BUY_FROM_MARKET

When the client chooses this action, the server asks to the client which column/row he wants to choose and consequently the buyFromMarket method of the Model is called.The resources taken from the market are shown all together at the beggining to the client, and then the MarketCLI will manage them . For each resource the client has to choose if he wants to keep it (choice == 1) ,to discard it (choice == 2), or move two depots before (choice == 0). If the client decides to discard it, all the other players will increase their position on the faithTrack of one.

There are some particular cases: if the client recives a Red Marble from the market, his FaithMarker will increase his position of one in the FaithTrack; If he receives a BLANK resource, it means that he has activated two TransformWhiteMarble's Leader Cards and he has to choose from the two resources indicated, otherwise in the case that he has activated only a TransformWhiteMarble's Leader Card, the white marble obtained by the market will be automatically converted into the relative Colored Marble.

BuyFromMarket

sendMsg(msg4)

flush()

addInQueue
notify(CONTR,queue(0))

receiveMsg(msg4)

notify(CONTR,msg4)

receiveMsg(msg4)

findUserRoom.notify(CONTR,msg4)

receiveMsg(msg4)

notify(CONT, msg4)

receiveMsg(msg4)

VUpdateMarket update

notify(VIEW, update)
checkUsername
player.BuyFromMarket()

receiveMsg(update)

flush()

updateMarket()
market= update.getMarket()

loop [foreach (Resources r : player.BuyFromMarket()]

   alt [if(r!=FAITHMARKER)]

resourcesToStore.add(r)

   [else]

VUpdateFaithTrach msg5

notify(VIEW, msg5)

receiveMsg(msg5)

sendMsg(msg5)

flush()

addInQueue
notify(VIEW, queue(0))

receiveMsg(msg5)

"You increased the position
in Faith Track"

VNotifyPositionIncreasedBy msg6

notify(VIEW, msg8)

receiveMsg(msg6)

send to all other player(not this)

sendMsg(msg6)

flush()

addInQueue
notify(VIEW, queue(0))

receiveMsg(msg6)

"Player X increased
his position in Faith Track"

alt [numberResource==resourcesToStore.size()]

VChooseDepots msg7

notify(VIEW, msg7)

receiveMsg(msg7)

sendMsg(msg7)

flush()

addInQueue
notify(VIEW, queue(0))

receiveMsg(msg7)

new()
setResourcesStored(false)
handleResources()

MarketCLI

alt [if resources(0).equals("BLANK")]

choseAndBuyFromMarket()

buyFromMarket(resource)

buyFromMarket(resource(0))

buyFromMarket(resource(0))

alt [choice == 0]

setWait(true)

CMoveResource msg8

sendMsg(msg8)

flush()

addInQueue
notify(CONTR,queue(0))

receiveMsg(msg8)

notify(CONTR,msg8)

receiveMsg(msg8)

findUserRoom.notify(CONTR,msg8)

receiveMsg(msg8)

notify(CONT, msg8)

receiveMsg(msg8)

player.moveResource(from, to)

VUpdateWarehouse update

notify(VIEW, update)

receiveMsg(update)

sendMsg(update)

flush()

addInQueue
notify(VIEW, queue(0))

receiveMsg(update)

updateWarehouse()
warehouse = update.getWarehouse
showWarehouse()

setWait(false)
handleResources()

buyFromMarket(resource(0))

[choise == 1]

CChooseResourceAndDepot msg9

sendMsg(msg9)

flush()

addInQueue
notify(CONTR,queue(0))

receiveMsg(msg9)

notify(CONTR,msg9)

receiveMsg(msg9)

findUserRoom.notify(CONTR,msg9)

receiveMsg(msg8)

notify(CONT, msg9)

receiveMsg(msg9)

player.addeResource(msg9.getResource)

alt [ok]

VUpdateWarehouse update

...
as before

[ko]

VNotValidDepot msg10

receiveMsg(msg10)

sendMsg(msg10)

flush()

addInQueue
notify(VIEW, queue(0))

receiveMsg(msg10)

askAgain()

[choice == 2]

CChooseDiscardResources msg11

sendMsg(msg11)

flush()

addInQueue
notify(CONTR,queue(0))

receiveMsg(msg11)

notify(CONTR,msg11)

receiveMsg(msg11)

findUserRoom.notify(CONTR,msg11)

receiveMsg(msg11)

allOtherPlayer.increasePosition()

VNotifyPositionIncreasedBy msg12

...

VUpdateFaithTrack update

...

checkEndGame()