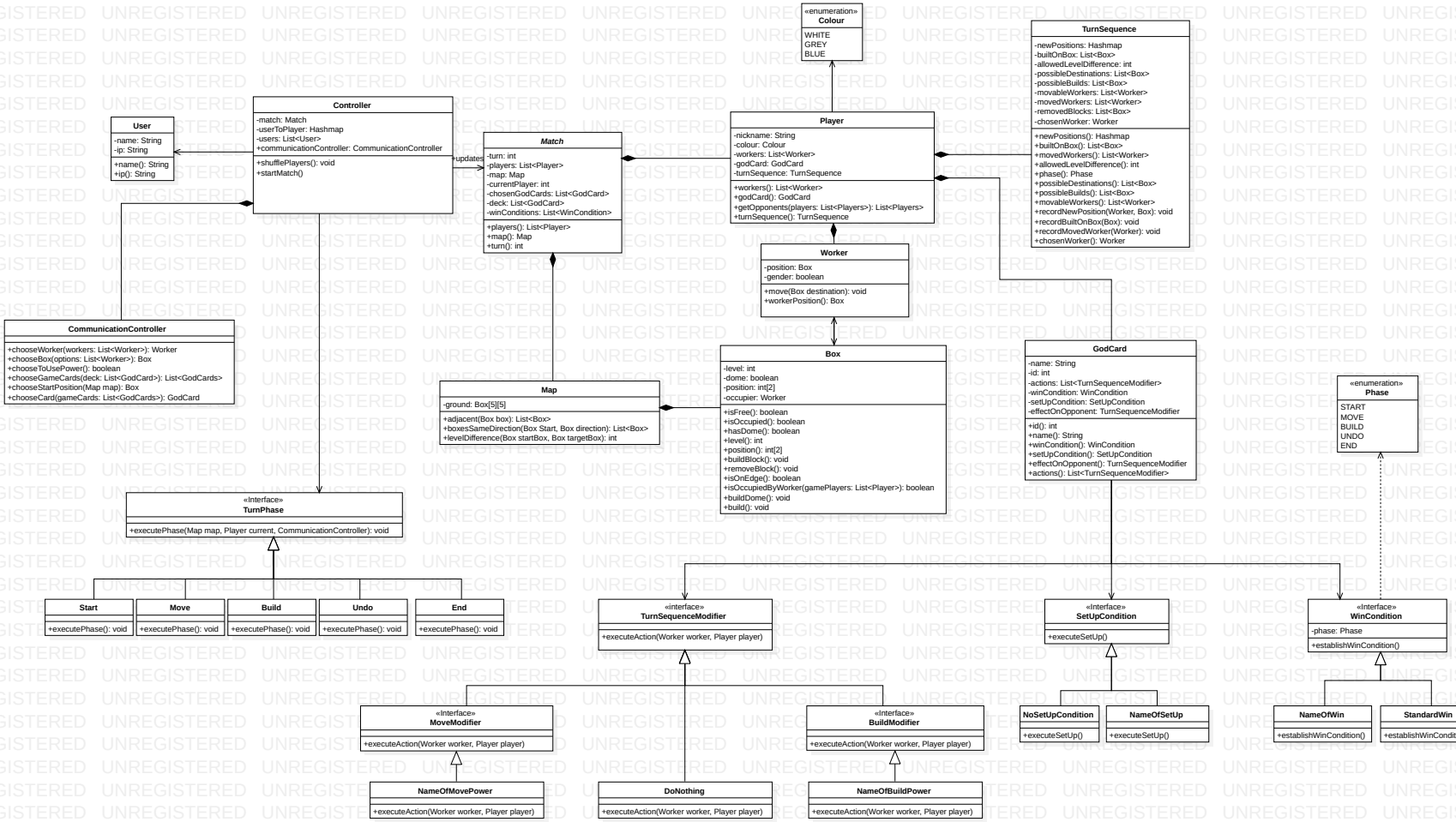
# Model.Main


«enumeration»
Colour
WHITE
GREY
BLUE
TurnSequence
-newPositions: Hashmap
-builtOnBox: List<Box>
-allowedLevelDifference: int
-possibleDestinations: List<Box>
-possibleBuilds: List<Box>
-movableWorkers: List<Worker>
-movedWorkers: List<Worker>
-removedBlocks: List<Box>
-chosenWorker: Worker
+newPositions(): Hashmap
+builtOnBox(): List<Box>
+movedWorkers(): List<Worker>
+allowedLevelDifference(): int
+phase(): Phase
+possibleDestinations(): List<Box>
+possibleBuilds(): List<Box>
+movableWorkers(): List<Worker>
+recordNewPosition(Worker, Box): void
+recordBuiltOnBox(Box): void
+recordMovedWorker(Worker): void
+chosenWorker(): Worker
Controller
-match: Match
-userToPlayer: Hashmap
-users: List<User>
+communicationController: CommunicationController
+shufflePlayers(): void
+startMatch()
User
-name: String
-ip: String
+name(): String
+ip(): String
+updates
Match
-turn: int
-players: List<Player>
-map: Map
-currentPlayer: int
-chosenGodCards: List<GodCard>
-deck: List<GodCard>
-winCondition: List<WinCondition>
+players(): List<Player>
+map(): Map
+turn(): int
Player
-nickname: String
-colour: Colour
-workers: List<Worker>
-godCard: GodCard
-turnSequence: TurnSequence
+workers(): List<Worker>
+godCard(): GodCard
+getOpponents(players: List<Players>): List<Players>
+turnSequence(): TurnSequence
Worker
-position: Box
-gender: boolean
+move(Box destination): void
+workerPosition(): Box
CommunicationController
+chooseWorker(workers: List<Worker>): Worker
+chooseBox(options: List<Worker>): Box
+chooseToUsePower(): boolean
+chooseGameCards(deck: List<GodCard>): List<GodCards>
+chooseStartPosition(Map map): Box
+chooseCard(gameCards: List<GodCards>): GodCard
Map
-ground: Box[5][5]
+adjacent(Box box): List<Box>
+boxesSameDirection(Box Start, Box direction): List<Box>
+levelDifference(Box startBox, Box targetBox): int
Box
-level: int
-dome: boolean
-position: int[2]
-occupier: Worker
+isFree(): boolean
+isOccupied(): boolean
+hasDome(): boolean
+level(): int
+position(): int[2]
+buildBlock(): void
+removeBlock(): void
+isOnEdge(): boolean
+isOccupiedByWorker(gamePlayers: List<Player>): boolean
+buildDome(): void
+build(): void
GodCard
-name: String
-id: int
-actions: List<TurnSequenceModifier>
-winCondition: WinCondition
-setUpCondition: SetUpCondition
-effectOnOpponent: TurnSequenceModifier
+id(): int
+name(): String
+winCondition(): WinCondition
+setUpCondition(): SetUpCondition
+effectOnOpponent(): TurnSequenceModifier
+actions(): List<TurnSequenceModifier>
«enumeration»
Phase
START
MOVE
BUILD
UNDO
END
«Interface»
TurnPhase
+executePhase(Map map, Player current, CommunicationController): void
Start
+executePhase(): void
Move
+executePhase(): void
Build
+executePhase(): void
Undo
+executePhase(): void
End
+executePhase(): void
«interface»
TurnSequenceModifier
+executeAction(Worker worker, Player player)
«Interface»
SetUpCondition
+executeSetUp()
«Interface»
WinCondition
-phase: Phase
+establishWinCondition()
«Interface»
MoveModifier
+executeAction(Worker worker, Player player)
«Interface»
BuildModifier
+executeAction(Worker worker, Player player)
NoSetUpCondition
+executeSetUp()
NameOfSetUp
+executeSetUp()
NameOfWin
+establishWinCondition()
StandardWin
+establishWinCondition()
NameOfMovePower
+executeAction(Worker worker, Player player)
DoNothing
+executeAction(Worker worker, Player player)
NameOfBuildPower
+executeAction(Worker worker, Player player)