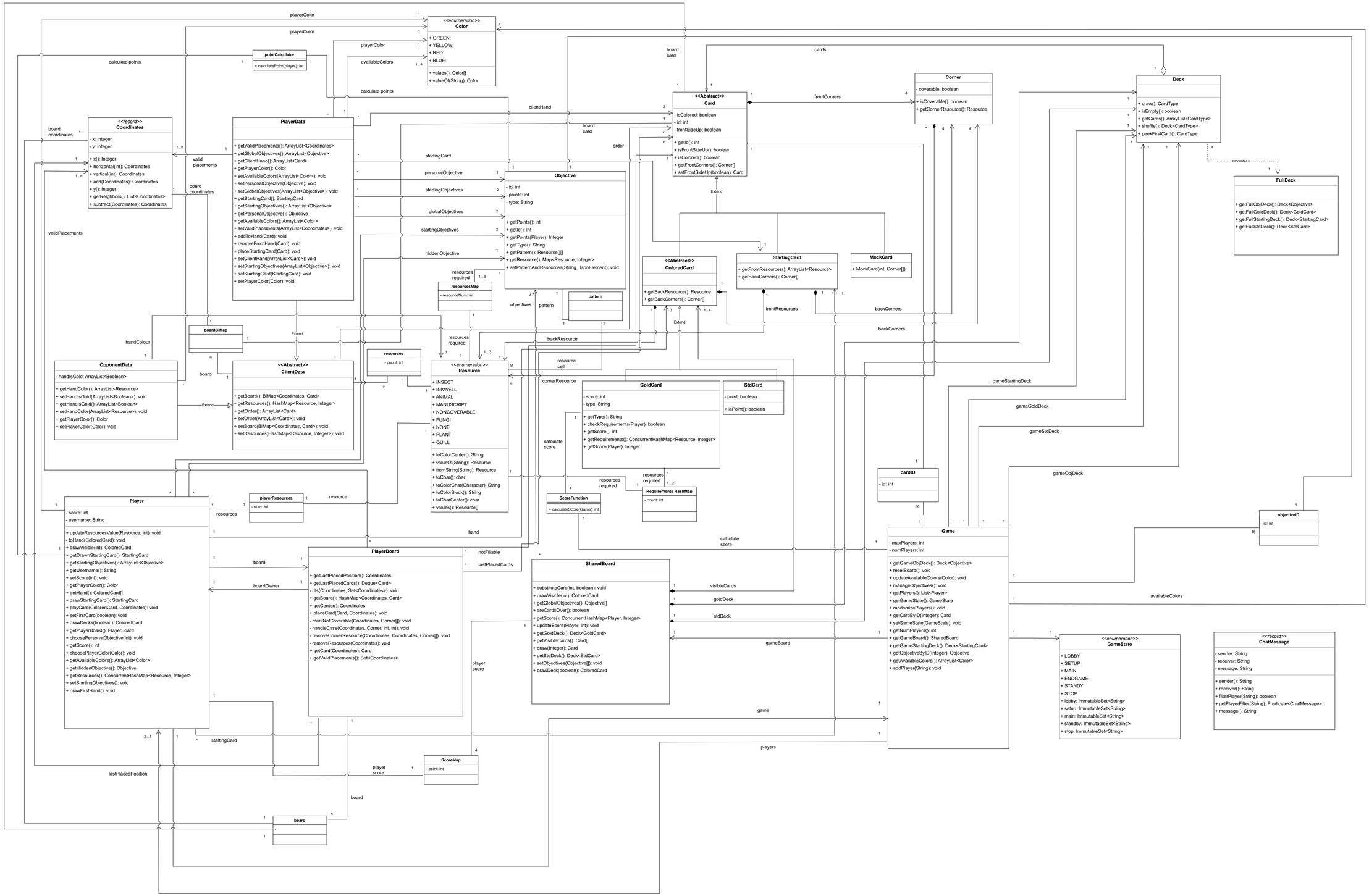
Color
+ GREEN:
+ YELLOW:
+ RED:
+ BLUE:
+ values(): Color[]
+ valueOf(String): Color
pointCalculator
+ calculatePoint(player): int
Coordinates
- x: Integer
- y: Integer
+ x(): Integer
+ horizontal(int): Coordinates
+ vertical(int): Coordinates
+ add(Coordinates): Coordinates
+ y(): Integer
+ getNeighbors(): List<Coordinates>
+ subtract(Coordinates): Coordinates
PlayerData
+ getValidPlacements(): ArrayList<Coordinates>
+ getGlobalObjectives(): ArrayList<Objective>
+ getClientHand(): ArrayList<Card>
+ getPlayerColor(): Color
+ setAvailableColors(ArrayList<Color>): void
+ setPersonalObjective(Objective): void
+ setGlobalObjectives(ArrayList<Objective>): void
+ getStartingCard(): StartingCard
+ getStartingObjectives(): ArrayList<Objective>
+ getPersonalObjective(): Objective
+ getAvailableColors(): ArrayList<Color>
+ setValidPlacements(ArrayList<Coordinates>): void
+ addToHand(Card): void
+ removeFromHand(Card): void
+ placeStartingCard(Card): void
+ setClientHand(ArrayList<Card>): void
+ setStartingObjectives(ArrayList<Objective>): void
+ setStartingCard(StartingCard): void
+ setPlayerColor(Color): void
Card
- isColored: boolean
- id: int
- frontSideUp: boolean
+ getId(): int
+ isFrontSideUp(): boolean
+ isColored(): boolean
+ getFrontCorners(): Corner[]
+ setFrontSideUp(boolean): Card
Corner
- coverable: boolean
+ isCoverable(): boolean
+ getCornerResource(): Resource
Deck
+ draw(): CardType
+ isEmpty(): boolean
+ getCards(): ArrayList<CardType>
+ shuffle(): Deck<CardType>
+ peekFirstCard(): CardType
FullDeck
+ getFullObjDeck(): Deck<Objective>
+ getFullGoldDeck(): Deck<GoldCard>
+ getFullStartingDeck(): Deck<StartingCard>
+ getFullStdDeck(): Deck<StdCard>
Objective
- id: int
- points: int
- type: String
+ getPoints(): int
+ getId(): int
+ getPoints(Player): Integer
+ getType(): String
+ getPattern(): Resource[][]
+ getResource(): Map<Resource, Integer>
+ setPatternAndResources(String, JsonElement): void
ColoredCard
+ getBackResource(): Resource
+ getBackCorners(): Corner[]
StartingCard
+ getFrontResources(): ArrayList<Resource>
+ getBackCorners(): Corner[]
MockCard
+ MockCard(int, Corner[]):
resourcesMap
- resourceNum: int
pattern
boardBiMap
OpponentData
- handIsGold: ArrayList<Boolean>
+ getHandColor(): ArrayList<Resource>
+ setHandIsGold(ArrayList<Boolean>): void
+ getHandIsGold(): ArrayList<Boolean>
+ setHandColor(ArrayList<Resource>): void
+ getPlayerColor(): Color
+ setPlayerColor(Color): void
ClientData
+ getBoard(): BiMap<Coordinates, Card>
+ getResources(): HashMap<Resource, Integer>
+ getOrder(): ArrayList<Card>
+ setOrder(ArrayList<Card>): void
+ setBoard(BiMap<Coordinates, Card>): void
+ setResources(HashMap<Resource, Integer>): void
resources
- count: int
Resource
+ INSECT
+ INKWELL
+ ANIMAL
+ MANUSCRIPT
+ NONCOVERABLE
+ FUNGI
+ NONE
+ PLANT
+ QUILL
+ toColorCenter(): String
+ valueOf(String): Resource
+ fromString(String): Resource
+ toChar(): char
+ toColorChar(Character): String
+ toColorBlock(): String
+ toCharCenter(): char
+ values(): Resource[]
GoldCard
- score: int
- type: String
+ getType(): String
+ checkRequirements(Player): boolean
+ getScore(): int
+ getRequirements(): ConcurrentHashMap<Resource, Integer>
+ getScore(Player): Integer
StdCard
- point: boolean
+ isPoint(): boolean
cardID
- id: int
ScoreFunction
+ calculateScore(Game): int
Requirements HashMap
- count: int
objectiveID
- id: int
Player
- score: int
- username: String
+ updateResourcesValue(Resource, int): void
- toHand(ColoredCard): void
+ drawVisible(int): ColoredCard
+ getDrawnStartingCard(): StartingCard
+ getStartingObjectives(): ArrayList<Objective>
+ getUsername(): String
+ setScore(int): void
+ getPlayerColor(): Color
+ getHand(): ColoredCard[]
+ drawStartingCard(): StartingCard
+ playCard(ColoredCard, Coordinates): void
+ setFirstCard(boolean): void
+ drawDecks(boolean): ColoredCard
+ getPlayerBoard(): PlayerBoard
+ choosePersonalObjective(int): void
+ getScore(): int
+ choosePlayerColor(Color): void
+ getAvailableColors(): ArrayList<Color>
+ getHiddenObjective(): Objective
+ getResources(): ConcurrentHashMap<Resource, Integer>
+ setStartingObjectives(): void
+ drawFirstHand(): void
playerResources
- num: int
PlayerBoard
+ getLastPlacedPosition(): Coordinates
+ getLastPlacedCards(): Deque<Card>
- dfs(Coordinates, Set<Coordinates>): void
+ getBoard(): HashMap<Coordinates, Card>
+ getCenter(): Coordinates
+ placeCard(Card, Coordinates): void
- markNotCoverable(Coordinates, Corner[]): void
- handleCase(Coordinates, Corner, int, int): void
- removeCornerResource(Coordinates, Coordinates, Corner[]): void
- removeResources(Coordinates): void
+ getCard(Coordinates): Card
+ getValidPlacements(): Set<Coordinates>
notFillable
lastPlacedCards
SharedBoard
+ substituteCard(int, boolean): void
+ drawVisible(int): ColoredCard
+ getGlobalObjectives(): Objective[]
+ areCardsOver(): boolean
+ getScore(): ConcurrentHashMap<Player, Integer>
+ updateScore(Player, int): void
+ getGoldDeck(): Deck<GoldCard>
+ getVisibleCards(): Card[]
+ draw(Integer): Card
+ getStdDeck(): Deck<StdCard>
+ setObjectives(Objective[]): void
+ drawDeck(boolean): ColoredCard
Game
- maxPlayers: int
- numPlayers: int
+ getGameObjDeck(): Deck<Objective>
+ resetBoard(): void
+ updateAvailableColors(Color): void
+ manageObjectives(): void
+ getPlayers(): List<Player>
+ getGameState(): GameState
+ randomizePlayers(): void
+ getCardByID(Integer): Card
+ setGameState(GameState): void
+ getNumPlayers(): int
+ getGameBoard(): SharedBoard
+ getGameStartingDeck(): Deck<StartingCard>
+ getObjectiveByID(Integer): Objective
+ getAvailableColors(): ArrayList<Color>
+ addPlayer(String): void
GameState
+ LOBBY
+ SETUP
+ MAIN
+ ENDGAME
+ STANDY
+ STOP
+ lobby: ImmutableSet<String>
+ setup: ImmutableSet<String>
+ main: ImmutableSet<String>
+ standby: ImmutableSet<String>
+ stop: ImmutableSet<String>
ChatMessage
- sender: String
- receiver: String
- message: String
+ sender(): String
+ receiver(): String
+ filterPlayer(String): boolean
+ getPlayerFilter(String): Predicate<ChatMessage>
+ message(): String
ScoreMap
- point: int
board
playerColor
availableColors
calculate points
clientHand
board card
cards
frontCorners
board coordinates
valid placements
startingCard
personalObjective
startingObjectives
globalObjectives
hiddenObjective
resources required
order
Extend
frontResources
backCorners
handColour
board
resource cell
backResource
cornerResource
calculate score
resource
hand
boardOwner
visibleCards
goldDeck
stdDeck
gameBoard
game
players
gameStartingDeck
gameGoldDeck
gameStdDeck
gameObjDeck
availableColors
lastPlacedPosition
player score
notFillable
lastPlacedCards