PEX Tools Users Guide

Roaming Gamer, LLC. Version 1.0 30 March 2014

Table of Contents

This Guide

This is brief users guide to get you started with the Roaming Gamer PEX Tools.

Note: If you don't like reading, just look at the examples. :)

The tools that this guide comes with are designed to allow you to load and create a particle emitter from any of the following formats:

- **Particle Designer 2** (http://71squared.com/ ;***.json, *.png**) - With this tool, you can export a JSON encoded table (emitter definition) and an image file (the particle used by the emitter). This is an OS X only too.
- **Starling** (http://onebyonedesign.com/flash/particleeditor/;***.pex, *.png**) - With this tool, you can export an XML file (emitter definition) and an image file (the particle used by the emitter). This is a browser tool.
- **Roaming Gamer Emitter Editor** (http://roaminggamer.com/makegames/;***.rg, *.png**) - With this tool you can produce a JSON encoded table (emitter definition) and an image file (the particle used by the emitter). **This tool is available for the Corona Simulator, iOS, Android.**

Note: The Roaming Gamer 'format' is simply the input format used by the Corona SDK and output from my own particle editor tool. It isn't a formal format. I just needed to call it something.

Loading The 'Tools'

1. Copy pex.lua to the root folder of your project or to a folder of your choice.
2. Write this code:

```
local pex = require "pex"
```

Loading and Creating an Emitter

Syntax

Particle Designer 2	`pex.loadPD2( group, x, y, fileName [ , params ] )`
Starling	`pex.loadStarling( group, x, y, fileName [ , params ] )`
Roaming Gamer	`pex.loadRG( group, x, y, fileName [ , params ] )`

Function Arguments

group	A display group to place the emitter in. Pass `nil` to place emitter in 'current stage'.
x, y	< x, y > position to place emitter.
fileName	Full path and filename with extension of emitter definition file.
params	Optional settings and over-rides. See 'Options Settings and Overrides' below for more details.

Examples

Particle Designer 2

1. Copy emitter file and particle file to root folder of your game.
2. Write this code:

```
local emitter = pex.loadPD2( nil, 200, 100, "Comet.json" )
```

Starling

1. Copy emitter file and particle file to root folder of your game.
2. Write this code:

```
local emitter = pex.loadStarling( nil, 200, 100, "particle3.pex" )
```

Roaming Gamer

1. Cope emitter file and particle file to root folder of your game.
2. Write this code:

```
local emitter = pex.loadRG( nil, 200, 100, "emitter16178.rg" )
```

Options Settings and Overrides

This library is designed to be flexible and to allow you to place your assets where you want them. i.e. You don't need to put the emitters and particle files in the root folder of your game.

Additionally, you can over-ride any paramater(s) of an emitter definition that you choose.

Specifying Emitter and Particle Sources

Same Folders + Default Particle

Sometimes, you simply want to place your emitter definitions and particle images in a folder and then use to the emitters as the are. i.e. You want to use the default particle specified by the emitter.

1. Make a folder and move your emitter definitions and particle images to it. For this example, the folder is 'emitters/ParticleDesigner2/'
2. Write this code:

```
local emitter = pex.loadPD2( nil, 200, 100,
                 "emitters/ParticleDesigner2/Comet.json",
                 { texturePath = "emitters/ParticleDesigner2/" } )
```

Same Folders + Alternate Particle

Sometimes, you simply want to place your emitter definitions and particle images in a folder, but you want to use a different particle (in the same folder)

1. Make a folder and move your emitter definitions and particle images to it. For this example, the folder is 'emitters/ParticleDesigner2/'
2. Write this code (uses 'texture.png' instead of original texture):

```
local emitter = pex.loadPD2( nil, 200, 100,
                 "emitters/ParticleDesigner2/Comet.json",
                 { texturePath = "emitters/ParticleDesigner2/",
                   altTexture = "texture.png" } )
```

Different Folders

Sometimes, you want to place your emitter definitions in one folder and your particle images in another.

1. Make a folder a folder for your emitters and place them there (ex: 'emitters/ParticleDesigner2/')
2. Make a folder for you particles and place them there (ex: 'images/')
3. Write this code (uses 'start.png' from the 'images/' folder instead of original texture):

```
local emitter = pex.loadPD2( nil, 200, 100,
                 "emitters/ParticleDesigner2/Comet.json",
                 { altTexture = "images/star.png" } )
```

Overriding Emitter Parameters

Finally, you may wish to specify an emitter in the editor and later change a few paramaters. Instead of re-editing the emitter, simply pass in overrides to the loader/builder as follows.

In this example, we'll modify the following parameters:

- Duration – Make the emitter run for 3500 milliseconds then stop.
- Particle Lifespan – Give particles a lifespan between 500 and 1500 milliseconds.

```
local emitter = pex.loadPD2( nil, 200, 100,
                 "emitters/ParticleDesigner2/Comet.json",
                 { altTexture = "images/star.png",
                   duration = 3.0,
                   particleLifespan = 1.0,
                   particleLifespanVariance 0.5 } )
```

Emitter Parameters

Emitter Settings - Basic

Field Name	MIN	MAX	Decimal Places
maxParticles	1	2000	0
sourcePositionVariancex	0	2048	0
angle	0	360	2
emitterType	0	1	0
duration	-1; 0	infinite	2
sourcePositionVariancey	0	2048	0
angleVariance	-360	360	2
EmitterSettings-Gravity			
gravityx	-2000	2000	2
speed	0	2000	0
radialAcceleration	-2000	2000	2
tangentialAcceleration	-2000	2000	2
gravityy	-2000	2000	2
speedVariance	0	2000	0
radialAccelVariance	-2000	2000	2
tangentialAccelVariance	-2000	2000	2

Emitter Settings - Radial

maxRadius	0	1000	2
minRadius	0	1000	2
rotatePerSecond	-1000	1000	2
maxRadiusVariance	0	1000	2
minRadiusVariance	0	1000	2
rotatePerSecondVariance	-1000	1000	2

Particle Settings

particleLifespan	0.05	10	2
startParticleSize	0	512	2
finishParticleSize	0	512	2
rotationStart	0	360	2

rotationEnd	0	7200	2
particleLifespanVariance	0	10	2
startParticleSizeVariance	0	512	2
finishParticleSizeVariance	0	512	2
rotationStartVariance	0	360	2
rotationEndVariance	0	7200	2
	Color Settings		
startColorRed	0	1	9
startColorGreen	0	1	9
startColorBlue	0	1	9
startColorVarianceRed	0	1	9
startColorVarianceGreen	0	1	9
startColorVarianceBlue	0	1	9
finishColorRed	0	1	9
finishColorGreen	0	1	9
finishColorBlue	0	1	9
finishColorVarianceRed	0	1	9
finishColorVarianceGreen	0	1	9
finishColorVarianceBlue	0	1	9
	Alpha / Blend Settings		
startColorAlpha	0	1	2
finishColorAlpha	0	1	2
blendFuncSource	1	5	0
startColorVarianceAlpha	0	1	2
finishColorVarianceAlpha	0	1	2
blendFuncDestination	1	5	0