ReflectionLog TugOfWar

```
//Create objects for the buttons and LEDs.
DigitalInput redButton = new DigitalInput();

DigitalOutput redLED = new DigitalOutput();

DigitalInput greenButton = new DigitalInput();

DigitalOutput greenLED = new DigitalOutput();

//Address the objects to their respective hub ports.
redButton.setHubPort(0);

redButton.setIsHubPortDevice(true);

redLED.setHubPort(1);

redLED.setIsHubPortDevice(true);

greenButton.setHubPort(5);

greenButton.setIsHubPortDevice(true);

greenLED.setHubPort(4);

greenLED.setIsHubPortDevice(true);

//open the devices.
redButton.open(1000);

redLED.open(1000);

greenButton.open(1000);

greenLED.open(1000);
```

This code creates four Phidget objects: two buttons (redButton and greenButton) and two LEDs (redLED and greenLED). It assigns each device to specific hub ports (0, 1, 5, and 4). The setIsHubPortDevice(true) method ensures the program recognizes these devices as connected through hub ports. The devices are then opened with a 1000ms timeout, allowing the program to communicate with the physical devices. This setup prepares the buttons and LEDs for use in the rest of the program.

```
//counters for both players
int redPressCount = 0;

int greenPressCount = 0;

//Turn both LEDs.
redLED.setState(true);

greenLED.setState(true);

//continue until one player wins
while (true) {

    //if red button is pressed, increment the red player's count and flash the red LED.
    if (redButton.getState()) {

        redPressCount++;

        System.out.println("Red Presses: " + redPressCount);

        //Flash the red LED (turn off for 100ms then back on).
        redLED.setState(false);

        Thread.sleep(100);

        redLED.setState(true);

        Thread.sleep(100);
    }

    //if green button is pressed, increment the green player's count and flash the green LED.
    if (greenButton.getState()) {

        greenPressCount++;

        System.out.println("Green Presses: " + greenPressCount);

        //flash the green LED (turn off for 100ms then back on).
        greenLED.setState(false);
        Thread.sleep(100);
        greenLED.setState(true);
        Thread.sleep(100);
    }
```

This code creates a simple game where two players, using red and green buttons, try to accumulate presses. Each player's button press increments their respective counter (redPressCount or greenPressCount), and the current count is printed to the console. When a button is pressed, the corresponding LED flashes by turning off for 100 milliseconds and then back on. The program runs in a loop, continuously checking for button presses and updating the LEDs and counters until one of the players wins.

```
//check if either player has won (10 presses).
if (redPressCount >= 10 || greenPressCount >= 10) {
    // Flash both LEDs once
    redLED.setState(true);

    greenLED.setState(true);

    //both LEDs on for 200ms..
    Thread.sleep(200);


    redLED.setState(false);

    greenLED.setState(false);

    //both LEDs off for 200ms.
    Thread.sleep(200);

    //flash the winner's LED 5 times.
    if (redPressCount >= 10) {

        for (int i = 0; i < 5; i++) {

            redLED.setState(true);

            Thread.sleep(200);  //red LED on for 200ms.


            redLED.setState(false);

            Thread.sleep(200);  //red LED off for 200ms.
        }

        System.out.println("Red is the winner!");

    } else {

        for (int i = 0; i < 5; i++) {

            greenLED.setState(true);

            Thread.sleep(200);  //green LED on for 200ms.
```

This code checks if either player (red or green) has won by reaching 10 button presses. If a player wins, both LEDs flash on and off for 200 milliseconds. The winner's LED (either red or green) then flashes five times, turning on for 200 milliseconds and off for 200 milliseconds, indicating the victory. After flashing the winning LED, a message is printed in the console declaring the winner. This process continues until one of the players achieves 10 button presses.

```
                }
                System.out.println("Green is the winner!");
            }

            //exit the game after displaying the winner.
            break;
        }

        //avoid over-counting.
        Thread.sleep(150);
    }
}
```

This code completes the game by announcing the winner (either "Red is the winner!" or "Green is the winner!") once one of the players reaches 10 button presses. After displaying the winner, it exits the game by breaking out of the loop. The Thread.sleep(150) ensures that the program avoids counting multiple presses too quickly, preventing over-counting. The game stops once a player wins, and no further button presses are processed after the winner is declared.