Buy a resource: explanation

When the user wants to buy some marbles from the market it selects the preferred column or row and confirms the action. A MarketBuyEvent is built from the UI and passed to the Network handler that will send to the server through a socket. The server will associate the event to the right match and send it to the proper controller. This queries the model through the market manager to get the set of marbles to buy. The model notifies the change to a proper ModelChangeHandler that, in turn, notifies all the players with a MarketStateEvent.

The controller gets the bought marbles and checks for red marbles in order to increment the position on the faith track. If there are some red marbles, the model is update by the FaithTrackManager and again the change is notified to all players thanks to a ModelChangeHandler.

Now the controller queries again the model to know if there are active some leader powers in order to know the complete set of resources that have been to be displaced.

At this point the controller sends through the network a ChoseMultipleExtraResourcesEvent to ask the user which resources he wants to get from white marbles. The reply is given in a SimpleChosenResourcesEvent that will be managed by the controller.

If there are some errors, the controller sends a playerActionEvent and re-sends the ChoseMultipleExtraResourcesEvent until the players gives back a valid answare.

Once the controller got all the resources, it sends to the user a OrganizeResourcesEvent. The view will show a form to ask the user where to put the resources bought from market. The displacement is provided to the controller again with a NewResourcesOrganizationEvent.

At this point the controller updates the model that, in turn, will notify all players with the new state of the deposit of the player that had performed the action. This is done with the events DashBoardStateEvent and DepositLeaderPowerStateEvent.

Inside the NewResourcesOrganizationEvent there is a field that specifies the resources to be discarded. If there are any, the controller asks the FaithTrackManager to update the model, incrementing all the other players positions on faith track. As all changes in model, this triggers a ModelChangeHandler that will notify all players with a FaithTrackEvent