User

-id: String   (Hexadecimal)
-completed: List<Strings>
-nextPuzzle: String
-notAccessible: List<String>

+updateNextPuzzle():void
+getCompletedList():List<String>
+getNextPuzzle():String
+getPlayerRank():String {using completed.length()}

Board

-id: int
-squares:HashMap(Point, Square) {point is upper left corner of the square}
-sections:List<List<Point>>
-rows:int
-columns:int
-invalidStars:HashSet<Point2D>
-validStars: HashSet<Point2D>
-solution: List<Point>
-numStars:int

<<constructor>> Board(rows:int, columns:int, sections:List<List<Point>>, solution:List<Point>, numStars:int, id:int):void
+updateSquare(point:Point, state:String):void
-checkRow(row:int):bool
-checkColumn(column:int):bool
-checkSection(point:Point):bool
-checkArea(point:Point):bool
-checkSquare(point:Point):bool {calls other check methods}
+getID():int
+isComplete():bool
+getSectionBoundaries(): List<Line>
+clearBoard(): void
+getSolution():List<Point>
+getNumStars():int

Square

-state:String {none, "dot", "star"}
-isStarValid:bool = true

#setState(state:String): void

User 1 → 1 Board

Board 1 → 1..* Square