

# Mobile Applications for Sensing and Control

EE P 523 Sp23 M W 4:00—5:50p ECE 045

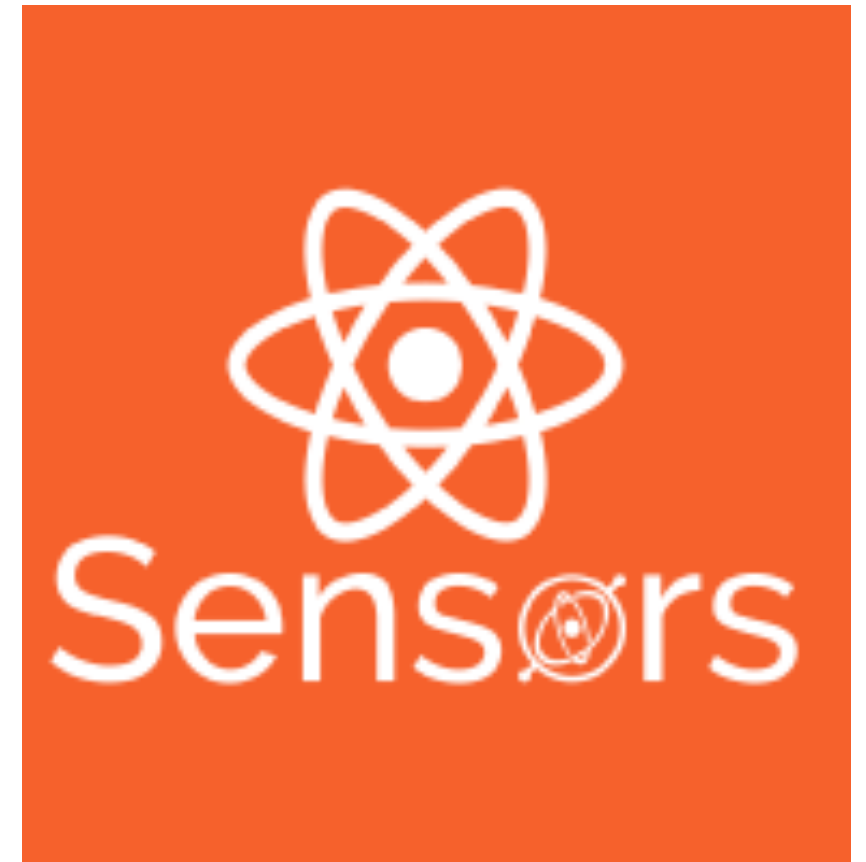
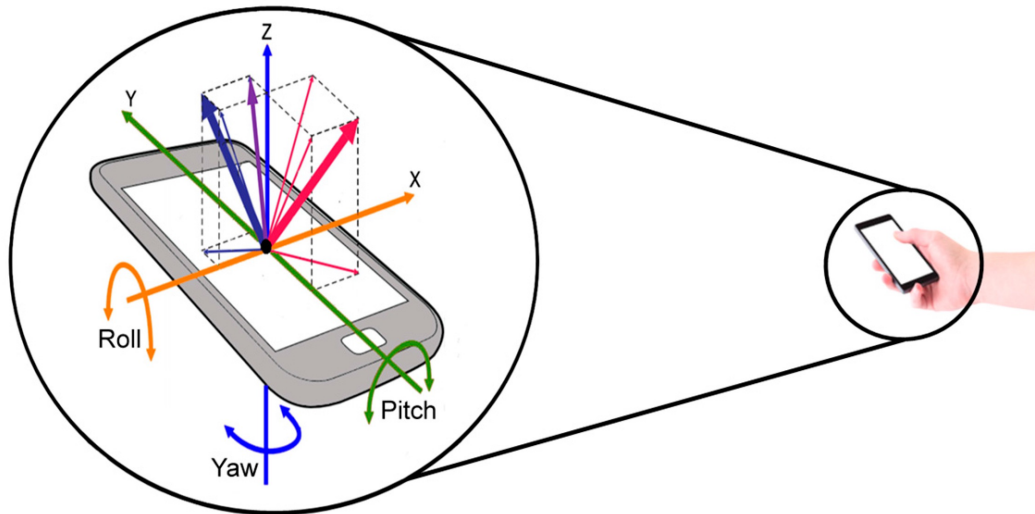
Sep Makhsous

Week 3

# React Native Sensors

---

- The React Native Sensors library is a package that provides easy-to-use APIs for accessing various device sensors in React Native apps, such as the accelerometer, gyroscope, magnetometer, and more.



# Sensors Library

- The library uses the native sensor APIs on the device, which provides accurate and reliable sensor data to your app.
- The library provides a simple and consistent API for all the sensors, making it easy to integrate sensor data into your app.
- Library: <https://react-native-sensors.github.io/>

# Sensors Library

- The React Native Sensors library provides a lot of customization options, such as setting the update interval, the maximum number of listeners, or enabling or disabling certain sensors, which allows you to fine-tune the behavior of the sensors in your app.
- One thing to keep in mind when using the React Native Sensors library is that some sensors may not be available on all devices, or their data may be affected by the device orientation or other external factors. Therefore, it's important to test your app on different devices and conditions to ensure the sensor data is accurate and reliable.

# Sensors Library Popular Applications

- Some common use cases for the React Native Sensors library include creating fitness apps, games, or augmented reality experiences that use sensor data to provide an immersive user experience.



# How to Install

- To use the React Native Sensors library in your app, you need to install it using `npm` or `yarn`, and then import the sensor classes you want to use, such as
  - Accelerometer: Measures acceleration forces on a device.
  - Gyroscope: Detects the device's rotation around its three primary axes.
  - Magnetometer: Measures the surrounding magnetic field.
  - Barometer: Detects atmospheric pressure changes.
  - Pedometer: Counts the number of steps taken by the user.
- `npm install react-native-sensors`
- `yarn add react-native-sensors.`

# Using the Subscribe Method (accelerometer)

- First Let's import the classes we need:

```
import React, { useState, useEffect } from 'react';  
import { Text, View } from 'react-native';  
import { accelerometer, setUpdateIntervalForType, SensorTypes } from  
'react-native-sensors';
```

- Then we need to create a state variable to store the accelerometer data in our functional component:

```
const [accelerometerData, setAccelerometerData] = useState({  
  x: 0,  
  y: 0,  
  z: 0,  
});
```

# Using the Subscribe Method (accelerometer)

- Next, set the update interval for the accelerometer sensor. This determines how frequently the sensor data will be updated. Here, we set it to update every 100ms:

```
setUpdateIntervalForType(SensorTypes.accelerometer, 100); // time in ms
```

- Now, create a function to handle accelerometer data updates:

```
const handleAccelerometerData = (data) => {  
  setAccelerometerData({  
    x: data.x.toFixed(2),  
    y: data.y.toFixed(2),  
    z: data.z.toFixed(2),  
  });  
};
```



# Using the Subscribe Method (accelerometer)

- Use the `useEffect` hook to subscribe to the accelerometer updates and clean up the subscription when the component is unmounted:

```
useEffect(() => {  
    const subscription = accelerometer.subscribe(handleAccelerometerData);  
  
    return () => {  
        subscription.unsubscribe();  
    };  
}, []);
```

# Using the Subscribe Method (accelerometer)

- Finally, display the accelerometer data in your component:

```
return (  
    <View>  
        <Text>Accelerometer Data:</Text>  
        <Text>X: {accelerometerData.x}</Text>  
        <Text>Y: {accelerometerData.y}</Text>  
        <Text>Z: {accelerometerData.z}</Text>  
    </View>  
);
```

Full code is provided on canvas. Keep in mind depending on your system you might need to make some small changes before you compile

# ICTE 2

- See: <https://canvas.uw.edu/courses/1633582/assignments/8288212>





