









First snapshot of the el

Last snapshot of the el

Inverted values (from/fromTo)

Play the computed animation





Last snapshot of the el

Inverted values (from/to)

First snapshot of the el

Play the computer animated

```

1  const playerQueue = targetEl.querySelector('.player-queue-landscape');
2  const trackPlayer = targetEl.querySelector('.track-player');
3
4  const playerRecPre = getRec(trackPlayer);
5  const playerQueueRecPre = getRec(playerQueue);
6
7  targetEl.classList.toggle('queue-active');
8
9  const playerRecPost = getRec(trackPlayer);
10 const playerQueueRecPost = getRec(playerQueue);
11
12 const playerDeltaX = playerRecPre.left - playerRecPost.left;
13 const playerDeltaY = playerRecPre.top - playerRecPost.top;
14
15 const playerQueueDeltaW = getQueueVectors(
16     playerQueueRecPre.width,
17     playerQueueRecPost.width
18 );
19 const playerQueueDeltaH = getQueueVectors(
20     playerQueueRecPre.height,
21     playerQueueRecPost.height
22 );
23
24 const playerQueueAnimation = createAnimation()
25     .addElement(playerQueue)
26     .fromTo(
27         'transform',
28         `scale3d(${playerQueueDeltaW.scaleFrom}, ${playerQueueDeltaH.scaleFrom}, ${playerQueueDeltaH.scaleFrom})`,
29         `scale3d(${playerQueueDeltaW.scaleTo}, ${playerQueueDeltaH.scaleTo}, ${playerQueueDeltaH.scaleTo})`
30     )
31     .fromTo(
32         'opacity',
33         playerQueueDeltaH.opacityFrom,
34         playerQueueDeltaH.opacityTo
35     );
36 animationChain.push(playerQueueAnimation);
37
38 const trackPlayerAnimation = createAnimation()
39     .addElement(trackPlayer)
40     .beforeStyles({
41         'transform-origin': 'top left',
42     })
43     .from('transform', `translate3d(${playerDeltaX}px, ${playerDeltaY}px, 0)`)
44     .afterClearStyles(['transform-origin']);
45
46 animationChain.push(trackPlayerAnimation);
47

```


