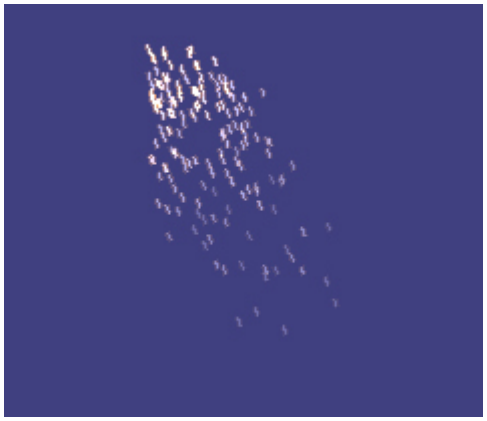
Creating Electrical Sparks

This very short tutorial will show you how to make electrical sparks with a spawner and a couple of FX.

The end result will look like this:



Setting up the configuration

```
[Sparks]
Spawner = SparksSpawner
LifeTime = 1.0

[SparksSpawner]
Object        = SparkObject
WaveSize      = 10
WaveDelay     = 0.05
ActiveObject  = 40
Rotation = -25 ~ -15 ; give a slight variance in the angle of each particle spark
UseRotation = true

[SparkObject]
Graphic = SparkGraphic
LifeTime = 1.0
Speed = (-50, -50, 0) ~ (50, 50, 0) ; when first created, have each spark shoot
                                    ; in random directions
FXList = SparkFallAwayFX # SparkFadeAwayFX ; use both FX so the sparks blow away
                                           ; and burn out

[SparkGraphic]
Texture = spark.png
BlendMode = add
Pivot = center
```

```
[SparkFallAwayFX] ; The simulated gravity on each spark
SlotList    = SparkFallAwayFXSlot
KeepInCache = true
Loop        = false

[SparkFallAwayFXSlot] ; this movement FX will pull the sparks down and right
                      ; like gravity and wind.
Type          = speed
Curve         = smooth
StartTime     = 0.0
EndTime       = 1.0
StartValue    = (0,0,0)
EndValue      = (150, 250, 0) ~ (140, 300, 0)
Period        = 1.0
Absolute      = false ; make the values relative so they move away from the
                      ; parent spawner, and not a fixed location on the
screen

[SparkFadeAwayFX] ; Have the sparks burn out to nothing
SlotList    = SparkFadeAwayFXSlot
KeepInCache = true
Loop        = false

[SparkFadeAwayFXSlot]
Type          = alpha
Curve         = linear
StartTime     = 0.0
EndTime       = 1.0
StartValue    = 1.0
EndValue      = 0.0
Period        = 1.0
Absolute      = true ; ensure absolute values for the alpha from 0.0 - 1.0
```

See the comments in the config above to see what part each FX plays on the particles.

Setting up Input

Just a quick mouse click handler to create a “Sparks” object on the screen. That will make the demo more fun to play with.

In your Init() method, add a handler for input so we can read the mouse to create sparks:

```
    orxEvent_AddHandler(orxEVENT_TYPE_INPUT, InputEventHandler);
```

Our event handler method with look like this:

```
    orxSTATUS orxFASTCALL InputEventHandler(const orxEVENT *_pstEvent) {

    if(orxInput_IsActive("Click") && orxInput_HasNewStatus("Click")) {
        orxVECTOR mousePosition = { 0,0,0 };
        orxVECTOR sparksPosition = { 0,0,0 };

        orxMouse_GetPosition(&mousePosition);

        orxOBJECT *sparks = orxObject_CreateFromConfig("Sparks");
        orxObject_GetPosition(sparks, &sparksPosition);

        sparksPosition.fX = mousePosition.fX;
        sparksPosition.fY = mousePosition.fY;

        orxObject_SetPosition(sparks, &sparksPosition);
    }

    return orxSTATUS_SUCCESS;
}
```

Need to define “Click” as our mouse button in the config:

```
[KeysForInput]
KEY_ESCAPE      = Quit
MOUSE_LEFT      = Click
```

Finished

All done. Click away and cause lots of electrical shorts.