Alien Aurora (FD23.js)

https://fdossena.com/?p=alienaurora/i.frag

Initialization

First, include fd23.js in your page's head section:

```
<script type="text/javascript" src="fd23.js"></script>
```

The animation can be drawn to a canvas element, we call this the target. For example, you could have something like this in the body:

```
<canvas id="myCanvas"></canvas>
```

Finally, to start the effect, put this code after the canvas has been declared:

```
<script type="text/javascript">
      var x=new FD23(myCanvas);
</script>
```

This will initialize the effect with the default settings. Feel free to customize the appearance of the canvas using CSS.

See example1.html for some code ready to copy-paste.

Multiple instances of FD23 can coexist on the same page without issues.

Destruction

The effect is only rendered while an element is visible on the screen, but if for some reason you need to remove the canvas from the page, you have to stop the effect first:

```
x.destroy();
//now you can remove the canvas
```

If the canvas is removed without stopping the effect first, it can impact the performance of the page.

If you don't have a reference to the instance of FD23 created during the initialization, you can get it from the canvas itself:

```
myCanvas.fd23instance.destroy();
//now you can remove the canvas
```

Customization

The easiest way to change the way the animation looks and plays is by passing a JSON string generated with editor.html during the initialization.

Example:

```
var x=new FD23(myCanvas,'{"hueCyclePeriod":"60"}');
```

Most settings can be also changed at any time while the animation is running by directly manipulating variables in the object.

Example:

```
x.targetHue=180; //smoothly change color to teal
```

Here's a full list of all the important variables in an FD23 instance.

Variable	Possible values	Animated?	Description
`targetHue`	0-360 default: 245	Yes	Base hue for all color calculations
`targetSaturation`	0-100 default: 75	Yes	Base saturation for all color calculations
`targetLightness`	0-100 default: 56	Yes	Base lightness for all color calculations
`targetAlpha`	0-100 default: 40	Yes	Alpha value for the particles (called Effect intensity in the editor)
`targetHueVariance`	Any positive value default: 43	Yes	How much the hue of the particles can deviate from the `targetHue` in positive or negative
`targetSaturationVariance`	Any positive value default: 0	Yes	How much the saturation of the particles can deviate from the `targetSaturation` in positive or negative
`targetLightnessVariance`	Any positive value default: 0	Yes	How much the lightness of the particles can deviate from the `targetLightness` in positive or negative
`colorAdjustmentSpeed`	Any positive value default: 1	No	The speed at which colors change when any of the previous variables are modified
`hueCyclePeriod`	0 (disabled) Any positive value (enabled) default: 0	No	When enabled, the hue is automatically changed over the specified length of time (in seconds)
`compensatePerceivedBrighness`	Boolean default: false	No	When enabled, the lightness value is automatically adjusted based on the perceived brightness (meaning that colors around yellow and green will be darkened a bit). Useful in combination with `hueCyclePeriod`. Don't use with high values for `targetHueVariance`
`nParticles`	Any positive value default: 5	No	The number of particles in the animation
`ringsPerParticle`	Any positive value default: 2	No	The number of rings in each particle. Expensive on firefox for values >=4
`targetGradientBias`	Any positive value default: 0.85	Yes	How rings are shaped and spaced in a particle (called Ring shape in the editor). Higher values make the rings shrink toward the center of the particle, lower values make them grow towards the outside of the particle, 0 will make the particle a single color
`targetSpeed`	Any positive value default: 1	Yes	Overall speed of the animation (called Animation speed in the editor)
`targetAnimationIntensity`	Any positive value default: 1	Yes	How much "instability" to add to the particle movement (called Animation variability in the editor)
`targetRingDistortionIntensity`	Any positive value default: 1	Yes	Intensity of the animation that moves the center of each particle, distorting the rings. High values can cause glitches
`speedAdjustmentSpeed`	Any positive value default: 1	No	The speed at which the animation changes when any of the previous 4 variables are modified (called Animation adjustment speed in the editor)
`resolutionScale`	Any positive value default: 1	No	Rendering quality for the animation. Lower values can speed up rendering on slower devices or devices without hardware acceleration at the expense of quality. No special upscaling techniques are used
`maxFps`	Any positive value or 0 to disable cap default: 0	No	The maximum number of frames per second that can be rendered or 0 to disable the limit. Note that regardless of this value, the FPS will always be limited by the refresh rate of the display
`targetTOffset`	Any value, initialized at 0	Yes	Moves the animation forwards/backwards in time and can be used to make the animation respond to events such as a mouse scroll or something being tapped. Ignored in the JSON config
`tOffsetIntensity`	Any positive value default: 3	No	How much the animation should respond to changes in `targetTOffset`
`tOffsetAdjustmentSpeed`	Any positive value default: 1	No	How quickly the animation should respond to changes in `targetTOffset`
`paused`	Boolean default: false	No	Pause/unpause the animation. Note that regardless of this value, the animation will be paused if the target canvas is outside the viewport. Ignored in the JSON config.

Compatibility

FD23 works with all modern browsers.

License

GNU LGPL, version 3 or newer.

https://www.gnu.org/licenses/lgpl-3.0.en.html

It pretty much boils down to:

- You can use this library in your project, free of charge, regardless of your project's license
- You can study, modify, redistribute this library
- Any modification you make to it must remain under the same GNU LGPL license or a compatible license