Production

-input : Map<ResourceType,Integer>
-output : Map<ResourceType,Integer>
-flag : boolean

+getInput() : void
+getOutput : void
+checkFlag(): boolean
+ToggleFlag() : void

CardDevelopment

-->(-List<ResourceRequirement> Costo)
-->(-Map<ResourceType,Integer> Costo)
-livello : Integer
-color : Color
-prod : Production

+getCosto() : Costo
+getLivello() : Integer
+getProd() : Production
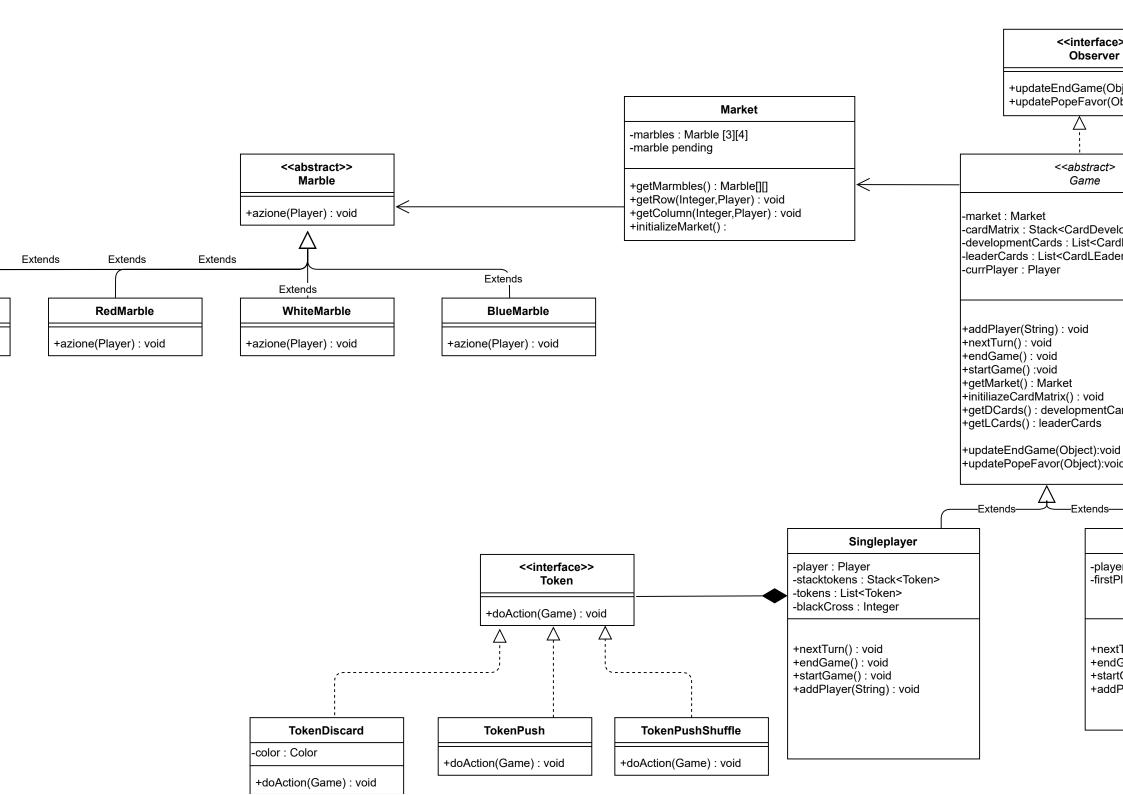

<<enumeration>>
ResourceType

YELLOW
GREY
VIOLET
BLUE
RED
WHITE
?

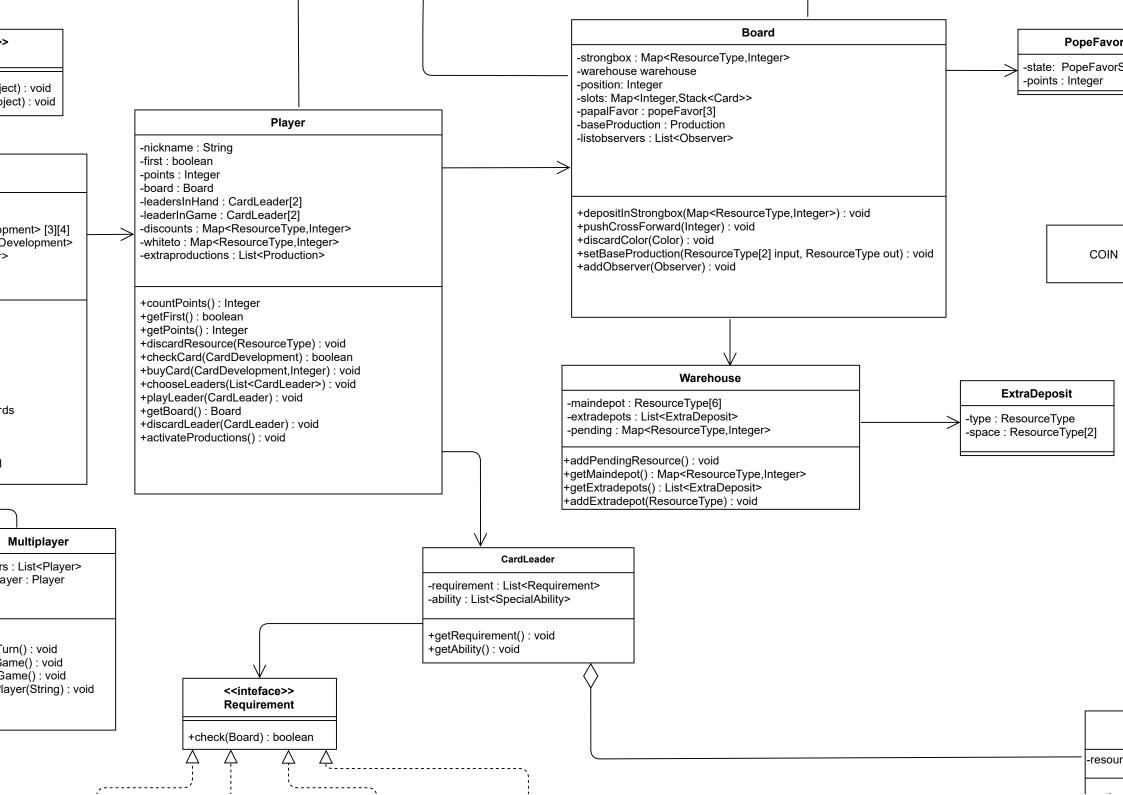| <<enumerate>> |
Color
GREEN
BLUE
YELLOW
VIOLET

| **PopeFavorState** |
<<enumeration>>
ACTIVE
UNACTIVE
DISCARDED

Extends

YellowMarble
+azione(Player) : void

VioletMarble
+azione(Player) : void

GreyMarble
+azione(Player) : void

Observer
<<interface>>

+updateEndGame(Obj
+updatePopeFavor(Ob

Market

-marbles : Marble [3][4]
-marble pending

+getMarmbles() : Marble[][]
+getRow(Integer,Player) : void
+getColumn(Integer,Player) : void
+initializeMarket() :

Marble
<>

+azione(Player) : void

Extends   Extends   Extends

Extends

Extends

RedMarble

+azione(Player) : void

WhiteMarble

+azione(Player) : void

BlueMarble

+azione(Player) : void

Game
<>

-market : Market
-cardMatrix : Stack<CardDevelo
-developmentCards : List<CardI
-leaderCards : List<CardLEader
-currPlayer : Player

+addPlayer(String) : void
+nextTurn() : void
+endGame() : void
+startGame() :void
+getMarket() : Market
+initiliazeCardMatrix() : void
+getDCards() : developmentCar
+getLCards() : leaderCards

+updateEndGame(Object):void
+updatePopeFavor(Object):voic

Extends   Extends

Token
<<interface>>

+doAction(Game) : void

Singleplayer

-player : Player
-stacktokens : Stack<Token>
-tokens : List<Token>
-blackCross : Integer

+nextTurn() : void
+endGame() : void
+startGame() : void
+addPlayer(String) : void

-player
-firstPl

+nextT
+endG
+startG
+addP

TokenDiscard

-color : Color

+doAction(Game) : void

TokenPush

+doAction(Game) : void

TokenPushShuffle

+doAction(Game) : void

Board

- -strongbox : Map<ResourceType,Integer>
- -warehouse warehouse
- -position: Integer
- -slots: Map<Integer,Stack<Card>>
- -papalFavor : popeFavor[3]
- -baseProduction : Production
- -listobservers : List<Observer>

- +depositInStrongbox(Map<ResourceType,Integer>) : void
- +pushCrossForward(Integer) : void
- +discardColor(Color) : void
- +setBaseProduction(ResourceType[2] input, ResourceType out) : void
- +addObserver(Observer) : void

PopeFavor

- -state: PopeFavorS
- -points : Integer

COIN

Player

- -nickname : String
- -first : boolean
- -points : Integer
- -board : Board
- -leadersInHand : CardLeader[2]
- -leaderInGame : CardLeader[2]
- -discounts : Map<ResourceType,Integer>
- -whiteto : Map<ResourceType,Integer>
- -extraproductions : List<Production>

- +countPoints() : Integer
- +getFirst() : boolean
- +getPoints() : Integer
- +discardResource(ResourceType) : void
- +checkCard(CardDevelopment) : boolean
- +buyCard(CardDevelopment,Integer) : void
- +chooseLeaders(List<CardLeader>) : void
- +playLeader(CardLeader) : void
- +getBoard() : Board
- +discardLeader(CardLeader) : void
- +activateProductions() : void

(left partial) >>
- ject) : void
- ject) : void

(left partial)
- opment> [3][4]
- Development>
- >

- rds

Multiplayer

- rs : List<Player>
- ayer : Player

- Turn() : void
- Game() : void
- Game() : void
- layer(String) : void

Warehouse

- -maindepot : ResourceType[6]
- -extradepots : List<ExtraDeposit>
- -pending : Map<ResourceType,Integer>

- +addPendingResource() : void
- +getMaindepot() : Map<ResourceType,Integer>
- +getExtradepots() : List<ExtraDeposit>
- +addExtradepot(ResourceType) : void

ExtraDeposit

- -type : ResourceType
- -space : ResourceType[2]

CardLeader

- -requirement : List<Requirement>
- -ability : List<SpecialAbility>

- +getRequirement() : void
- +getAbility() : void

<<inteface>> Requirement

- +check(Board) : boolean

-resour

State

<>
SpecialAbility

ce : ResourceType:

-co -qu	
+c	

CardRequirement

-olor : Color
-uantity : Integer

-heck(Board) : boolean

LevelCardRequirement

-color : Color
-quantity : Integer
-level : Integer

+check(Board) : boolean

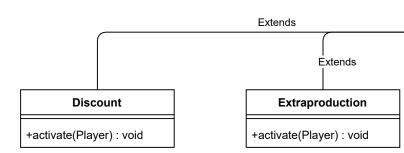**FaithRequirement**

-position: Integer

+check(Board) : boolean

ResourceRequirement

-resource: ResourceType
-quantity: Integer

+check(Board) : boolean

Extends

Extends

Discount

+activate(Player) : void

Extraproduction

+activate(Player) : void

te(Player) : void

─Extends─

Extends

WhiteTo

+activate(Player) : void

ExtraDeposit

+activate(Player) : void