

Credit Name: CSE 2920 - CSE Project C

Assignment: AttachDetachEvents

How has your program changed from planning to coding to now? Please Explain

```
import com.phidget22.*;
public class AttachDetachEvents {

    static boolean turnRedLEDon = false;

    public static void main(String[] args) throws Exception {

        //Create
        TemperatureSensor temperatureSensor = new TemperatureSensor();
        DigitalInput redButton = new DigitalInput();
        DigitalOutput redLED = new DigitalOutput();

        //Address
        redButton.setHubPort(0);
        redButton.setIsHubPortDevice(true);
        redLED.setHubPort(1);
        redLED.setIsHubPortDevice(true);

        //Open
        temperatureSensor.open(1000);
        redButton.open(1000);
        redLED.open(1000);
```

*i made a global variable to turn red led on/off.
Then i set up temp sensor and red button and red LED.*

```
//Data Event
temperatureSensor.addTemperatureChangeListener(new TemperatureSensorTemperatureChangeListener() {
    public void onTemperatureChange(TemperatureSensorTemperatureChangeEvent e) {
        //Print temperature
        System.out.println("Temperature: " + e.getTemperature() + "°C");
    }
});

//Attach Event
temperatureSensor.addAttachListener(new AttachListener() {
    public void onAttach(AttachEvent e) {
        System.out.println("Attach Temperature Sensor!");
    }
});

//Detach Event
temperatureSensor.addDetachListener(new DetachListener() {
    public void onDetach(DetachEvent e) {
        System.out.println("Detach Temperature Sensor!");
    }
});
```

Then using a simple smart phidget event, i printed temperature as it changes. Then i added an attach and detach event which print that the temp sensor was either attached or detached. This was done using the `addAttachListener` and the `addDetachListener` which detects if the sensor was connected or disconnected.

```
//Data event for button + LED
redButton.addStateChangeListener(new DigitalInputStateChangeListener() {
    public void onStateChange(DigitalInputStateChangeEvent e) {
        if(e.getState()) {
            turnRedLEDOn = true;
        }
        else {
            turnRedLEDOn = false;
        }
    }
});
```

I then made a data event for the button and led which turns the red LED on or off as the button state changes.

```
//Attach event for button
redButton.addAttachListener(new AttachListener() {
    public void onAttach(AttachEvent e) {
        System.out.println("Attach Red Button!");
    }
});

//Detach event for LED
redButton.addDetachListener(new DetachListener() {
    public void onDetach(DetachEvent e) {
        System.out.println("Detach Red Button!");
    }
});

//Attach event for button
redLED.addAttachListener(new AttachListener() {
    public void onAttach(AttachEvent e) {
        System.out.println("Attach Red LED!");

        try {
            // Reopen the red LED
            redLED.setHubPort(1);
            redLED.setIsHubPortDevice(true);
            redLED.open(1000); // Ensure it is opened
        } catch (PhidgetException ex) {
            System.err.println("Error reinitializing Red LED: " + ex.getMessage());
        }
    }
});

//Detach event for button
redLED.addDetachListener(new DetachListener() {
    public void onDetach(DetachEvent e) {
        System.out.println("Detach Red LED!");
    }
});
```

Then i made similar attach and detach events for the red button and red LED. However, using exactly the same attach event for the LED results in errors, requiring me to research why this happens.

To understand why this happens, there are data events for both the button and temperature sensor why reinitialised their conditions allowing them to work. However there is no such thing for the LED. Therefore, the red LED is not reinitialised and it needs to be done manually. Thats why there is a try catch statement in the red LED attach listener which tries to open the redLED again but also shows the exceptions when it was not able to reinitialise the red LED.

```
//keep program running
while (true) {
    try {
        Thread.sleep(150);
        if (redLED.getAttached()) { // Check if the device is attached
            redLED.setState(turnRedLEDOn);
        } else {
            System.out.println("Red LED is not attached.");
        }
    } catch (PhidgetException ex) {
        System.err.println("Error in while loop: " + ex.getMessage());
    }
}
```

Again, continuing on from reinitialising the red LED it becomes necessary to to check if the device is attached so that the appropriate output can be displayed. If the redLED is attached then the redLED can set its state based on changes made to the red button. Otherwise a message displays that the red LED is not attached. The temperature does not need to be included in the while loop as it is controlled by its state changes.