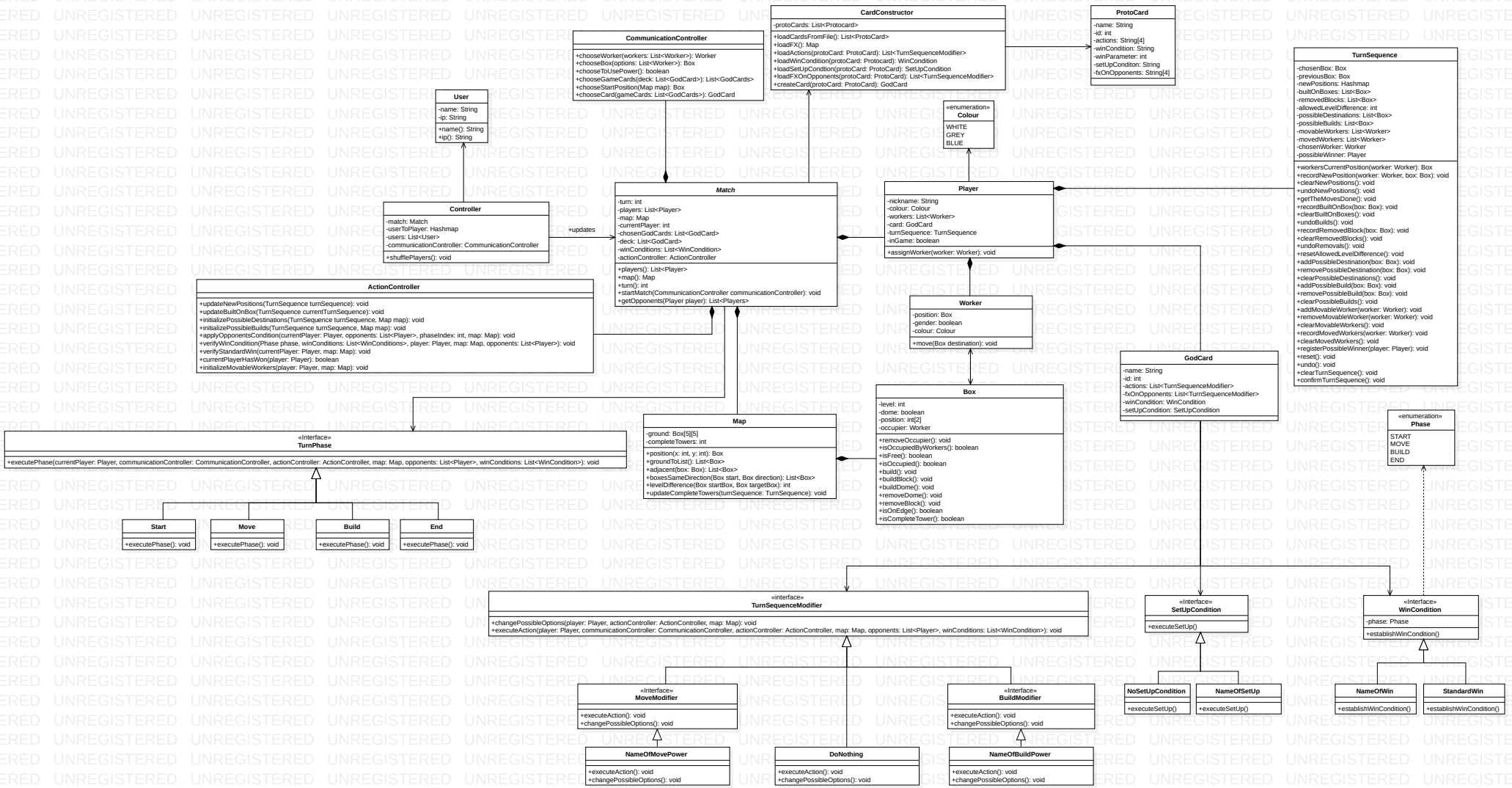
Model.main


CardConstructor
-protoCards: List<Protocard>
+loadCardsFromFile(): List<ProtoCard>
+loadFX(): Map
+loadActions(protoCard: ProtoCard): List<TurnSequenceModifier>
+loadWinCondition(protoCard: Protocard): WinCondition
+loadSetUpCondtion(protoCard: ProtoCard): SetUpCondition
+loadFXOnOpponents(protoCard: ProtoCard): List<TurnSequenceModifier>
+createCard(protoCard: ProtoCard): GodCard
ProtoCard
-name: String
-id: int
-actions: String[4]
-winCondition: String
-winParameter: int
-setUpCondition: String
-fxOnOpponents: String[4]
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
User
-name: String
-ip: String
+name(): String
+ip(): String
«enumeration»
Colour
WHITE
GREY
BLUE
TurnSequence
-chosenBox: Box
-previousBox: Box
-newPositions: Hashmap
-builtOnBoxes: List<Box>
-removedBlocks: List<Box>
-allowedLevelDifference: int
-possibleDestinations: List<Box>
-possibleBuilds: List<Box>
-movableWorkers: List<Worker>
-movedWorkers: List<Worker>
-chosenWorker: Worker
-possibleWinner: Player
+workersCurrentPosition(worker: Worker): Box
+recordNewPosition(worker: Worker, box: Box): void
+clearNewPositions(): void
+undoNewPositions(): void
+getTheMovesDone(): void
+recordBuiltOnBox(box: Box): void
+clearBuiltOnBoxes(): void
+undoBuilds(): void
+recordRemovedBlock(box: Box): void
+clearRemovedBlocks(): void
+undoRemovals(): void
+resetAllowedLevelDifference(): void
+addPossibleDestination(box: Box): void
+removePossibleDestination(box: Box): void
+clearPossibleDestinations(): void
+addPossibleBuild(box: Box): void
+removePossibleBuild(box: Box): void
+clearPossibleBuilds(): void
+addMovableWorker(worker: Worker): void
+removeMovableWorker(worker: Worker): void
+clearMovableWorkers(): void
+recordMovedWorkers(worker: Worker): void
+clearMovedWorkers(): void
+registerPossibleWinner(player: Player): void
+reset(): void
+undo(): void
+clearTurnSequence(): void
+confirmTurnSequence(): void
Controller
-match: Match
-userToPlayer: Hashmap
-users: List<User>
-communicationController: CommunicationController
+shufflePlayers(): void
+updates
Match
-turn: int
-players: List<Player>
-map: Map
-currentPlayer: int
-chosenGodCards: List<GodCard>
-deck: List<GodCard>
-winConditions: List<WinCondition>
-actionController: ActionController
+players(): List<Player>
+map(): Map
+turn(): int
+startMatch(CommunicationController communicationController): void
+getOpponents(Player player): List<Player>
Player
-nickname: String
-colour: Colour
-workers: List<Worker>
-card: GodCard
-turnSequence: TurnSequence
-inGame: boolean
+assignWorker(worker: Worker): void
ActionController
+updateNewPositions(TurnSequence turnSequence): void
+updateBuiltOnBox(TurnSequence currentTurnSequence): void
+initializePossibleDestinations(TurnSequence turnSequence, Map map): void
+initializePossibleBuilds(TurnSequence turnSequence, Map map): void
+applyOpponentsCondition(currentPlayer: Player, opponents: List<Player>, phaseIndex: int, map: Map): void
+verifyWinCondition(Phase phase, winConditions: List<WinConditions>, player: Player, map: Map, opponents: List<Player>): void
+verifyStandardWin(currentPlayer: Player, map: Map): void
+currentPlayerHasWon(player: Player): boolean
+initializeMovableWorkers(player: Player, map: Map): void
Worker
-position: Box
-gender: boolean
-colour: Colour
+move(Box destination): void
GodCard
-name: String
-id: int
-actions: List<TurnSequenceModifier>
-fxOnOpponents: List<TurnSequenceModifier>
-winCondition: WinCondition
-setUpCondition: SetUpCondition
Box
-level: int
-dome: boolean
-position: int[2]
-occupier: Worker
+removeOccupier(): void
+isOccupiedByWorkers(): boolean
+isFree(): boolean
+isOccupied(): boolean
+build(): void
+buildBlock(): void
+buildDome(): void
+removeDome(): void
+removeBlock(): void
+isOnEdge(): boolean
+isCompleteTower(): boolean
Map
-ground: Box[5][5]
-completeTowers: int
+position(x: int, y: int): Box
+groundToList(): List<Box>
+adjacent(box: Box): List<Box>
+boxesSameDirection(Box start, Box direction): List<Box>
+levelDifference(Box startBox, Box targetBox): int
+updateCompleteTowers(turnSequence: TurnSequence): void
«Interface»
TurnPhase
+executePhase(currentPlayer: Player, communicationController: CommunicationController, actionController: ActionController, map: Map, opponents: List<Player>, winConditions: List<WinCondition>): void
«enumeration»
Phase
START
MOVE
BUILD
END
Start
+executePhase(): void
Move
+executePhase(): void
Build
+executePhase(): void
End
+executePhase(): void
«interface»
TurnSequenceModifier
+changePossibleOptions(player: Player, actionController: ActionController, map: Map): void
+executeAction(player: Player, communicationController: CommunicationController, actionController: ActionController, map: Map, opponents: List<Player>, winConditions: List<WinCondition>): void
«Interface»
SetUpCondition
+executeSetUp()
«Interface»
WinCondition
-phase: Phase
+establishWinCondition()
«Interface»
MoveModifier
+executeAction(): void
+changePossibleOptions(): void
«Interface»
BuildModifier
+executeAction(): void
+changePossibleOptions(): void
NoSetUpCondition
+executeSetUp()
NameOfSetUp
+executeSetUp()
NameOfWin
+establishWinCondition()
StandardWin
+establishWinCondition()
NameOfMovePower
+executeAction(): void
+changePossibleOptions(): void
DoNothing
+executeAction(): void
+changePossibleOptions(): void
NameOfBuildPower
+executeAction(): void
+changePossibleOptions(): void