

LaneStatusView

-jp:JPanel
-psv:PinSetterView
-lv:LaneView
-lane:Lane

+laneNum:int
+laneShowing:boolean
+psShowing:boolean

+LaneStatusView(Lane,int):
+showLane();
+actionPerformed(e:ActionE
+recieveLaneEvent(LaneEv
+receivePinsetterEvent(pe)

LaneView

-roll:int
-initDone:boolean
+cur:int
+bowlIt:Iterator
lane:Lane

+LaneView(Lane,int):
+show()
+hide()
-makeFrame(party:Party)
+receiveLaneEvent(lc:LaneE
+actionPerformed(e:ActionE

LaneEvent

-p:Party
+frame:int
+ball:int
+bowler:Bowler
+cumulScore:int[][]
+score:HashMap
+index:int
+frameNum:int
+curScores:int[]
+mechProb:bool

+LaneEvent(all var)
+getter() //for all var

Lane

- party:Party
-setter:Pinsetter
-scores:HashMap
-subscribers:Vector
-gamelsHalted:boolean
-partyAssigned:boolean
-gameFinished
-bowlerIterator:Iterator
-ball:int
-bowlIndex:int
-frameNumber:int
-tenthFrameStrike:bool
-curScores:int[]
-cumulScores:int[][]
-canThrowAgain:bool
-finalScores:int[][]
-gamNumber:int
-currentThrower:Bowler

+ Lane():
+run():
+receivePinsetterEvent(pe:PinsetterEvent)
+resetBowlerIterator()
+resetScores()
+assignParty(theParty:Party)
+markScore(Bowler,int,int,int):
+lanePublish()
+getScore(Bowler,int)
+isPartyAssigned()

<<Interface>>
LaneOberser

+ receiveLaneEvent(lc:LaneEvent)

<<Interface>>
ActionListener

<<Interface>>
LaneServer

+ subscribe(toAdd LaneObserver):

<<Interface>>
LaneEventInterface

+ getFrameNum():int
+getScore():Hashmap
+getCurScores():int[]
+getIndex():int
+getBall():int
+getCumlScore():int[][]
+getParty():Party
+getBowler():Bowler

