

AI used: Chatgpt_5.2

Prompt

I have a basic p5.js blob that “breathes” using Perlin noise. The Week 2 side quest wants us to redesign the blob’s movement and environment to express a specific emotion, and optionally add a mischief mechanic. I’m not sure what emotion to choose that’s feasible without a lot of assets.

Response

First, choose an emotion (panic is recommended).

panic: fast, shaking, sudden turn, escaping the mouse, bouncing off a wall; background flashing, warning particles

frustration: stopping and starting, getting stuck, hitting a wall, heavy rebound; oppressive background, noise

joy: rhythmic bouncing, spinning, chasing a target; colorful particles, soft background

(2) Create a "mini-map/small object"

For example, put several squares/circles in an array as "collision objects".

(3) Mischievous mechanism

The simplest version:

blob hits an object → the object is pushed away (speed = direction of impact * force)

or

blob hits an object → the object disappears and scores ("steal")

Prompt

How would I actually translate "panic" into variables? I don't want it to be just on or off.

Response

You can compute the distance between the blob and the mouse and map that distance to a panic intensity between 0 and 1. That intensity can then scale speed, wobble, jitter, colour, and screen shake. like `panic = map(distance, far, near, 0, 1)`

Prompt

How would I actually translate "panic" into variables? I don't want it to be just on or off.

response

You can compute the distance between the blob and the mouse and map that distance to a panic intensity between 0 and 1. That intensity can then scale speed, wobble, jitter, colour, and screen shake.

Prompt

So something like `panic = map(distance, far, near, 0, 1)`?

response

Yes, and you should clamp it. A small helper function like `clamp(value, min, max)` will help prevent negative or over-1 values.

prompt

For movement, I want the blob to run away from the mouse. I'm not great at vector math. What's the simplest way?

response

Compute a vector from the mouse to the blob (`blob.x - mouseX, blob.y - mouseY`), normalize it, and apply acceleration in that direction. Use velocities (`vx, vy`) so movement feels physical rather than teleporting.

prompt

When the mouse is really close, I sometimes get weird snapping or NaN values. I think it's dividing by zero?

response

That's likely from normalizing a zero-length vector. When computing the magnitude, wrap it in `max(epsilon, magnitude)` so you never divide by zero.

