LRES – The Lightweight Roleplaying Experience System

1. About the LRES

The LRES is a system that distributes experience points based on roleplaying activity. The roleplaying activity is measured by the amount of spoken text/emotes that are not too short and do not include certain substrings, together with the amounts of PCs listening to it. It is not a goal of this system to reward lone players speaking only to themselves or the trees. One of the main goals of thi ssystem was to build a system that can be easily installed in any module or persistent world. Another goal of the system is to minimize the load on the heartbeat of the module.

2. How it works

Each PC speaking gets a dialog point for every other PC (or specially set up object) that hears what he says, as long as the string is not too short, does not include certain OOC marking substrings and he is not in an OOC area. If these dialog points raise above a certain mark, and the last time he got XP, or the time he logged on, is longer than the time limit, he is given a random amount of XP out of an intervall. Each time this happens, all his dialog points get removed, except for a residuum that will carry over. To get XP through the LRES again, he must again reach the minimum amount of dialog points necessary, and the minimum amount of time between two XP allotments must have passed. This minimum time can be set to be randomly taken out of an intervall for each player. A DM can block a player from getting XP through the LRES. If this block is removed, he is reseted, as if he just got XP. For every PC logging in the minimal amount of dialopg points necessary gets raised by a fixed amount of points. All these values are runtime customizable. It is also possible to have a linear factor for the XPs given, so that a player with a greater minimal time gets more XP. The factor is calculated by (random time to wait)/(niminal time to wait).

3. Configuring the LRES with the LRES-Configuration Wand

If used on a DM or through use on self by a DM the configuration menu with the following options appear:

3.1: The minimal XP values:
This value is the lowest amount of XP that is given.
3.2: The maximal XP value:
This value is the maximum amount of XP given.
3.3: Minimal time between two givings:
This is the minimal time that must pass between two XP givings, or the login and the first giving. If no randomization of time is active, this is the amount everybody must wait.
3.4: Maximal time between two XP givings:
This is the maximal time a player has to wait between two XP givings (provided he has enough dialog points of course).
3.5: The maximal residuum:
This is the maximal amount of dialog points that carries over after a XP giving.
3.6: Amount of minutes between two heartbeat runs to check if points need to be distributed:
This is the amount of time in minutes that pass between to runs of the check if someone will get XP. This sets the granularity of the XP giving mechanism. Leaving it in the 2-3 minutes range would be best, as the granularity gets to big if set too high.
3.7: The minimal dialog points needed:
This is the actual(absolut, raises from playernumber are already included) amount of dialog points a player needs to be allowed to get XP. This value is absolut. So players already logged in do not change this value. Players logging in after the change will raise this amount normally.
3.8: Raise of dialog points per PC:
This is the amount the minimal amount of dialog points gets raised for every player logging in.
3.9: Reset to defaults:
Resets all values that are changeable to the default values from mvd_02_const. After that the raise of dialog points for players logged in will get applied.
3.10: Activate/Deactivate random time for the players:
If activated, the minimal time a player must wait will get randomly choosen from the [min. time, max. time] intervall for every player after every XP allotment (or log in).
3.11: Activate/Deactivate linear factor:
If activated the random amount of XP a player will get is multiplied with
(random min. time to wait)/(minimal time to wait). This will reward players with longer wait times with slightly more XP.

3.12: Display current values:
This displays the cofigurable values of the LRES.
3.13: Force XP run:
This forces the XP distribution run, the same way as it is down if toggled through the heartbeat.
3.14: Activate/Deactivate LRES:
This option lets you activate/deactivate the LRES. If the LRES is activated after deactivation, all player will be reseted.

If the LRES-Configuration Wand is used on a player by a DM its current values are diplayed and the following options are available:

3.15: Block PC:
This will block the PC from any XP givings through this script. (A blocked player relogging will get XP again, as no status is saved at log out).
3.16: Unblock PC:
This will lift the block on a PC. This also resets the PC, meaning he will start with no more dialog points than the current residuum as has to wait the full minimal time calculated for him at unblocking.
3.17: Reset PC:
This will reset the PC (see above) without changing his blocked or unblocked status.

4. Installation

To install the LRES the erf file needs to be included. The placeable „LRES Main Hub“ needs to be put somewhere in the module. The configuration wand is optional, but should be included if you want to tune the LRES during runtime or block/unblock players. The best location for the LRES Main Hub would be an DM area that no player can reach. The wand and the LRES Main Hub are under Special->Custom1 under their apropriate object types.
Also in the following events/scripts the mentioned function must be called, and the include file included, as the following table shows:

Object	Event/Script	Include	Function
Module	OnHeartBeat	mvd_02_modheart	MvD_02_ModuleHeartBeat()
Module	OnClientEnter	mvd_02_init	MvD_02_PlayerEnter()
Module	OnClientLeave	mvd_02_init	MvD_02_PlayerExit()
Module	OnModuleLoad	mvd_02_init	MvD_02_ModInit()
Script	nw_g0_conversat	mvd_02_conversat	MvD_02_Conversat()

For performance reasons you might want to cache the scripts mvd_02_userevent, mvd_02_giver, mvd_02_conversat, nw_g0_conversat and nw_ch_ac4 (see the NWN familiar bug workaround).

If a NSC or another object that can listen should be configured as a dialogpoint giver the following events must call the functions according to the table:

Event	Include	Function
OnSpawn	mvd_02_init	MvD_02_NSCInit(OBJECT_SELF)
OnConversation	mvd_02_conversat	MvD_02_ConversatNPC()

It is not recommended to set NSCs or other non-player objects as listeners, as this can lead to abuse.

4.1 Installing the NWN Familiar Bug Workaround

The NWN Familiar Bug is a bug that is triggered if the master of a familiar hears its own familiar speak and is listening, while possessing the familiar. If that happens the focus automagically resets to the master and the possession is broken. This is a NWN Bug and has nothing to do with the LRES.
To install the workaround, do the following:

1. Change the function call SummonFamiliar() in nw_s2_familiar to MvD_02_SummonFamiliar(OBJECT_SELF). Also include the file mvd_02_familiar wit a normal include.
2. Change every occurence of SummonFamiliar(oMaster) in the non NWN Scripts to MvD_02_SummonFamiliar(oMaster). The oMaster is a general term here, MvD_02_SummonFamiliar gets the same parameters as the SummonFamiliar function. Also you need to include mvd_02_familiar in these scripts.
3. At the end of the file nw_ch_ac4 call the function MvD_02_ConversatFamiliar(). For that you need to include the file mvd_02_conversat.

4.2 Changing the OOC tests and OOC area marker

In the script mvd_02_conversat the two functions MvD_02_OOCMatchCase and MvD_02_AACMatchNoCase test if a spoken text/emote should be flagged as OOC and not give out dialog points. They test if at least one of the OOC constants from mvd_02_const match. MvD_02_OOCMatchCase does a case sensitive check against all constants of the form cMvD_02_sOOC[number]. MvD_02_OOCMatchNoCase does a case insensitive check against the constants of the form cMvD_02_sOOCWord[number]. To include more tests simply add another constant and else-if clause in the appropriate function. To remove a test simply comment the else-if claus out in the appropriate function. To simply change some tests, just change the value of the constant in mvd_02_const. All case insensitive constants need to be lowercase.
Every area whose name includes the substring defined in cMvD_02_sOOCAreaName is marked as an OOC area. In these areas no dialog points are given. To change the marker, simply change cMvD_02_sOOCAreaName.

4.3 Important Constants in mvd_02_const

Name	Default	Description
cMvD_02_iListenPatternNumber	5000	The number of the listen pattern used
cMvD_02_iPCRandomTime	TRUE	TRUE: Each player must wait a random time between XP allotments. FALSE: No random time
cMvD_02_iStrMinLength	2	The minimal length a string must have to count
cMvD_02_iCountPerPC	5	The amount the minimal needed dialog points gets raised per PC
cMvD_02_iMinDialogCount	25	Minimal aount of dialog points needed for 0 players
cMvD_02_iHeartBeatTime	2	Every which minute the script checks if XP need to be distributed
cMvD_02_iMinTimeDiff	7	Minimal time after which a PC can get another XP allotment
cMvD_02_iMaxTimeDiff	15	Maximal time after which a PC can get another XP allotment
cMvD_02_iMaxResiduum	10	The maximal amount of dialog points that carry over
cMvD_02_iMinXP	10	The minimal amount of XP
cMvD_02_iMaxXP	60	The maximal amount of XP
cMvD_02_iIsXPMult	FALSE	If the linearfactor for XP should be activated
cMvD_02_sOOCX		A case sensitive substring that marks OOC
cMvD_02_sOOCWord		A case insensitive substring that marks OOC
cMvD_02_sHubTag	LRESMainHub	This constant defines the name for the LRES Main Hub
cMvD_02_UserEventNr	5000	The number of the UserDefinedEvent that gets send to the LRES Main Hub to trigger a distribution run
cMvD_02_sOOCAreaName	OOC	This defines a OOC area marking substring. I.e. If an area name contains this substring it is an OOC area where no dialog points are given

5. Example Module

Contained in the Zipfile is a sample module with a NSC set up to listen for dialogs like a player. Also a DM area with the configuration wand and the LRES Main Hub. All scripts not starting with mvd_02 are modified scripts. All modifications are done according to the installation section of this manual. If a modified script has no header by me it is a modified version of the original Bioware script. In these modified scripts all additions and changes can be found by searching for "MvD-". All additions are commentary enclosed, all changes are described by commentary.

Usage: Stand before the NSC ans speak some strings (30 strings would suffice. Please bear in mind that strings flagged as OOC do not raise your dialog points. The minimal amount of dialog points you need in the example module is 30 for one player. For every other player this amount get raised by 5). Now wait between 7 and 17 minutes and you should get XP. Please note that your first and second string might not be noticed by the NSC as the spawn scripts sometimes fires a bit late. This is due to the way NWN works and has nothing to do with the LRES.

6. Version and planned features

LRES version: LRES-0.9-Beta9.9
Manual version: 0.9.7

planned features:
currently none

7. Copyright

8. Disclaimer

This package, its contents and the contents thereof is provided as is. The author is not responsible for any loss, discomfort or any other wanted or unwanted consequences arising of the use or misuse of this package. By using any part of the LRES you absolve the author of any responsibility.

9. Contact

Web: http://de.geocities.com/mvd_moragon
e-mail: mvd_moragon@yahoo.de

Bugreports are welcome.
Spam, flames etc will be routed to /dev/null