SetupManager

+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type

+ operation1(params):returnType
- operation2(params)
- operation3()

<<Enumeration>>
TurnPhase

START_TURN
MOVE_OR_BUILD
MOVE
BUILD
END_TURN

TurnManager

+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type

+ operation1(params):returnType
- operation2(params)
- operation3()

Game

+ state:GameState
+ turnManager:TurnManager
- gameOver:bool

+ run()

<<abstract>>
Action

- worker:Worker
- targetCell:Cell
- activate: Boolean
- currentPhase: TurnPhase
- nextPhase: TurnPhase

+ execute(params): None
+ getNextPhase(): List<Action>
+ activate(): None
+ deactivate(): None

MoveAction

- worker:Worker
- targetCell:Cell
-prevCell: Cell

+ operation1(params):returnType
- operation2(params)
- operation3()

BuildAction

+ worker:Worker
+ targetCell:Cell
- build:type

+ operation1(params):returnType
- operation2(params)
- operation3()

Cell

- x : int
- y : int
- tower : List<Block>
- occupant: Worker

+ getPostion
+ getLevel(): int
+ buildBlock(level: int) : None
+ isComplete(): Boolean
+ buildDome(): None
+ getOccupant(): Worker
+setOccupant(worker: Worker): None
+clearOccupant(): None
+ isOccupied(): Boolean

Board

- cells : List<Cell>

+ getCell(x: int, y: int): Cell
+ getMovableCellsFrom(currentCell: Cell):List<Cell>
+ getBuildableCellsFrom(currentCell: Cell): List<Cell>

GameState

- board:Board
- players:List<Player>

+ applyAction(agentIndex, action)
- isWin()

1
1
1
1

Block

-level: int
- isDome: boolean

0..*

Player

- name:String
- workers:List<Workers>
- godCard:GodCard

+ getName():String
+ getWorkers(): List<Worker>
+ getWorkerByID(id): Worker
+ getGodCard(): GodCard
+allowableAction(gameState: GameState)

Worker

- position: Cell
- owner: Player

+ getPosition():returnType
- allowableAction(gameState: GameState)
- operation3()

<<abstract>>
GodCard

- name: String
- numPlayer:List<Int>
- description: String

+ getName():String
- getDescription(): String
+setup():
+ modifyActions(player: Player, gameState: GameState, List<Action> baseActions): List<Action>
+checkWin(): boolean

GodCardFactory

+ godCardRegistry: Map<String, Supplier<God>>

+ registerGod(name: String, constructor: Supplier<God>)
+ createGod(String name): God
+ getAvailalbleGodCard(numPlayer: Int): List<String>