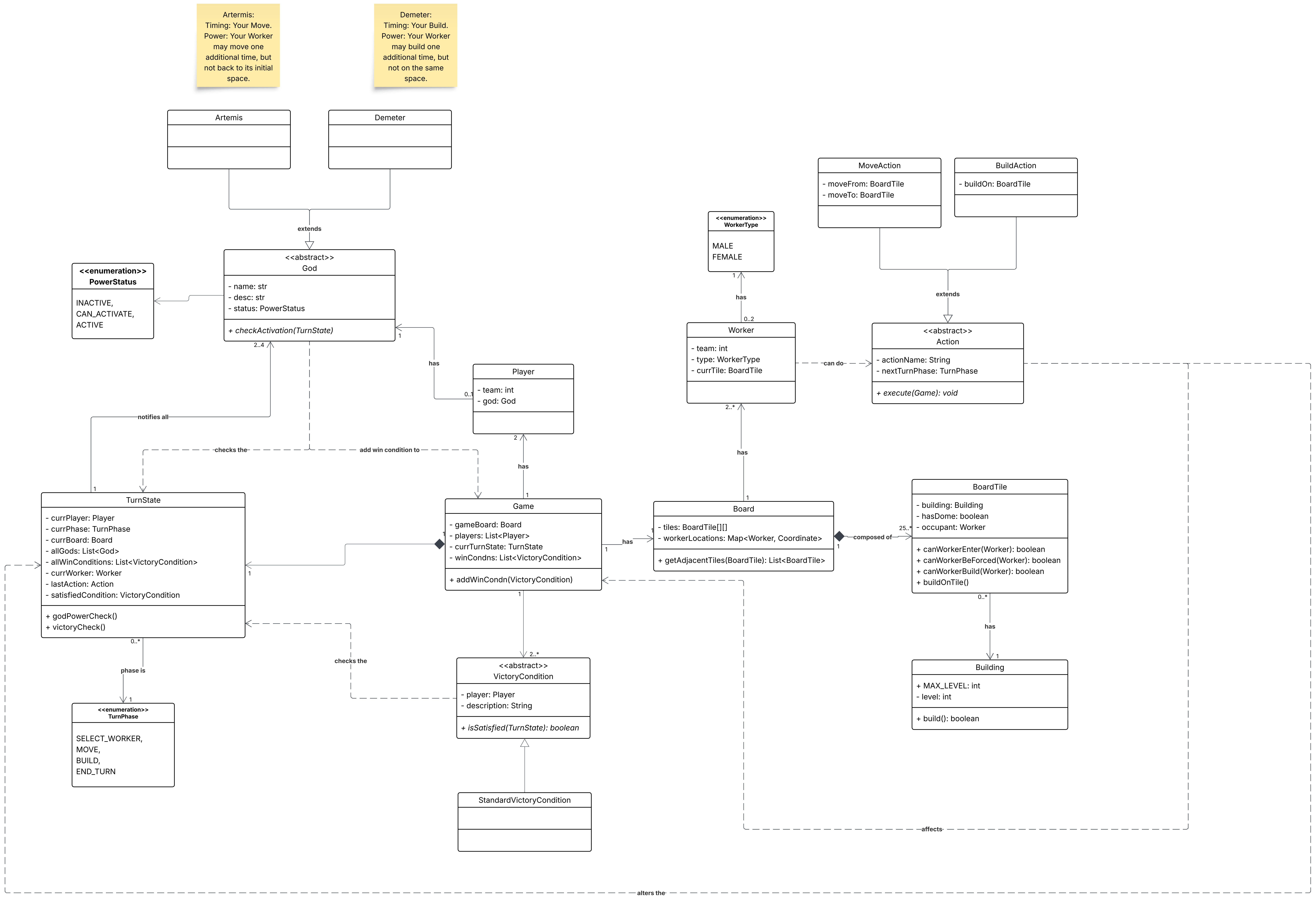
Artemis: Timing: Your Move. Power: Your Worker may move one additional time, but not back to its initial space.
Demeter: Timing: Your Build. Power: Your Worker may build one additional time, but not on the same space.
Artemis
Demeter
extends
<<abstract>> God
- name: str
- desc: str
- status: PowerStatus
+ checkActivation(TurnState)
<<enumeration>> PowerStatus
INACTIVE, CAN_ACTIVATE, ACTIVE
2..4
1
has
0..1
Player
- team: int
- god: God
2
has
1
notifies all
checks the
add win condition to
1
TurnState
- currPlayer: Player
- currPhase: TurnPhase
- currBoard: Board
- allGods: List<God>
- allWinConditions: List<VictoryCondition>
- currWorker: Worker
- lastAction: Action
- satisfiedCondition: VictoryCondition
+ godPowerCheck()
+ victoryCheck()
0..*
phase is
1
<<enumeration>> TurnPhase
SELECT_WORKER, MOVE, BUILD, END_TURN
Game
- gameBoard: Board
- players: List<Player>
- currTurnState: TurnState
- winCondns: List<VictoryCondition>
+ addWinCondn(VictoryCondition)
1
1
checks the
2..*
<<abstract>> VictoryCondition
- player: Player
- description: String
+ isSatisfied(TurnState): boolean
StandardVictoryCondition
has
1
1
Board
- tiles: BoardTile[][]
- workerLocations: Map<Worker, Coordinate>
+ getAdjacentTiles(BoardTile): List<BoardTile>
has
2..*
Worker
- team: int
- type: WorkerType
- currTile: BoardTile
0..2
has
1
<<enumeration>> WorkerType
MALE
FEMALE
can do
MoveAction
- moveFrom: BoardTile
- moveTo: BoardTile
BuildAction
- buildOn: BoardTile
extends
<<abstract>> Action
- actionName: String
- nextTurnPhase: TurnPhase
+ execute(Game): void
composed of
1
25..*
BoardTile
- building: Building
- hasDome: boolean
- occupant: Worker
+ canWorkerEnter(Worker): boolean
+ canWorkerBeForced(Worker): boolean
+ canWorkerBuild(Worker): boolean
+ buildOnTile()
0..*
has
1
Building
+ MAX_LEVEL: int
- level: int
+ build(): boolean
affects
alters the