Defining another behavior of an agent in a new scenario - make AgentScript and BehaviorScript functions derivable

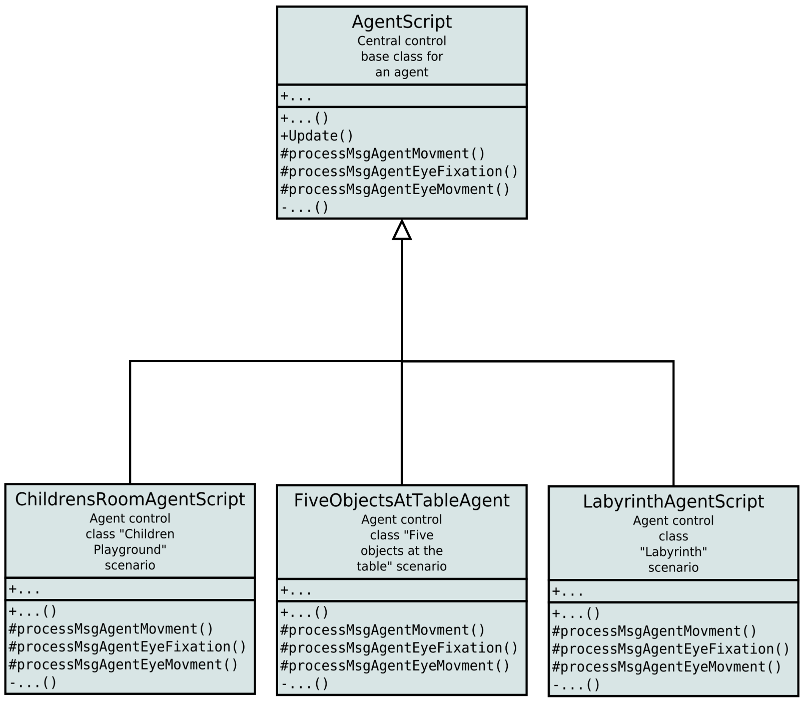
General idea

is the splitting of the control loop and message handling. The control loop of the agent is completely handled by the base classes and the user has no influence on it. The messages should have the option to be processed scenario dependent. For this, the user created a derived class of AgentScript respective BehaviorScript which specifies how the agent and VR reacts to messages.

The update() function in the AgentScript base class contains the control loop of the agent (sending Camera picture, walk, handle messages, etc.)

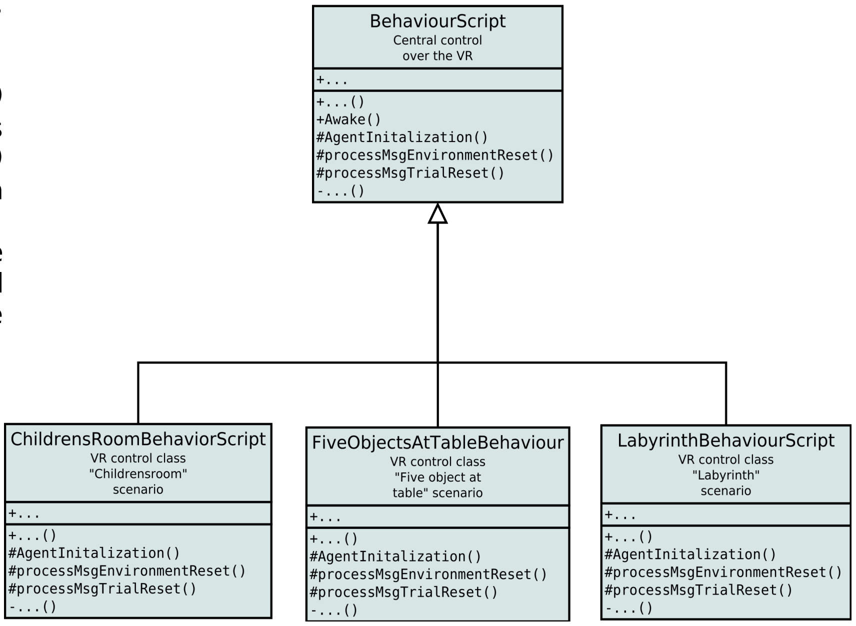
For each protobuf message, there exists an own protected virtual function.
For example *processMsgAgentMovment* is called when the agent receives the message MsgAgentMovment by the client.

The user therefore has to create a new class for his scenario, where he must define the processMsg* functions to implement the desired behavior of the agent.


AgentScript
Central control base class for an agent
+...
+...()
+Update()
#processMsgAgentMovment()
#processMsgAgentEyeFixation()
#processMsgAgentEyeMovment()
-...()
ChildrensRoomAgentScript
Agent control class "Children Playground" scenario
+...
+...()
#processMsgAgentMovment()
#processMsgAgentEyeFixation()
#processMsgAgentEyeMovment()
-...()
FiveObjectsAtTableAgent
Agent control class "Five objects at the table" scenario
+...
+...()
#processMsgAgentMovment()
#processMsgAgentEyeFixation()
#processMsgAgentEyeMovment()
-...()
LabyrinthAgentScript
Agent control class "Labyrinth" scenario
+...
+...()
#processMsgAgentMovment()
#processMsgAgentEyeFixation()
#processMsgAgentEyeMovment()
-...()

Analog to the AgentScript control class, the BehaviourScript controls the VR.

The agent(s) are spawned by the *awake()* function, but the way how this is done, is specified in the *AgentInitalization()* function, which must be defined in derived classes.
And like the AgentScript, the BehaviourScript has protected virtual functions for the Reset messages for the VR.


BehaviourScript
Central control over the VR
+...
+...()
+Awake()
#AgentInitalization()
#processMsgEnvironmentReset()
#processMsgTrialReset()
-...()
ChildrensRoomBehaviorScript
VR control class "Childrensroom" scenario
+...
+...()
#AgentInitalization()
#processMsgEnvironmentReset()
#processMsgTrialReset()
-...()
FiveObjectsAtTableBehaviour
VR control class "Five object at table" scenario
+...
+...()
#AgentInitalization()
#processMsgEnvironmentReset()
#processMsgTrialReset()
-...()
LabyrinthBehaviourScript
VR control class "Labyrinth" scenario
+...
+...()
#AgentInitalization()
#processMsgEnvironmentReset()
#processMsgTrialReset()
-...()