

```

<!DOCTYPE html><html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Lumipod — Glow-Care Pet</title>
  <!-- Tailwind via CDN -->
  <script src="https://cdn.tailwindcss.com"></script>
  <meta name="description" content="Raise a bioluminescent creature by balancing hunger,
energy, hygiene and mood. Your choices change its glow, color and evolutions." />
  <link rel="icon" href="data:image/svg+xml,<svg xmlns='http://www.w3.org/2000/svg'
viewBox='0 0 64 64'><circle cx='32' cy='32' r='30' fill='%231a1a2e'/><circle cx='32' cy='32'
r='18' fill='%2338bdf8'/></svg>" />
  <style>
    @keyframes floaty { 0% { transform: translateY(0px); } 50% { transform: translateY(-6px);}
100% { transform: translateY(0px); } }
    @keyframes pulseGlow { 0% { filter: drop-shadow(0 0 0px rgba(255,255,255,0.4)); } 50% {
filter: drop-shadow(0 0 12px rgba(255,255,255,0.9)); } 100% { filter: drop-shadow(0 0 0px
rgba(255,255,255,0.4)); } }
    .floaty { animation: floaty 3s ease-in-out infinite; }
    .pulseGlow { animation: pulseGlow 2.5s ease-in-out infinite; }
    .btn { @apply px-3 py-2 rounded-2xl shadow-md font-medium transition active:scale-95; }
    .pill { @apply px-2 py-1 rounded-full text-xs font-semibold; }
  </style>
</head>
<body class="min-h-screen bg-gradient-to-b from-slate-900 via-slate-800 to-slate-900
text-slate-100">
  <div class="max-w-4xl mx-auto p-4 sm:p-6">
    <!-- Header -->
    <header class="flex items-center justify-between mb-4">
      <h1 class="text-2xl sm:text-3xl font-extrabold tracking-tight">Lumipod <span
class="text-cyan-300">Glow-Care</span></h1>
      <div class="flex items-center gap-2">
        <button id="saveBtn" class="btn bg-cyan-600 hover:bg-cyan-500">Save</button>
        <button id="resetBtn" class="btn bg-rose-600 hover:bg-rose-500">Reset</button>
      </div>
    </header><!-- Status Bar / Day-Night -->
    <section class="grid grid-cols-2 md:grid-cols-4 gap-2 mb-4">
      <div class="bg-slate-800/60 rounded-2xl p-3">
        <div class="flex items-center justify-between mb-1"><span>Day</span><span
id="dayCount" class="pill bg-slate-700">1</span></div>
        <div class="text-sm opacity-80">Cycle: <span id="timeOfDay">Morning</span></div>
      </div>
      <div class="bg-slate-800/60 rounded-2xl p-3">
        <div class="flex items-center justify-between mb-1"><span>Glow</span><span
id="glowTier" class="pill bg-cyan-700">Spark</span></div>
        <div class="w-full h-2 bg-slate-700 rounded-full overflow-hidden"><div id="glowBar"
class="h-full bg-cyan-400" style="width: 20%"></div></div>
      </div>
    </section>
  </div>

```

```

<div class="bg-slate-800/60 rounded-2xl p-3">
  <div class="flex items-center justify-between mb-1"><span>XP</span><span
id="xpLabel" class="pill bg-amber-700">0</span></div>
  <div class="w-full h-2 bg-slate-700 rounded-full overflow-hidden"><div id="xpBar"
class="h-full bg-amber-400" style="width: 0%"></div></div>
</div>
<div class="bg-slate-800/60 rounded-2xl p-3">
  <div class="flex items-center justify-between mb-1"><span>Stage</span><span
id="stageLabel" class="pill bg-fuchsia-700">Sprout</span></div>
  <div class="text-sm opacity-80">Trait: <span id="traitLabel">Curious</span></div>
</div>
</section>

<!-- Main Card -->
<main class="grid md:grid-cols-3 gap-4">
  <!-- Pet Canvas -->
  <section class="md:col-span-2 bg-slate-800/60 rounded-3xl p-4 sm:p-6 shadow-xl">
    <div class="flex items-center justify-between mb-3">
      <div class="flex items-center gap-2"><span class="pill bg-slate-700">Pet</span><span
id="petName" class="font-semibold">Nova</span></div>
      <div id="moodBadge" class="pill bg-emerald-700">Chill</div>
    </div>

    <div class="relative aspect-[16/10] bg-gradient-to-br from-slate-900 to-slate-800
rounded-2xl border border-slate-700 overflow-hidden">
      <!-- background stars -->
      <svg class="absolute inset-0 w-full h-full opacity-30" viewBox="0 0 600 400"
preserveAspectRatio="none">
        <defs>
          <radialGradient id="halo" cx="50%" cy="50%" r="50%">
            <stop offset="0%" stop-color="#ffffff" stop-opacity="1"></stop>
            <stop offset="100%" stop-color="#38bdf8" stop-opacity="0"></stop>
          </radialGradient>
        </defs>
        <g fill="#fff">
          <circle cx="40" cy="60" r="1"/>
          <circle cx="200" cy="100" r="1"/>
          <circle cx="120" cy="200" r="1"/>
          <circle cx="320" cy="50" r="1"/>
          <circle cx="540" cy="140" r="1"/>
          <circle cx="420" cy="300" r="1"/>
          <circle cx="80" cy="340" r="1"/>
        </g>
      </svg>

      <!-- Lumipod creature -->
      <div id="creature" class="absolute left-1/2 top-1/2 -translate-x-1/2 -translate-y-1/2 floaty
pulseGlow">

```

```

<svg width="220" height="220" viewBox="0 0 220 220"
xmlns="http://www.w3.org/2000/svg">
  <defs>
    <radialGradient id="bodyGrad" cx="50%" cy="40%" r="60%">
      <stop offset="0%" stop-color="#a5f3fc"/>
      <stop offset="100%" stop-color="#06b6d4"/>
    </radialGradient>
  </defs>
  <!-- halo -->
  <circle cx="110" cy="108" r="60" fill="url(#halo)" opacity="0.35"></circle>
  <!-- body -->
  <ellipse id="body" cx="110" cy="110" rx="62" ry="70" fill="url(#bodyGrad)" />
  <!-- face -->
  <circle id="eyeL" cx="90" cy="100" r="8" fill="#0f172a"/>
  <circle id="eyeR" cx="130" cy="100" r="8" fill="#0f172a"/>
  <path id="mouth" d="M 90 130 Q 110 145 130 130" stroke="#0f172a" stroke-width="5"
fill="none" stroke-linecap="round"/>
  <!-- antennae -->
  <path d="M70 70 Q 80 40 100 50" stroke="#a5f3fc" stroke-width="6" fill="none"
stroke-linecap="round"/>
  <circle cx="100" cy="50" r="6" fill="#a5f3fc"/>
  <path d="M150 70 Q 140 40 120 50" stroke="#a5f3fc" stroke-width="6" fill="none"
stroke-linecap="round"/>
  <circle cx="120" cy="50" r="6" fill="#a5f3fc"/>
</svg>
</div>

```

```

<!-- ground -->
<div class="absolute bottom-0 left-0 right-0 h-14 bg-gradient-to-t from-cyan-900/40
to-transparent"></div>
</div>

```

```

<!-- Stats -->
<div class="mt-4 grid grid-cols-2 sm:grid-cols-4 gap-3">
  <div class="bg-slate-900/60 rounded-2xl p-3">
    <div class="flex items-center justify-between mb-1"><span>Hunger</span><span
id="hungerLabel" class="pill bg-slate-700">50</span></div>
    <div class="w-full h-2 bg-slate-700 rounded-full overflow-hidden"><div id="hungerBar"
class="h-full bg-emerald-400" style="width: 50%;"></div></div>
  </div>
  <div class="bg-slate-900/60 rounded-2xl p-3">
    <div class="flex items-center justify-between mb-1"><span>Energy</span><span
id="energyLabel" class="pill bg-slate-700">50</span></div>
    <div class="w-full h-2 bg-slate-700 rounded-full overflow-hidden"><div id="energyBar"
class="h-full bg-violet-400" style="width: 50%;"></div></div>
  </div>
</div>

```

```
<div class="flex items-center justify-between mb-1"><span>Hygiene</span><span
id="hygieneLabel" class="pill bg-slate-700">50</span></div>
<div class="w-full h-2 bg-slate-700 rounded-full overflow-hidden"><div id="hygieneBar"
class="h-full bg-cyan-400" style="width: 50%"></div></div>
</div>
<div class="bg-slate-900/60 rounded-2xl p-3">
  <div class="flex items-center justify-between mb-1"><span>Mood</span><span
id="moodLabel" class="pill bg-slate-700">50</span></div>
  <div class="w-full h-2 bg-slate-700 rounded-full overflow-hidden"><div id="moodBar"
class="h-full bg-amber-400" style="width: 50%"></div></div>
</div>
</div>
```

```
<!-- Actions -->
<div class="mt-4 grid grid-cols-2 sm:grid-cols-4 gap-3">
  <button id="feedBtn" class="btn bg-emerald-600 hover:bg-emerald-500">Feed</button>
  <button id="playBtn" class="btn bg-amber-600 hover:bg-amber-500">Play</button>
  <button id="washBtn" class="btn bg-cyan-600 hover:bg-cyan-500">Wash</button>
  <button id="napBtn" class="btn bg-violet-600 hover:bg-violet-500">Nap</button>
</div>
```

```
<!-- Mini-quests -->
<div class="mt-4 bg-slate-900/60 rounded-2xl p-4">
  <div class="flex items-center justify-between mb-2">
    <h3 class="font-semibold">Explorations</h3>
    <span id="questCooldown" class="text-xs opacity-70">Ready</span>
  </div>
  <div class="grid sm:grid-cols-3 gap-2">
    <button data-quest="cave" class="quest btn bg-slate-700 hover:bg-slate-600">Crystal
Cave</button>
    <button data-quest="reef" class="quest btn bg-slate-700 hover:bg-slate-600">Singing
Reef</button>
    <button data-quest="grove" class="quest btn bg-slate-700 hover:bg-slate-600">Moonlit
Grove</button>
  </div>
  <p id="questLog" class="mt-3 text-sm opacity-90">Find rare shards to boost glow and
unlock evolutions.</p>
</div>
</section>
```

```
<!-- Sidebar -->
<aside class="space-y-4">
  <div class="bg-slate-800/60 rounded-3xl p-4">
    <h3 class="font-semibold mb-2">Lore</h3>
    <p class="text-sm opacity-90">Lumipods are pocket-sized, bioluminescent lifeforms from
the abyss. They respond to care with changing colors and brightness. Healthy bonds unlock
<span class="text-cyan-300">Glow Evolutions</span>.</p>
  </div>
```

```

<div class="bg-slate-800/60 rounded-3xl p-4">
  <h3 class="font-semibold mb-2">Achievements</h3>
  <ul id="achievements" class="list-disc list-inside text-sm space-y-1 opacity-90">
    <li id="ach_firstFeed" class="opacity-60">First Meal</li>
    <li id="ach_firstQuest" class="opacity-60">First Exploration</li>
    <li id="ach_glow100" class="opacity-60">Blinding Glow</li>
  </ul>
</div>
<div class="bg-slate-800/60 rounded-3xl p-4">
  <h3 class="font-semibold mb-2">Settings</h3>
  <label class="block text-sm mb-2">Name
    <input id="nameInput" class="mt-1 w-full rounded-xl bg-slate-900/70 border
border-slate-700 px-3 py-2" placeholder="Nova"/>
  </label>
  <button id="renameBtn" class="btn bg-slate-700 hover:bg-slate-600
w-full">Rename</button>
  <div class="mt-3 text-xs opacity-70">Autosaves every few seconds. No accounts, just
on-device storage.</div>
</div>
</aside>
</main>

```

```

<footer class="mt-6 text-center text-xs opacity-70">© <span id="year"></span> Lumipod —
An original virtual-pet concept by Cian & Amelia. Not affiliated with any other
franchise.</footer>

```

```

</div> <script>
// ---- Core State ----
const state = {
  day: 1,
  time: 0, // 0..3 -> Morning, Day, Evening, Night
  hunger: 50,
  energy: 50,
  hygiene: 50,
  mood: 50,
  glow: 20,
  xp: 0,
  stage: 'Sprout', // Sprout -> Bloom -> Radiant -> Mythic
  trait: 'Curious',
  name: 'Nova',
  cooldown: 0,
  achievements: { firstFeed:false, firstQuest:false, glow100:false },
};

const $ = sel => document.querySelector(sel);
const clamp = (v,min=0,max=100)=> Math.max(min, Math.min(max, v));
const lerp = (a,b,t)=> a+(b-a)*t;

```

```

// ---- Persistence ----
const STORAGE_KEY = 'lumipod_save_v1';
const load = () => {
  try { const saved = JSON.parse(localStorage.getItem(STORAGE_KEY)); if(saved)
Object.assign(state, saved);} catch(e){}
};
const save = () => localStorage.setItem(STORAGE_KEY, JSON.stringify(state));

// ---- UI Bindings ----
const $day = $('#dayCount');
const $time = $('#timeOfDay');
const $glowBar = $('#glowBar');
const $glowTier = $('#glowTier');
const $xpBar = $('#xpBar');
const $xpLabel = $('#xpLabel');
const $stageLabel = $('#stageLabel');
const $traitLabel = $('#traitLabel');
const $petName = $('#petName');
const $moodBadge = $('#moodBadge');

const statEls = {
  hunger: { bar: $('#hungerBar'), label: $('#hungerLabel') },
  energy: { bar: $('#energyBar'), label: $('#energyLabel') },
  hygiene:{ bar: $('#hygieneBar'), label: $('#hygieneLabel') },
  mood:   { bar: $('#moodBar'), label: $('#moodLabel') },
};

const achEls = {
  firstFeed: $('#ach_firstFeed'),
  firstQuest: $('#ach_firstQuest'),
  glow100: $('#ach_glow100'),
};

// Creature face dynamics
const eyeL = document.getElementById('eyeL');
const eyeR = document.getElementById('eyeR');
const mouth = document.getElementById('mouth');
const body = document.getElementById('body');

function moodDescriptor() {
  const avg = (state.mood + state.energy) / 2;
  if(avg > 75) return 'Ecstatic';
  if(avg > 55) return 'Chill';
  if(avg > 35) return 'Worried';
  return 'Gloomy';
}

function glowTier(g){

```

```

    if(g >= 90) return 'Nova';
    if(g >= 65) return 'Flare';
    if(g >= 40) return 'Spark';
    return 'Dim';
}

```

```

function maybeStageUp(){
    const thresholds = [0, 50, 120, 240]; // xp gates
    const labels = ['Sprout','Bloom','Radiant','Mythic'];
    let idx = 0; while(idx < thresholds.length && state.xp >= thresholds[idx]) idx++;
    state.stage = labels[Math.max(0, idx-1)];
}

```

```

function updateFace(){
    const smile = clamp(state.mood, 10, 90);
    // mouth curve: higher mood = bigger smile
    const m = `M 90 130 Q 110 ${lerp(145,120, state.mood/100)} 130 130`;
    mouth.setAttribute('d', m);
    // eye size slightly increases with mood
    const r = lerp(6, 10, state.mood/100);
    eyeL.setAttribute('r', r);
    eyeR.setAttribute('r', r);
    // body hue shifts with glow: cyan -> teal -> violet
    const glowT = state.glow/100;
    const c1 = `hsl(${lerp(190, 260, glowT)} 90% 85%)`;
    const c2 = `hsl(${lerp(185, 250, glowT)} 90% 45%)`;
    const grad = document.getElementById('bodyGrad');
    grad.children[0].setAttribute('stop-color', c1);
    grad.children[1].setAttribute('stop-color', c2);
}

```

```

function render(){
    $day.textContent = state.day;
    $time.textContent = ['Morning','Day','Evening','Night'][state.time];
    $glowBar.style.width = `${state.glow}%`;
    $glowTier.textContent = glowTier(state.glow);
    $xpBar.style.width = `${Math.min(100, state.xp % 120 / 1.2)}%`;
    $xpLabel.textContent = state.xp;
    $stageLabel.textContent = state.stage;
    $traitLabel.textContent = state.trait;
    $petName.textContent = state.name;
    $moodBadge.textContent = moodDescriptor();

    for(const k of ['hunger','energy','hygiene','mood']){
        const v = clamp(state[k]);
        statEls[k].bar.style.width = `${v}%`;
        statEls[k].label.textContent = Math.round(v);
    }
}

```

```

// Achievements UI
achEls.firstFeed.style.opacity = state.achievements.firstFeed ? 1 : 0.6;
achEls.firstQuest.style.opacity = state.achievements.firstQuest ? 1 : 0.6;
achEls.glow100.style.opacity = state.achievements.glow100 ? 1 : 0.6;

updateFace();
}

// ---- Game Loop ----
function tick(){
  // time progresses; every 15 seconds change time segment
  if(!document.hidden){ state.time = (state.time + 1) % 4; if(state.time===0) state.day++; }
  // natural decay & recovery
  state.hunger = clamp(state.hunger - 3);
  state.energy = clamp(state.energy - (state.time===3 ? -4 : 2)); // rest more at night
  state.hygiene = clamp(state.hygiene - 2);
  // mood responds to average needs
  const needAvg = (state.hunger + state.energy + state.hygiene)/3;
  state.mood = clamp(lerp(state.mood, needAvg, 0.25));
  // glow is guided by mood & cleanliness
  const targetGlow = clamp((state.mood*0.6) + (state.hygiene*0.4));
  state.glow = clamp(lerp(state.glow, targetGlow, 0.2));
  if(state.glow>=100 && !state.achievements.glow100){ state.achievements.glow100=true;
toast('Achievement: Blinding Glow!'); }

  maybeStageUp();
  if(state.cooldown>0) state.cooldown--;
  $('#questCooldown').textContent = state.cooldown>0 ? `Cooldown: ${state.cooldown}s` :
'Ready';
  render(); save();
}

let loop = null;
function startLoop(){ if(loop) clearInterval(loop); loop = setInterval(tick, 5000); } // 5s per
segment

// ---- Actions ----
function toast(msg){
  const t = document.createElement('div');
  t.className = 'fixed bottom-4 left-1/2 -translate-x-1/2 bg-slate-900/90 border
border-slate-700 rounded-2xl px-4 py-2 text-sm shadow-xl';
  t.textContent = msg; document.body.appendChild(t);
  setTimeout(()=>{ t.remove(); }, 1600);
}

function grantXP(n){ state.xp += n; maybeStageUp(); }

```



```

$('#feedBtn').addEventListener('click', ()=>{
  state.hunger = clamp(state.hunger + 20);
  state.hygiene = clamp(state.hygiene - 5);
  grantXP(5);
  if(!state.achievements.firstFeed){ state.achievements.firstFeed=true; toast('Achievement:
First Meal'); }
  toast(`${state.name} enjoyed a luminous nibble!`);
  render(); save();
});

```

```

$('#playBtn').addEventListener('click', ()=>{
  state.mood = clamp(state.mood + 18);
  state.energy = clamp(state.energy - 10);
  grantXP(6);
  toast(`${state.name} played with stardust!`);
  render(); save();
});

```

```

$('#washBtn').addEventListener('click', ()=>{
  state.hygiene = clamp(state.hygiene + 22);
  state.mood = clamp(state.mood + 6);
  grantXP(4);
  toast(`${state.name} sparkles after a wash!`);
  render(); save();
});

```

```

$('#napBtn').addEventListener('click', ()=>{
  state.energy = clamp(state.energy + 24);
  state.mood = clamp(state.mood + 4);
  state.hunger = clamp(state.hunger - 6);
  grantXP(4);
  toast(`${state.name} had a cozy nap.`);
  render(); save();
});

```

// Quests

```

document.querySelectorAll('.quest').forEach(btn=>{
  btn.addEventListener('click', ()=>{
    if(state.cooldown>0){ toast('Exploration on cooldown.');
```

}

```

    const q = btn.dataset.quest;
    state.cooldown = 20; // seconds
    const outcomes = {
      cave: [
        ['Found a Prism Shard! Glow++', ()=>{ state.glow=clamp(state.glow+15);
grantXP(12);} ],
        ['Got spooked by echoes... Mood--', ()=>{ state.mood=clamp(state.mood-10);} ],
        ['Met a miner moth. Hygiene--', ()=>{ state.hygiene=clamp(state.hygiene-12);} ],
      ],

```

```

    reef: [
      ['Danced with songfish! Mood++', ()=>{ state.mood=clamp(state.mood+16);
grantXP(10);} ],
      ['Salty splash. Hygiene--', ()=>{ state.hygiene=clamp(state.hygiene-8);} ],
      ['Sunbeams charged you. Energy++', ()=>{ state.energy=clamp(state.energy+14);} ],
    ],
    grove: [
      ['Moon nectar feast! Hunger++', ()=>{ state.hunger=clamp(state.hunger+18);
grantXP(8);} ],
      ['Fireflies taught tricks. XP++', ()=>{ grantXP(18);} ],
      ['Thorny mishap... Mood--', ()=>{ state.mood=clamp(state.mood-8);} ],
    ]
  };
  const roll = outcomes[q][Math.floor(Math.random()*outcomes[q].length)];
  roll[1]();
  $('#questLog').textContent = roll[0];
  if(!state.achievements.firstQuest){ state.achievements.firstQuest=true;
toast('Achievement: First Exploration'); }
  render(); save();
});
});

// Rename
$('#renameBtn').addEventListener('click', ()=>{
  const val = $('#nameInput').value.trim();
  if(val){ state.name = val; toast(`Hello, ${state.name}!`); render(); save(); }
});

// Save / Reset
$('#saveBtn').addEventListener('click', ()=>{ save(); toast('Game saved.'); });
$('#resetBtn').addEventListener('click', ()=>{
  if(confirm('Reset your Lumipod? This clears local progress.')){
    localStorage.removeItem(STORAGE_KEY); window.location.reload();
  }
});

// Visibility pause/resume loop for battery friendliness
document.addEventListener('visibilitychange', ()=>{
  if(document.hidden){ clearInterval(loop); } else { startLoop(); }
});

// Init
load();
document.getElementById('year').textContent = new Date().getFullYear();
render(); startLoop();
</script></body>
</html>

```