Object - Class Diagram

Class Move

- startLocation: HashMap<Integer, Integer>
- direction: Integer
- tileNum: Integer
- tiles: List<Tile>
- specialTiles: List<Special Tile>
- specialTileLocation:

+ exchangeTile(): Boolean
+ removeTile(HashMap<Integer, Integer>()): Void
+ setTile(Tile tile): Void
+ setStartLocation(HashMap<Integer, Integer>()): Void
+ setDirection(Integer): Void
+ getSpecialTile(): SpecialTile
+ setSpecialTile(SpecialTile st): Void
+ getTileNumber(): Integer
+ getDirection(): Integer
+ getStartLocation(): HashMap<Integer, Integer>
+ getSpecialTileLocation(): HashMap<Integer, Integer>
+ getTiles(): List<Tile>
+ getLocation(): HashMap<Integer, Integer>

Class Board

- emptyNum: Integer
- locations: HashMap<Integer, Integer>()
- word: String

+ setSpecialTile(Move move): Void
+ setTile(Move move): Void
+ runSpecialTile(SpecialTile st): Void
+ removeTile(HashMap<Integer, Integer>()): Void
+ removeSpecialTile(SpecialTile st): Void
+ isOnStar(Move move): Boolean
+ isAdjacent(): Boolean
+ isOnSpecialTile(): Boolean
+ isOnBoard(HashMap<Integer, Integer>()): Boolean
+ getTile(Integer x, y): Tile
+ getWord(Set<Tile> tile): String
+ checkInLine(): Boolean
+ checkValidation(String word): Boolean

Class Scrabble

- players: List<Player>
- tilePackage: TilePackage
- board: Board
- playerMaxNum: Integer
- playerNum: Integer
- counterNum: Integer

+ startGame(): Void
+ endGame(): Void
+ addPlayer(): Void
+ runGame(): Void
+ getPlayers(): List<Player>
+ getCounter(): Integer
+ getCurrentPlayer(): Player
+ getNextPlayer(): Player
+ getPlayerNum(): Integer
+ getTile(): Tile
+ getWinner(): Player
+ getScore(Player player): Integer
+ giveTiles(Integer num, Player player): Void
+ exchangeTiles(Set<Tile> tiles, Player player): Void
+ isEndOfRound(): Boolean
+ setTiles(Move move): Boolean
+ skipOrder():) Void
+ lastMove(Player player): Move
+ doesPlayerStop(): Boolean
+ hasWinner(Player winner): Boolean
+ playMove(Move move): Boolean

Class Shop

- specialTiles: List<SpecialTile>
- numOfST: Integer

+ buySpecialTileByIndex(Integer index, Player player): Void

Class Player

- name: String
- score: Integer
- tileNum: Integer
- tileMaxNum: final Integer
- inventory: Inventory
- Move: move

+ updateScore(Integer score): Void
+ updateTiles(Move move): Set<Tile>
+ getWord(Set<Tiles> tiles): String
+ getScore(Set<Tiles> tiles): Integer
+ skipOrder(): Void
+ exchangeTiles(Set<Tiles> oldTiles, newTiles): Void
+ challenge(): Boolean
+ getTiles(Move move): Set<Tiles> newTiles
+ getMove(): Move
+ runMove(Move): Void
+ checkTilesValidation(Move move): Boolean

<<Interface>> SpecialTile

+ runSpecialTile(): Void
+ deactivate(): Void
+ setOwner(Player player): Void
+ getName(): String
+ getOwner(): Player
+ getPrice(): Integer

<<Interface>> Multipler

+ setValue(String word): Void
+ getName(): String

Class Tile

- letter: Character
- value: Integer

+ getLetter(): Character
+ getValue(): Integer

Class Tile Package

- tiles: Set<Tile>
- num: Integer

+ getTiles(Integer): Set<Tile>
+ isEmpty(): Boolean
+ getNum(): Integer
+ addTiles(Tile tile): Void

Abstract Class BaseSpecialTile

- name: String
- price: Integer
- owner: Player
- location: HashMap<Integer, Integer>

Abstract Class BaseMultipler

- name: String

+ getName(): String
+ updateTiles(Set<Tile>): Void

Buy

Get Tiles

Extends

Class Reverse-player-order

Class Nagative-points

Class Boom

Class My-tile

Class Unknown-tile

Class DoubleLetter

Class DoubleWord

Class TripleLetter

Class TripleWord