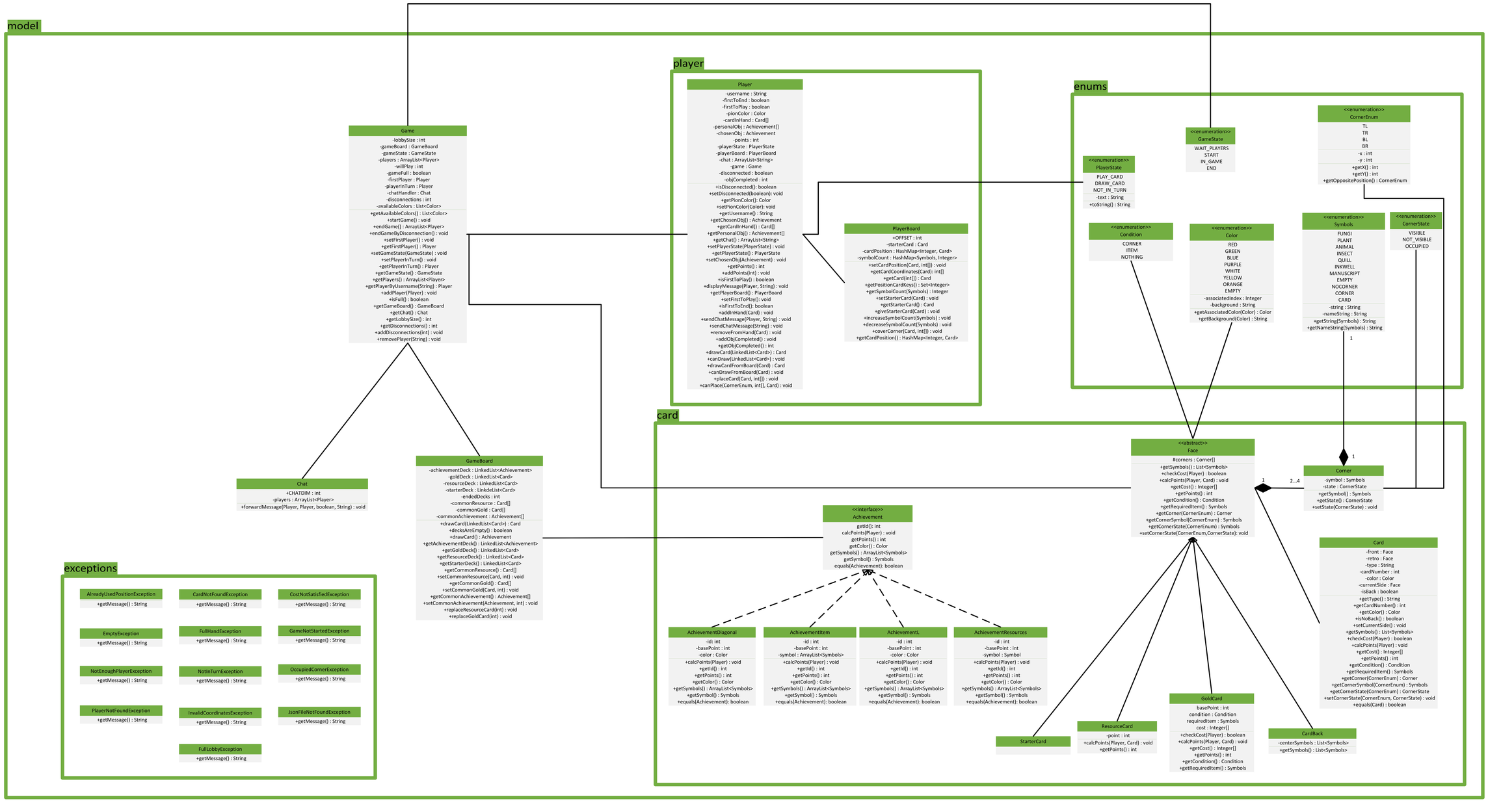
model
Game
-lobbySize : int
-gameBoard : GameBoard
-gameState : GameState
-players : ArrayList<Player>
-willPlay : int
-gameFull : boolean
-firstPlayer : Player
-playerInTurn : Player
-chatHandler : Chat
-disconnections : int
-availableColors : List<Color>
+getAvailableColors() : List<Color>
+startGame() : void
+endGame() : ArrayList<Player>
+endGameByDisconnection() : void
+setFirstPlayer() : void
+getFirstPlayer() : Player
+setGameState(GameState) : void
+setPlayerInTurn() : void
+getPlayerInTurn() : Player
+getGameState() : GameState
+getPlayers() : ArrayList<Player>
+getPlayerByUsername(String) : Player
+addPlayer(Player) : void
+isFull() : boolean
+getGameBoard() : GameBoard
+getChat() : Chat
+getLobbySize() : int
+getDisconnections() : int
+addDisconnections(int) : void
+removePlayer(String) : void
Chat
+CHATDIM : int
-players : ArrayList<Player>
+forwardMessage(Player, Player, boolean, String) : void
GameBoard
-achievementDeck : LinkedList<Achievement>
-goldDeck : LinkedList<Card>
-resourceDeck : LinkedList<Card>
-starterDeck : LinkedList<Card>
-endedDecks : int
-commonResource : Card[]
-commonGold : Card[]
-commonAchievement : Achievement[]
+drawCard(LinkedList<Card>) : Card
+decksAreEmpty() : boolean
+drawCard() : Achievement
+getAchievementDeck() : LinkedList<Achievement>
+getGoldDeck() : LinkedList<Card>
+getResourceDeck() : LinkedList<Card>
+getStarterDeck() : LinkedList<Card>
+getCommonResource() : Card[]
+setCommonResource(Card, int) : void
+getCommonGold() : Card[]
+setCommonGold(Card, int) : void
+getCommonAchievement() : Achievement[]
+setCommonAchievement(Achievement, int) : void
+replaceResourceCard(int) : void
+replaceGoldCard(int) : void
player
Player
-username : String
-firstToEnd : boolean
-firstToPlay : boolean
-pionColor : Color
-cardInHand : Card[]
-personalObj : Achievement[]
-chosenObj : Achievement
-points : int
-playerState : PlayerState
-playerBoard : PlayerBoard
-chat : ArrayList<String>
-game : Game
-disconnected : boolean
-objCompleted : int
+isDisconnected(): boolean
+setDisconnected(boolean): void
+getPionColor(): Color
+setPionColor(Color): void
+getUsername() : String
+getChosenObj() : Achievement
+getCardInHand() : Card[]
+getPersonalObj() : Achievement[]
+getChat() : ArrayList<String>
+setPlayerState(PlayerState) : void
+getPlayerState() : PlayerState
+setChosenObj(Achievement) : void
+getPoints() : int
+addPoints(int) : void
+isFirstToPlay() : boolean
+displayMessage(Player, String) : void
+getPlayerBoard() : PlayerBoard
+setFirstToPlay(): void
+isFirstToEnd(): boolean
+addInHand(Card) : void
+sendChatMessage(Player, String) : void
+sendChatMessage(String) : void
+removeFromHand(Card) : void
+addObjCompleted() : void
+getObjCompleted() : int
+drawCard(LinkedList<Card>) : Card
+canDraw(LinkedList<Card>) : void
+drawCardFromBoard(Card) : Card
+canDrawFromBoard(Card) : void
+placeCard(Card, int[]) : void
+canPlace(CornerEnum, int[], Card) : void
PlayerBoard
+OFFSET : int
-starterCard : Card
-cardPosition : HashMap<Integer, Card>
-symbolCount : HashMap<Symbols, Integer>
+setCardPosition(Card, int[]) : void
+getCardCoordinates(Card): int[]
+getCard(int[]) : Card
+getPositionCardKeys() : Set<Integer>
+getSymbolCount(Symbols) : Integer
+setStarterCard(Card) : void
+getStarterCard() : Card
+giveStarterCard(Card) : void
+increaseSymbolCount(Symbols) : void
+decreaseSymbolCount(Symbols) : void
+coverCorner(Card, int[]) : void
+getCardPosition() : HashMap<Integer, Card>
enums
<<enumeration>>
PlayerState
PLAY_CARD
DRAW_CARD
NOT_IN_TURN
-text : String
+toString() : String
<<enumeration>>
GameState
WAIT_PLAYERS
START
IN_GAME
END
<<enumeration>>
CornerEnum
TL
TR
BL
BR
-x : int
-y : int
+getX() : int
+getY() : int
+getOppositePosition() : CornerEnum
<<enumeration>>
Condition
CORNER
ITEM
NOTHING
<<enumeration>>
Color
RED
GREEN
BLUE
PURPLE
WHITE
YELLOW
ORANGE
EMPTY
-associatedIndex : Integer
-background : String
+getAssociatedColor(Color) : Color
+getBackground(Color) : String
<<enumeration>>
Symbols
FUNGI
PLANT
ANIMAL
INSECT
QUILL
INKWELL
MANUSCRIPT
EMPTY
NOCORNER
CORNER
CARD
-string : String
-nameString : String
+getString(Symbols) : String
+getNameString(Symbols) : String
<<enumeration>>
CornerState
VISIBLE
NOT_VISIBLE
OCCUPIED
card
<<abstract>>
Face
#corners : Corner[]
+getSymbols() : List<Symbols>
+checkCost(Player) : boolean
+calcPoints(Player, Card) : void
+getCost() : Integer[]
+getPoints() : int
+getCondition() : Condition
+getRequiredItem() : Symbols
+getCorner(CornerEnum) : Corner
+getCornerSymbol(CornerEnum) : Symbols
+getCornerState(CornerEnum) : Symbols
+setCornerState(CornerEnum,CornerState): void
Corner
-symbol : Symbols
-state : CornerState
+getSymbol() : Symbols
+getState() : CornerState
+setState(CornerState) : void
1
2...4
1
1
<<interface>>
Achievement
getId(): int
calcPoints(Player) : void
getPoints() : int
getColor() : Color
getSymbols() : ArrayList<Symbols>
getSymbol() : Symbols
equals(Achievement): boolean
AchievementDiagonal
-id: int
-basePoint : int
-color : Color
+calcPoints(Player) : void
+getId() : int
+getPoints() : int
+getColor() : Color
+getSymbols() : ArrayList<Symbols>
+getSymbol() : Symbols
+equals(Achievement): boolean
AchievementItem
-id : int
-basePoint : int
-symbol : ArrayList<Symbols>
+calcPoints(Player) : void
+getId() : int
+getPoints() : int
+getColor() : Color
+getSymbols() : ArrayList<Symbols>
+getSymbol() : Symbols
+equals(Achievement): boolean
AchievementL
-id : int
-basePoint : int
-color : Color
+calcPoints(Player) : void
+getId() : int
+getPoints() : int
+getColor() : Color
+getSymbols() : ArrayList<Symbols>
+getSymbol() : Symbols
+equals(Achievement): boolean
AchievementResources
-id : int
-basePoint : int
-symbol : Symbol
+calcPoints(Player) : void
+getId() : int
+getPoints() : int
+getColor() : Color
+getSymbols() : ArrayList<Symbols>
+getSymbol() : Symbols
+equals(Achievement): boolean
StarterCard
ResourceCard
-point : int
+calcPoints(Player, Card) : void
+getPoints() : int
GoldCard
basePoint : int
condition : Condition
requiredItem : Symbols
cost : Integer[]
+checkCost(Player) : boolean
+calcPoints(Player, Card) : void
+getCost() : Integer[]
+getPoints() : int
+getCondition() : Condition
+getRequiredItem() : Symbols
CardBack
-centerSymbols : List<Symbols>
+getSymbols() : List<Symbols>
Card
-front : Face
-retro : Face
-type : String
-cardNumber : int
-color : Color
-currentSide : Face
-isBack : boolean
+getType() : String
+getCardNumber() : int
+getColor() : Color
+isNoBack() : boolean
+setCurrentSide() : void
+getSymbols() : List<Symbols>
+checkCost(Player) : boolean
+calcPoints(Player) : void
+getCost() : Integer[]
+getPoints() : int
+getCondition() : Condition
+getRequiredItem() : Symbols
+getCorner(CornerEnum) : Corner
+getCornerSymbol(CornerEnum) : Symbols
+getCornerState(CornerEnum) : CornerState
+setCornerState(CornerEnum, CornerState) : void
+equals(Card) : boolean
exceptions
AlreadyUsedPositionException
+getMessage() : String
CardNotFoundException
+getMessage() : String
CostNotSatisfiedException
+getMessage() : String
EmptyException
+getMessage() : String
FullHandException
+getMessage() : String
GameNotStartedException
+getMessage() : String
NotEnoughPlayerException
+getMessage() : String
NotInTurnException
+getMessage() : String
OccupiedCornerException
+getMessage() : String
PlayerNotFoundException
+getMessage() : String
InvalidCoordinatesException
+getMessage() : String
JsonFileNotFoundException
+getMessage() : String
FullLobbyException
+getMessage() : String