* Attributes
* roundsPlayed :int
* width: int
* height: int
* Methods
* + Boolean isLoadGame()
* + Board getBoard()
* + int getWidth()
* + int getHeight()
* Attributes
* //b :Board
* //villages: Set<Village>
* //tiles: Set<Tile>
* Methods
* + void newGame(ge :GameEngine, players :Set<Player>)
* + void setBoard(b: Board)
* + void findVillages(tiles: Set<Tile>)  
  + Set<Tile> getTiles()
* + void setPlayers(tiles: Set<Tile>)
* + Set<Player> getPlayers()
* + void beginTurn(p: Player)
* + void upgradeVillage(v: Village, newLevel: VillageType)
* + void takeoverTile(dest: Tile)
* + void checkSize(tiles: Set<Tile>, v: Village)
* + Player getActivePlayer()
* + void merge(first: Village, second: Village)
* + void removeVillage(v: Village)
* + void moveUnit(u: Unit, t: Tile)
* + boolean checkDestTile(dest: Tile, u: Unit)
* + void checkForTrample(path: Set<Tile>)
* Attributes
* myType: LandType
* //struct: Structure
* //vil: Village
* Methods
* + void addStruct(s: Structure)
* + void removeStruct()
* + void setLandType(lt: LandType)
* + LandType getLandType()
* + void setOwner(p: Player)
* + Player getOwner()
* + Set<Tile> getNeighbours()
* + Unit getUnit()
* + void upgradeStruct(vl: VillageLevel)
* + void getVillage()
* + boolean hasStructure()
* + void setUnit(u: Unit)
* + void removeUnit(u: Unit)
* Attributes
* myType: VillageType
* gold: int
* wood: int
* structTile: Tile
* // tiles: Set<Tile>
* // units: Set<Unit>
* areCultivating: boolean
* areBuilding: boolean
* Methods
* + Set<Tile> getTiles()
* + void delete()
* + void setStructTile(t: Tile)
* + Tile getStructTile()
* + void addTile(t: Tile)
* + void addTiles(tiles: Set<Tile>)
* + void removeTile(dest: Tile)
* + Set<Unit> getUnits()
* + void removeUnit(u: Unit)
* + void refresh()
* + void tombPhase(tiles: Set<Tile>)
* + void buildPhase(tiles: Set<Tile>)
* + void incomePhase(tiles: Set<Tile>)
* + void paymentPhase(tiles: Set<Tile>)
* + boolean areCultivating()
* + boolean areBuilding()
* + void addGold(gold: int)
* + void addWood()
* + void decreaseGold(gold: int)
* + int getGold()
* + void setResources(gold: int, wood: int)
* + void addResources(gold: int, wood: int)
* + void buildRoad(u: Unit)
* + int getWood()
* + void setVillageType(type: VillageType)
* + void decreaseWood(wood: int)
* + void upgradeUnit(u: Unit, newLevel: UnitType)
* + int[] getResources()
* + void checkForBreak(tiles: Set<Tile>)
* + Village checkForMerge()
* + Player getOwner()
* + void setTiles(tiles: Set<Tile>)
* + void newPath()
* + Set<Tile> findPath(start: Tile, dest: Tile)
* + void addUnit(u: Unit)
* + Set<Tile> getPath()
* Attributes
* //tiles: Set<Tile>
* Method
* + void addTile(t: Tile, x: int, y: int)
* + Set<Tile> getTiles()
* Attributes
* isActive: boolean
* //villages: Set<Village>
* Methods
* + void setActive()
* + Set<Village> getVillages()
* + void takeTurn()
* + removeVillage(v: Village)
* + void incrementLosses()
* + void incrementWins()
* + void addVillage(v: Village)
* Attributes
* - myType: UnitType
* - currentAction: ActionType
* - //vil: Village
* Methods
* + Tile getTile()
* + ActionType getActionType()
* + void setActionType(at: ActionType)
* + UnitType getUnitType()
* + int getSalary()
* + void setUnitType(ut: UnitType)
* + Village getVillage()
* + void setVillage(v: Village)
* + Player getOwner()

Attributes

- myType: StructureType

Methods

+ void upgrade(VillageType)

* Attributes
* timestamp: Time
* message: String
* Attributes
* now: Time