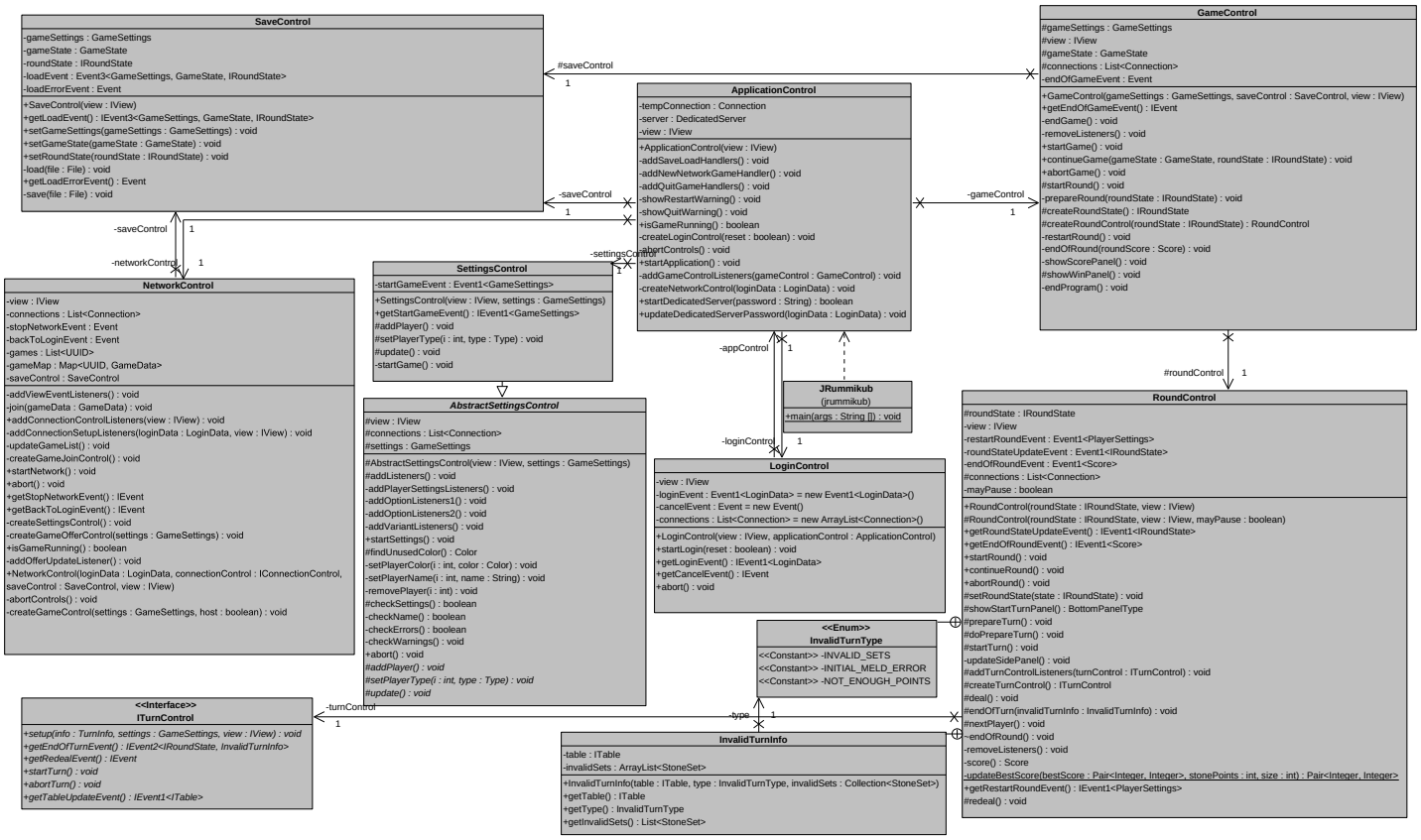
SaveControl
-gameSettings : GameSettings
-gameState : GameState
-roundState : IRoundState
-loadEvent : Event3<GameSettings, GameState, IRoundState>
-loadErrorEvent : Event
+SaveControl(view : IView)
+getLoadEvent() : IEvent3<GameSettings, GameState, IRoundState>
+setGameSettings(gameSettings : GameSettings) : void
+setGameState(gameState : GameState) : void
+setRoundState(roundState : IRoundState) : void
-load(file : File) : void
+getLoadErrorEvent() : Event
-save(file : File) : void
#saveControl
1
GameControl
#gameSettings : GameSettings
#view : IView
#gameState : GameState
#connections : List<Connection>
-endOfGameEvent : Event
+GameControl(gameSettings : GameSettings, saveControl : SaveControl, view : IView)
+getEndOfGameEvent() : IEvent
-endGame() : void
-removeListeners() : void
+startGame() : void
+continueGame(gameState : GameState, roundState : IRoundState) : void
+abortGame() : void
#startRound() : void
-prepareRound(roundState : IRoundState) : void
#createRoundState() : IRoundState
#createRoundControl(roundState : IRoundState) : RoundControl
-restartRound() : void
-endOfRound(roundScore : Score) : void
-showScorePanel() : void
#showWinPanel() : void
-endProgram() : void
ApplicationControl
-tempConnection : Connection
-server : DedicatedServer
-view : IView
+ApplicationControl(view : IView)
-addSaveLoadHandlers() : void
-addNewNetworkGameHandler() : void
-addQuitGameHandlers() : void
-showRestartWarning() : void
-showQuitWarning() : void
+isGameRunning() : boolean
-createLoginControl(reset : boolean) : void
-abortControls() : void
+startApplication() : void
-addGameControlListeners(gameControl : GameControl) : void
-createNetworkControl(loginData : LoginData) : void
+startDedicatedServer(password : String) : boolean
+updateDedicatedServerPassword(loginData : LoginData) : void
-saveControl
1
-gameControl
1
-saveControl
1
-networkControl
1
-settingsControl
1
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
SettingsControl
-startGameEvent : Event1<GameSettings>
+SettingsControl(view : IView, settings : GameSettings)
+getStartGameEvent() : IEvent1<GameSettings>
#addPlayer() : void
#setPlayerType(i : int, type : Type) : void
#update() : void
-startGame() : void
AbstractSettingsControl
#view : IView
#connections : List<Connection>
#settings : GameSettings
#AbstractSettingsControl(view : IView, settings : GameSettings)
#addListeners() : void
-addPlayerSettingsListeners() : void
-addOptionListeners1() : void
-addOptionListeners2() : void
-addVariantListeners() : void
+startSettings() : void
#findUnusedColor() : Color
-setPlayerColor(i : int, color : Color) : void
-setPlayerName(i : int, name : String) : void
-removePlayer(i : int) : void
#checkSettings() : boolean
-checkName() : boolean
-checkErrors() : boolean
-checkWarnings() : void
+abort() : void
#addPlayer() : void
#setPlayerType(i : int, type : Type) : void
#update() : void
-appControl
1
JRummikub
(jrummikub)
+main(args : String []) : void
-loginControl
1
LoginControl
-view : IView
-loginEvent : Event1<LoginData> = new Event1<LoginData>()
-cancelEvent : Event = new Event()
-connections : List<Connection> = new ArrayList<Connection>()
+LoginControl(view : IView, applicationControl : ApplicationControl)
+startLogin(reset : boolean) : void
+getLoginEvent() : IEvent1<LoginData>
+getCancelEvent() : IEvent
+abort() : void
#roundControl
1
RoundControl
#roundState : IRoundState
-view : IView
-restartRoundEvent : Event1<PlayerSettings>
-roundStateUpdateEvent : Event1<IRoundState>
-endOfRoundEvent : Event1<Score>
#connections : List<Connection>
-mayPause : boolean
+RoundControl(roundState : IRoundState, view : IView)
#RoundControl(roundState : IRoundState, view : IView, mayPause : boolean)
+getRoundStateUpdateEvent() : IEvent1<IRoundState>
+getEndOfRoundEvent() : IEvent1<Score>
+startRound() : void
+continueRound() : void
+abortRound() : void
#setRoundState(state : IRoundState) : void
#showStartTurnPanel() : BottomPanelType
#prepareTurn() : void
#doPrepareTurn() : void
#startTurn() : void
-updateSidePanel() : void
#addTurnControlListeners(turnControl : ITurnControl) : void
#createTurnControl() : ITurnControl
#deal() : void
#endOfTurn(invalidTurnInfo : InvalidTurnInfo) : void
#nextPlayer() : void
-endOfRound() : void
-removeListeners() : void
-score() : Score
-updateBestScore(bestScore : Pair<Integer, Integer>, stonePoints : int, size : int) : Pair<Integer, Integer>
+getRestartRoundEvent() : IEvent1<PlayerSettings>
#redeal() : void
<<Enum>>
InvalidTurnType
<<Constant>> -INVALID_SETS
<<Constant>> -INITIAL_MELD_ERROR
<<Constant>> -NOT_ENOUGH_POINTS
<<Interface>>
ITurnControl
+setup(info : TurnInfo, settings : GameSettings, view : IView) : void
+getEndOfTurnEvent() : IEvent2<IRoundState, InvalidTurnInfo>
+getRedealEvent() : IEvent
+startTurn() : void
+abortTurn() : void
+getTableUpdateEvent() : IEvent1<ITable>
-turnControl
1
-type
1
InvalidTurnInfo
-table : ITable
-invalidSets : ArrayList<StoneSet>
+InvalidTurnInfo(table : ITable, type : InvalidTurnType, invalidSets : Collection<StoneSet>)
+getTable() : ITable
+getType() : InvalidTurnType
+getInvalidSets() : List<StoneSet>