame() : void
+startRound() : void
+redeal() : void
+updateRoundState(roundState : IRoundState) : void
+updateTable(table : ITable) : void
+endTurn(state : IRoundState, invalidTurnInfo : InvalidTurnInfo) : void
+nextPlayer() : void
+startTurn() : void
-sendGameOffer() : void
-requests() : void
-emitLater(event : Event) : void
-emitLater(event : Event1<T>, arg : T) : void
-createMessage(extension : PacketExtension) : Message
-createJRummikubExtension() : DefaultPacketExtension
-processPacket(packet : Packet) : void
-handleMessageTypes(extension : DefaultPacketExtension, sender : String, messageType : String) : void
-messagesDuringGame(extension : DefaultPacketExtension, sender : String, messageType : String) : void
-messagesDuringRound(extension : DefaultPacketExtension, messageType : String) : void
-gameControl
1
-settingsControl
1
NetworkGameControl
-host : boolean
-backEvent : Event
+NetworkGameControl(gameSettings : GameSettings, saveControl : SaveControl, view : IView, connectionControl : IConnectionControl, host : boolean)
+getBackEvent() : IEvent
#startRound() : void
#createRoundState() : IRoundState
#createRoundControl(roundState : IRoundState) : RoundControl
#showWinPanel() : void
-connectionLost() : void
NetworkSettingsControl
-nickname : String
-offerGameEvent : Event1<GameSettings>
-backEvent : Event
+NetworkSettingsControl(nickname : String, view : IView, settings : GameSettings)
+getOfferGameEvent() : Event1<GameSettings>
+getBackEvent() : Event
-offerGame() : void
#update() : void
#setPlayerType(i : int, type : Type) : void
-addFirstPlayer() : void
#addPlayer() : void
-connectionControl
1
-connectionControl
1
<<Interface>>
IConnectionControl
+getNickname() : String
+connect() : void
+disconnect() : void
+getConnectedEvent() : IEvent
+getConnectionFailedEvent() : IEvent1<LoginError>
+getGameOfferEvent() : IEvent1<GameData>
+getGameWithdrawalEvent() : IEvent1<UUID>
+getGameJoinEvent() : IEvent1<String>
+getGameLeaveEvent() : IEvent1<String>
+getGameJoinAckEvent() : IEvent1<Boolean>
+getChangeColorEvent() : IEvent2<String, Color>
+getGameStartEvent() : IEvent
+getRoundStartEvent() : IEvent
+getRedealEvent() : IEvent
+getRoundStateUpdateEvent() : IEvent1<IRoundState>
+getTableUpdateEvent() : IEvent1<ITable>
+getTurnEndEvent() : IEvent2<IRoundState, InvalidTurnInfo>
+getNextPlayerEvent() : IEvent
+getTurnStartEvent() : IEvent
+getParticipantLeftEvent() : IEvent1<String>
+getConnectionLostEvent() : IEvent
+offerGame(data : GameData) : void
+withdrawGame() : void
+getCurrentGame() : GameData
+setCurrentGame(game : GameData) : void
+joinGame(game : GameData) : void
+leaveGame() : void
+ackJoinGame(recipient : String, ack : boolean) : void
+changeColor(color : Color) : void
+startGame() : void
+startRound() : void
+redeal() : void
+updateRoundState(roundState : IRoundState) : void
+updateTable(table : ITable) : void
+endTurn(state : IRoundState, invalidTurnInfo : InvalidTurnInfo) : void
+nextPlayer() : void
+startTurn() : void
#connectionControl
1
AbstractGameBeginControl
#connections : List<Connection>
#gameData : GameData
#view : IView
#backEvent : Event
+AbstractGameBeginControl(connection : IConnectionControl, view : IView, gameData : GameData, settingsMode : SettingsMode)
-addViewListeners(view : IView, gameData : GameData) : void
#goBack() : void
+getBackEvent() : IEvent
#abort() : void
#updateSettingsPanel() : void
GameOfferControl
-gameStartEvent : Event1<GameSettings>
+GameOfferControl(connectionControl : IConnectionControl, settings : GameSettings, view : IView)
-checkPlayers() : boolean
-addConnectionControlListeners(connectionControl : IConnectionControl, settings : GameSettings) : void
-handleLeave(nickname : String) : void
+getStartGameEvent() : IEvent1<GameSettings>
-startGame() : void
-removeVacant(settings : GameSettings) : void
+startGameOffer() : void
#goBack() : void
gameOfferControl
1
LeaveListener
+voiceRevoked(arg0 : String) : void
+voiceGranted(arg0 : String) : void
+ownershipRevoked(arg0 : String) : void
+ownershipGranted(arg0 : String) : void
+nicknameChanged(arg0 : String, arg1 : String) : void
+moderatorRevoked(arg0 : String) : void
+moderatorGranted(arg0 : String) : void
+membershipRevoked(arg0 : String) : void
+membershipGranted(arg0 : String) : void
+left(participant : String) : void
+kicked(participant : String, actor : String, reason : String) : void
+joined(arg0 : String) : void
+banned(participant : String, arg1 : String, arg2 : String) : void
+adminRevoked(arg0 : String) : void
+adminGranted(arg0 : String) : void
ConnectRunner
+run() : void
-doConnect() : LoginError
-doLogin() : LoginError
-doJoin() : LoginError
-connectionControl
1
-connectionControl
1
NetworkRoundControl
-currentlyActive : boolean
-inited : boolean
+NetworkRoundControl(roundState : IRoundState, view : IView, connectionControl : IConnectionControl, startActive : boolean)
-updateActive() : void
#addTurnControlListeners(turnControl : ITurnControl) : void
#createTurnControl() : ITurnControl
#prepareTurn() : void
#startTurn() : void
#deal() : void
#nextPlayer() : void
#endOfTurn(invalidTurnInfo : InvalidTurnInfo) : void
#redeal() : void
DisconnectListener
+connectionClosed() : void
+connectionClosedOnError(arg0 : Exception) : void
+reconnectingIn(arg0 : int) : void
+reconnectionFailed(arg0 : Exception) : void
+reconnectionSuccessful() : void
-connectionControl
1
NetworkTurnControl
-stateUpdateEvent : Event1<IRoundState>
+getStateUpdateEvent() : IEvent1<IRoundState>
+doStartTurn() : void
-endOfTurn(roundState : IRoundState, invalidTurnInfo : InvalidTurnInfo) : void
-redeal() : void
#timeOut() : void
+NetworkTurnControl(connectionControl : IConnectionControl)