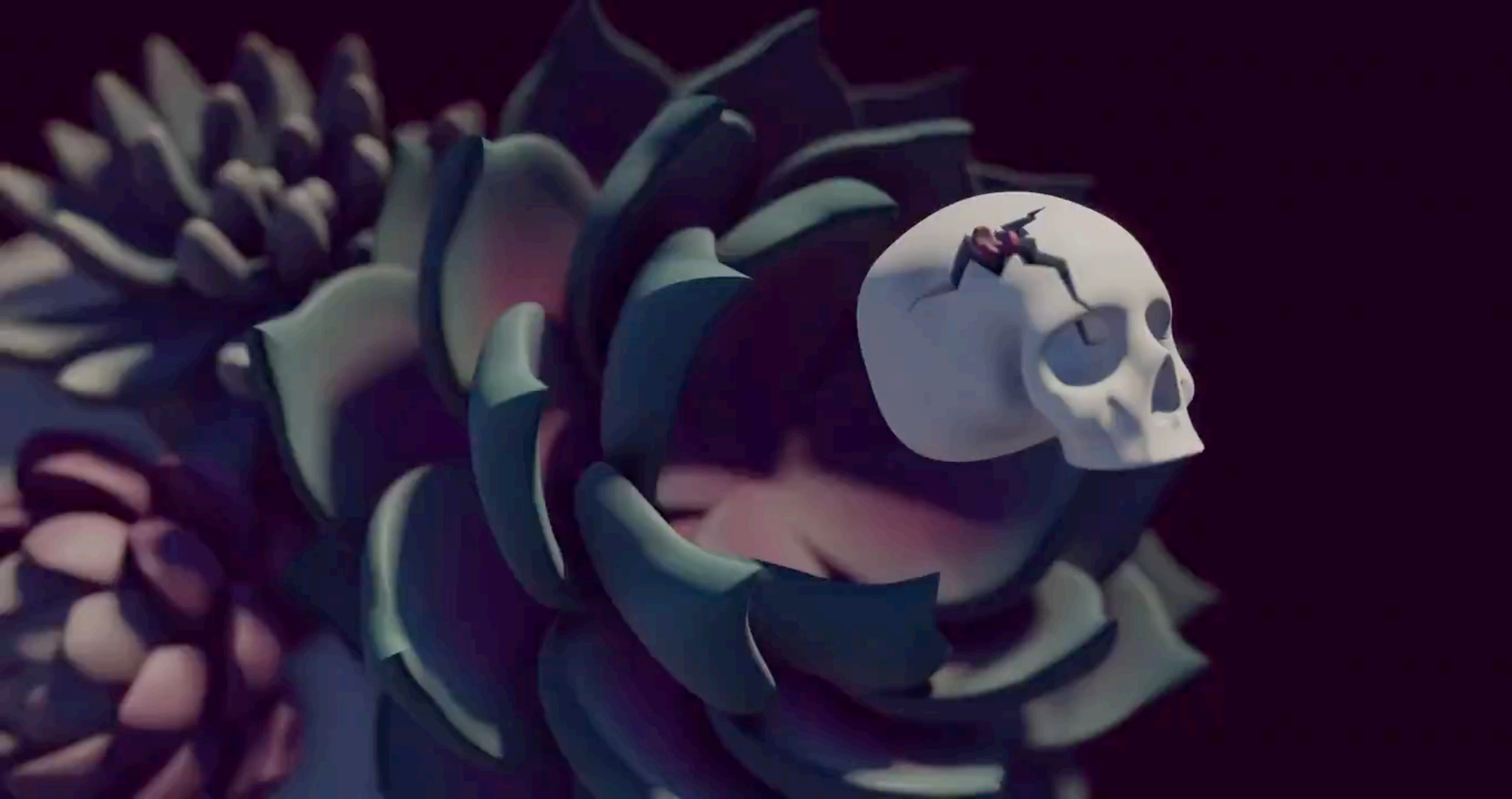

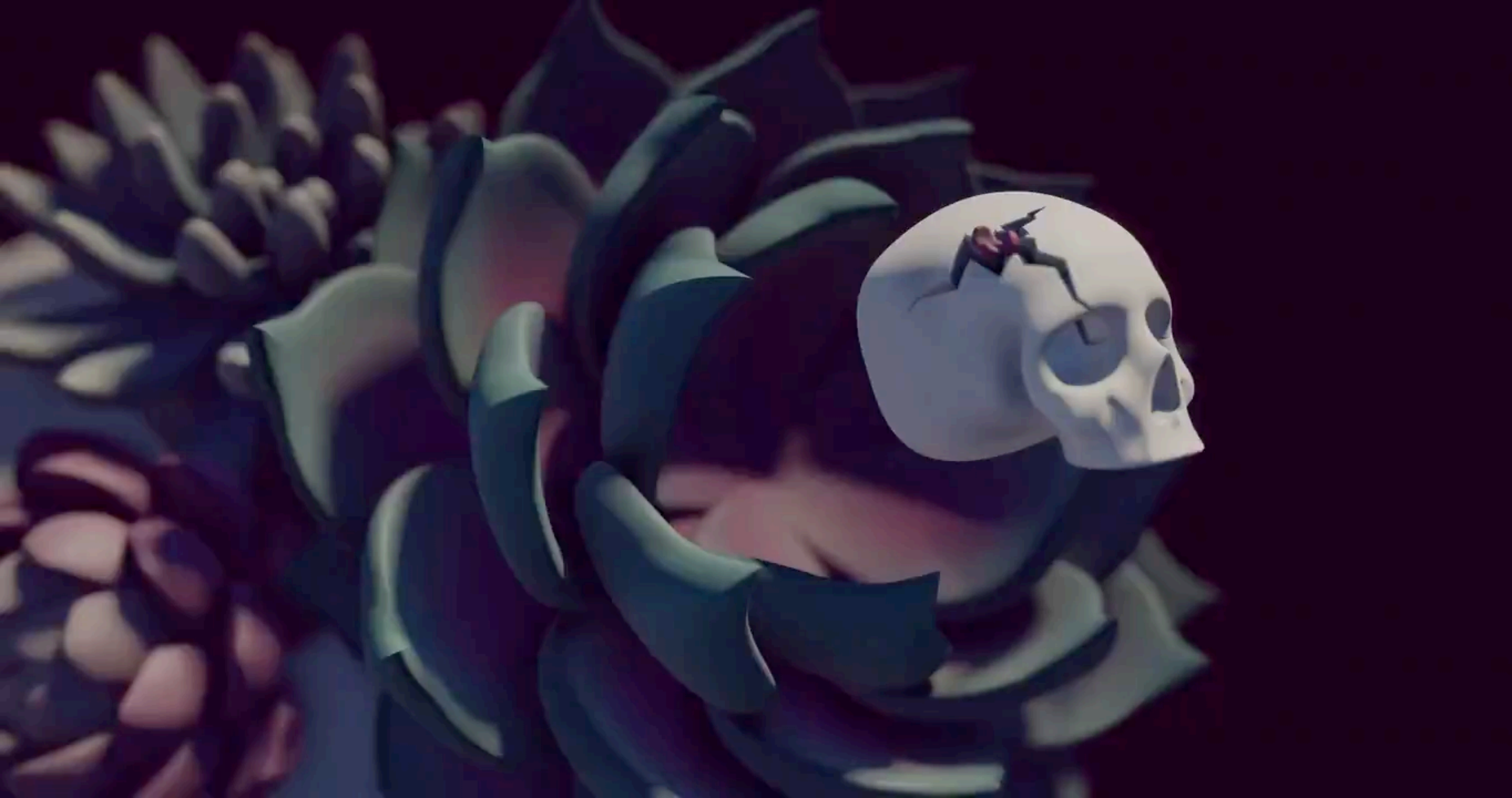

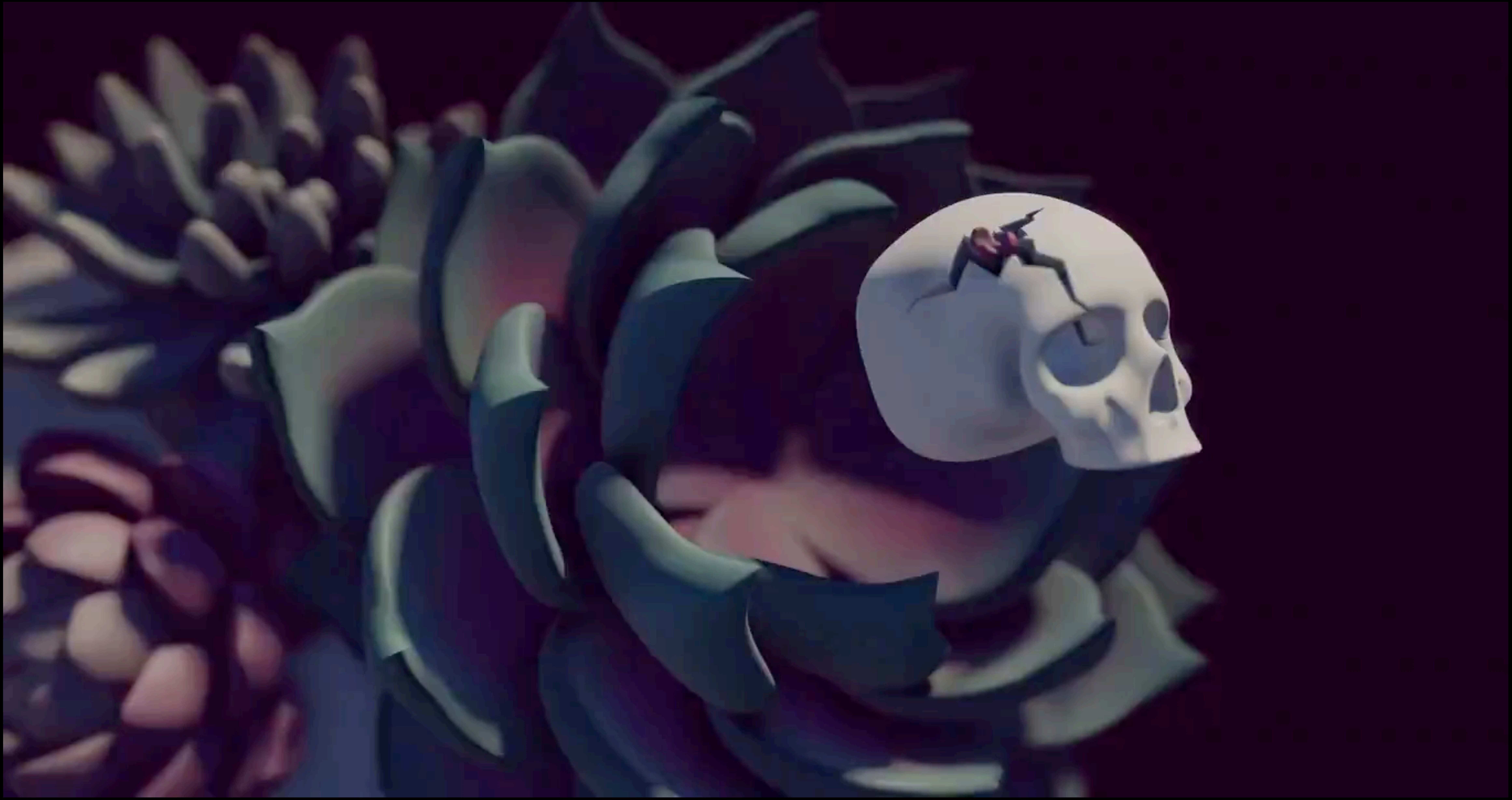

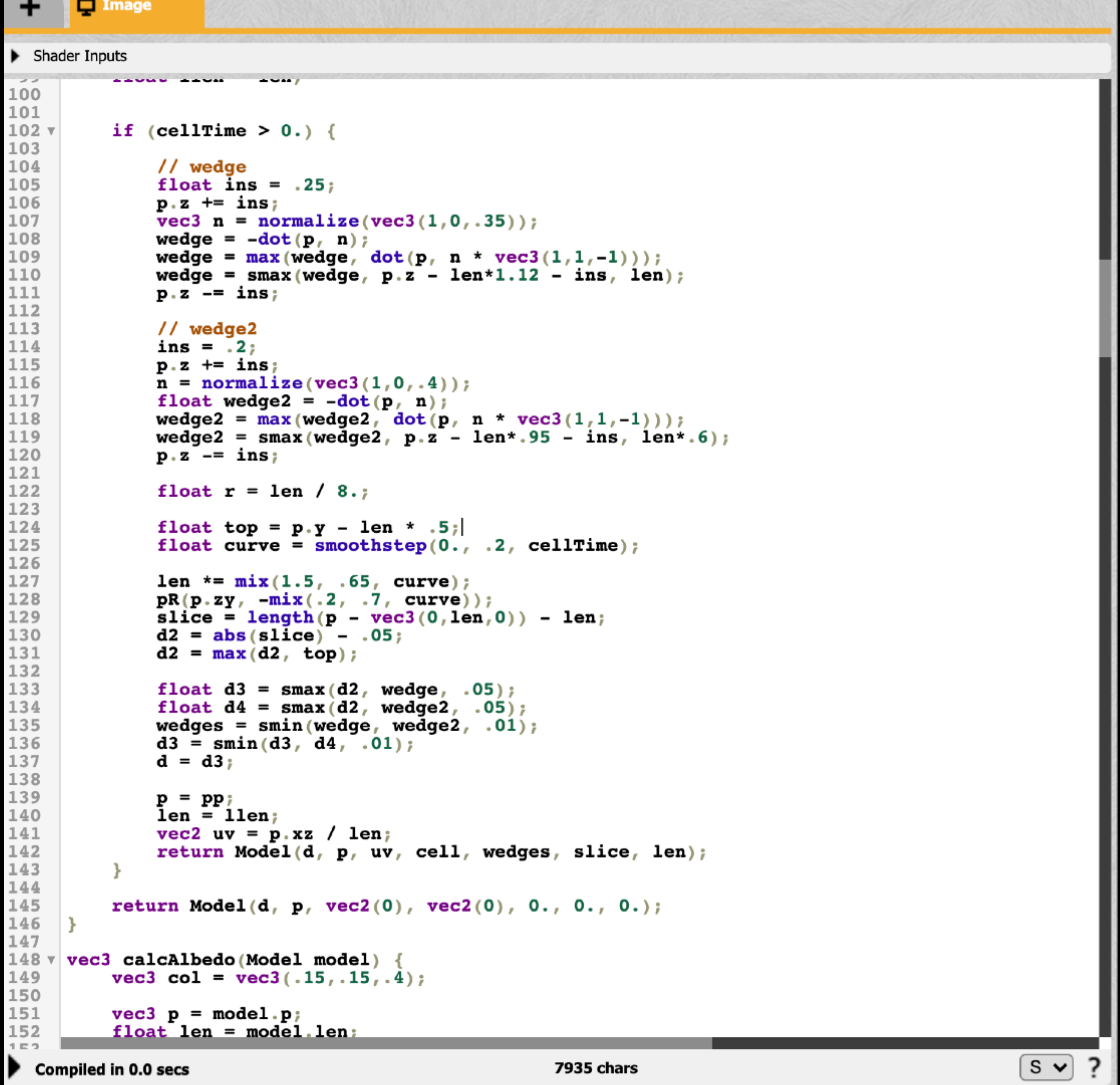

```
    float llen = len;

    if (cellTime > 0.) {

        // wedge
        float ins = .25;
        p.z += ins;
        vec3 n = normalize(vec3(1,0,.35));
        wedge = -dot(p, n);
        wedge = max(wedge, dot(p, n * vec3(1,1,-1)));
        wedge = smax(wedge, p.z - len*1.12 - ins, len);
        p.z -= ins;

        // wedge2
        ins = .2;
        p.z += ins;
        n = normalize(vec3(1,0,.4));
        float wedge2 = -dot(p, n);
        wedge2 = max(wedge2, dot(p, n * vec3(1,1,-1)));
        wedge2 = smax(wedge2, p.z - len*.95 - ins, len*.6);
        p.z -= ins;

        float r = len / 8.;

        float top = p.y - len * .5;
        float curve = smoothstep(0., .2, cellTime);

        len *= mix(1.5, .65, curve);
        pR(p.zy, -mix(.2, .7, curve));
        slice = length(p - vec3(0,len,0)) - len;
        d2 = abs(slice) - .05;
        d2 = max(d2, top);

        float d3 = smax(d2, wedge, .05);
        float d4 = smax(d2, wedge2, .05);
        wedges = smin(wedge, wedge2, .01);
        d3 = smin(d3, d4, .01);
        d = d3;

        p = pp;
        len = llen;
        vec2 uv = p.xz / len;
        return Model(d, p, uv, cell, wedges, slice, len);
    }

    return Model(d, p, vec2(0), vec2(0), 0., 0., 0.);
}

vec3 calcAlbedo(Model model) {
    vec3 col = vec3(.15,.15,.4);

    vec3 p = model.p;
    float len = model.len;
```

Compiled in 0.0 secs

7935 chars