

UML Class Diagram

Model / Controller

Constants
+ version: String
+ bookshelfX: int
+ bookshelfY: int
+ livingRoomBoardX: int
+ livingRoomBoardY: int
+ playersLowerBound: int
+ playersUpperBound: int
+ IDLowerBound: int
+ tileVariants: int
+ tilesPerType: int
+ minCommonGoalCards: int
+ maxCommonGoalCard: int
+ totalNumberOfCommonGoalCards: int
+ minPick: int
+ maxPick: int
+ endGameToken: int
- personalGoalCards: List<Map<String, ?>>
- commonGoalCardsDescription: Map<String, String>
+ defaultRMIRegistryPort: int
+ defaultSocketPort: int
+ defaultRMName: String
+ pingpongTimeout: long
+ connectionLostTimeout: long
+ walkoverTimeout: long
+ getInvalidPositions(int): Set<Point>
+ getScoringTokens(int): int[]
+ getPersonalGoalCards(): List<Map<String, ?>>
+ getCGCDescriptionByID(int): String
+ getPersonalGoalCardsPoints(int): int
+ getAdjacentTilesPoints(int): int

GameUtils
+ checkMatrixSize(T[][]): void
+ findGroup(Point, Tile[], boolean[]): int
+ checkAdjacentTile(Point, Tile[], Direction): Tile
+ checkIfTileCanBeTaken(Tile[], Point...): boolean
- checkContiguity(ToIntFunction<Point>, Point): boolean
+ checkIfColumnHasEnoughSpace(Tile[], int, int): boolean

Utils
+ extractRandomDsWithoutDuplicates(int, int): int[]
+ iterateInResourceDirectory(String, Consumer<String>): void

TextualUtils
+ nextInt(Scanner): int
+ nextInt(Scanner, int, int, String): int
+ nextIntInterruptible(Scanner): int
+ nextIntInterruptible(Scanner, int, int, String): int
+ isN(Scanner, String): boolean
+ printPoints(List<Point>): void

<<enumeration>> Direction
UP DOWN LEFT RIGHT

