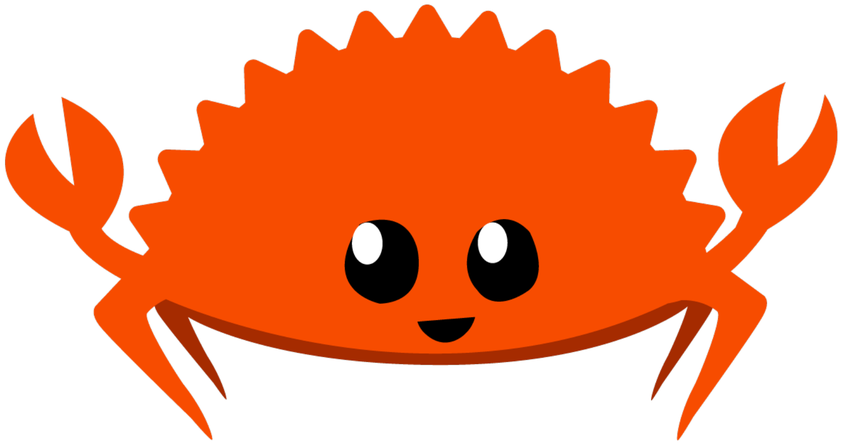
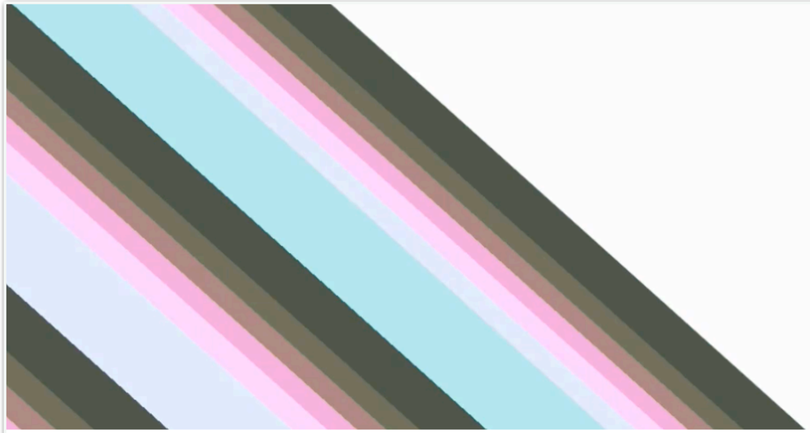
Rust Oslo

Lightning talks / Async Rust
Nov 26, 2019



Nannou

https://nannou.cc/



HOME DOCS GUIDE POSTS DONATE CODE

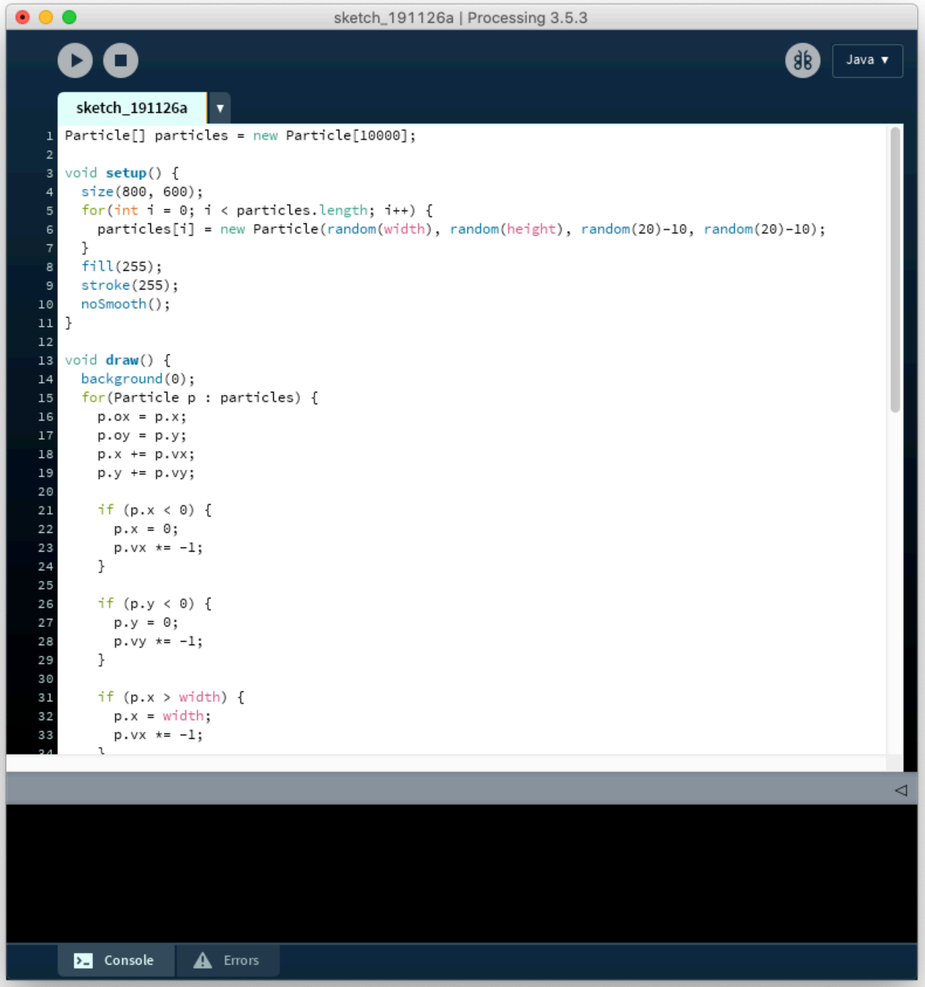
An open-source creative-coding framework for Rust

Graphics LEDs Lasers Audio GUI

Nannou is a library that aims to make it easy for artists to express themselves with **simple, fast, reliable** code.

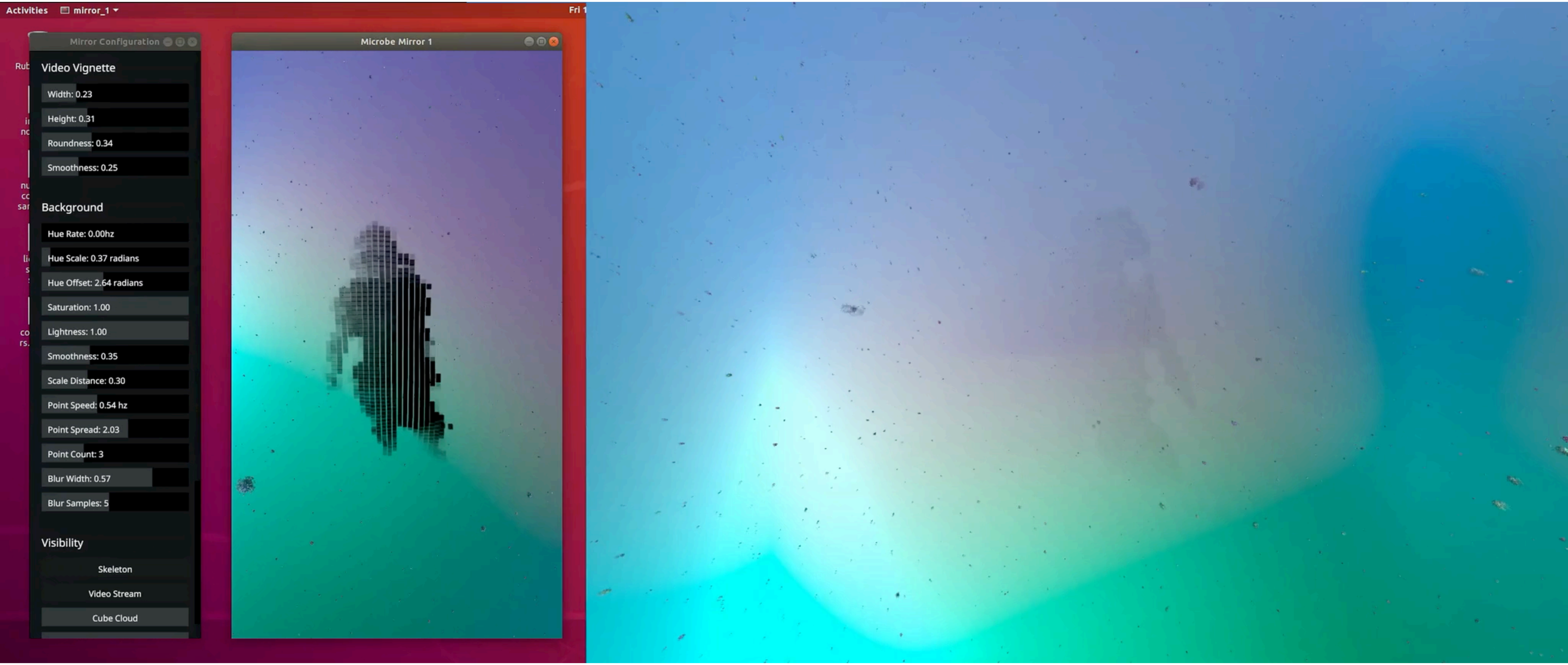
Batteries Included

One of the beauties of being a creative coder is that we have the potential to create works in a wide range of domains. Nannou aims to give equal priority to a full suite of creative I/O including graphics, multi-windowing, audio, lasers, lighting and more.


sketch_191126a | Processing 3.5.3
Java
sketch_191126a
Particle[] particles = new Particle[10000];
void setup() {
size(800, 600);
for(int i = 0; i < particles.length; i++) {
particles[i] = new Particle(random(width), random(height), random(20)-10, random(20)-10);
}
fill(255);
stroke(255);
noSmooth();
}
void draw() {
background(0);
for(Particle p : particles) {
p.ox = p.x;
p.oy = p.y;
p.x += p.vx;
p.y += p.vy;
if (p.x < 0) {
p.x = 0;
p.vx *= -1;
}
if (p.y < 0) {
p.y = 0;
p.vy *= -1;
}
if (p.x > width) {
p.x = width;
p.vx *= -1;
Console
Errors

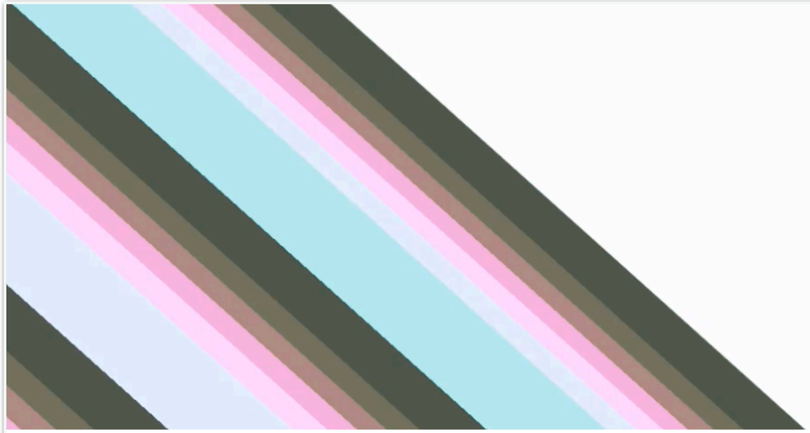
```
6
7   [dependencies]
8   nannou = "0.12"
9
```

```rust
use nannou::prelude::*;

struct Model {
}

fn main() {
    nannou::app(model)
        .update(update)
        .simple_window(view)
        .run();
}

fn model(_app: &App) -> Model {
    Model { }
}

fn update(_app: &App, _model: &mut Model, _update: Update) {
}

fn view(app: &App, _model: &Model, frame: &Frame) {
    let draw : Draw  = app.draw();

    draw.background().color(BLUE);

    draw.to_frame(app, &frame).unwrap();
}
```


Activities
mirror_1
Fri 1
Mirror Configuration
Video Vignette
Width: 0.23
Height: 0.31
Roundness: 0.34
Smoothness: 0.25
Background
Hue Rate: 0.00hz
Hue Scale: 0.37 radians
Hue Offset: 2.64 radians
Saturation: 1.00
Lightness: 1.00
Smoothness: 0.35
Scale Distance: 0.30
Point Speed: 0.54 hz
Point Spread: 2.03
Point Count: 3
Blur Width: 0.57
Blur Samples: 5
Visibility
Skeleton
Video Stream
Cube Cloud
Microbe Mirror 1

```rust
use nannou::prelude::*;
use nannou::rand::random_range;
use nannou::state::mouse::ButtonPosition::Up;

struct Model {
    particles: Vec<Particle>,
}

struct Particle {
    x: f32,
    y: f32,
    ox: f32,
    oy: f32,
    vx: f32,
    vy: f32,
}

fn main() {
    nannou::app(model)
        .update(update)
        .simple_window(view)
        .run();
}

fn model(_app: &App) -> Model {
    let count : usize  = 10000;
    let mut particles : Vec<Particle>  = Vec::with_capacity( capacity: count);
    for _i in 0..count {
        particles.push( value: Particle {
            x: random_range( min: -1000.,  max: 1000.),
            y: random_range( min: -1000.,  max: 1000.),
            ox: 0.,
            oy: 0.,
            vx: random_range( min: -10.,  max: 10.),
            vy: random_range( min: -10.,  max: 10.),
        });
    }
    Model { particles }
}

fn update(app: &App, model: &mut Model, _update: Update) {
    let pressed : bool  = app.mouse.buttons.left() ≠ &Up;
    let mouse_pos : Point2<S>  = app.mouse.position();

    for mut p : &mut Particle  in &mut model.particles {
        p.ox = p.x;
        p.oy = p.y;
        p.x += p.vx;
        p.y += p.vy;
        p.vx *= 0.98;
        p.vy *= 0.98;

        if pressed {
            let dist = mouse_pos.distance(pt2(p.x, p.y));
            p.vx -= 3. / dist * (p.x - mouse_pos.x);
            p.vy -= 3. / dist * (p.y - mouse_pos.y);
        }
    }
}

fn view(app: &App, model: &Model, frame: &Frame) {
    let draw : Draw  = app.draw();

    draw.background().color(BLACK);

    for p : &Particle  in &model.particles {
        draw.line()
            .start(pt2(p.ox, p.oy))
            .end(pt2(p.x, p.y))
            .weight(1.0)
            .color(rgb(
                r: (abs(p.vx) * 0.05).min(1.),
                g: (abs(p.vy) * 0.05).min(1.),
                b: 0.5,
            ));
    }

    draw.to_frame(app, &frame).unwrap();
}
```

https://nannou.cc/



HOME DOCS GUIDE POSTS DONATE CODE

An open-source creative-coding framework for Rust

Graphics LEDs Lasers Audio GUI

Nannou is a library that aims to make it easy for artists to express themselves with **simple, fast, reliable** code.

Batteries Included

One of the beauties of being a creative coder is that we have the potential to create works in a wide range of domains. Nannou aims to give equal priority to a full suite of creative I/O including graphics, multi-windowing, audio, lasers, lighting and more.