

# \*.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 receive 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` -  $\text{ParticleRate} = 1/\text{dt}$ , 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  $0.25=1$  and  $0.65=3$  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 separate 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` & `RotVelMax` - 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](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.

---

Revision #3

Created 20 November 2020 14:55:54 by CrushyGreenRen

Updated 17 May 2023 10:45:40 by CrushyGreenRen