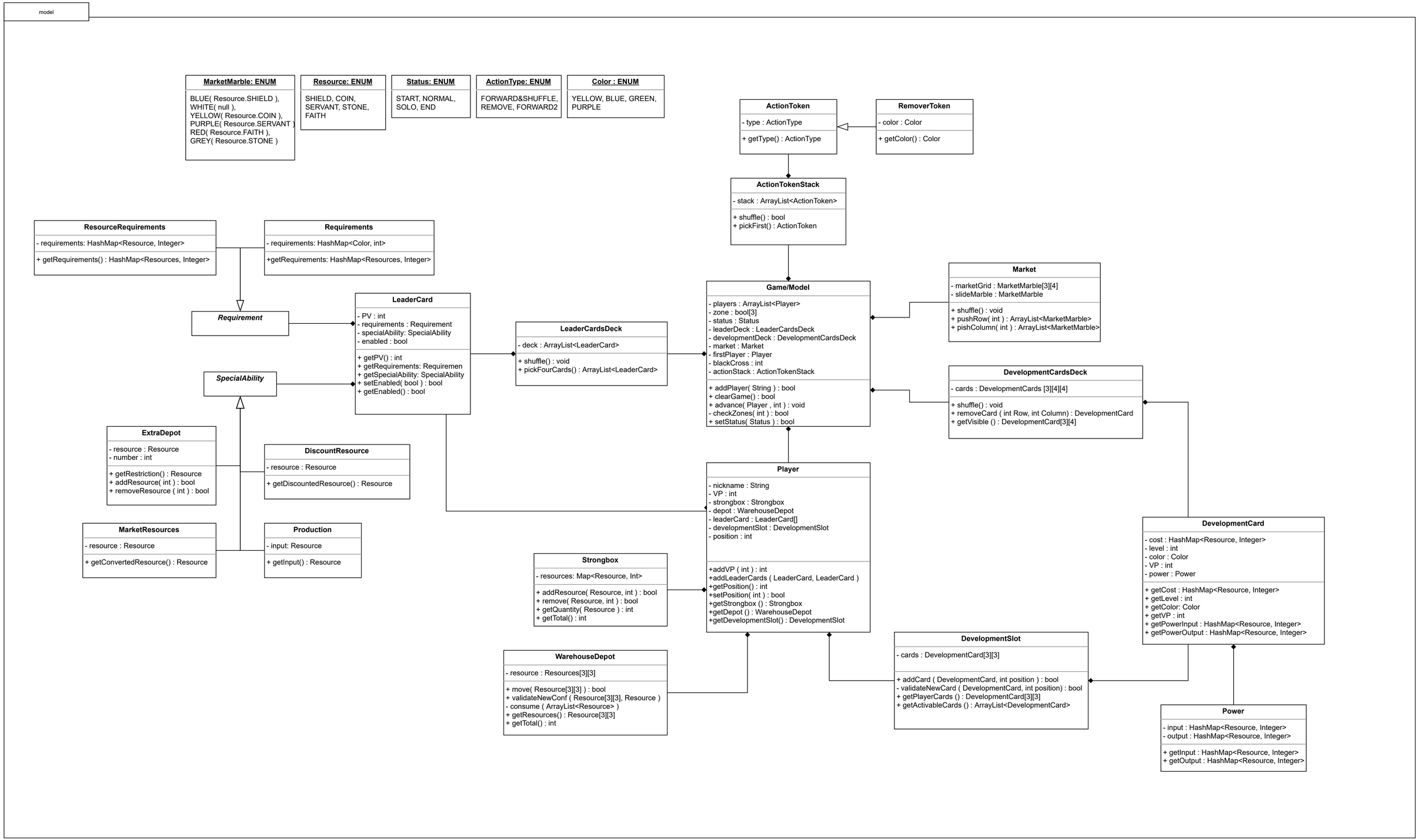
model
MarketMarble: ENUM
BLUE(Resource.SHIELD),
WHITE(null),
YELLOW(Resource.COIN),
PURPLE(Resource.SERVANT)
RED(Resource.FAITH),
GREY(Resource.STONE)
Resource: ENUM
SHIELD, COIN,
SERVANT, STONE,
FAITH
Status: ENUM
START, NORMAL,
SOLO, END
ActionType: ENUM
FORWARD&SHUFFLE,
REMOVE, FORWARD2
Color : ENUM
YELLOW, BLUE, GREEN,
PURPLE
ActionToken
- type : ActionType
+ getType() : ActionType
RemoverToken
- color : Color
+ getColor() : Color
ActionTokenStack
- stack : ArrayList<ActionToken>
+ shuffle() : bool
+ pickFirst() : ActionToken
ResourceRequirements
- requirements: HashMap<Resource, Integer>
+ getRequirements() : HashMap<Resources, Integer>
Requirements
- requirements: HashMap<Color, int>
+getRequirements: HashMap<Resources, Integer>
Requirement
LeaderCard
- PV : int
- requirements : Requirement
- specialAbility: SpecialAbility
- enabled : bool
+ getPV() : int
+ getRequirements: Requiremen
+ getSpecialAbility: SpecialAbility
+ setEnabled(bool) : bool
+ getEnabled() : bool
LeaderCardsDeck
- deck : ArrayList<LeaderCard>
+ shuffle() : void
+ pickFourCards() : ArrayList<LeaderCard>
Game/Model
- players : ArrayList<Player>
- zone : bool[3]
- status : Status
- leaderDeck : LeaderCardsDeck
- developmentDeck : DevelopmentCardsDeck
- market : Market
- firstPlayer : Player
- blackCross : int
- actionStack : ActionTokenStack
+ addPlayer(String) : bool
+ clearGame() : bool
+ advance(Player , int) : void
- checkZones(int) : bool
+ setStatus(Status) : bool
Market
- marketGrid : MarketMarble[3][4]
- slideMarble : MarketMarble
+ shuffle() : void
+ pushRow(int) : ArrayList<MarketMarble>
+ pishColumn(int) : ArrayList<MarketMarble>
SpecialAbility
DevelopmentCardsDeck
- cards : DevelopmentCards [3][4][4]
+ shuffle() : void
+ removeCard (int Row, int Column) : DevelopmentCard
+ getVisible () : DevelopmentCard[3][4]
ExtraDepot
- resource : Resource
- number : int
+ getRestriction() : Resource
+ addResource(int) : bool
+ removeResource (int) : bool
DiscountResource
- resource : Resource
+ getDiscountedResource() : Resource
Player
- nickname : String
- VP : int
- strongbox : Strongbox
- depot : WarehouseDepot
- leaderCard : LeaderCard[]
- developmentSlot : DevelopmentSlot
- position : int
+addVP (int) : int
+addLeaderCards (LeaderCard, LeaderCard)
+getPosition() : int
+setPosition(int) : bool
+getStrongbox () : Strongbox
+getDepot () : WarehouseDepot
+getDevelopmentSlot() : DevelopmentSlot
MarketResources
- resource : Resource
+ getConvertedResource() : Resource
Production
- input: Resource
+ getInput() : Resource
Strongbox
- resources: Map<Resource, Int>
+ addResource(Resource, int) : bool
+ remove(Resource, int) : bool
+ getQuantity(Resource) : int
+ getTotal() : int
DevelopmentCard
- cost : HashMap<Resource, Integer>
- level : int
- color : Color
- VP : int
- power : Power
+ getCost : HashMap<Resource, Integer>
+ getLevel : int
+ getColor: Color
+ getVP : int
+ getPowerInput : HashMap<Resource, Integer>
+ getPowerOutput : HashMap<Resource, Integer>
WarehouseDepot
- resource : Resources[3][3]
+ move(Resource[3][3]) : bool
+ validateNewConf (Resource[3][3], Resource)
- consume (ArrayList<Resource>)
+ getResources() : Resource[3][3]
+ getTotal() : int
DevelopmentSlot
- cards : DevelopmentCard[3][3]
+ addCard (DevelopmentCard, int position) : bool
- validateNewCard (DevelopmentCard, int position) : bool
+ getPlayerCards () : DevelopmentCard[3][3]
+ getActivableCards () : ArrayList<DevelopmentCard>
Power
- input : HashMap<Resource, Integer>
- output : HashMap<Resource, Integer>
+ getInput : HashMap<Resource, Integer>
+ getOutput : HashMap<Resource, Integer>