GameStatistics

+endingRound: int

+winner: RoundState.Winner +isWinByComparison: boolean +winningMove: DecisionType

Engine

```
+roundState: RoundState
+firstPlayer: Player
+secondPlayer: player
-currentPlayer: Player
-opponentPlayer: Player
+iteration: int
+statistics: GameStatistics
+log: StringBuilder
-endOfLine: String
+Engine(in firstPlayer:Player,in secondPlayer:Player)
+Engine(in firstPlayer:Player,in secondPlayer:Player,
        in roundStateCopy:RoundState)
-giveSpaceToPlayer(in firstPlayer:Player,
                   in secondPlayer:Player)
-initVariables()
+run()
+iterate()
+setCurrentAndOpponentSpace()
+getDecisionFromCurrentPlayer(out decision:DecisionType)
+executeDecision(in decision:DecisionType)
-guardPlay(in guardChoice:CardType)
-priestPlay()
-baronPlay()
-handmaidPlay()
-princePlay()
-kingPlay()
-countessPlay()
-princessPlay()
+endGame(in winner:Player)
+logResult()
```

playerAlgorithm.player

