

# Process & Decision Documentation

## Project/Assignment Decisions

I ended up making my blob angry. To showcase this I made it red, because anger is generally associated with red. I also wanted to make it seem erratic, and the use of noise helped this. The biggest part of how I made it seem angry was its pulsing, the blob occasionally pulses, which is meant to represent an angry outburst. This is represented with sudden jagged spikes as well as a change in colour and size.

## GenAI Documentation

Gen AI was used to generate the base code for the pulsing behaviour of the blob. The prompts were as follows:

*(pasted original code) change this simple blob code to make it red. Make it pulse very few seconds, the pulse should drastically reduce its vertices briefly as well as make the vertices jut out more, the pulse should also change it to a brighter red. Optionally add a button that controls when the pulse happens*

*make the blob change slowly from red to a slightly darker red and back and forth while not pulsing, also make the vertices more spread out while pulsing, kind of like a jagged star*

*make the blobs pulse bigger/more spread out*

The gen AI outputs provided the foundation for the blobs behaviours, but the AI's code did not demonstrate the pulsing behaviour I was looking for.

## My changes to the AI code:

- Adjusted the colours
- Adjusted the on screen text
- Adjusted loop timing
- Adjusted pulse frequency
- 
- Adjusted blob behaviour based on the AI's variables
  - The number of points
  - Jaggedness of points during pulsing
  - Count of points
  - Duration of pulsing

## The Unchanged AI Code From my Last Prompt for Reference:

```
let blob = {
  x: 240,
  y: 160,

  r: 28,      // Base radius
  points: 48, // Smooth circle resolution

  wobble: 8,
  wobbleFreq: 0.8,

  t: 0,
  tSpeed: 0.01,

  // Pulse state
  pulsing: false,
  pulseT: 0,
  pulseDur: 0.45,
  pulseGap: 180,
  frameLast: 0,

  // Idle color breathing
  colorT: 0,
  colorSpeed: 0.005,
};

let pulseBtn;

function setup() {
  createCanvas(480, 320);
  noStroke();
  textFont("sans-serif");
  textSize(14);

  pulseBtn = createButton("Pulse");
  pulseBtn.position(10, height + 10);
  pulseBtn.mousePressed(triggerPulse);
}

// Manually start a pulse
function triggerPulse() {
  blob.pulsing = true;
  blob.pulseT = 0;
}

function draw() {
  background(240);
```

```

// Advance noise time
blob.t += blob.tSpeed;

// Auto pulse every few seconds
if (!blob.pulsing && frameCount - blob.frameLast > blob.pulseGap) {
  triggerPulse();
}

// --- Pulse timing ---
let p = 0;
if (blob.pulsing) {
  blob.pulseT += deltaTime / 1000;
  p = constrain(blob.pulseT / blob.pulseDur, 0, 1);

  if (p >= 1) {
    blob.pulsing = false;
    blob.frameLast = frameCount;
  }
}

// Smooth pulse curve (0 → 1 → 0)
let pulseAmt = sin(p * PI);

// --- Shape exaggeration during pulse ---
// Much fewer points = star-like silhouette
let pts = floor(lerp(blob.points, 8, pulseAmt));

// Strong outward spikes
let wob = lerp(blob.wobble, 46, pulseAmt);

// Higher frequency sharpens spikes
let freq = lerp(blob.wobbleFreq, 3.0, pulseAmt);

// Overall size expansion during pulse
let baseR = lerp(blob.r, 52, pulseAmt);

// --- Idle color breathing ---
blob.colorT += blob.colorSpeed;
let idleAmt = (sin(blob.colorT) + 1) * 0.5;

let idleCol = lerpColor(
  color(140, 0, 0),
  color(190, 0, 0),
  idleAmt
);

// Bright flash color for pulse

```

```

let pulseCol = color(255, 70, 70);

// Blend idle → pulse color
let col = lerpColor(idleCol, pulseCol, pulseAmt);

// --- Draw blob ---
fill(col);
beginShape();

for (let i = 0; i < pts; i++) {
  const a = (i / pts) * TAU;

  // Angle-based Perlin noise sample
  const n = noise(
    cos(a) * freq + 100,
    sin(a) * freq + 100,
    blob.t
  );

  // Large radial displacement for explosive pulse
  const r = baseR + map(n, 0, 1, -wob, wob);

  vertex(
    blob.x + cos(a) * r,
    blob.y + sin(a) * r
  );
}

endShape(CLOSE);

fill(0);
text(
  "Idle: slow red breathing · Pulse: larger jagged star burst",
  10,
  18
);
}

```