Model::Main

PoliticCardDeck
+generateRandomPoliticCard(): PoliticCard
+distributePoliticCards(): ArrayList<PoliticCard>

Council
-councillors: Queue<Councillors>

KingCouncil
-councilors: Queue<Councilors>

Tile
-bonus: ArrayList<String>

ogni tipo di tile avrà
un suo costruttore
a cui sarà passato
il numero di bonus
voluti che
perscherà
dall'enumerazione
andando a riempire
l'array di bonus

Map
-map: Graph<City>

+connectCities(connections: HashMap<City City>): void
+countDistance(cityFrom: City, cityTo: City): int
+findKingCity(): City
+moveKing(cityDestination: City): void
+getCitiesConnectedTo(city: City): ArrayList<City>
+setAllNotVisited(): void
+isEligibleForColorBonus(owner: Player): boolean

Client
-ipAddress: String
-int: port
-coins: int
-assistants: int
-turnNumber: int
-politicCards: ArrayList<PoliticCard>
-victoryPoints: int
-usedPermitTiles: ArrayList<PermitTile>
-unusedPermitTiles: ArrayList<PermitTile>
-emporiums: int
-controlledCities: ArrayList<City>

+hasBuiltLastEmporium(): boolean
+performActions(): void
+mainActions(): void
+quickActions(): void
+drawPoliticCard(): void
+electCoucillor(): void
+acquireBusinessPermitTile(): void
+buildEmporiumWithPermitTile(): void
+buildEmporiumWithKingsHelp(): void
+engageAssistant(): void
+changeBusinessPermitTiles(): void
+sendAssistantToElectCouncillor(): void
+performAdditionalMainAction(): void

PoliticCard
-color: String

RegionBonusTile
-points: int
-region: Region

«enumeration»
RegionName
HILLS
MOUNTAINS
COAST

RewardToken

KingRewardDeck
-deck: ArrayList<KingRewardTile>

+getKingRewars(): RewardTile

«enumeration»
CityColors
GOLD
SILVER
BRONZE
STEEL

ColorBonusTile
-points: int
-color: String

KingRewardTiles
-label: String
-points: int

Player
-userName: String
-password: String
-matchesWon: int
-rageQuits: int

+getUserName(): String
+getPassword(): String
+getMatchesWon(): int
+getRageQuits(): int

Server
-ip: String
-port: int
-registeredUser: ArrayList<Player>

+launchMatch(): void

Councillor
-color: String

«enumeration»
CouncillorColors
PINK
ORANGE
BLUE
PURPLE
BLACK
WHITE
MULTICOLOR

Region
-name: String
-council: Council
-cities: ArrayList<City>
-deck: PermitTileDeck
-regionBonus: RegionBonusTile

+electCouncillor(councillor: Councillor): void
+checkCouncilSatisfaction(politicCards: ArrayList<PoliticCard>): boolean
+isEligibleForRegionBonus(owner: Player): boolean
+winRegionBonus(owner: Player): RegionBonusTile

City
-name: String
-color: String
-emporiums: ArrayList<Emporium>
-region: Region
-coordinate: Point
-rewardToken: RewardToken
-kingIsHere: boolean
-visited: boolean

+getKingIsHere(): boolean
+getRegion(): Region
+getColor(): String
+setKingIsHere(value: boolean): void
+buildEmporium(owner: Player): void
+countEmporiums(owner: Player): int
+winBonus(): String
+countOthersEmporiums(owner: Player): int
+checkPresenceOfEmporiums(owner: Player): boolean
+setVisited(): void

«enumeration»
RewardTokenType
ASSISTANT
POLITIC
VICTORYTRACK
COINS
NOBILITYTRACK

MatchHandler
-id: int
-date: Date

+mapSetup(): void

Market

CoinsManager

Connection

ClientHandler

«enumeration»
CityNames
ARKON
BURGEN
CASTRUM
DORFUL
ESTI
FRAMEK
GRADEN
HELLAR
INDUR
JUVELAR
KULTOS
LYRAM
MERKATIM
NARIS
OSIUM
PEHEATH
QUARNGATE
REWOOD
STEWOLDS
TEEWICH
URUKHAI
VANHELSING
WOOWICK
XERO
YUMA
ΩMEGA

Point
-x: int
-y: int

Emporium
-owner: String
-cityName: String

NobilityTrack
-track: ArrayList<NobilityCell>

NobilityCell
-bonus: ArrayList<String>

«enumeration»
NobilityTrackBonusType
ASSISTANT
VICTORYTRACK
POLITIC
COINS
NOBILITYTRACK
DRANPERMITTILE
BONUSPERMITTILE
TWOEMPORIUMCITY

PermitTileDeck
-deck: Queue<PermitTile>
-uncoveredPermitTile1: PermitTile
-uncoveredPermitTile2: PermitTile

+switchPermitTiles(): void
+drawPermitTile(): PermitTile

PermitTile
-cities: ArrayList<Cities>
-region: Region

«enumeration»
PermitTileBonusType
ASSISTANT
VICTORYTRACK
POLITIC
NOBILITYTRACK
NEWMAINACTION
COINS