Luxor 1 / Amun Rising

Documentation for Luxor 1 / Amun Rising

Getting Started with Luxor 1/Amun Rising Modding

So, you decided on making a Luxor 1 / Amun Rising mod? This page is dedicated to newcomers to modding the game. Luxor Amun Rising is often the modded game of choice due to it being an improvement (and added features) over Luxor 1, and has the same engine as said game.

What You Need

- A copy of the game. (Steam (might be broken for you) | Big Fish Games (is in a bundle pack))
- QuickBMS

Extraction

Luxor 1 and Amun Rising are packed within a MJZ file, which can be opened with QuickBMS. Luxor 1 and Amun Rising share the same engine, although Amun Rising is more recommended to be modded because of it's difficulties and it's ability to use `\n` in dialog boxes.

1. Download QuickBMS from here
2. Download the MumboJumbo script from here
3. Open QuickBMS and select the mumbojumbo script, and hit Select.
4. Go to your Luxor 1 (or Amun Rising) folder and select **data.mjz** in the `data` folder, then hit Open.
5. When another file select box appears to ask where to extract, **go up one folder** (this will take you to the game folder itself) and hit Save.
6. Repeat step 3.
7. Repeat step 4, but this time, **select english.mjz** and hit Open.
8. Repeat step 5.

Video explanation:

https://www.youtube.com/embed/CBskMSDVUHI

Now that is out of the way, you can either:

- Have seperate folders for each mod and copy the game files there
- Have mods be in their own folder and swap mods by renaming data folders

If you opt for the first option, make sure to copy Steam.dll for each copy (Steam), or copy the bundle pack executable and launch Luxor 1/Amun Rising from there. (Bundle Pack)

If you opt for the first option, simply rename the data folder to something else, or take it out, then add the mod's data folder.

If you opt for the second option, here is a visual guide: luxor-installation.png

What's Next?

Know your basic goals. If you're making a mod, at least have a general idea of where to go, and what to do.

Before you just jump in thinking you're going to make the next high quality mod. **stop**. Those aren't easy to do. First mods are often mediocre, and that's okay! What's important is that you learn and improve over time.

Start small. Level design is one of the most important aspects of most mods, as they aim to create a new experience for the player, so start with that. Since this is likely your first mod, try aiming for the standard vanilla level order: 25 maps, 88 levels. Focus on the extra things (UI, menus, sound) later on.

Assuming that you are reading this because you are a new modder in the scene, try making maps first as practice, but keep these in mind when you finally make the decision to create your very first mod.

Now that you know what to do, it's time to learn about making maps.

*.psys files

PSYS files are the visual effect files used in the Luxor 1 engine (used in 1 and Amun Rising). They can make or break the game's aesthetics, and if done right, can add to a level / the game, and can be stunning.

How to include effects to a ui file

Create a new uiParticleSystem child in your ui file (yes, this means that you can add effects to a level!).

```
    Child child_name = uiParticleSystem
    {
        X = 0
        Y = 0
        Depth = YourDepthHere
        File = your\path\here.psys
    }
```

Make sure your uiParticleSystem child name is unique. Not doing so could make the game load the wrong uiParticleSystem child.

One thing to note is that unlike sprites, which have their anchor point in the top-left corner, their anchor point is at the center. This also goes for the emitter in the psys file.

For example, setting an effect to x0 y0 will still go to the top-left corner as any other child does, but only the lower-right portion of the effect will be displayed in that position.

An easier way of saying this is: Treat psys file positioning as you would position the frog in Zuma.

Anatomy of a .PSYS File

We will look at a specific part of the "extra life" psys that happens when you receieve an extra life.

```
Emitter ankh
{
Type = Sprite
```

```
	Flags = EF_NONE
	StartParticles = 2
	MaxParticles = 2
	ParticleRate = 0.000000
	Sprite = data\sprites\particles\ankh.spr
	ColorRate = 0.000000
	AnimMin = 0
	AnimMax = 0
	AnimRate = 30
	FadeInEndTime = 0.000000
	FadeOutStartTime = 0.800000
	LifespanMin = 2.000000
	LifespanMax = 2.000000
	RenderDelay = 0.000000
	FirstFrameDelay = 0.000000
	PosX = 0.000000
	PosY = 0.000000
	SpawnRadiusMin = 0.000000 0.000000
	SpawnRadiusMax = 0.000000 0.000000
	StartVelMin = 0.000000 -25.000000
	StartVelMax = 0.000000 -25.000000
	Acc = 0.000000 0.000000
	DevDelay = 0.000000
	DevAngle = 0.000000 0.000000
	EmitterVelMin = 0.000000 0.000000
	EmitterVelMax = 0.000000 0.000000
	EmitterAcc = 0.000000 0.000000
	EmitterStart = 0.000000 0.000000
	EmitterLifespan = 0.250000 0.250000
}
```

At first glance it looks slightly simple - and it is simple!

Type

Not touched too much yet, keep it as Sprite.

Flags

These flags determine what this effect should do.

- `EF_NONE` - If your particle does not need any of these flags, use this flag.
- `EF_SPRITE_ANIM_LOOP` - makes the animated sprite loop, if it is animated.
- `EF_SPRITE_RANDOM_FRAME` - takes a random frame from the sprite, if it is animated.
- `EF_VEL_POSRELATIVE` - Particles' velocity will depend on initial particle position.
- `EF_LIFESPAN_INFINITE` - makes the particles' lifespan infinite.
- `EF_POS_RELATIVE` - If this flag is set then position of all particles will all the time be relative to emitter's position (for example, gem shadow moves together with gem, but sparks may not).
- `EF_ELIFESPAN_INFINITE` - Makes the emitter lifespan infinite.
- `EF_USE_COLOR_RATE` - makes the sprite use the palette from an image file (for example: the wild powerup's symbol changes colors taken from the data\bitmaps\powerups\wild_pal.jpg file)
- `EF_PALETTE_LOOP` - makes the palette loop back to the start. Used with EF_USE_COLOR_RATE.
- `EF_VEL_ORBIT` - makes the particles move along a circular trajectory.
- `EF_VEL_DEVIATION` - allows for DevAngle and DevDelay to be used.
- `EF_SOFTWARE` - unknown
- `EF_HARDWARE` - unknown

The Rest

- `StartParticles` - the amount of particles it will start with.
- `MaxParticles` - particle cap
- `ParticleRate` - ParticleRate=1/dt, where dt is time period (in seconds) between the particle spawning. If ParticleRate=1 then particle will be spawned each second. If ParticleRate=10 then particle will be spawned each 100 milliseconds (10 times per second).
- `Sprite` - the path to the spr file
- `Palette` - if EF_USE_COLOR_RATE is enabled, this will make the sprite color itself to the palette file defined here
- `Spline` - Requires a valid .obj path to be used. When it is used, the emitter will travel on the path specified. It does not loop and will immediately stop movement at the path's end.
- `ColorRate` - if EF_USE_COLOR_RATE is enabled, this will set the speed of the color change
- `AnimMin` - unknown/not touched yet
- `AnimMin` - unknown/not touched yet
- `AnimMax` - this will make the palette loop back to the start. Used with EF_USE_COLOR_RATE.
- `AnimRate` - animation speed
- `FadeInEndTime` & `FadeOutStartTime` - Both between 0 and 1. Relatively to the particle lifespan. For example, particle lifespan time is 5 seconds, FadeInEndTime=0.2 and FadeOutStartTime=0.6. Then the real (not relative to the lifespan) time before the Fading In End and Fading Out Start will be 0.2*5=1 and 0.6*5=3 seconds respectively.

- `LifespanMin` & `LifespanMax` - minimum lifespan (in seconds) of a single particle
- `RenderDelay` - untouched, likely self-explanatory
- `FirstFrameDelay` - Delay before the animation start in milliseconds.
- `PosX` & `PosY` - X and Y positions of the emitter.
- `SpawnRadiusMin` & `SpawnRadiusMax` - The minimum/maximum (in separate X and Y percentages) of the spawn radius.
- `StartVelMin` & `StartVelMax` - The minimum/maximum (in seperate X and Y percentages) of the velocity (in where do the particles go).
- `Acc` - The speed + relative position (in X and Y percentages) of how fast will the particles accelerate.
- `RotMin` & `RotMax` - The minimum/maximum degree of rotation.
- `RotVelMin` & `RotVelMin` - The minimum/maximum speed of how fast it rotates.
- `DragMin` & `DragMax` - unknown
- `DevDelay` & `DevAngle` - Works only if EF_VEL_DEVIATION flag is set. The particles' moving direction will start to be changing after the DevDelay time will pass.
- `EmitterVelMin` & `EmitterVelMax` - Emitter velocity. If there is a set Spline, these will be the speeds of the emitter moving along the spline.
- `EmitterAcc` - Emitter acceleration. Usually, this is double of the EmitterVelMin and EmitterVelMax values.
- `EmitterStart` - Emitter spawn delay.
- `EmitterLifespan` - Emitter lifespan.

Examples

When done right, you can do awesome effects like these:

An extra from Luxor Mod 2020:

https://www.youtube.com/embed/IIEvL_YCh_8

Sniper Bullet:

https://www.youtube.com/embed/TrQIfTqQ2Jw

Keep In Mind

- If it's something such as a ball collapse, make sure the particle count isn't high, as too much can overload the game. Keep it at the default value if you're not too sure!
- Luxor 1/AR does not have blending modes, so try to make things look not weird.
- The Scorpion shares a collapse with the Wild Ball, so try to make the wild ball collapse psys not colored.

- When adding temporary particles in a ui file, such as a dialog box, keep in mind that particle effects that don't have the `EF_LIFESPAN_INFINITE` flag do not reload.

Creating Maps

Now that you have decided to create a mod (or make one or two levels), it's time to create a map.

What You'll Need

- A graphics editor (Paint.net, GIMP, Adobe Photoshop, just something that you can do well in.)
- Nocturnal Owl De-vertice Editor, found **here**
- Some form of text editor. Notepad can do, although you may want to use Notepad++ for clarity.

Creating your Path Concept

Luxor spheres are 32 x 32 pixels thick. You can either have a 32 px thick stroke, or thinner. It doesn't matter, as long as the player has a general idea of how the path goes. Here's an image of a simple map: rows, but it's not plain rows. luxor-path-example-1.png

Now, there are a few things to note when making paths. You should avoid going down 500px or lower, as this is where the shooter area is. Clipping between the shooter and the spheres is close here.

Now that that's out of the way...

Creating the Background

You don't need to be a god at art to create a background. Just make sure it's decent and all. There are plenty of tutorials on the internet if you aren't much of an art prodigy - but try experimenting with plugins and effects, and you'll probably have something going.

The path should be visible to the player, so that they have a general idea of how does the path go.

For demonstration purposes, we'll go with a dirt and rock background.
luxor-path-example-1-decorated.png

Creating the Level Folder

1. Navigate to `data\maps` of your game. You should see a bunch of folders with no spaces and full of level names.
2. Create a new folder and name it to your level's name. For this tutorial, we'll name it `RockyRoad`.
3. Export the background you just made as a JPG and name it `background.jpg`.

Creating the path in N.O.D.E

Now is the time to download and extract N.O.D.E if you haven't already. If you haven't already, get it **here**.

Before all of that, it's recommended to edit the background a little bit to guide you on what are you doing. Make the path lighter and add a 40px section (not including rounded tips) to the end of the path, as shown: luxor-path-example-1-decorated-nodeready.png Save this version of the background to the N.O.D.E folder, replacing `background.jpg` in the same folder as N.O.D.E.

This is where the tool comes in. Either open the exe file, the swf file or the html file.

4. Click on Start, and if it is your first time booting, enter a username. Ignore all of the other buttons, you will want to click on **New Path**. NODE-tutorial-1.png
5. Disable hotkeys, then name the level in the "Map Name" box. Enable the hotkeys again and focus on a text box that only requires numbers (such as the X/Y manual adding). You will want to remember 2 hotkeys: **H** for making the next node hidden / visible, and **Del** for deleting your last placed vertice. uQuNODE-tutorial-2.png
6. Now trace the path. Use Magnifier and lower your mouse DPI if you need to do so. (For the 40px ending. N.O.D.E automatically hides your last two vertices on saving the path because of how the hidden nodes function work in Luxor. For demonstration purposes, here it is with the two vertices already placed in as hidden.) NODE-tutorial-3.png
7. Click on Save and you should see what looks like an obj file. Copy the contents. NODE-tutorial-4.pngGet a plaintext editor, paste the contents there and save it as `path.obj` in your map folder, in this case `data\maps\RockyRoad`.
8. Back to N.O.D.E, click on **Toggle OBJ & UI/GVF**. This changes the dialog box's layout. Click on the pyramid that suits the level's ending best. In this case, we'll go with the right-facing pyramid, because the danger zone (the ending area of a path) goes to the left. NODE-tutorial-5.pngHere's a handy protip: Virtually all Luxor 1/AR files are in **plaintext** - this includes UI, level settings, sprite file settings, et cetera.
9. Copy the top text box's contents into a plaintext editor. You will want to remove this line. Save this file as `map.ui` in the level folder.

```
GLSprite = 0 0 GameBackgroundSprites data\maps\RockyRoad\mask.spr
```

9. Now, create a new file. Paste this in, replacing `RockyRoad` with your folder name. Save this in your level folder as `background.spr`.

```
data\maps\RockyRoad\background.jpg
none
800 600
1
1
0 0
```

Applying and Testing your Level File

10. Head over to `data\levels` and open `level_1_1` (Stage 1-1, or the first level). Replace "DIE KHUFU DIE" (or "URAEUS NEFERTARI") with your folder name **in all capital letters with spaces**. level-apply.png
11. Now open up the game, choose a difficulty if you're in Amun Rising and click on Start. You should now see your new level. rocky-road.png

And that's it! You've created your first Luxor level. Congratulations! You probably want to have an in-depth look of the lvl file format, which can be found HERE.

Tuning Level Settings

Level files are what loads the map and difficulty settings, They can be found in `data\levels`. You will want to look at the `level_x_x.lvl` files.

Level Files

Here is an example: Luxor Amun Rising's 1-1.

```
mapFile = "DIE KHUFU DIE"

// Ball Colors
spawnColor_1 = true
spawnColor_2 = true
spawnColor_3 = true
spawnColor_4 = true
spawnColor_5 = false
spawnColor_6 = false
spawnColor_7 = false
spawnStreak = 225

// Powerups
powerup_reverse   = true
powerup_slow      = true
powerup_stop      = true
powerup_shotspeed = true
powerup_lightning = true
powerup_bomb      = true
powerup_colorbomb = true
powerup_wild      = true
powerup_scorpion  = true

// Rewards
reward_gem_1      = true
reward_gem_2      = true
reward_gem_3      = true
```

```
reward_gem_4      = false
reward_gem_5      = false
reward_gem_6      = false
reward_gem_7      = false
reward_gem_8      = false
reward_gem_9      = false
reward_gem_10     = false
reward_gem_11     = false
reward_gem_12     = false
reward_gem_13     = false
reward_gem_14     = false
reward_gem_15     = false

// Win Condition (collapses and board clear)
winCondition = 100

// Vise Groups
viseGroupCount = 30

// Vise Speed
viseMaxSpeed = 800.0
viseMidMaxSpeed = 100.0
viseMidMinSpeed = 100.0
viseMinSpeed = 5.0
viseSpeedMaxBzLerp = 0.9 0.9
viseSpeedMidBzLerp = 0.25 0.75
viseSpeedMinBzLerp = 0.25 0.75

// Path Distances
viseSpawnDistance_1 = 0.6
midStartDistance_1 = 0.2
midEndDistance_1 = 0.6
```

Parameters

- `mapFile` - the folder name of your map with spaces and all in capital letters
- `spawnColor_n` - which sphere colors are activated. 1 is blue, 2 is yellow, 3 is red, 4 is green, 5 is purple, 6 is white and 7 is black.

- `spawnStreak` - This was supposed to increase / decrease the chances of clusters in a pusher train, though this parameter is **hardcoded.**
- `powerup_x` - Which powerups spawn. If you want to change a chance of a powerup spawning, edit `powerups.txt`.
- `reward_gem_x` - Which gems to spawn if a pusher train has been destroyed (not merged). For a reference of which gem gives a specific amount of points, refer to `powerups.txt`.
- `winCondition` - The amount of **spheres destroyed** to fill up the progress bar.
- `viseMaxSpeed` - the maximum speed of the spheres
- `viseMidMaxSpeed` - the middle-maximum speed of the spheres
- `viseMidMinSpeed` - the middle-minimum speed of the spheres
- `viseMinSpeed` - the minimum speed of the spheres, which is triggered on danger zone.
- `viseSpeed(Max/Mid/Min)BzLerp` - transitions(?)
- `viseSpawnDistance_n` - from 0 (0%) to 1 (100%), if there are no spheres in this area, spawn a new pusher train
- `midStartDistance_n` - from 0 (0%) to 1 (100%), trigger viseMidMaxSpeed here.
- `midEndDistance_n` - from 0 (0%) to 1 (100%), end viseMidMinSpeed here and trigger the viseMinSpeed.

For each path (up to 2), you must have a viseSpawnDistance, midStartDistance and midEndDistance. If there are no values for those, default values will be applied (unknown).

A better visual explanation: Luxor-lvl-demo.png

Every level slot you define in stage_select.uis must have their `level_x_x.lvl` file. The main menu level file is `level_0_0.lvl`.

powerups.txt

This txt file handles global powerup spawn chances and gem scoring. Everything here is self-explanatory.

```
// Powerups File
// Defines global powerup spawning chances and scoring


// Powerup Spawning
spawn_reverse      = 1000
spawn_slow         = 1000
spawn_stop         = 1000
spawn_speed_shot   = 1000
spawn_lightning    = 500
spawn_bomb         = 500
spawn_color_bomb   = 500
spawn_wild         = 1000
```

```
spawn_scorpion     = 500

// Powerup Scoring
scoring_coin       = 250
scoring_gem_1      = 1000
scoring_gem_2      = 2000
scoring_gem_3      = 3000
scoring_gem_4      = 4000
scoring_gem_5      = 5000
scoring_gem_6      = 6000
scoring_gem_7      = 7000
scoring_gem_8      = 8000
scoring_gem_9      = 9000
scoring_gem_10     = 10000
scoring_gem_11     = 11000
scoring_gem_12     = 12000
scoring_gem_13     = 13000
scoring_gem_14     = 14000
scoring_gem_15     = 15000
```