§

implementing the chart with

bit.ly/canvas_spring

100.02

```
function Canvas({ x }) {
  const canvasRef = useRef(null);
  const ctx = canvasRef.current?.getContext("2d");

  useEffect(() => {
    const draw = (x) => {
      if (!ctx) return;
      ctx.clearRect(0, 0, WIDTH, HEIGHT);
      ctx.beginPath();
      ctx.arc(x, 20, radius, 0, Math.PI * 2, false);
      ctx.fillStyle = "red";
      ctx.fill();
    };
    draw(x);
  }, [x, ctx]);

  return <canvas ref={canvasRef} style={{ width: "100%" }} />;
}

export const AnimatedCanvas = animated(Canvas);
```

100.02

```
function Canvas({ x }) {
  const canvasRef = useRef(null);
  const ctx = canvasRef.current?.getContext("2d");

  useEffect(() => {
    const draw = (x) => {
      if (!ctx) return;
      ctx.clearRect(0, 0, WIDTH, HEIGHT);
      ctx.beginPath();
      ctx.arc(x, 20, radius, 0, Math.PI * 2, false);
      ctx.fillStyle = "red";
      ctx.fill();
    };
    draw(x);
  }, [x, ctx]);

  return <canvas ref={canvasRef} style={{ width: "100%" }} />;
}

export const AnimatedCanvas = animated(Canvas);
```