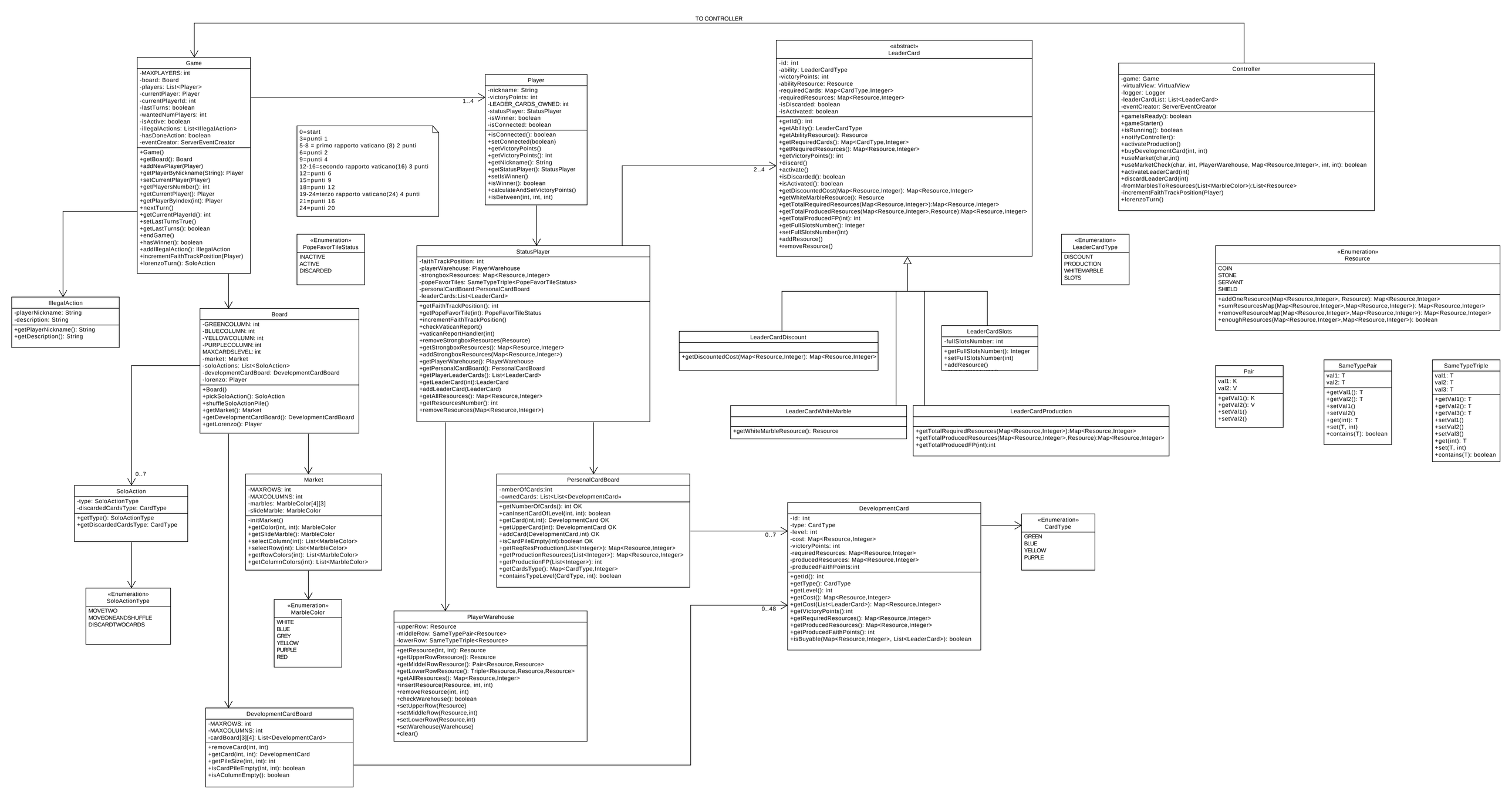
TO CONTROLLER
Game
-MAXPLAYERS: int
-board: Board
-players: List<Player>
-currentPlayer: Player
-currentPlayerId: int
-lastTurns: boolean
-wantedNumPlayers: int
-isActive: boolean
-illegalActions: List<IllegalAction>
-hasDoneAction: boolean
-eventCreator: ServerEventCreator
+Game()
+getBoard(): Board
+addNewPlayer(Player)
+getPlayerByNickname(String): Player
+setCurrentPlayer(Player)
+getPlayersNumber(): int
+getCurrentPlayer(): Player
+getPlayerByIndex(int): Player
+nextTurn()
+getCurrentPlayerId(): int
+setLastTurnsTrue()
+getLastTurns(): boolean
+endGame()
+hasWinner(): boolean
+addIllegalAction(): IllegalAction
+incrementFaithTrackPosition(Player)
+lorenzoTurn(): SoloAction
1..4
Player
-nickname: String
-victoryPoints: int
-LEADER_CARDS_OWNED: int
-statusPlayer: StatusPlayer
-isWinner: boolean
-isConnected: boolean
+isConnected(): boolean
+setConnected(boolean)
+getVictoryPoints()
+getVictoryPoints(): int
+getNickname(): String
+getStatusPlayer(): StatusPlayer
+setIsWinner()
+isWinner(): boolean
+calculateAndSetVictoryPoints()
+isBetween(int, int, int)
0=start
3=punti 1
5-8 = primo rapporto vaticano (8) 2 punti
6=punti 2
9=punti 4
12-16=secondo rapporto vaticano(16) 3 punti
12=punti 6
15=punti 9
18=punti 12
19-24=terzo rapporto vaticano(24) 4 punti
21=punti 16
24=punti 20
«abstract»
LeaderCard
-id: int
-ability: LeaderCardType
-victoryPoints: int
-abilityResource: Resource
-requiredCards: Map<CardType,Integer>
-requiredResources: Map<Resource,Integer>
-isDiscarded: boolean
-isActivated: boolean
+getId(): int
+getAbility(): LeaderCardType
+getAbilityResource(): Resource
+getRequiredCards(): Map<CardType,Integer>
+getRequiredResources(): Map<Resource,Integer>
+getVictoryPoints(): int
+discard()
+activate()
+isDiscarded(): boolean
+isActivated(): boolean
+getDiscountedCost(Map<Resource,Integer): Map<Resource,Integer>
+getWhiteMarbleResource(): Resource
+getTotalRequiredResources(Map<Resource,Integer>):Map<Resource,Integer>
+getTotalProducedResources(Map<Resource,Integer>,Resource):Map<Resource,Integer>
+getTotalProducedFP(int): int
+getFullSlotsNumber(): Integer
+setFullSlotsNumber(int)
+addResource()
+removeResource()
2..4
Controller
-game: Game
-virtualView: VirtualView
-logger: Logger
-leaderCardList: List<LeaderCard>
-eventCreator: ServerEventCreator
+gameIsReady(): boolean
+gameStarter()
+isRunning(): boolean
+notifyController():
+activateProduction()
+buyDevelopmentCard(int, int)
+useMarket(char,int)
+useMarketCheck(char, int, PlayerWarehouse, Map<Resource,Integer>, int, int): boolean
+activateLeaderCard(int)
+discardLeaderCard(int)
-fromMarblesToResources(List<MarbleColor>):List<Resource>
-incrementFaithTrackPosition(Player)
+lorenzoTurn()
«Enumeration»
PopeFavorTileStatus
INACTIVE
ACTIVE
DISCARDED
StatusPlayer
-faithTrackPosition: int
-playerWarehouse: PlayerWarehouse
-strongboxResources: Map<Resource,Integer>
-popeFavorTiles: SameTypeTriple<PopeFavorTileStatus>
-personalCardBoard:PersonalCardBoard
-leaderCards:List<LeaderCard>
+getFaithTrackPosition(): int
+getPopeFavorTile(int): PopeFavorTileStatus
+incrementFaithTrackPosition()
+checkVaticanReport()
+vaticanReportHandler(int)
+removeStrongboxResources(Resource)
+getStrongboxResources(): Map<Resource,Integer>
+addStrongboxResources(Map<Resource,Integer>)
+getPlayerWarehouse(): PlayerWarehouse
+getPersonalCardBoard(): PersonalCardBoard
+getPlayerLeaderCards(): List<LeaderCard>
+getLeaderCard(int):LeaderCard
+addLeaderCard(LeaderCard)
+getAllResources(): Map<Resource,Integer>
+getResourcesNumber(): int
+removeResources(Map<Resource,Integer>)
«Enumeration»
LeaderCardType
DISCOUNT
PRODUCTION
WHITEMARBLE
SLOTS
«Enumeration»
Resource
COIN
STONE
SERVANT
SHIELD
+addOneResource(Map<Resource,Integer>, Resource): Map<Resource,Integer>
+sumResourcesMap(Map<Resource,Integer>,Map<Resource,Integer>): Map<Resource,Integer>
+removeResourceMap(Map<Resource,Integer>,Map<Resource,Integer>): Map<Resource,Integer>
+enoughResources(Map<Resource,Integer>,Map<Resource,Integer>): boolean
IllegalAction
-playerNickname: String
-description: String
+getPlayerNickname(): String
+getDescription(): String
Board
-GREENCOLUMN: int
-BLUECOLUMN: int
-YELLOWCOLUMN: int
-PURPLECOLUMN: int
MAXCARDSLEVEL: int
-market: Market
-soloActions: List<SoloAction>
-developmentCardBoard: DevelopmentCardBoard
-lorenzo: Player
+Board()
+pickSoloAction(): SoloAction
+shuffleSoloActionPile()
+getMarket(): Market
+getDevelopmentCardBoard(): DevelopmentCardBoard
+getLorenzo(): Player
LeaderCardDiscount
+getDiscountedCost(Map<Resource,Integer): Map<Resource,Integer>
LeaderCardSlots
-fullSlotsNumber: int
+getFullSlotsNumber(): Integer
+setFullSlotsNumber(int)
+addResource()
LeaderCardWhiteMarble
+getWhiteMarbleResource(): Resource
LeaderCardProduction
+getTotalRequiredResources(Map<Resource,Integer>):Map<Resource,Integer>
+getTotalProducedResources(Map<Resource,Integer>,Resource):Map<Resource,Integer>
+getTotalProducedFP(int):int
Pair
val1: K
val2: V
+getVal1(): K
+getVal2(): V
+setVal1()
+setVal2()
SameTypePair
val1: T
val2: T
+getVal1(): T
+getVal2(): T
+setVal1()
+setVal2()
+get(int): T
+set(T, int)
+contains(T): boolean
SameTypeTriple
val1: T
val2: T
val3: T
+getVal1(): T
+getVal2(): T
+getVal3(): T
+setVal1()
+setVal2()
+setVal3()
+get(int): T
+set(T, int)
+contains(T): boolean
0..7
SoloAction
-type: SoloActionType
-discardedCardsType: CardType
+getType(): SoloActionType
+getDiscardedCardsType: CardType
Market
-MAXROWS: int
-MAXCOLUMNS: int
-marbles: MarbleColor[4][3]
-slideMarble: MarbleColor
-initMarket()
+getColor(int, int): MarbleColor
+getSlideMarble(): MarbleColor
+selectColumn(int): List<MarbleColor>
+selectRow(int): List<MarbleColor>
+getRowColors(int): List<MarbleColor>
+getColumnColors(int): List<MarbleColor>
PersonalCardBoard
-nmberOfCards:int
-ownedCards: List<List<DevelopmentCard»
+getNumberOfCards(): int OK
+canInsertCardOfLevel(int, int): boolean
+getCard(int,int): DevelopmentCard OK
+getUpperCard(int): DevelopmentCard OK
+addCard(DevelopmentCard,int) OK
+isCardPileEmpty(int):boolean OK
+getReqResProduction(List<Integer>): Map<Resource,Integer>
+getProductionResources(List<Integer>): Map<Resource,Integer>
+getProductionFP(List<Integer>): int
+getCardsType(): Map<CardType,Integer>
+containsTypeLevel(CardType, int): boolean
0..7
DevelopmentCard
-id: int
-type: CardType
-level: int
-cost: Map<Resource,Integer>
-victoryPoints: int
-requiredResources: Map<Resource,Integer>
-producedResources: Map<Resource,Integer>
-producedFaithPoints:int
+getId(): int
+getType(): CardType
+getLevel(): int
+getCost(): Map<Resource,Integer>
+getCost(List<LeaderCard>): Map<Resource,Integer>
+getVictoryPoints():int
+getRequiredResources(): Map<Resource,Integer>
+getProducedResources(): Map<Resource,Integer>
+getProdcedFaithPoints(): int
+isBuyable(Map<Resource,Integer>, List<LeaderCard>): boolean
«Enumeration»
CardType
GREEN
BLUE
YELLOW
PURPLE
«Enumeration»
SoloActionType
MOVETWO
MOVEONEANDSHUFFLE
DISCARDTWOCARDS
«Enumeration»
MarbleColor
WHITE
BLUE
GREY
YELLOW
PURPLE
RED
PlayerWarehouse
-upperRow: Resource
-middleRow: SameTypePair<Resource>
-lowerRow: SameTypeTriple<Resource>
+getResource(int, int): Resource
+getUpperRowResource(): Resource
+getMiddelRowResource(): Pair<Resource,Resource>
+getLowerRowResource(): Triple<Resource,Resource,Resource>
+getAllResources(): Map<Resource,Integer>
+insertResource(Resource, int, int)
+removeResource(int, int)
+checkWarehouse(): boolean
+setUpperRow(Resource)
+setMiddleRow(Resource,int)
+setLowerRow(Resource,int)
+setWarehouse(Warehouse)
+clear()
0..48
DevelopmentCardBoard
-MAXROWS: int
-MAXCOLUMNS: int
-cardBoard[3][4]: List<DevelopmentCard>
+removeCard(int, int)
+getCard(int, int): DevelopmentCard
+getPileSize(int, int): int
+isCardPileEmpty(int, int): boolean
+isAColumnEmpty(): boolean