Lua Functions for use in the Dialogue System

These Lua Functions are all made for different systems to talk to each other, between the dialogue system, quest system, and inventory system. For example, if you need a dialogue option to give you a quest, look at “Dialogue System to Quest System”. If you need an NPC to be able to check your inventory during dialogue, look at “Dialogue System to Inventory System”, etc.

These are meant to be used in the dialogue entry Scripts and Conditions.

For examples on how to use these, see the tutorial videos as well as the example scene.

Dialogue System to Quest System

Lua Function List

Function	**Description**
Quests on Default Player	
CanOfferQuest(giverID, questID)	Checks if the quest giver can offer a quest to the player.
GiveQuest(giverID, questID)	Tells the quest giver with *giverID* to give the quest with *questID* to the player.
GiveAllQuests(giverID)	Tells the quest giver with *giverID* to give all of its quests to the player.
HasQuest(questID)	Returns true if the player has a quest with *questID*.
CurrentQuestState(questID)	Returns the state* of a quest in the player’s journal
SetQuestState(questID, state)	Sets the state* of a quest in the player’s journal
GetQuestNodeState(questID, nodeID)	Gets the state* of a quest node in the player’s journal
SetQuestNodeState(questID, nodeID, state)	Sets the state* of a quest node in the player’s journal
GetQuestNodeNumberState(questID, nodeNumber)	Gets the state* of a quest node (specifies the node *number*, not its ID)
SetQuestNodeNumberState(questID, nodeNumber, state)	Gets the state* of a quest node (specifies the node *number*, not its ID)
GetQuestCounterValue(questID, counterName)	Gets the value of a quest counter
SetQuestCounterValue(questID, counterName, value)	Sets the value of a quest counter

Message System	
SendMessageSystem(message, parameter)	Sends a message to the Message System
SendMessageSystemString(message, parameter, string)	Sends a message with a string value
SendMessageSystemInt(message, parameter, int)	Sends a message with an integer value
Misc	
SetQuestIndicator(questID, entityID, state)	Sets an entity’s quest indicator for a specified quest**

Possible Fields

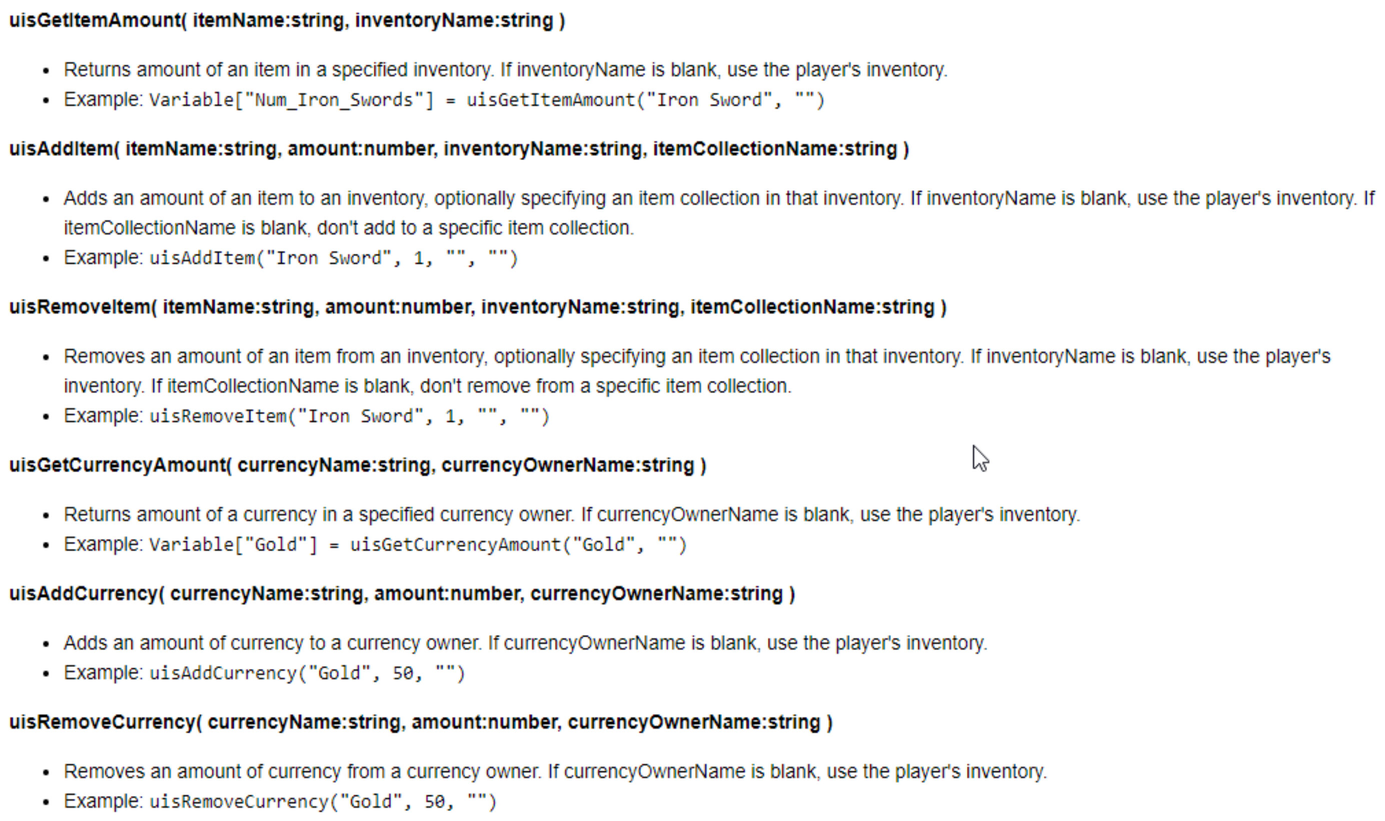
* Dialogue System quest state strings:

Dialogue System Quest State String	Description
“unassigned”	Quest is in waiting to start or disabled state
“active”	Quest is in active state
“success”	Quest is in successful state
“failure”	Quest is in failed state
“abandoned”	Quest is in abandoned state

** Quest indicator state strings:

Indicator State String (any case, no spaces)
“None”
“OfferDisabled”
“TalkDisabled”
“InteractDisabled”
“Offer”
“Talk”
“Interact”
“Custom0” - “Custom9”

Dialogue System to Inventory System



uisGetItemAmount(itemName:string, inventoryName:string)

- Returns amount of an item in a specified inventory. If inventoryName is blank, use the player's inventory.
- Example: `Variable["Num_Iron_Swords"] = uisGetItemAmount("Iron Sword", "")`

uisAddItem(itemName:string, amount:number, inventoryName:string, itemCollectionName:string)

- Adds an amount of an item to an inventory, optionally specifying an item collection in that inventory. If inventoryName is blank, use the player's inventory. If itemCollectionName is blank, don't add to a specific item collection.
- Example: `uisAddItem("Iron Sword", 1, "", "")`

uisRemoveItem(itemName:string, amount:number, inventoryName:string, itemCollectionName:string)

- Removes an amount of an item from an inventory, optionally specifying an item collection in that inventory. If inventoryName is blank, use the player's inventory. If itemCollectionName is blank, don't remove from a specific item collection.
- Example: `uisRemoveItem("Iron Sword", 1, "", "")`

uisGetCurrencyAmount(currencyName:string, currencyOwnerName:string)

- Returns amount of a currency in a specified currency owner. If currencyOwnerName is blank, use the player's inventory.
- Example: `Variable["Gold"] = uisGetCurrencyAmount("Gold", "")`

uisAddCurrency(currencyName:string, amount:number, currencyOwnerName:string)

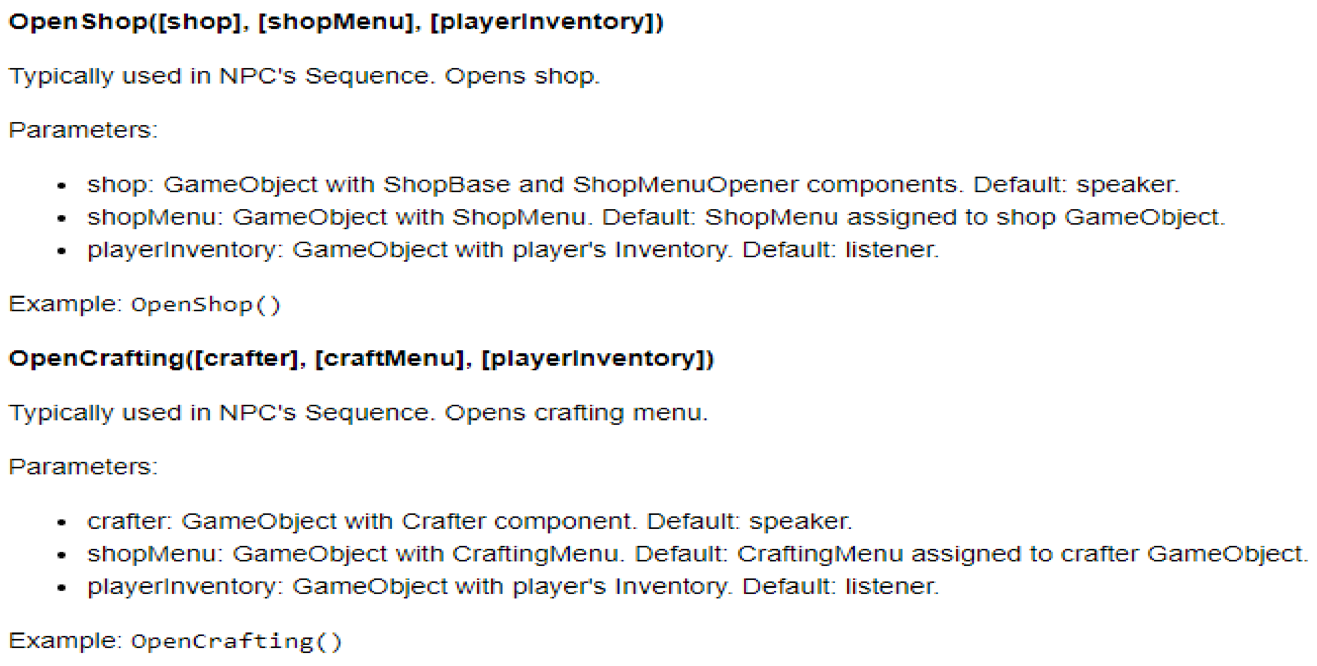
- Adds an amount of currency to a currency owner. If currencyOwnerName is blank, use the player's inventory.
- Example: `uisAddCurrency("Gold", 50, "")`

uisRemoveCurrency(currencyName:string, amount:number, currencyOwnerName:string)

- Removes an amount of currency from a currency owner. If currencyOwnerName is blank, use the player's inventory.
- Example: `uisRemoveCurrency("Gold", 50, "")`

Sequences

The following two items are used in the "Sequence" section of the dialogue system, while the above scripts are used in the Script section



OpenShop([shop], [shopMenu], [playerInventory])

Typically used in NPC's Sequence. Opens shop.

Parameters:

- shop: GameObject with ShopBase and ShopMenuOpener components. Default: speaker.
- shopMenu: GameObject with ShopMenu. Default: ShopMenu assigned to shop GameObject.
- playerInventory: GameObject with player's Inventory. Default: listener.

Example: `OpenShop()`

OpenCrafting([crafter], [craftMenu], [playerInventory])

Typically used in NPC's Sequence. Opens crafting menu.

Parameters:

- crafter: GameObject with Crafter component. Default: speaker.
- shopMenu: GameObject with CraftingMenu. Default: CraftingMenu assigned to crafter GameObject.
- playerInventory: GameObject with player's Inventory. Default: listener.

Example: `OpenCrafting()`