## User

- color: CatanColor
- name: String
- playerIndex: int
- playerId: int

## ResourceList

- brick:int
- ore:int
- sheep: int
- wheat: int
- wood: int

## Chat

- chatMessages: MessageList
- log: MessageList

## MessageList

- lines: list[MessageLine]

## MessageLine

- message: String
- source: String

## Bank

- resoures: ResourceList

+canGetResource(Resource)

## Player

- discarded: boolean
- victoryPoints: int

- cities: int
- roads:int
- settlements:int
- soldiers: int

- newDevCards: DevCardList
- oldDevCards: DevCardList
- playedDevCard: boolean

- resources:ResourceStorage

+canPlayDevCard(DevCardType)
+canBuildRoad(ResourceList)
+canBuildSettlement(ResourceList)
+canBuildCity(ResourceList)
+canBuyDevCard(ResourceList)

## UserManager

- users: list [User]
- turnTracker: TurnTracker

## ClientModel (GameInstance)

- bank: Bank
- chat: ChatObject
- map: Map
- userManager: UserManager
- tradeOffer: TradeOffer
- version: int
- winner: int

## ResourceStorage

- resoures: ResourceList

+canAddResource(Resource)

## DevCardList

- monopoly: int
- monument: int
- roadBuilding: int
- soldier: int
- yearOfPlenty: int

## TurnTracker

- currentTurn:int
- status: String
- longestRoad: int
- largestArmy: int

0..4

0..*

1

0..*

1

## Map

- hexes: list [Hex]
- roads: list [Road]
- settlements: list [VertexObject]
- cities: list [VertexObject]
- radius: int
- robber: Robber

+ canBuildRoad(Hex)
+ canBuildSettlement(Hex)
+ canBuildCity(Hex)
+ canMoveRobber(Hex)

## Robber

- Location: HexLocation
- Number: number

+ MoveTo(Hex)

## VertexObject

- owner: int
- type: (City or Settlement)
- resourceRatio: int
- port: PortType

+ onPort()

## Harbor

- resource: String
- ratio: int

## Hex

- location: HexLocation
- resource: String (optional)
- number: int (optional)
- roadMap: Map<Edge, Road>
- harbor: Harbor (optional)
- communityMap: <Vertex, VertexObject>

+ getAdjacentEdges()
+ getAdjacentVertices()
+ getNeighbors()
+ canBuildRoad(Hex)
+ canBuildSettlement(Hex)
+ canBuildCity(Hex)

## Road

- owner: int

## Poller

proxy: Proxy
serializer: Serializer

+ poll()
- updateModel(ClientModel)

## Serializer

+ toJSON(ClientModel)
+ from JSON(JSON):
ClientModel

< uses

uses
v

## Interface "Proxy"

Poller uses Serializer to populate the client model from JSON, Proxy to get JSON, and DTOs to represent objects, not represented by clientModel

The interface is split into 2 pieces for convenience of placing it into UML.

## "Interface" staticActionsProxy

+ login(username, password): User
+ registerNewUser(username, password):User
+ listGames(): GameContainer
+ createGames(name, randomTiles, randomNumbers, randomPorts): Game
+ joinGame(gameId, Color)
+ saveGames(gameId, fileName)
+ loadGame()
+ retrieveCurrentState(versionNumber)
+ resetGame()
+ getCommands(): array[Command]
+ postGameCommands(array[Command])
+ listAllTypes(): array[AITypes]
+ addAIPlayer(AIType)
+ changeLogLevel(LogLevel)

## "Interface" MoveProxy

+ sendChat(content:String)
+ acceptTrade(willAccept: boolean)
+ discardCards(discardedCards: ResourceHand)
+ rollNumber(): integer
+ buildRoad(free:boolean, roadLocation: EdgeLocation)
+buildSettlement(free:boolean, vertexLocation:VertexLocation)
+ buildCitty(VertexLocation)
+ offerTrade(offer:ResourceHand, receiver:playerIndex)
+ maritimeTrade(ratio:integer, inputResource:Resource,
outputResource:Resource)
+ robPlayer(location:HexLocation, victimIndex:playerIndex)
+ finishTurn()
+ buyDevCard()
+ playSoldier(location:HexLocation, victimIndex:playerIndex)
+ playYearOfPlenty(resource1:Resource, resource2:Resource)
+ playRoadBuilding(spot1:EdgeLocation, spot2:EdgeLocation)
+ playMonopoly(resource:Resource)
+ playMonument()

Common Data Types:
playerIndex
edgeLocation
vertexLocation
resource
resourceHand

universal command
properties:
type: String (type of move)
playerIndex: integer (0-3)

## GameContainer

- games: array[Game]

## Game

- title: String
- id: int
- players: array[Player]

## TradeOffer

- sender: int
- receiver: int
- offer: ResourceList
- give: Resource
- get: Resource

+ canAddResources(User)
+ addResources(ResourceType)
+ canRemoveResources(User)
+ removeResources(ResourceType)