Got it ✅ — you want an **Angular + GSAP** example where:

- A circle follows the mouse smoothly.
- When you click, a new circle is dropped at the click position with an animation.

Here's a **complete runnable Angular example** (Angular 15+ syntax, works in Angular 19 too) with **GSAP**.

1️⃣ Install GSAP

```
npm install gsap
```

2️⃣ Angular Component Code

`mouse-follow.component.ts`

```
import { Component, HostListener, OnInit, Renderer2, ElementRef }
from '@angular/core';
import { gsap } from 'gsap';

@Component({
  selector: 'app-mouse-follow',
  templateUrl: './mouse-follow.component.html',
  styleUrls: ['./mouse-follow.component.css']
})
export class MouseFollowComponent implements OnInit {
  private follower!: HTMLElement;

  constructor(private renderer: Renderer2, private el: ElementRef) {}

  ngOnInit(): void {
    // Create the follower circle dynamically
    this.follower = this.renderer.createElement('div');
    this.renderer.addClass(this.follower, 'follower');
    this.renderer.appendChild(this.el.nativeElement, this.follower);
  }
```

```
// Mouse move listener
@HostListener('document:mousemove', ['$event'])
onMouseMove(event: MouseEvent) {
  gsap.to(this.follower, {
    x: event.clientX - 15, // center offset
    y: event.clientY - 15,
    duration: 0.2,
    ease: 'power2.out'
  });
}

// Mouse click listener
@HostListener('document:click', ['$event'])
onClick(event: MouseEvent) {
  const drop = this.renderer.createElement('div');
  this.renderer.addClass(drop, 'drop-circle');
  this.renderer.appendChild(this.el.nativeElement, drop);

  // Position the drop circle
  gsap.set(drop, {
    x: event.clientX - 10,
    y: event.clientY - 10,
    scale: 0
  });

  // Animate the drop circle
  gsap.to(drop, {
    scale: 1,
    opacity: 1,
    duration: 0.3,
    ease: 'back.out(2)'
  });

  // Fade out and remove after animation
  gsap.to(drop, {
    opacity: 0,
```

```
      duration: 0.5,
      delay: 0.8,
      onComplete: () =>
this.renderer.removeChild(this.el.nativeElement, drop)
    });
  }
}
```

mouse-follow.component.html

```
<!-- Empty, circles are created dynamically -->
```

mouse-follow.component.css

```
.follower {
  position: fixed;
  width: 30px;
  height: 30px;
  background: rgba(0, 150, 255, 0.8);
  border-radius: 50%;
  pointer-events: none;
  z-index: 9999;
}

.drop-circle {
  position: fixed;
  width: 20px;
  height: 20px;
  background: rgba(255, 100, 100, 0.9);
  border-radius: 50%;
  pointer-events: none;
  z-index: 9998;
}
```

3️⃣ How It Works

- **Follower Circle**: Created once in `ngOnInit`, moves smoothly to mouse position using `gsap.to`.
- **Drop Circles**: Created on each click, animated in with a scale effect, then faded out and removed.
- **Renderer2**: Used for safe DOM manipulation in Angular.
- **GSAP**: Handles smooth animations with easing.

4️⃣ Demo Behavior

- Move your mouse → blue circle follows smoothly.
- Click anywhere → red circle appears at click point, pops in, fades out.

If you want, I can extend this so the **drop circles fall with gravity** using **GSAP physics** for a more realistic effect.
Do you want me to add that?