

```

1  import { Component, OnInit } from '@angular/core';
2  import { animate, state, style, transition, trigger } from '@angular/animations';
3  import { Observable } from 'rxjs/Observable';
4  import { Subscription } from 'rxjs/Subscription';
5  import { Router } from '@angular/router';
6
7  @Component({
8    selector: 'app-intro',
9    templateUrl: './intro.component.html',
10   styleUrls: ['./intro.component.scss'],
11   animations: [
12     trigger('frameOpacity', [
13       state('frameOpacityStart', style({ transform: 'translate(0, -165%) rotate3d(0,
14         1, 0, -90deg)', opacity: 1 })),
15       state('frameOpacityEnd', style({ transform: 'translate(0, -165%) rotate3d(0,
16         1, 0, 0deg)', opacity: 1 })),
17       transition('* => frameOpacityEnd', animate('150ms ease-out')),
18     ]),
19     trigger('picOpacity', [
20       state('picOpacityStart', style({ opacity: 0 })),
21       state('picOpacityEnd', style({ opacity: 1 })),
22       transition('* => picOpacityEnd', animate('1000ms 1000ms ease-out')),
23     ]),
24     trigger('brandOpacity', [
25       state('brandOpacityStart', style({ opacity: 0 })),
26       state('brandOpacityEnd', style({ opacity: 1 })),
27       transition('brandOpacityStart => brandOpacityEnd', animate('800ms 800ms')),
28     ]),
29     trigger('visibilityChanged', [
30       state('shown', style({ opacity: 1 })),
31       state('hidden', style({ opacity: 0, display: 'none' })),
32       transition('shown => hidden', animate('500ms 300ms')),
33     ])
34   ]
35 })
36
37 export class IntroComponent implements OnInit {
38   frameOpacityState: string[];
39   picOpacityState: string[];
40   brandOpacityState: string;
41   visibilityState: string;
42   polaroidCount: number[];
43   polaroidNumber = 20;
44   private timerSubscription: Subscription;
45
46   constructor(private router: Router) { }
47
48   // set polaroid frame opacity timer and initialize arrays and animations
49   ngOnInit() {
50     this.polaroidCount = Array.from(Array(this.polaroidNumber).keys());
51     this.frameOpacityState =
52       Array.from(Array(this.polaroidNumber).fill('frameOpacityStart'));
53     this.picOpacityState =
54       Array.from(Array(this.polaroidNumber).fill('picOpacityStart'));
55     this.brandOpacityState = 'brandOpacityStart';
56     this.visibilityState = 'shown';
57     const timer = Observable.timer(1000, 100);
58     this.timerSubscription = timer.subscribe((t) => {
59       this.frameOpacityState[t] = 'frameOpacityEnd';
60       if (t === this.polaroidNumber - 1) { this.timerSubscription.unsubscribe(); }
61     });
62   }
63
64   // frame opacity done callback
65   startPicOpacity(e, index: number) {
66     if (e.toState === 'frameOpacityEnd') {
67       this.picOpacityState[index] = 'picOpacityEnd';
68       if (index === this.polaroidNumber - 1) {
69         this.brandOpacityState = 'brandOpacityEnd';
70       }
71     }
72   }
73 }

```

```
70     // animation finished callback
71     changeVisibility(e) {
72         if (e.toState === 'brandOpacityEnd') { this.visibilityState = 'hidden'; }
73     }
74
75     // redirect to a different page after fade out callback
76     redirectHome(e) {
77         if (e.toState === 'hidden') { this.router.navigate(['/cv']); }
78     }
79 }
80
```