Framer Motion Bouncing Ball Animation

Written by Seth Corker on Benevolent Bytes

What we're making

A tweet showing the bouncing animation we created

We're creating this bouncing animation using easeOut easing and yoyo. This creates a simple bounce which continues to loop. We also make use of keyframes instead of variants to specify the exact changes we wish to animate between. This is useful when changing the colour, we achieve this with some strategic transition properties.

How to achieve the bounce and colour change

If you'd like to see a video tutorial, here's one I prepared - it's about 4 minutes long and explains the process too.

A 4 minute video tutorial showing how we setup the animation from start to finish

There are a couple of things to note in order to create this animation. Here is our BouncingBall component, first you'll notice the animate props use arrays instead of a single value or variant and second - we the magic happens in the transition prop.

```
export default function BouncingBall() {
  return (
    <div
      style={{
        width: "2rem",
        height: "2rem",
        display: "flex",
        justifyContent: "space-around",
      }}
    >
      <motion.span
        style={ballStyle}
        transition={bounceTransition}
        animate={{
          y: ["100%", "-100%"],
          backgroundColor: ["#ff6699", "#6666ff"],
        }}
      />
    </div>
  )
}
```

Using Keyframes in Framer Motion

In previous Framer Motion animation tutorials I used variants or animated the properties directly. This time easy property within our animate object is a assigned an array value. This tells the motion component to treat the value changes as keyframes and sequentially set them. So, the `y` position will start at `100%` and at the next frame it will become `-100%`. We do the same thing with the `backgroundColor`.

Making the animation loop

The transition property is the most important part of this animation. We define an object called `bounceTransition`, here we define how each property we are animating, actually performs the animation. The bounce is easy, we set `yoyo` to `Infinity` which means the animation will loop, when it reaches the end it will reverse the animation and continue playing. We set `ease` to `easeOut` to create the 'bounce'. This works well because it's smooth in the one part but has a sudden stop which produces the 'bounce' rather than a smooth movement which `linear` or `easeInOut` easing would give us.

The colour change works by setting the same props as the `y` position animation and changing the duration to `0` so it's instantaneous and setting `repeatDelay` to twice the duration of our bounce animation (our bounce

is 400ms so our delay is 800ms). We have two `backgroundColor` keyframes which will last 400ms each and continue to repeat. This creates the colour swap when the ball bounces.

```
const bounceTransition = {
  y: {
    duration: 0.4,
    yoyo: Infinity,
    ease: "easeOut",
  },
  backgroundColor: {
    duration: 0,
    yoyo: Infinity,
    ease: "easeOut",
    repeatDelay: 0.8,
  },
}
```

Where to go from here?

The animation achieves the effect but a good next step would be to apply some traditional animation techniques like squish and stretch to give it a less mechanical feel. This emphasises the motion by squishing the ball on impact and stretching it when it's in the air.

Resources

- To see the full source code, checkout the repo on GitHub (this also contains the other loading animation code from previous tutorials)
- Check out my playlist of video tutorials covering animation in Framer Motion
- Take a look at the official Framer Motion documentation