pkg(model)

ScoreTrack
- Num_player : int
+ ScoreTrack(name1 : String, name2 : String)
+ ScoreTrack(nome1 : String, nome2 : String, nome3 : String)
+ ScoreTrack(nome1 : int, nome2 : int, nome3 : int, nome4 : int)
+ updateScoreTrack(name : String, POINTS : int) : void
+ printTable() : void

Game
+ numPlayers : int
+ currentPlayerIndex : int
+ startGame(usernames : String) : void
+ dealSecretObjective(player : Player) : ObjectiveCard[]
+ placeSecretObjective(player : Player, secretObjective : ObjectiveCard) : void
+ drawCard(playerName : String, message : Message) : void
+ playCard(playerName : String, message : int) : void
- calculateFinalPoints() : void
- declare_winner() : void
+ get_next_winner_name() : String

1   1   isTracked
2..4   + players
1
3

pkg(player)

<<enum>>
color
+ red : color
+ green : color
+ blue : color
+ black : color

<<definition>>
Private class used to memorize the cards placed by the player and the Points on which they are placed

PlacedCards
- nextIndex : int
+ put(p : Point, card : PlayableCard) : void
+ containsPoint(p : Point) : boolean
+ get(p : Point) : PlayableCard

Player
- name : String
- currentPoints : int
- currentTurn : int
- numberOfScoredObjectives : int
+ getStartingCard() : PlayableCard
+ getName() : int
+ getNumberOfScoredObjectives() : int
+ getPoints() : int
+ receiveDrawnCard(card : PlayableCard) : void
+ placeCard(cardID : int, xCoordinate : int, yCoordinate : int, isFacingUp : boolean) : void
- isPlacingPointValid(point : Point) : boolean
+ setSecretObjective(secretObjective : ObjectiveCard) : void
+ placeStartingCard(isFacingUp : boolean) : void
+ calculateCommonObjectives(objectives : ObjectiveCard[]) : void
+ calculateSecretObjective() : void
- calculatePointsOnPlacement(card : PlayableCard) : int
+ quantityOfCards() : int
+ viewCurrentHand() : int
+ viewVisibleSymbols() : int

1..*
0..*   - placedCards   1
1   - playingField   2..3   1
1

PlayingField
- visibleSymbols : int
+ addPlacedCard(card : PlayableCard) : void
+ isGoldCardPlaceable(goldCard : GoldCard) : boolean
+ isPositionAvailable(point : Point) : boolean
+ getVisibleSymbols() : TokenType
+ getVisibleTokenType(requested : TokenType) : int
+ calculateGoldPoints(card : GoldCard) : int
+ calculateObjectivePoints(objective : ObjectiveCard) : int

pkg(cards)

Card
~ ID : int
+ getID() : int
+ printCardInfo() : String

0..40   contains
1

Deck
+ Deck(cards : int)
+ getNumRemaining() : int
+ getTopCardID() : int
+ getFirstVisible() : Card
+ getSecondVisible() : Card
+ shuffle() : void
+ draw(choice : int) : Card
+ setFirstVisible() : void
+ setSecondVisible() : void
+ setVisibleAfterDraw(otherDeck : Deck) : void

1   - cardDecks
1

- positions   *

Point
- yCoord : int
- xCoord : int
+ getX() : int
+ getY() : int

0..1
- position
1

0..*   - personalHandCards
- cards
1   startingCard

PlayableCard
- isFacingUp : boolean
- placedInTurn : int
+ getCardColour() : CardType
+ isFacingUp() : boolean
+ getPlacedTopRight() : TokenType
+ getPlacedTopLeft() : TokenType
+ getPlacedBottomLeft() : TokenType
+ getPlacedBottomRight() : TokenType
+ getPosition() : Point
+ placeCard(p : Point, turn : int, isFacingUp : boolean) : void

8   - corners

- secretObjective

ObjectiveCard
- awardedPoints : int
- isPositionalObjective : boolean
+ isPositional() : boolean
+ getPoints() : int
+ getPositionalRequirements() : ObjectiveSequence
+ getListRequirements() : TokenType

0..3   - listRequirements   0..1

GoldCard
- requiredFungi : int
- requiredPlant : int
- requiredAnimal : int
- requiredInsect : int
- awardedPoints : int
+ getPoints() : int
+ getPointsCondition() : TokenType
+ getPlacementConditions() : TokenType

ResourceCard
- pointsOnPlacement : int
+ getPointsOnPlacement() : int

StartingCard
+ getCentralSymbols() : TokenType

1..3

- pointsCondition   *
- positionalRequirements
0..1

pkg(enums)

- centralSymbols

<<enum>>
TokenType
+ empty : TokenType
+ blocked : TokenType
+ fungi : TokenType
+ insect : TokenType
+ animal : TokenType
+ plant : TokenType
+ scroll : TokenType
+ quill : TokenType
+ ink : TokenType

6
86

<<enum>>
ObjectiveSequence
+ redDiagonal : ObjectiveSequence
+ greenAntiDiagonal : ObjectiveSequence
+ blueDiagonal : ObjectiveSequence
+ purpleAntiDiagonal : ObjectiveSequence
+ twoRedOneGreen : ObjectiveSequence
+ twoGreenOnePurple : ObjectiveSequence
+ twoBlueOneRed : ObjectiveSequence
+ twoPurpleOneBlue : ObjectiveSequence

<<enum>>
CardType
+ fungi : CardType
+ insect : CardType
+ animal : CardType
+ plant : CardType
+ starter : CardType

86
- colour