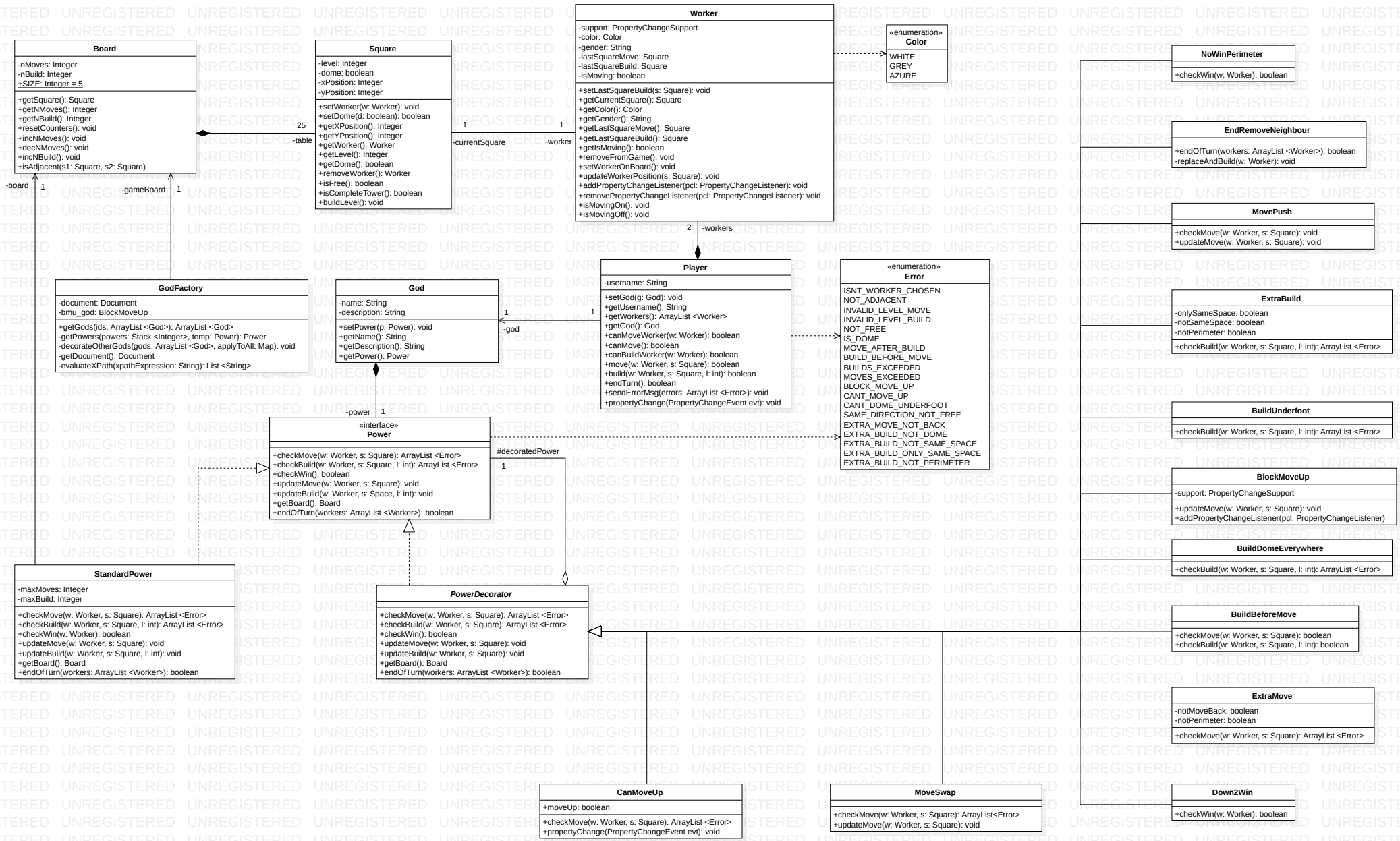
Model.main


Board
-nMoves: Integer
-nBuild: Integer
+SIZE: Integer = 5
+getSquare(): Square
+getNMoves(): Integer
+getNBuild(): Integer
+resetCounters(): void
+incNMoves(): void
+decNMoves(): void
+incNBuild(): void
+isAdjacent(s1: Square, s2: Square)
Square
-level: Integer
-dome: boolean
-xPosition: Integer
-yPosition: Integer
+setWorker(w: Worker): void
+setDome(d: boolean): boolean
+getXPosition(): Integer
+getYPosition(): Integer
+getWorker(): Worker
+getLevel(): Integer
+getDome(): boolean
+removeWorker(): Worker
+isFree(): boolean
+isCompleteTower(): boolean
+buildLevel(): void
Worker
-support: PropertyChangeSupport
-color: Color
-gender: String
-lastSquareMove: Square
-lastSquareBuild: Square
-isMoving: boolean
+setLastSquareBuild(s: Square): void
+getCurrentSquare(): Square
+getColor(): Color
+getGender(): String
+getLastSquareMove(): Square
+getLastSquareBuild(): Square
+getIsMoving(): boolean
+removeFromGame(): void
+setWorkerOnBoard(): void
+updateWorkerPosition(s: Square): void
+addPropertyChangeListener(pcl: PropertyChangeListener): void
+removePropertyChangeListener(pcl: PropertyChangeListener): void
+isMovingOn(): void
+isMovingOff(): void
«enumeration»
Color
WHITE
GREY
AZURE
NoWinPerimeter
+checkWin(w: Worker): boolean
EndRemoveNeighbour
+endOfTurn(workers: ArrayList <Worker>): boolean
-replaceAndBuild(w: Worker): void
MovePush
+checkMove(w: Worker, s: Square): void
+updateMove(w: Worker, s: Square): void
25
-table
1
-currentSquare
1
-worker
-board
1
-gameBoard
1
2
-workers
GodFactory
-document: Document
-bmu_god: BlockMoveUp
+getGods(ids: ArrayList <God>): ArrayList <God>
-getPowers(powers: Stack <Integer>, temp: Power): Power
-decorateOtherGods(gods: ArrayList <God>, applyToAll: Map): void
-getDocument(): Document
-evaluateXPath(xpathExpression: String): List <String>
God
-name: String
-description: String
+setPower(p: Power): void
+getName(): String
+getDescription(): String
+getPower(): Power
1
-god
1
Player
-username: String
+setGod(g: God): void
+getUsername(): String
+getWorkers(): ArrayList <Worker>
+getGod(): God
+canMoveWorker(w: Worker): boolean
+canMove(): boolean
+canBuildWorker(w: Worker): boolean
+move(w: Worker, s: Square): boolean
+build(w: Worker, s: Square, l: int): boolean
+endTurn(): boolean
+sendErrorMsg(errors: ArrayList <Error>): void
+propertyChange(PropertyChangeEvent evt): void
«enumeration»
Error
ISNT_WORKER_CHOSEN
NOT_ADJACENT
INVALID_LEVEL_MOVE
INVALID_LEVEL_BUILD
NOT_FREE
IS_DOME
MOVE_AFTER_BUILD
BUILD_BEFORE_MOVE
BUILDS_EXCEEDED
MOVES_EXCEEDED
BLOCK_MOVE_UP
CANT_MOVE_UP
CANT_DOME_UNDERFOOT
SAME_DIRECTION_NOT_FREE
EXTRA_MOVE_NOT_BACK
EXTRA_BUILD_NOT_DOME
EXTRA_BUILD_NOT_SAME_SPACE
EXTRA_BUILD_ONLY_SAME_SPACE
EXTRA_BUILD_NOT_PERIMETER
ExtraBuild
-onlySameSpace: boolean
-notSameSpace: boolean
-notPerimeter: boolean
+checkBuild(w: Worker, s: Square, l: int): ArrayList <Error>
-power
1
«interface»
Power
+checkMove(w: Worker, s: Square): ArrayList <Error>
+checkBuild(w: Worker, s: Square, l: int): ArrayList <Error>
+checkWin(): boolean
+updateMove(w: Worker, s: Square): void
+updateBuild(w: Worker, s: Space, l: int): void
+getBoard(): Board
+endOfTurn(workers: ArrayList <Worker>): boolean
#decoratedPower
1
BuildUnderfoot
+checkBuild(w: Worker, s: Square, l: int): ArrayList <Error>
BlockMoveUp
-support: PropertyChangeSupport
+updateMove(w: Worker, s: Square): void
+addPropertyChangeListener(pcl: PropertyChangeListener)
BuildDomeEverywhere
+checkBuild(w: Worker, s: Square, l: int): ArrayList <Error>
StandardPower
-maxMoves: Integer
-maxBuild: Integer
+checkMove(w: Worker, s: Square): ArrayList <Error>
+checkBuild(w: Worker, s: Square, l: int): ArrayList <Error>
+checkWin(w: Worker): boolean
+updateMove(w: Worker, s: Square): void
+updateBuild(w: Worker, s: Square, l: int): void
+getBoard(): Board
+endOfTurn(workers: ArrayList <Worker>): boolean
PowerDecorator
+checkMove(w: Worker, s: Square): ArrayList <Error>
+checkBuild(w: Worker, s: Square): ArrayList <Error>
+checkWin(): boolean
+updateMove(w: Worker, s: Square): void
+updateBuild(w: Worker, s: Square): void
+getBoard(): Board
+endOfTurn(workers: ArrayList <Worker>): boolean
BuildBeforeMove
+checkMove(w: Worker, s: Square): boolean
+checkBuild(w: Worker, s: Square, l: int): boolean
ExtraMove
-notMoveBack: boolean
-notPerimeter: boolean
+checkMove(w: Worker, s: Square): ArrayList <Error>
CanMoveUp
+moveUp: boolean
+checkMove(w: Worker, s: Square): ArrayList <Error>
+propertyChange(PropertyChangeEvent evt): void
MoveSwap
+checkMove(w: Worker, s: Square): ArrayList<Error>
+updateMove(w: Worker, s: Square): void
Down2Win
+checkWin(w: Worker): boolean