

Kaydırdıkça şekil değiştiren parçacık teması nasıl yapılır

dec5.tech ana sayfasını Claude Code ile nasıl yaptım. Kopyalanabilir promptlar, tek dosyalık çalışan demo ve karşılaştığım hatalar.

SÜRE

30-45 dk

SEVİYE

Orta

ADIM

5

PROMPT

6

Sitenin ana sayfasındaki beyin, kaydırdıkça dağılıp ampule, dünyaya ve logoya dönüşüyor. Bunu yorumlarda çok soran oldu. Ama yorumlarda prompt kopyalanamıyor, o yüzden hepsini buraya koydum. Her bloğun sağ üstündeki **Kopyala** butonuna bas, yapıştır, devam et.

Ne yapacağız

Ekranda binlerce küçük üçgen var. Hepsi aynı anda birkaç şekle birden ait. Sayfayı kaydırdıkça bir şekilden diğerine akıyorlar. Fikir [Dala](#) adlı bir siteden geliyor, ben kendi siteme uyarladım.

Sonuçta elinde üç şey olacak:

1. Tarayıcıda hemen açabileceğin tek dosyalık bir demo
2. Kendi sitene uygulamak için sırayla vereceğin promptlar
3. Parçacıkların yerine kendi görsellerini koyma yöntemi

MALZEMELER

- [Claude Code](#). Ben bununla yaptım, promptlar ona göre.
- Bir web sitesi projesi. Benimki Next.js, ama demo düz HTML ile de çalışıyor.
- Ekran görüntüsü alabilmesi için Claude Code'a Playwright MCP bağlı olsun. Olmazsa 2. adımı sen elle yaparsın.

ARAÇLAR

Claude Code

Playwright MCP

three.js

01 Referansı bul

Beğendiğin bir site bul. Ben [Refero Styles](#) kullanıyorum. Orada gerçek sitelerin renk, font ve bileşen kuralları hazır duruyor.

Ama dikkat: benim ilk hatam buydu. Refero'daki renk tablosunu uyguladım, site sadece renk değiştirdi, hiç benzemedi. Bir sitenin asıl havası renginde değil, **hareketinde**.

02 Önce siteyi izlet, sonra yaptır

En önemli adım bu. Claude'a "şu siteye benzet" deme. Önce siteyi açıp izlemesini, kaydırmasını ve ne olduğunu sana anlatmasını iste.

PROMPT

Şu siteyi Playwright ile aç: <SİTE ADRESİ>

1. Hiçbir şeye dokunmadan 10 saniye izle, 3 saniyede bir ekran görüntüsü al.
2. Sonra sayfayı baştan sona adım adım kaydır, her adımda ekran görüntüsü al.
3. Bana bir tablo çıkar: her kaydırma aşamasında ekranda hangi şekil var, metin nerede duruyor, geçişler nasıl oluyor.

Henüz kod yazma. Sadece ne gördüğünü anlat.

Tabloyu okuyunca neyi kopyalayacağın netleşiyor. Dala'da çıkan akış şuydu: beyin → dağılma → ampul → dünya → logo.

03 Önce tek dosyada dene

Kendi projene girmeden mekanizmayı gör. Aşağıdaki kodu

`parcacik.html` diye kaydet ve tarayıcıda aç. Kurulum yok, three.js internetten geliyor.

Kaydırdıkça küre dağılıyor, halkaya, sonra küpe dönüşüyor. [Canlı hali burada.](#)

HTML

```
<!doctype html>
<html lang="tr">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-
scale=1" />
  <title>Parçacık Demo</title>
  <style>
    html, body { margin: 0; background: #000; color: #fff;
font-family: system-ui, sans-serif; }
    #scene { position: fixed; inset: 0; }
    section { min-height: 140vh; display: flex; align-items:
center; padding: 0 8vw; position: relative; }
    h2 { font-size: clamp(2rem, 6vw, 4.5rem); font-weight:
400; letter-spacing: -0.04em; max-width: 12ch; }
  </style>
</head>
<body>
  <div id="scene"></div>
  <section data-morph="0"><h2>Küre</h2></section>
  <section data-morph="1"><h2>Dağıl</h2></section>
  <section data-morph="2"><h2>Halka</h2></section>
  <section data-morph="3"><h2>Küp</h2></section>

  <script type="importmap">
    { "imports": { "three":
"https://cdn.jsdelivr.net/npm/three@0.184.0/build/three.modul
e.js" } }
  </script>
  <script type="module">
    import * as THREE from "three";

    const N = 5000;
    const COLORS = ["#8052ff", "#ffb829", "#ffffff",
"#e056fd", "#2fd3b5"];

    // Her şekil için N nokta üret. Aynı parçacık her şekilde
bir yer tutuyor.
    const sphere = (i) => {
      const u = Math.random() * 2 - 1, a = Math.random() *
```

```

Math.PI * 2, s = Math.sqrt(1 - u * u);
    return [s * Math.cos(a) * 1.6, u * 1.6, s * Math.sin(a)
* 1.6];
};
    const chaos = () => [(Math.random() - 0.5) * 12,
(Math.random() - 0.5) * 7, (Math.random() - 0.5) * 7];
    const torus = () => {
        const a = Math.random() * Math.PI * 2, b =
Math.random() * Math.PI * 2;
        const x = (1.4 + 0.45 * Math.cos(b)) * Math.cos(a), y =
0.45 * Math.sin(b), z = (1.4 + 0.45 * Math.cos(b)) *
Math.sin(a);
        return [x, y * Math.cos(1.1) - z * Math.sin(1.1), y *
Math.sin(1.1) + z * Math.cos(1.1)]; // biraz eđ
    };
    const cube = () => {
        const p = [Math.random() * 2 - 1, Math.random() * 2 -
1, Math.random() * 2 - 1];
        p[Math.floor(Math.random() * 3)] = Math.random() < 0.5
? -1 : 1; // yüzeye yapıştır
        return p.map((v) => v * 1.2);
    };
    const shapes = [sphere, chaos, torus, cube];

    const geo = new THREE.BufferGeometry();
    shapes.forEach((fn, s) => {
        const arr = new Float32Array(N * 3);
        for (let i = 0; i < N; i++) arr.set(fn(i), i * 3);
        geo.setAttribute("aP" + s, new
THREE.BufferAttribute(arr, 3));
    });
    geo.setAttribute("position", geo.getAttribute("aP0"));
    const col = new Float32Array(N * 3), seed = new
Float32Array(N * 2), c = new THREE.Color();
    for (let i = 0; i < N; i++) {
        c.set(COLORS[i % COLORS.length]);
        col.set([c.r, c.g, c.b], i * 3);
        seed.set([Math.random() * 6.28, Math.random()], i * 2);
    }
    geo.setAttribute("aColor", new THREE.BufferAttribute(col,
3));
    geo.setAttribute("aSeed", new THREE.BufferAttribute(seed,

```

```

2));

    const uniforms = { uMorph: { value: 0 }, uTime: { value:
0 }, uPixel: { value: 1 } };
    const mat = new THREE.ShaderMaterial({
        uniforms,
        transparent: true,
        depthWrite: false,
        blending: THREE.AdditiveBlending,
        vertexShader: `
            attribute vec3 aP0; attribute vec3 aP1; attribute
vec3 aP2; attribute vec3 aP3;
            attribute vec3 aColor; attribute vec2 aSeed;
            uniform float uMorph; uniform float uTime; uniform
float uPixel;
            varying vec3 vColor; varying float vRot; varying
float vSize;
            vec3 shape(float i) {
                if (i < 0.5) return aP0; if (i < 1.5) return aP1;
if (i < 2.5) return aP2; return aP3;
            }
            void main() {
                float i0 = floor(uMorph), f = fract(uMorph);
                float t = smoothstep(0.0, 1.0, clamp(f * 1.6 -
aSeed.y * 0.6, 0.0, 1.0)); // parçacıklar sırayla kalkar
                vec3 p = mix(shape(i0), shape(min(i0 + 1.0, 3.0)),
t);

                float a = uTime * 0.15; p = vec3(p.x * cos(a) + p.z
* sin(a), p.y, -p.x * sin(a) + p.z * cos(a));
                p.x += 1.3; // metne yer aç: şekli sağa al
                vec4 mv = viewMatrix * vec4(p, 1.0);
                gl_PointSize = min(0.05 * uPixel / -mv.z, 64.0);
                gl_Position = projectionMatrix * mv;
                vColor = aColor; vRot = aSeed.x + uTime * 0.3;
vSize = gl_PointSize;
            }`,
        fragmentShader: `
            varying vec3 vColor; varying float vRot; varying
float vSize;

            float seg(vec2 p, vec2 a, vec2 b) { vec2 pa = p - a,
ba = b - a; return length(pa - ba * clamp(dot(pa, ba) /
dot(ba, ba), 0.0, 1.0)); }

```

```

    void main() {
        vec2 p = gl_PointCoord * 2.0 - 1.0; float c =
cos(vRot), s = sin(vRot); p = mat2(c, -s, s, c) * p;
        vec2 a = vec2(0.0, 0.82), b = vec2(-0.71, -0.41), d
= vec2(0.71, -0.41);
        float e = min(min(seg(p, a, b), seg(p, b, d)),
seg(p, d, a)); // içi boş üçgen
        float px = 2.0 / vSize, alpha = 1.0 - smoothstep(px
* 0.5, px * 1.5, e);
        if (alpha < 0.01) discard;
        gl_FragColor = vec4(vColor, alpha * 0.8);
    },
});

const scene = new THREE.Scene();
scene.add(new THREE.Points(geo, mat));
const camera = new THREE.PerspectiveCamera(40, 1, 0.1,
50);
camera.position.z = 6;
const renderer = new THREE.WebGLRenderer({ alpha: true
});

document.getElementById("scene").appendChild(renderer.domElem
ent);

function resize() {
    const dpr = Math.min(devicePixelRatio, 2);
    renderer.setPixelRatio(dpr);
    renderer.setSize(innerWidth, innerHeight);
    camera.aspect = innerWidth / innerHeight;
    camera.updateProjectionMatrix();
    uniforms.uPixel.value = (innerHeight * dpr) / (2 *
Math.tan((camera.fov * Math.PI) / 360));
}
resize();
addEventListener("resize", resize);

// Kaydırma → hangi şekil: ekranın ortasına en yakın iki
bölüm arasında yumuşak geçiş.
let target = 0;
function onScroll() {
    const y = scrollY + innerHeight / 2;

```

```

const marks = [...document.querySelectorAll("[data-morph]")]
  .map((el) => ({
    y: el.offsetTop + el.offsetHeight / 2,
    i: Number(el.dataset.morph),
  }));
target = marks[0].i;
for (let k = 0; k < marks.length - 1; k++) {
  if (y > marks[k].y) {
    const t = Math.min(1, Math.max(0, ((y - marks[k].y) / (marks[k + 1].y - marks[k].y) - 0.25) / 0.5));
    target = marks[k].i + (marks[k + 1].i - marks[k].i) * t;
  }
}
addEventListener("scroll", onScroll, { passive: true });
onScroll();

const clock = new THREE.Clock();
renderer.setAnimationLoop(() => {
  uniforms.uTime.value = clock.getElapsedTime();
  uniforms.uMorph.value += (target - uniforms.uMorph.value) * 0.06;
  renderer.render(scene, camera);
});
</script>
</body>
</html>

```

Nasıl çalışıyor, kısaca:

- **Her parçacığın her şekilde bir adresi var.** `aP0`, `aP1`, `aP2`, `aP3` aynı parçacığın dört şekildeki yeri.
- **Kaydırma bir sayıya dönüşüyor.** `uMorph` 0 iken küre, 1 iken dağınık, 2 iken halka. Arada iki şekil karışıyor.
- **Parçacıklar aynı anda kalkmıyor.** `aSeed.y` her birine küçük bir gecikme veriyor. Şekil kayarak değil, dağılıp yeniden toplanarak değişiyor.

- **Üçgenleri shader çiziyor.** Her nokta ekranda küçük bir kare. Fragment shader o karenin içine boş bir üçgen çiziyor.

04 Kendi sitene uygula

Demo çalıştıysa asıl projeye geç. Promptları sırayla ver. Her birinden sonra sonucu tarayıcıda gör, sonra diğerine geç.

PROMPT

Ana sayfaya, sayfanın arkasında sabit duran (position: fixed) tam ekran bir three.js parçacık sahnesi ekle.

- Her parçacık aynı anda birkaç şekle ait olsun. Şekiller: beyin, dağınık kaos, ampul, dünya küresi, en sonda sitenin logosu.
- Her bölümün bir data-morph="<şekil numarası>" özelliği olsun. Ekranın ortasına en yakın bölüm hangisiyse parçacıklar o şekle aksın.
- Geçişte parçacıklar aynı anda değil, küçük gecikmelerle kalsın.
- Parçacıklar içi boş üçgen olsun, renkler: mor, amber, beyaz, pembe, camgöbeği.
- Kameraya yakın olanlar bulanıklaşsın (alan derinliği).
- prefers-reduced-motion açıksa animasyon olmasın.

PROMPT

Logo şeklini logo dosyasından üret: <LOGO DOSYASININ YOLU>
Görseli bir canvas'a çiz, dolu pikselleri bul, parçacıkları o piksellere rastgele dağıt.

PROMPT

Sayfayı Playwright ile masaüstünde (1440 genişlik) ve mobilde (390 genişlik) aç. Her data-morph bölümüne kaydırıp ekran görüntüsü al ve bana göster. Şekiller metnin üstüne biniyorsa ya da ekrandan taşıyorsa düzelt.

Son prompt çok önemli. Claude kendi sonucunu görmeden "tamam" derse, çoğu zaman bir şey eksik kalıyor.

05 Parçacıkların yerine kendi görsellerin

Benim sitemde beyni Instagram kapaklarım oluşturuyor. Bunu da ayrı bir adım olarak iste:

PROMPT

Parçacıkların bir kısmını, üçgenlerle birlikte, <GÖRSEL KLASÖRÜ> içindeki görsellerden oluşan küçük kartlar yap.

- Görselleri tek bir texture atlas'ta topla (her biri küçük boyutta, toplam birkaç yüz KB'ı geçmesin).
- Kartlar da aynı şekillere aksın.
- Kartlar dik dursun, dönmesin. Kenarları hafif yuvarlak olsun.
- Masaüstünde 150, mobilde 80 kart yeterli.

Tam boy görselleri yükleme. Benim 40 kapağım 6.8MB tutuyordu. Küçük sürümler 300KB'a indi.

PÜF NOKTALARI

Her şey bembeyaz, şekil seçilmiyor. Parçacıklar üst üste binip beyaza patlıyor. Parçacık sayısını düşür, çizgiyi incelt, arka yüzdeki parçacıkları soldur.

PROMPT

Parçacıklar çok parlak, şekil seçilmiyor. Parçacık sayısını azalt, üçgen çizgisini 1 piksele indir, şeklin arka yüzündeki parçacıkların opaklığını düşür.

Sahne hiç görünmüyor. Canvas'ı `z-index: -10` ile arkaya aldıysan, body'nin siyah arka planı onun üstüne boyanıyor. Canvas'a negatif z-index verme, onu sayfada içerikten önce koy, içeriğe `position: relative` ver.

Mobilde her şey tek bir beyaz kütle. Şekil küçülüyor ama üçgenler aynı boyda kalıyor. Dar ekranda parçacık boyutunu ve opaklığını düşür, şekli metnin üstüne al.

BİTTİ Mİ?

- ☐ Referans siteyi izlettim ve kaydirttim, akışı tablo olarak aldım
- ☐ Tek dosyalık demoyu açtım, çalıştı
- ☐ Sahne kendi sitemde arkada duruyor, metin okunuyor
- ☐ Masaüstü ve mobilde ekran görüntüsüyle kontrol ettim
- ☐ Hareket azaltma ayarı açıkken sayfa sabit duruyor

Takıldığı yeri Instagram'da yaz, dökümana eklerim.