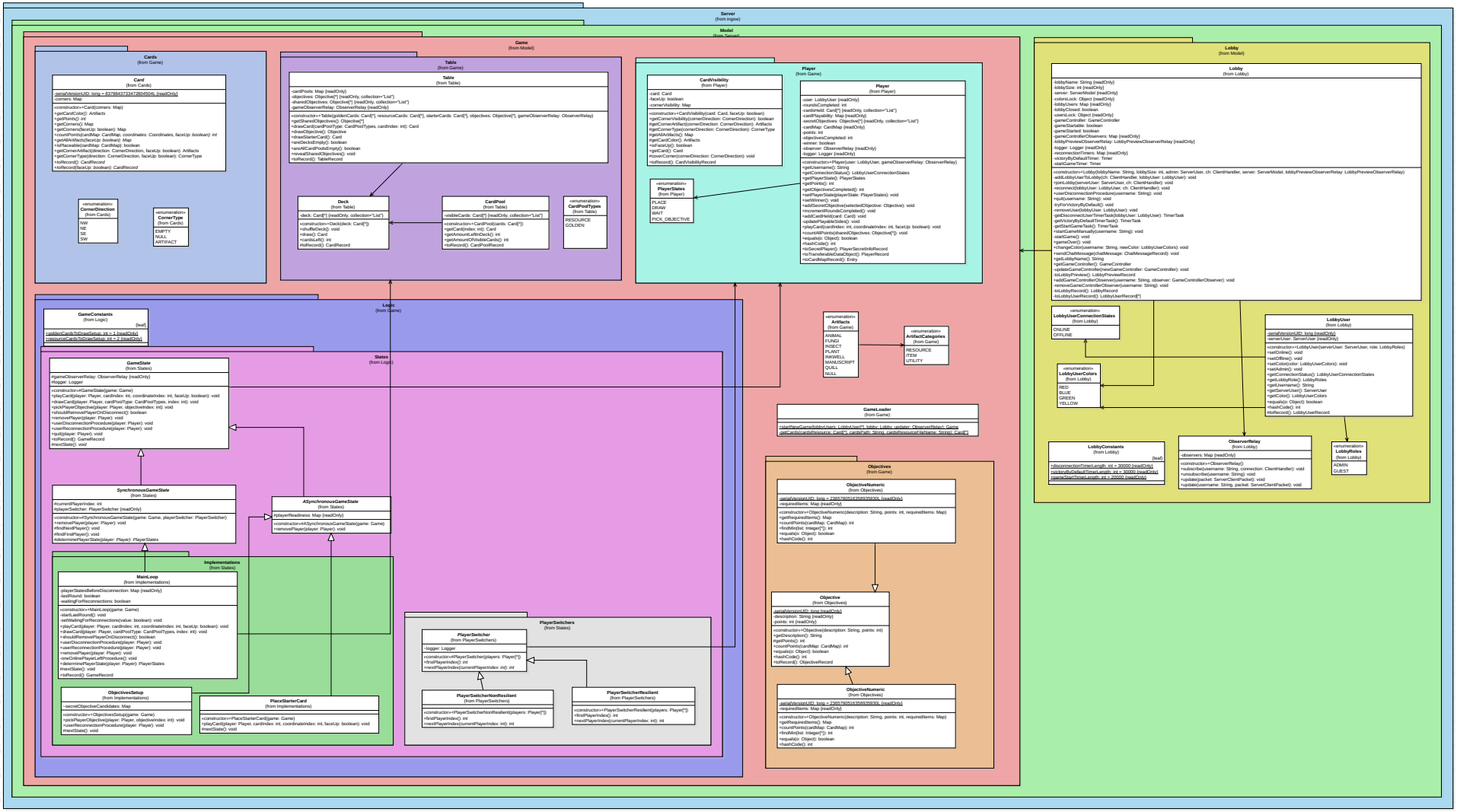
Server
(from ingsw)
Model
(from Server)
Game
(from Model)
Cards
(from Game)
Card
(from Cards)
-serialVersionUID: long = 8378643734728045045L (readOnly)
-corners: Map
+constructor(+Card(corners: Map))
+getCardColor(): Artifacts
+getPoints(): int
+getCorners(): Map
+getCorner(faceUp: boolean): Map
+countPoints(cardMap: CardMap, coordinates: Coordinates, faceUp: boolean): int
+getAllArtifacts(faceUp: boolean): Map
+isPlaceable(cardMap: CardMap): boolean
+getCornerArtifact(direction: CornerDirection, faceUp: boolean): Artifacts
+getCornerType(direction: CornerDirection, faceUp: boolean): CornerType
+toRecord(): CardRecord
+toRecord(faceUp: boolean): CardRecord
«enumeration»
CornerDirection
(from Cards)
NW
NE
SE
SW
«enumeration»
CornerType
(from Cards)
EMPTY
NULL
ARTIFACT
Table
(from Game)
Table
(from Table)
-cardPools: Map (readOnly)
-objectives: Objective[*] (readOnly, collection="List")
-sharedObjectives: Objective[*] (readOnly, collection="List")
-gameObserverRelay: ObserverRelay (readOnly)
+constructor+Table(goldenCards: Card[*], resourceCards: Card[*], starterCards: Card[*], objectives: Objective[*], gameObserverRelay: ObserverRelay)
+getSharedObjectives(): Objective[*]
+drawCard(cardPoolType: CardPoolTypes, cardIndex: int): Card
+drawObjective(): Objective
+drawStarterCard(): Card
+areDecksEmpty(): boolean
+areAllCardPoolsEmpty(): boolean
+revealCardsOnDisplay(): void
+toRecord(): TableRecord
Deck
(from Table)
-deck: Card[*] (readOnly, collection="List")
+constructor+Deck(deck: Card[*])
+shuffleDeck(): void
+draw(): Card
+cardsLeft(): int
+toRecord(): CardRecord
CardPool
(from Table)
-visibleCards: Card[*] (readOnly, collection="List")
+constructor+CardPool(cards: Card[*])
+getCard(index: int): Card
+getAmountLeftInDeck(): int
+getAmountOfVisibleCards(): int
+toRecord(): CardPoolRecord
«enumeration»
CardPoolTypes
(from Table)
RESOURCE
GOLDEN
Player
(from Game)
CardVisibility
(from Player)
-card: Card
-faceUp: boolean
-cornerVisibility: Map
+constructor+CardVisibility(card: Card, faceUp: boolean)
+getCornerVisibility(cornerDirection: CornerDirection): boolean
+getCornerArtifact(cornerDirection: CornerDirection): Artifacts
+getCornerType(cornerDirection: CornerDirection): CornerType
+getAllArtifacts(): Map
+getCardColor(): Artifacts
+isFaceUp(): boolean
+getCard(): Card
+toverCornerCornerDirection: CornerDirection): void
+toRecord(): CardVisibilityRecord
Player
(from Player)
-user: LobbyUser (readOnly)
-roundsCompleted: int
-cardsHeld: Card[*] (readOnly, collection="List")
-cardPlayability: Map (readOnly)
-secretObjectives: Objective[*] (readOnly, collection="List")
-cardMap: CardMap (readOnly)
-points: int
-objectivesCompleted: int
-winner: boolean
-observer: ObserverRelay (readOnly)
-logger: Logger (readOnly)
+constructor+Player(user: LobbyUser, gameObserverRelay: ObserverRelay)
+getUsername(): String
+getConnectionStatus(): LobbyUserConnectionStates
+getPlayerState(): PlayerStates
+getPoints(): int
+getObjectivesCompleted(): int
+setPlayerState(playerState: PlayerStates): void
+setWinner(): void
+addSecretObjective(selectedObjective: Objective): void
+incrementRoundsCompleted(): void
+addCardHeld(card: Card): void
-updatePlayableSides(): void
+playCard(cardIndex: int, coordinateIndex: int, faceUp: boolean): void
+countAllPoints(sharedObjectives: Objective[*]): void
+equals(o: Object): boolean
+hashCode(): int
+toSecretPlayer(): PlayerSecretInfoRecord
+toTransferableDataObject(): PlayerRecord
+toCardMapRecord(): Entry
«enumeration»
PlayerStates
(from Player)
PLACE
DRAW
WAIT
PICK_OBJECTIVE
Logic
(from Game)
GameConstants
(from Logic)
(leaf)
+goldenCardsToDrawSetup: int = 1 (readOnly)
+resourceCardsToDrawSetup: int = 2 (readOnly)
States
(from Logic)
GameState
(from States)
#gameObserverRelay: ObserverRelay (readOnly)
#logger: Logger
+constructor+GameState(game: Game)
+playCard(player: Player, cardIndex: int, coordinateIndex: int, faceUp: boolean): void
+drawCard(player: Player, cardPoolType: CardPoolTypes, index: int): void
+pickPlayerObjective(player: Player, objectiveIndex: int): void
+shouldRemovePlayerOnDisconnect(): boolean
+removePlayer(player: Player): void
+userDisconnectionProcedure(player: Player): void
+userReconnectionProcedure(player: Player): void
+quit(player: Player): void
+toRecord(): GameRecord
+nextState(): void
SynchronousGameState
(from States)
#currentPlayerIndex: int
#playerSwitcher: PlayerSwitcher (readOnly)
+constructor+SynchronousGameState(game: Game, playerSwitcher: PlayerSwitcher)
+removePlayer(player: Player): void
#findNextPlayer(): void
#findFirstPlayer(): void
#determinePlayerState(player: Player): PlayerStates
ASynchronousGameState
(from States)
#playerReadiness: Map (readOnly)
+constructor+ASynchronousGameState(game: Game)
+removePlayer(player: Player): void
Implementations
(from States)
MainLoop
(from Implementations)
-playerStatesBeforeDisconnection: Map (readOnly)
-lastRound: boolean
-waitingForReconnections: boolean
+constructor+MainLoop(game: Game)
-startLastRound(): void
-setWaitingForReconnections(value: boolean): void
+playCard(player: Player, cardIndex: int, coordinateIndex: int, faceUp: boolean): void
+drawCard(player: Player, cardPoolType: CardPoolTypes, index: int): void
+shouldRemovePlayerOnDisconnect(): boolean
+userDisconnectionProcedure(player: Player): void
+userReconnectionProcedure(player: Player): void
+removePlayer(player: Player): void
-oneOnlinePlayerLeftProcedure(): void
+determinePlayerState(player: Player): PlayerStates
+nextState(): void
+toRecord(): GameRecord
ObjectivesSetup
(from Implementations)
-secretObjectiveCandidates: Map
+constructor+ObjectivesSetup(game: Game)
+pickPlayerObjective(player: Player, objectiveIndex: int): void
+userReconnectionProcedure(player: Player): void
+nextState(): void
PlaceStarterCard
(from Implementations)
+constructor+PlaceStarterCard(game: Game)
+playCard(player: Player, cardIndex: int, coordinateIndex: int, faceUp: boolean): void
+nextState(): void
PlayerSwitchers
(from States)
PlayerSwitcher
(from PlayerSwitchers)
-logger: Logger
+constructor+PlayerSwitcher(players: Player[*])
+firstPlayerIndex(): int
+nextPlayerIndex(currentPlayerIndex: int): int
PlayerSwitcherNonResilient
(from PlayerSwitchers)
+constructor+PlayerSwitcherNonResilient(players: Player[*])
+firstPlayerIndex(): int
+nextPlayerIndex(currentPlayerIndex: int): int
PlayerSwitcherResilient
(from PlayerSwitchers)
+constructor+PlayerSwitcherResilient(players: Player[*])
+firstPlayerIndex(): int
+nextPlayerIndex(currentPlayerIndex: int): int
«enumeration»
Artifacts
(from Game)
ANIMAL
FUNGI
INSECT
PLANT
INKWELL
MANUSCRIPT
QUILL
NULL
«enumeration»
ArtifactCategories
(from Game)
RESOURCE
ITEM
UTILITY
GameLoader
(from Game)
+startNewGame(lobbyUsers: LobbyUser[*], lobby: Lobby, updater: ObserverRelay): Game
-setCardsLoad(cardsResource: Card[*], cardPath: String, cardsResourceFileName: String): Card[*]
Objectives
(from Game)
ObjectiveNumeric
(from Objectives)
-serialVersionUID: long = 2365780516398939L (readOnly)
-requiredItems: Map (readOnly)
+constructor+ObjectiveNumeric(description: String, points: int, requiredItems: Map)
+getRequiredItems(): Map
+countPoints(cardMap: CardMap): int
+findMeld(list: Integer[*]): int
+equals(o: Object): boolean
+hashCode(): int
Objective
(from Objectives)
-serialVersionUID: long (readOnly)
-description: String (readOnly)
-points: int (readOnly)
+constructor+Objective(description: String, points: int)
+getDescription(): String
+getPoints(): int
+countPoints(cardMap: CardMap): int
+equals(o: Object): boolean
+hashCode(): int
+toRecord(): ObjectiveRecord
ObjectiveNumeric
(from Objectives)
-serialVersionUID: long = 2365780516398939L (readOnly)
-requiredItems: Map (readOnly)
+constructor+ObjectiveNumeric(description: String, points: int, requiredItems: Map)
+getRequiredItems(): Map
+countPoints(cardMap: CardMap): int
+findMeld(list: Integer[*]): int
+equals(o: Object): boolean
+hashCode(): int
Lobby
(from Model)
Lobby
(from Lobby)
-lobbyName: String (readOnly)
-lobbySize: int (readOnly)
-server: ServerModel (readOnly)
-colorsLock: Object (readOnly)
-lobbyUsers: Map (readOnly)
-lobbyClosed: boolean
-usersLock: Object (readOnly)
-gameController: GameController
-gameStartable: boolean
-gameStarted: boolean
-gameControllerObservers: Map (readOnly)
-lobbyPreviewObserverRelay: LobbyPreviewObserverRelay (readOnly)
-logger: Logger (readOnly)
-reconnectionTimers: Map (readOnly)
-victoryByDefaultTimer: Timer
-startGameTimer: Timer
+constructor+Lobby(lobbyName: String, lobbySize: int, admin: ServerUser, ch: ClientHandler, server: ServerModel, lobbyPreviewObserverRelay: LobbyPreviewObserverRelay)
+addLobbyUserToLobby(ch: ClientHandler, lobbyUser: LobbyUser): void
+joinLobbyServerUser: ServerUser, ch: ClientHandler): void
+reconnect(lobbyUser: LobbyUser, ch: ClientHandler): void
+userDisconnectionProcedure(username: String): void
+quit(username: String): void
+isForVictoryByDefault(): void
+removeUser(lobbyUser: LobbyUser): void
-getDisconnectUserTimerTask(lobbyUser: LobbyUser): TimerTask
-getVictoryByDefaultTimerTask(): TimerTask
-getStartGameTask(): TimerTask
+startGameManually(username: String): void
-startGame(): void
+gameOver(): void
+changeColor(username: String, newColor: LobbyUserColors): void
+sendChatMessage(chatMessage: ChatMessageRecord): void
+getLobbyName(): String
+getGameController(): GameController
-updateGameControllerNewGameController: GameController): void
-toLobbyPreview(): LobbyPreviewRecord
+addGameControllerObserver(username: String, observer: GameControllerObserver): void
-removeGameControllerObserver(username: String): void
-toLobbyRecord(): LobbyRecord
-toLobbyUserRecord(): LobbyUserRecord[*]
«enumeration»
LobbyUserConnectionStates
(from Lobby)
ONLINE
OFFLINE
LobbyUser
(from Lobby)
-serialVersionUID: long (readOnly)
-serverUser: ServerUser (readOnly)
+constructor+LobbyUser(serverUser: ServerUser, role: LobbyRoles)
+setOnline(): void
+setOffline(): void
+setColor(color: LobbyUserColors): void
+setAdmin(): void
+getConnectionStatus(): LobbyUserConnectionStates
+getLobbyRole(): LobbyRoles
+getUsername(): String
+getServerUser(): ServerUser
+getColor(): LobbyUserColors
+equals(o: Object): boolean
+hashCode(): int
+toRecord(): LobbyUserRecord
«enumeration»
LobbyUserColors
(from Lobby)
RED
BLUE
GREEN
YELLOW
LobbyConstants
(from Lobby)
(leaf)
+disconnectionTimerLength: int = 30000 (readOnly)
+victoryByDefaultTimerLength: int = 30000 (readOnly)
+startGameTimerLength: int = 20000 (readOnly)
ObserverRelay
(from Lobby)
-observers: Map (readOnly)
+constructor+ObserverRelay()
+subscribe(username: String, connection: ClientHandler): void
+unsubscribe(username: String): void
+update(packet: ServerClientPacket): void
+update(username: String, packet: ServerClientPacket): void
«enumeration»
LobbyRoles
(from Lobby)
ADMIN
GUEST