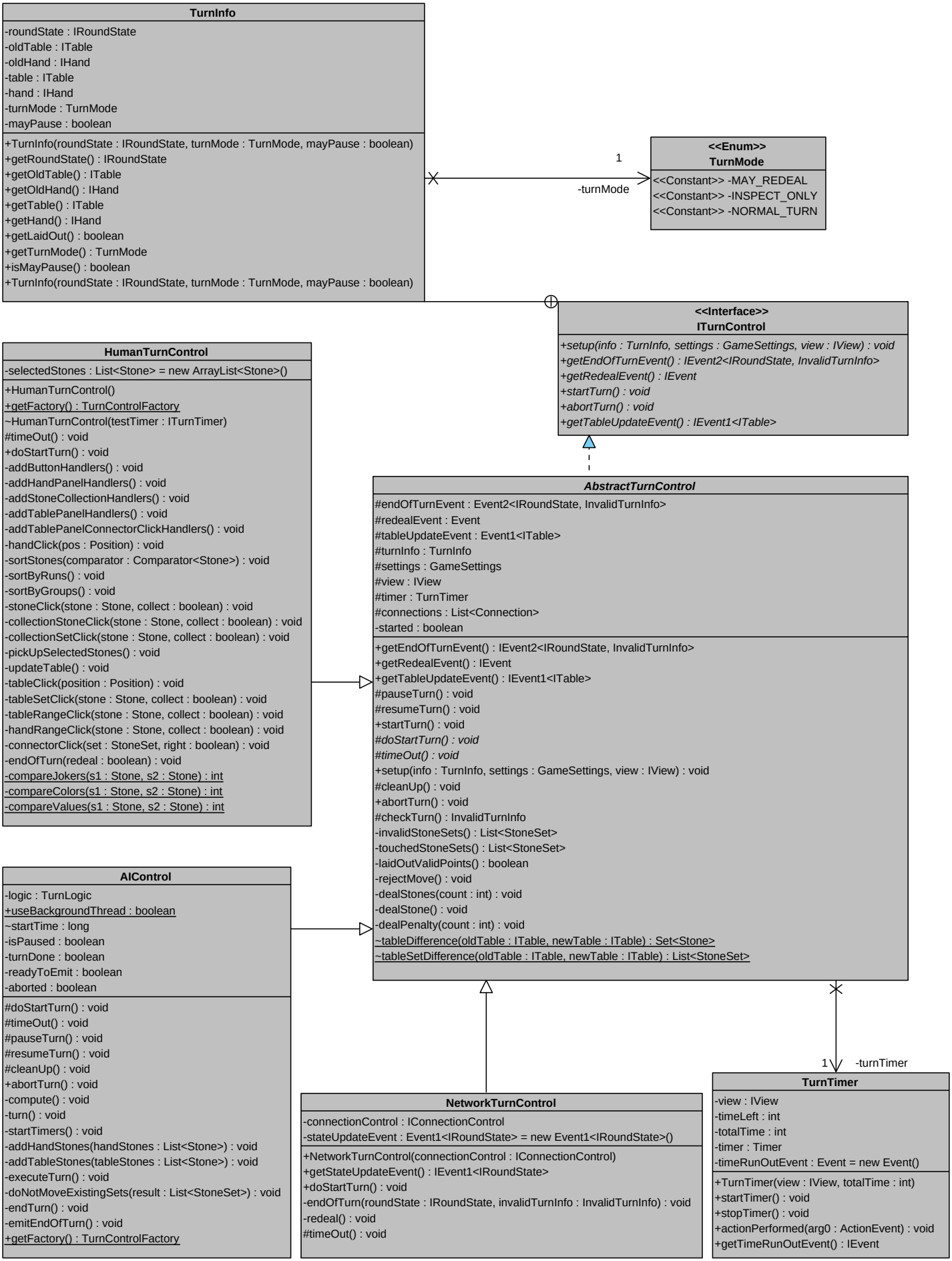
TurnInfo
-roundState : IRoundState
-oldTable : ITable
-oldHand : IHand
-table : ITable
-hand : IHand
-turnMode : TurnMode
-mayPause : boolean
+TurnInfo(roundState : IRoundState, turnMode : TurnMode, mayPause : boolean)
+getRoundState() : IRoundState
+getOldTable() : ITable
+getOldHand() : IHand
+getTable() : ITable
+getHand() : IHand
+getLaidOut() : boolean
+getTurnMode() : TurnMode
+isMayPause() : boolean
+TurnInfo(roundState : IRoundState, turnMode : TurnMode, mayPause : boolean)
1
-turnMode
<<Enum>>
TurnMode
<<Constant>> -MAY_REDEAL
<<Constant>> -INSPECT_ONLY
<<Constant>> -NORMAL_TURN
<<Interface>>
ITurnControl
+setup(info : TurnInfo, settings : GameSettings, view : IView) : void
+getEndOfTurnEvent() : IEvent2<IRoundState, InvalidTurnInfo>
+getRedealEvent() : IEvent
+startTurn() : void
+abortTurn() : void
+getTableUpdateEvent() : IEvent1<ITable>
HumanTurnControl
-selectedStones : List<Stone> = new ArrayList<Stone>()
+HumanTurnControl()
+getFactory() : TurnControlFactory
~HumanTurnControl(testTimer : ITurnTimer)
#timeOut() : void
+doStartTurn() : void
-addButtonHandlers() : void
-addHandPanelHandlers() : void
-addStoneCollectionHandlers() : void
-addTablePanelHandlers() : void
-addTablePanelConnectorClickHandlers() : void
-handClick(pos : Position) : void
-sortStones(comparator : Comparator<Stone>) : void
-sortByRuns() : void
-sortByGroups() : void
-stoneClick(stone : Stone, collect : boolean) : void
-collectionStoneClick(stone : Stone, collect : boolean) : void
-collectionSetClick(stone : Stone, collect : boolean) : void
-pickUpSelectedStones() : void
-updateTable() : void
-tableClick(position : Position) : void
-tableSetClick(stone : Stone, collect : boolean) : void
-tableRangeClick(stone : Stone, collect : boolean) : void
-handRangeClick(stone : Stone, collect : boolean) : void
-connectorClick(set : StoneSet, right : boolean) : void
-endOfTurn(redeal : boolean) : void
-compareJokers(s1 : Stone, s2 : Stone) : int
-compareColors(s1 : Stone, s2 : Stone) : int
-compareValues(s1 : Stone, s2 : Stone) : int
AbstractTurnControl
#endOfTurnEvent : Event2<IRoundState, InvalidTurnInfo>
#redealEvent : Event
#tableUpdateEvent : Event1<ITable>
#turnInfo : TurnInfo
#settings : GameSettings
#view : IView
#timer : TurnTimer
#connections : List<Connection>
-started : boolean
+getEndOfTurnEvent() : IEvent2<IRoundState, InvalidTurnInfo>
+getRedealEvent() : IEvent
+getTableUpdateEvent() : IEvent1<ITable>
#pauseTurn() : void
#resumeTurn() : void
+startTurn() : void
#doStartTurn() : void
#timeOut() : void
+setup(info : TurnInfo, settings : GameSettings, view : IView) : void
#cleanUp() : void
+abortTurn() : void
#checkTurn() : InvalidTurnInfo
-invalidStoneSets() : List<StoneSet>
-touchedStoneSets() : List<StoneSet>
-laidOutValidPoints() : boolean
-rejectMove() : void
-dealStones(count : int) : void
-dealStone() : void
-dealPenalty(count : int) : void
~tableDifference(oldTable : ITable, newTable : ITable) : Set<Stone>
~tableSetDifference(oldTable : ITable, newTable : ITable) : List<StoneSet>
AIControl
-logic : TurnLogic
+useBackgroundThread : boolean
~startTime : long
-isPaused : boolean
-turnDone : boolean
-readyToEmit : boolean
-aborted : boolean
#doStartTurn() : void
#timeOut() : void
#pauseTurn() : void
#resumeTurn() : void
#cleanUp() : void
+abortTurn() : void
-compute() : void
-turn() : void
-startTimers() : void
-addHandStones(handStones : List<Stone>) : void
-addTableStones(tableStones : List<Stone>) : void
-executeTurn() : void
-doNotMoveExistingSets(result : List<StoneSet>) : void
-endTurn() : void
-emitEndOfTurn() : void
+getFactory() : TurnControlFactory
NetworkTurnControl
-connectionControl : IConnectionControl
-stateUpdateEvent : Event1<IRoundState> = new Event1<IRoundState>()
+NetworkTurnControl(connectionControl : IConnectionControl)
+getStateUpdateEvent() : IEvent1<IRoundState>
+doStartTurn() : void
-endOfTurn(roundState : IRoundState, invalidTurnInfo : InvalidTurnInfo) : void
-redeal() : void
#timeOut() : void
1
-turnTimer
TurnTimer
-view : IView
-timeLeft : int
-totalTime : int
-timer : Timer
-timeRunOutEvent : Event = new Event()
+TurnTimer(view : IView, totalTime : int)
+startTimer() : void
+stopTimer() : void
+actionPerformed(arg0 : ActionEvent) : void
+getTimeRunOutEvent() : IEvent