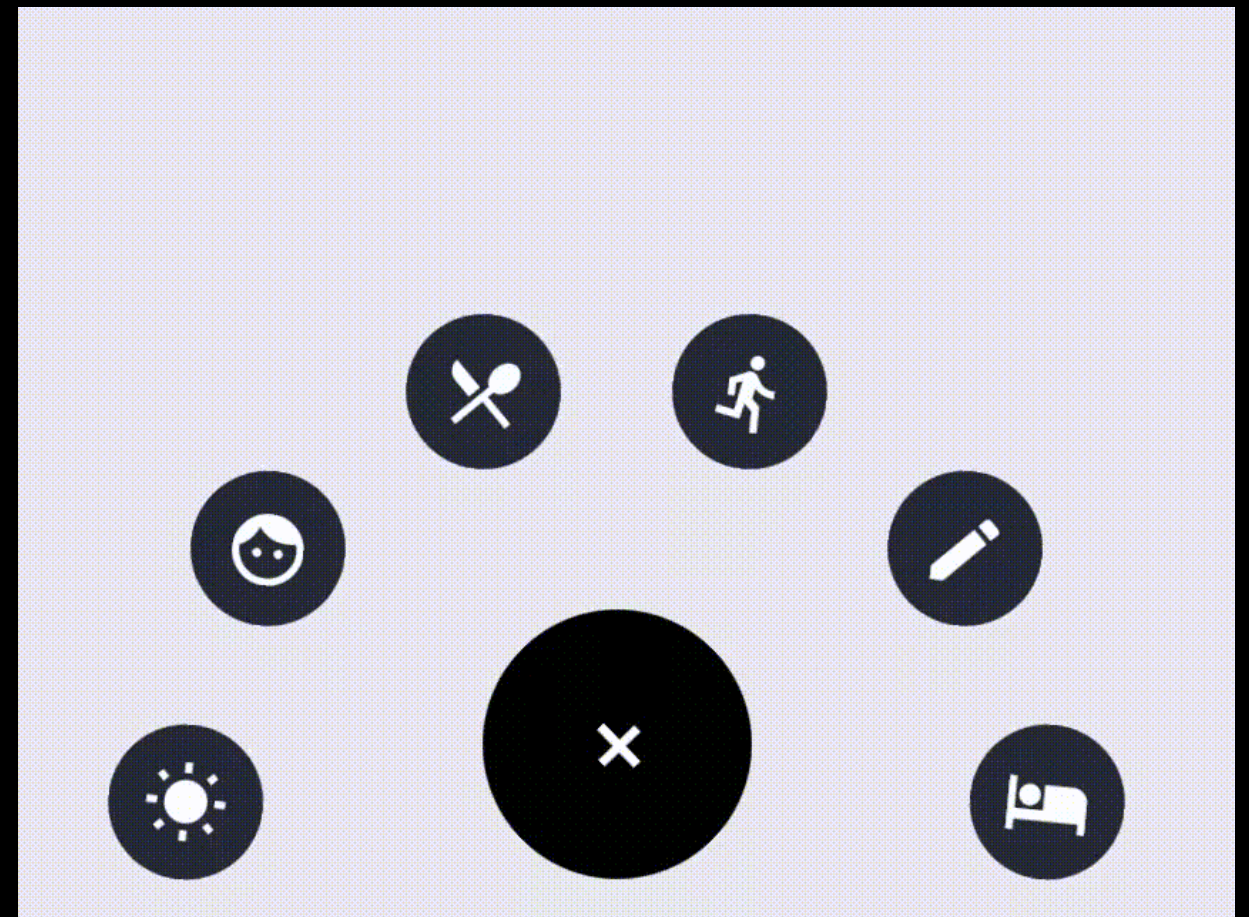
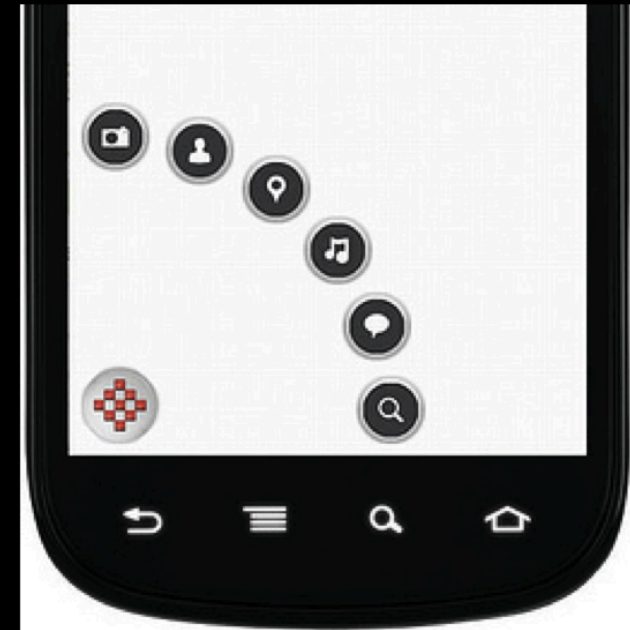
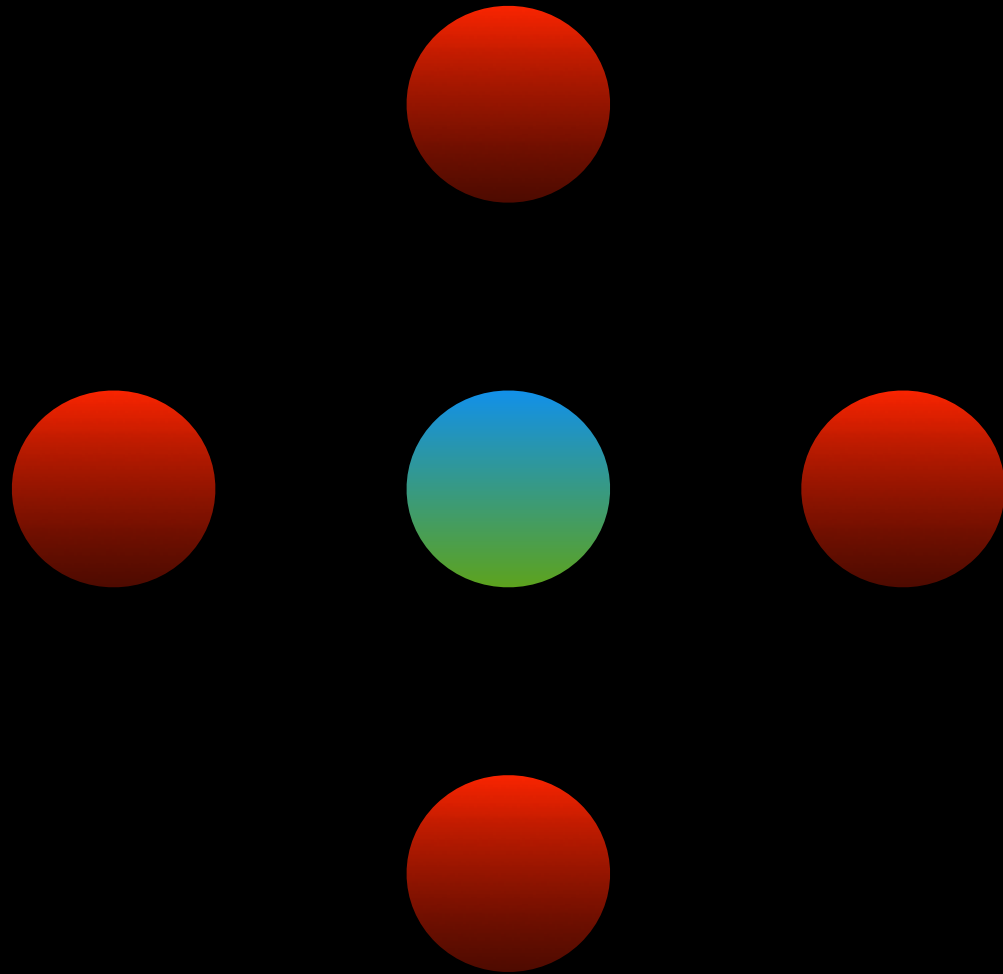


DANIEL GRAHAM

REACT NATIVE ANIMATIONS

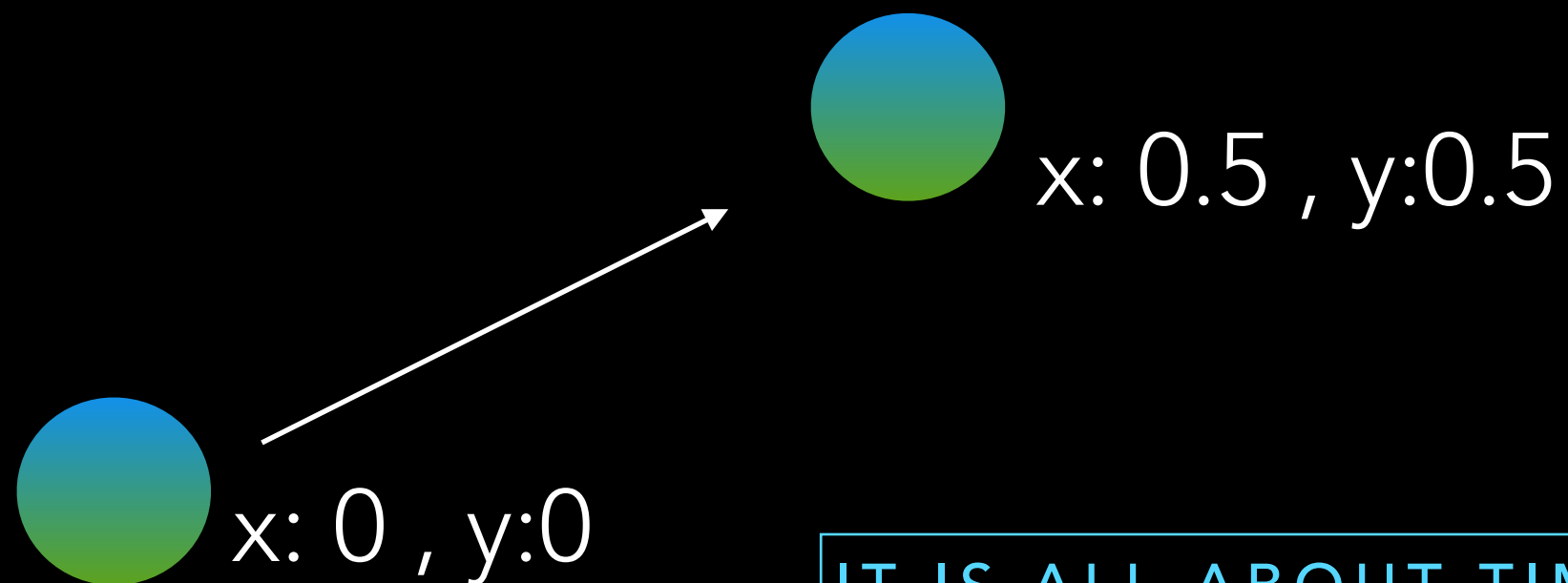
LET'S BUILD A FIT SPINNER



ANIMATION FRAME OF REFERENCE

The style sheet determine the layout of the screen

Animation is just modifying the these style sheet properties over time.

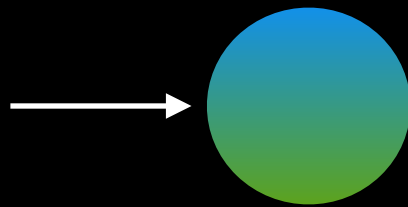


IT IS ALL ABOUT TIMING

LET'S START BY THE CENTER OF OUR SPINNER

Opacity:0

Opacity:1



```
constructor(){  
  super()  
  this.state = {  
    // Initial value for opacity: 0  
    opacityAnimatedValue: new Animated.Value(0),  
  }  
}
```


A VARIABLE THAT
THE ANIMATED LIBRARY
CAN ADJUST

<https://snack.expo.io/@professorxii/fadeparti>

LET'S START BY THE CENTER OF OUR SPINNER

THE FUNCTION THAT IS RESPONSIBLE ADJUSTING THE ANIMATION VALUE

AN OBJECT ON HOW JUST ADJUST THE VARIABLE



```
componentDidMount() {  
  Animated.timing( // Animate over time  
    this.state.opacityAnimatedValue, // The animated value to drive  
    {  
      toValue: 1, // Animate to opacity: 1 (opaque)  
      duration: 10000, // Make it take a while  
    },  
  ).start(); // Starts the animation  
}
```

<https://snack.expo.io/@professorxii/fadeparti>

LET'S START BY THE CENTER OF OUR SPINNER

EXTRACT THE ANIMATED VALUE PROPERTY

```
render() {  
  let { opacityAnimatedValue } = this.state;
```

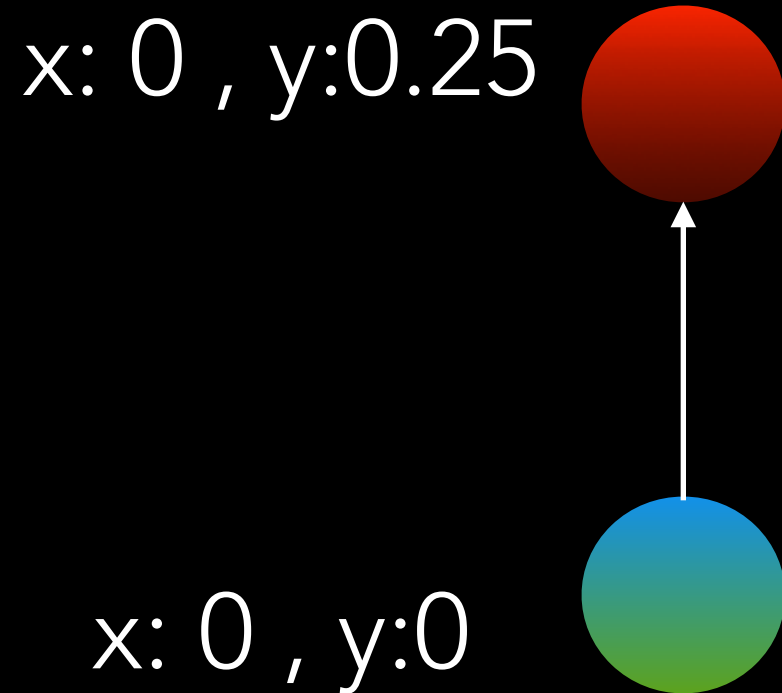
```
  return (  
    <View style={styles.container}>  
      <Animated.View style={ { ...styles.centerCircle , opacity: opacityAnimatedValue } }/>  
    </View>  
  );  
}
```

DECONSTRUCT THE STYLE OBJECT AND CREATE
A NEW OBJECT WITH OPACITY

<https://snack.expo.io/@professorxii/fadeparti>

EXPLODE ON PRESS

Have one of slide up
When the center circle is pressed



Animation Curves

<https://easings.net/en>

<https://snack.expo.io/@professorxii/fadepartii>

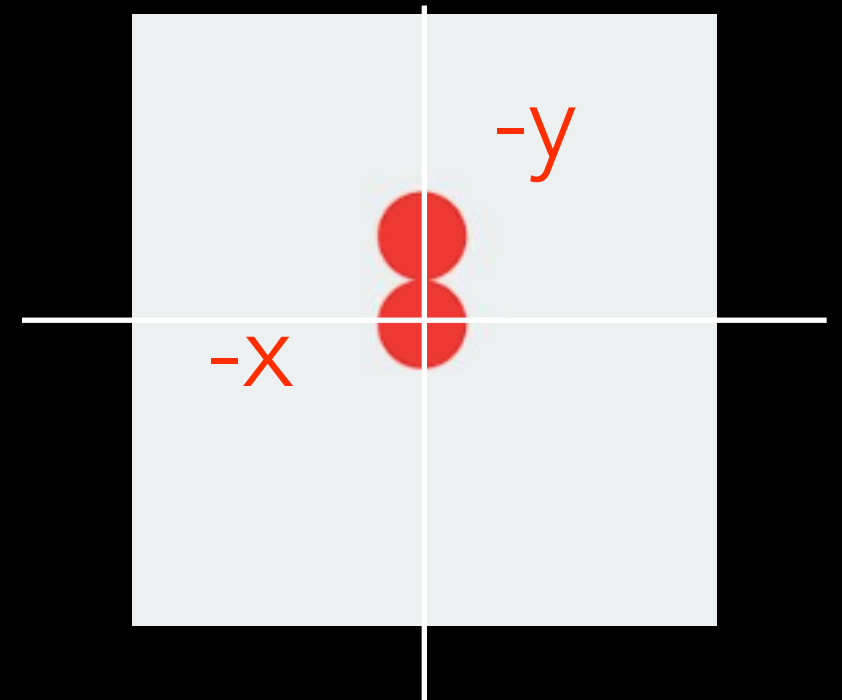
EXPLODE ON PRESS

```
constructor(){  
  super()  
  this.state = {  
    opacityAnimatedValue: new Animated.Value(0), // Initial value for opacity: 0  
    positionAnimatedValue: new Animated.ValueXY({x: 0, y: 0}), //Initial value for position  
    circleVisible: false  
  }  
}
```

<https://snack.expo.io/@professorxii/fadepartii>

EXPLODE ON PRESS

```
moveCirlces(){  
  this.setState({  
    circleVisible: true  
  })  
  console.log("clicked")  
  Animated.timing(this.state.positionAnimatedValue, {  
    toValue: {x:0, y: 200},  
    easing: Easing.bounce,  
    duration: 2000,  
  }).start()  
}
```



Animation Curves

<https://easings.net/en>

```
positionAnimatedValue: new Animated.ValueXY({x: 0, y: -44}), //Initial value for position
```

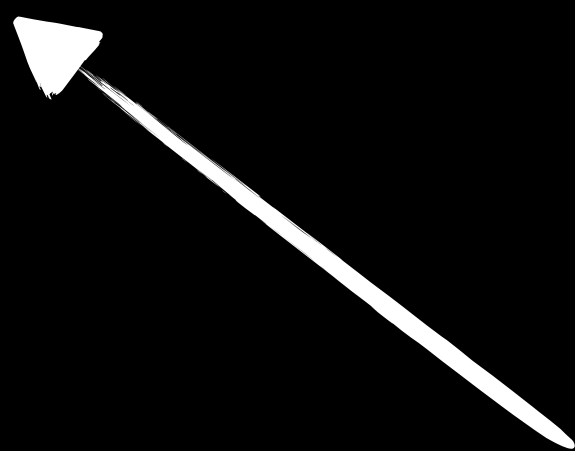
<https://snack.expo.io/@professorxii/fadepartii>

<https://snack.expo.io/@professorxii/fadepartii>

```
render() {  
  let { opaciytAnimatedValue } = this.state  
  let { positionAnimatedValue } = this.state  
  let { circleVisible } = this.state
```

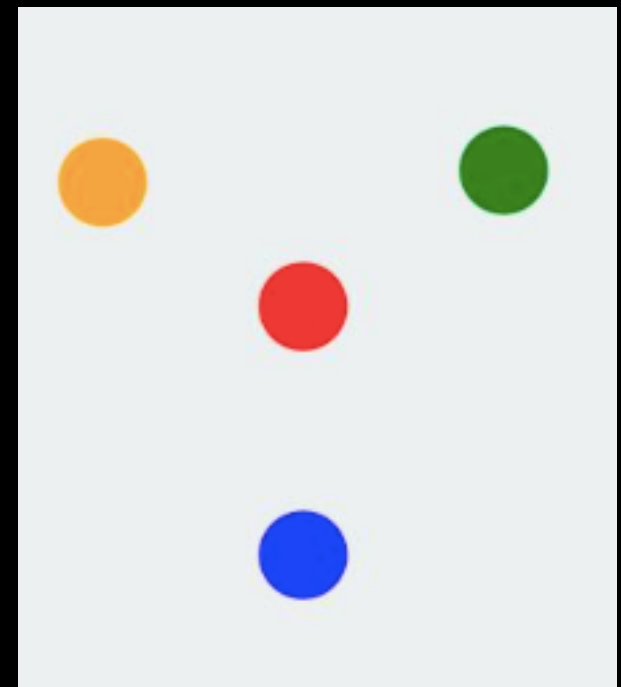
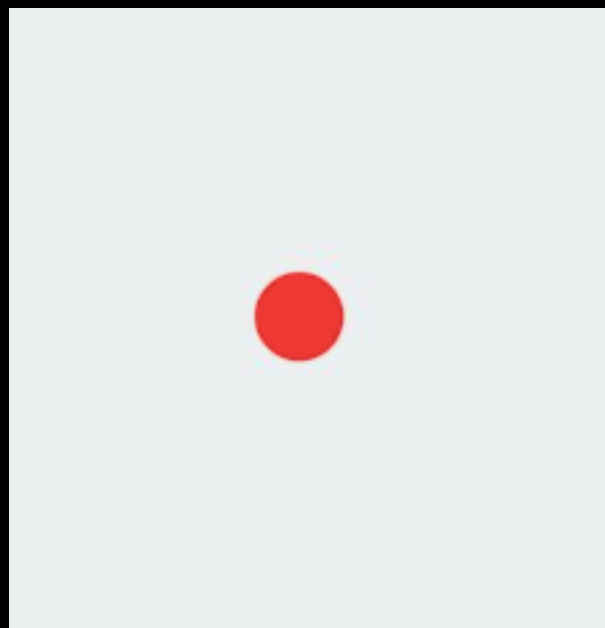
```
  return (  
    <View style={styles.container}>
```

```
      <TouchableOpacity onPress={() => this.moveCirlces()}>  
        <Animated.View style={ { ...styles.centerCirlce , opacity: opaciytAnimatedValue } }/>  
      </TouchableOpacity>  
      {circleVisible &&  
        <Animated.View style={ { ...styles.centerCirlce ,  
          opacity: opaciytAnimatedValue,  
          ...positionAnimatedValue.getLayout() } }/>  
      }  
    </View>  
  );  
}
```



Gets the css for
Positions the view

LETS ADD MORE

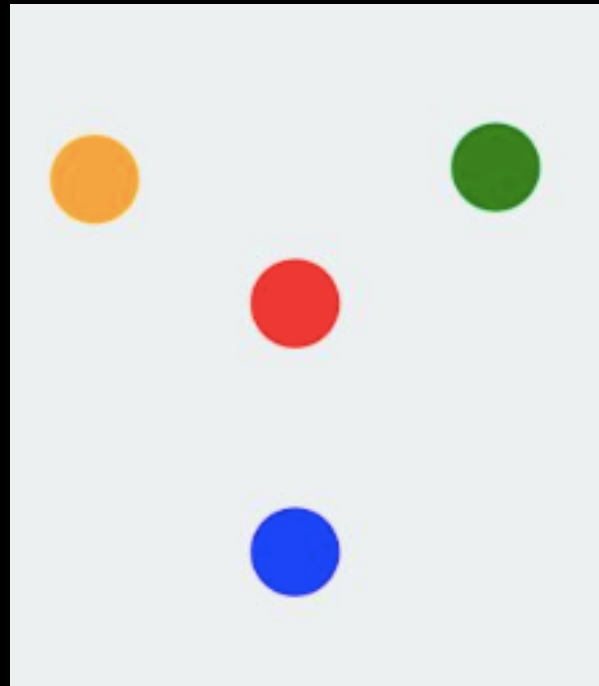


<https://snack.expo.io/@professorxii/fadepartiii>

```

constructor(){
  super()
  this.state = {
    opaciytAnimatedValue: new Animated.Value(0), // Initial value for opacity: 0
    positionAnimatedValue1: new Animated.ValueXY({x: 0, y: -44}), //Inital value for position
    positionAnimatedValue2: new Animated.ValueXY({x: 0, y: -88}), //Inital value for position
    positionAnimatedValue3: new Animated.ValueXY({x: 0, y: -132}), //Inital value for position
    circleVisible: false
  }
}

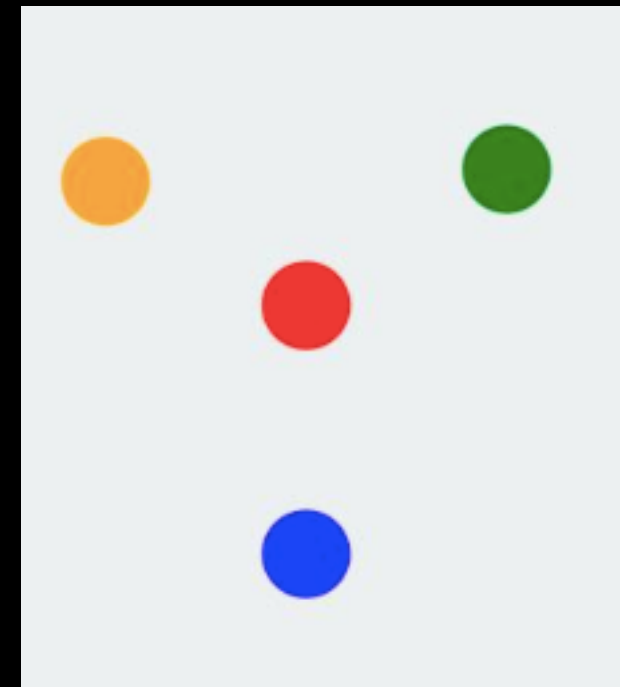
```



```
moveCirlces(){  
  this.setState({  
    circleVisible: true  
  })  
  console.log("clicked")  
  Animated.timing(this.state.positionAnimatedValue1, {  
    toValue: {x: 0, y: 80},  
    easing: Easing.bounce,  
    duration: 2000,  
  }).start()
```

```
  Animated.timing(this.state.positionAnimatedValue2, {  
    toValue: {x: -100, y: -150},  
    easing: Easing.bounce,  
    duration: 2000,  
  }).start()
```

```
  Animated.timing(this.state.positionAnimatedValue3, {  
    toValue: {x: 100, y: -200},  
    easing: Easing.bounce,  
    duration: 2000,  
  }).start()
```



```
}
```

<https://snack.expo.io/@professorxii/fadepartiii>

Top



Left

Right

Bottom

```
x = r*cos(t) + h;  
y = r*sin(t) + k;
```

```
const startX = 150  
const startY = 10  
const radius = 100
```

```
moveCirlces(angles){  
  this.setState({  
    circleVisible: true  
  })  
  Animated.timing(this.state.positionAnimatedValue1, {  
    toValue: {x: radius*Math.cos(angles[0]) + startX , y: radius*Math.sin(angles[0]) + startY},  
    easing: Easing.bounce,  
    duration: 2000,  
  }).start()
```

<https://snack.expo.io/@professorxii/fadepartiv>

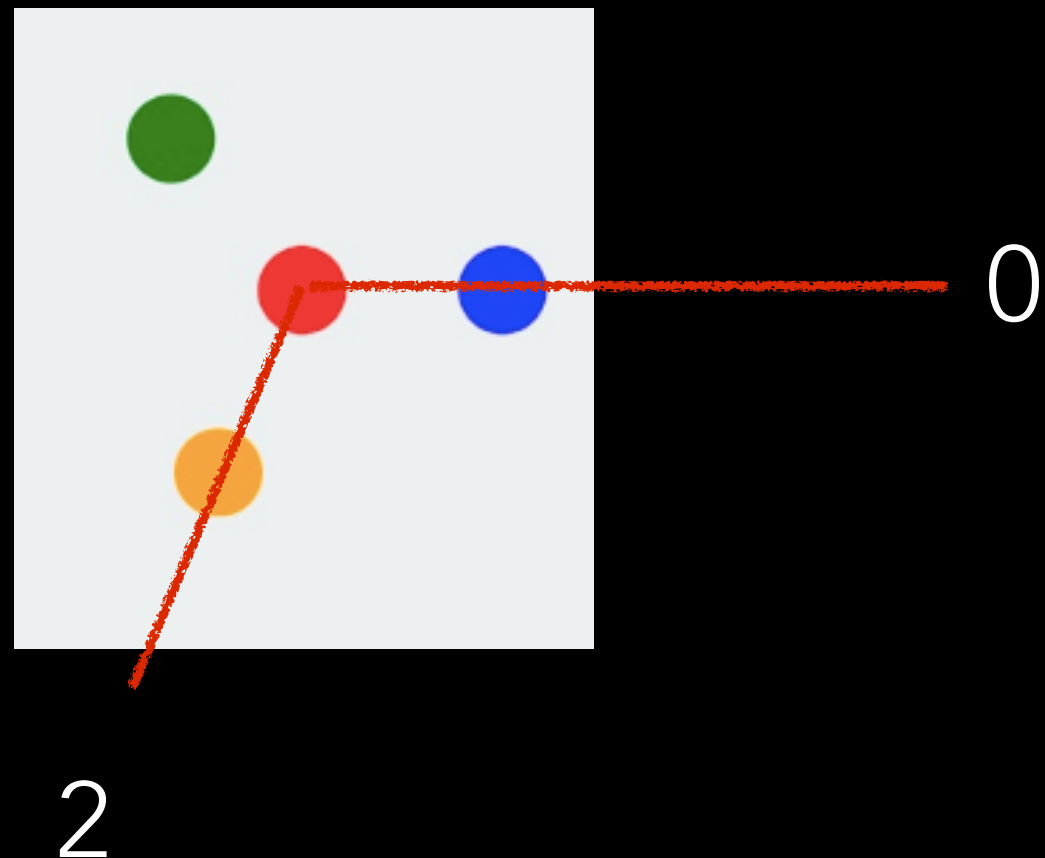
```
return (
```

```
<View style={styles.container}>
```

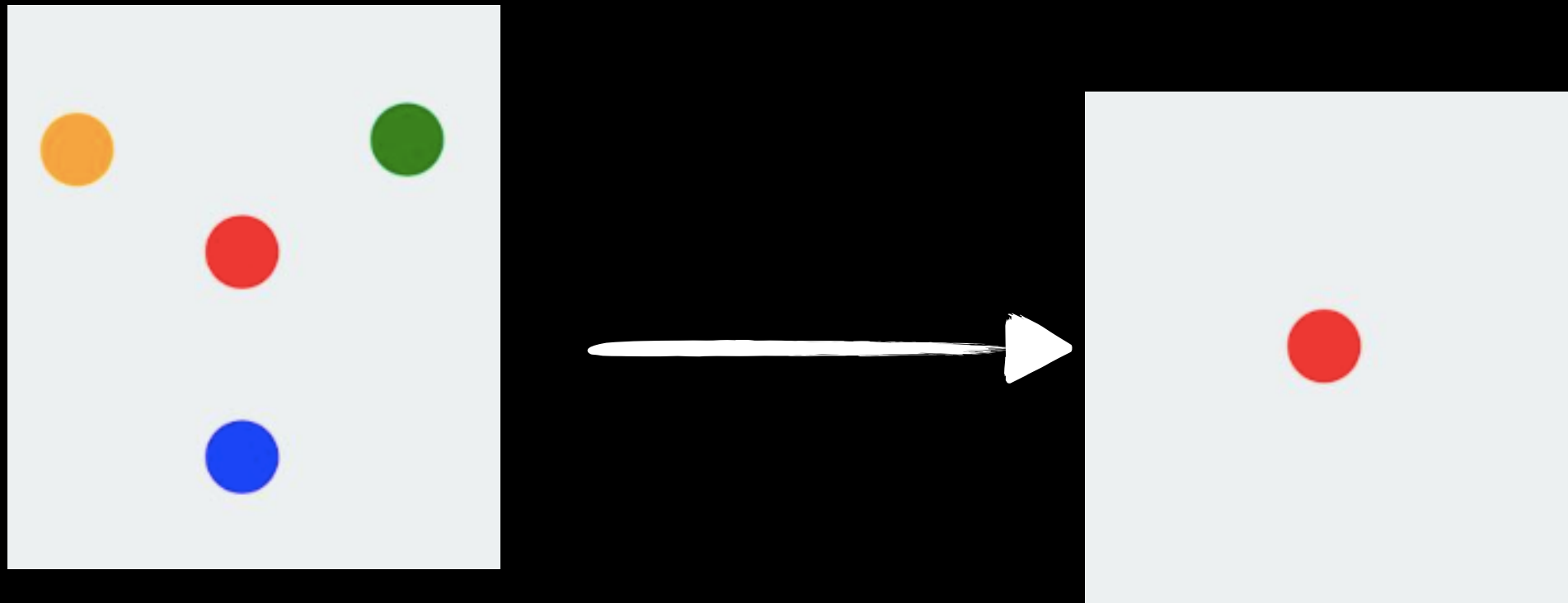
```
<TouchableOpacity style={styles.centerOpacity} onPress={() => this.moveCirlces([0, 2, 4])}>
```

```
<Animated.View style={ { ...styles.centerCirlce , opacity: opaciytAnimatedValue } }/>
```

```
</TouchableOpacity>
```



LETS MODIFY SO THAT IT COLLAPSES WHEN WE PRESS IT AGAIN



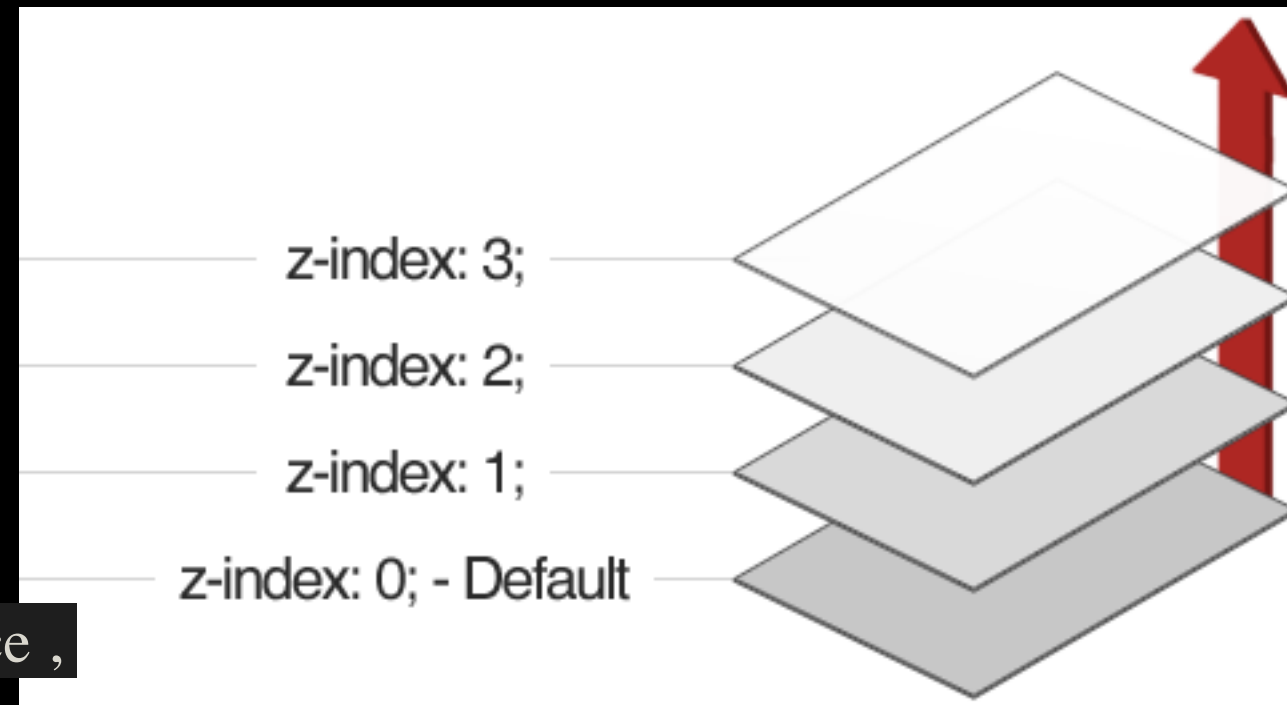
```
moveCirlces(angles, radius){  
  this.setState((state) => {  
    return {circleVisible: true, collapsed: !state.collapsed };  
  })  
}
```

```
<TouchableOpacity style={styles.centerOpacity} onPress={() =>  
  this.moveCirlces([0, 2, 4], this.state.collapsed ? radiusGobal : 0)}>
```

BUT THERE IS PROBLEM

```
{circleVisible &&  
  <Animated.View style={ { ...styles.greenCirlce ,  
    opacity: opaciytAnimatedValue,  
    ...positionAnimatedValue3.getLayout() } }/>  
}
```

```
<TouchableOpacity style={styles.centerOpacity} onPress={() =>  
  this.moveCirlces([0, 2, 4], this.state.collapsed ? radiusGobal : 0)}>  
  <Animated.View style={ { ...styles.centerCirlce , opacity: opaciytAnimatedValue } }/>  
</TouchableOpacity>
```



You could change the Z index
Or just change the rendering order

<https://snack.expo.io/@professorxii/fadeparty>

ANIMATED API

Key Functions:

Animation Timing:

Animation Curves

<https://easings.net/en>

```
Animated.timing(this.state.xPosition, {  
  toValue: 100,  
  easing: Easing.back(),  
  duration: 2000,  
}).start();
```

The simplest workflow for creating an animation is to create an `Animated.Value`, hook it up to one or more style attributes of an animated component, and then drive updates via animations using **`Animated.timing()`**:

LET'S BUILD THIS WEATHER APP





EVENTS



15.04.2016

NIGHT IN DISCO STL

18 LITTLE WEST 12TH ST, NEW
YORK CITY, NY

ATTENDING
16/20

DON'T

I'M IN

