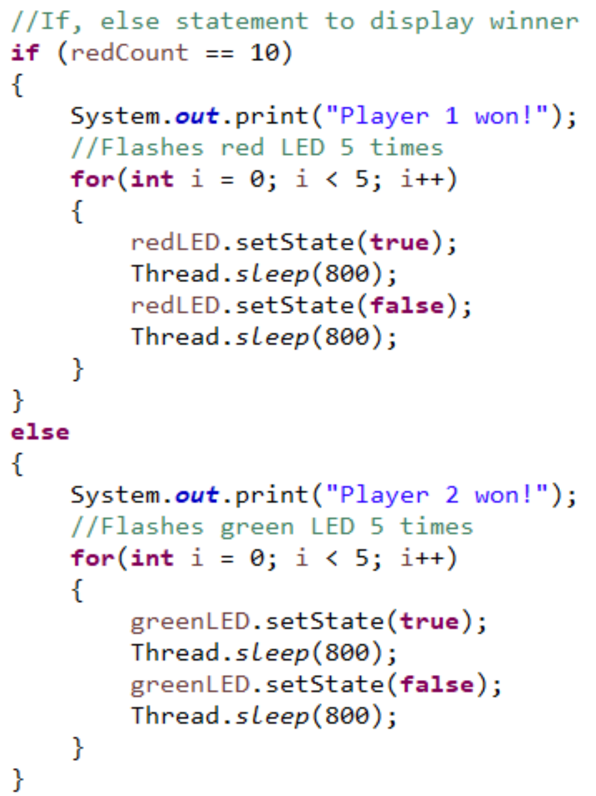
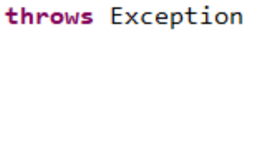


```
//If, else statement to display winner
if (redCount == 10)
{
    System.out.print("Player 1 won!");
    //Flashes red LED 5 times
    for(int i = 0; i < 5; i++)
    {
        redLED.setState(true);
        Thread.sleep(800);
        redLED.setState(false);
        Thread.sleep(800);
    }
}
else
{
    System.out.print("Player 2 won!");
    //Flashes green LED 5 times
    for(int i = 0; i < 5; i++)
    {
        greenLED.setState(true);
        Thread.sleep(800);
        greenLED.setState(false);
        Thread.sleep(800);
    }
}
```

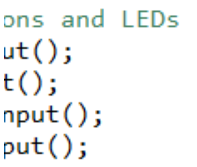
w? Please explain?

h the value of 0 to count the number of times each button is pressed.



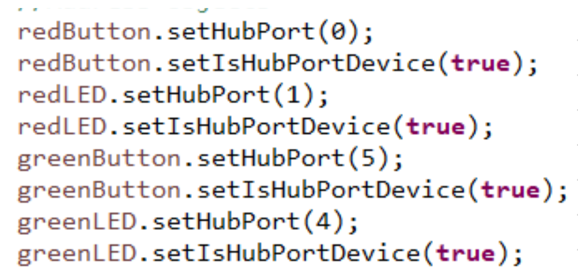
```
throws Exception
```

g a class to read the button's and LED's state.



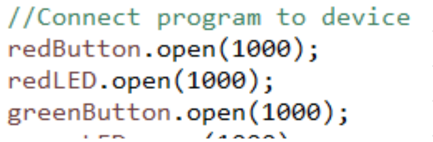
```
ons and LEDs
ut();
t();
nput();
put();
```

t, setting the setIsHubPortDevice as true.

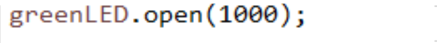


```
redButton.setHubPort(0);
redButton.setIsHubPortDevice(true);
redLED.setHubPort(1);
redLED.setIsHubPortDevice(true);
greenButton.setHubPort(5);
greenButton.setIsHubPortDevice(true);
greenLED.setHubPort(4);
greenLED.setIsHubPortDevice(true);
```

Opened a connection between the program and the physical device. Will timeout if after 1 sec(1000 miliseconds) and throw execption if it cannot locate the device.

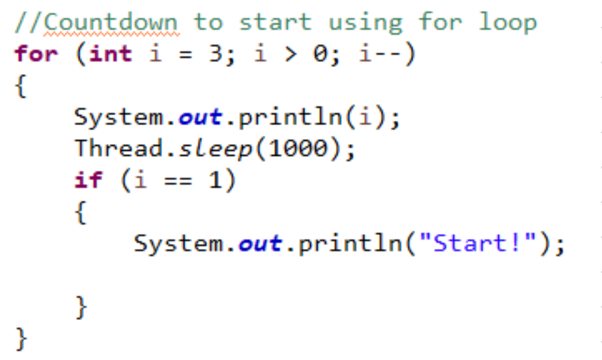


```
//Connect program to device
redButton.open(1000);
redLED.open(1000);
greenButton.open(1000);
```



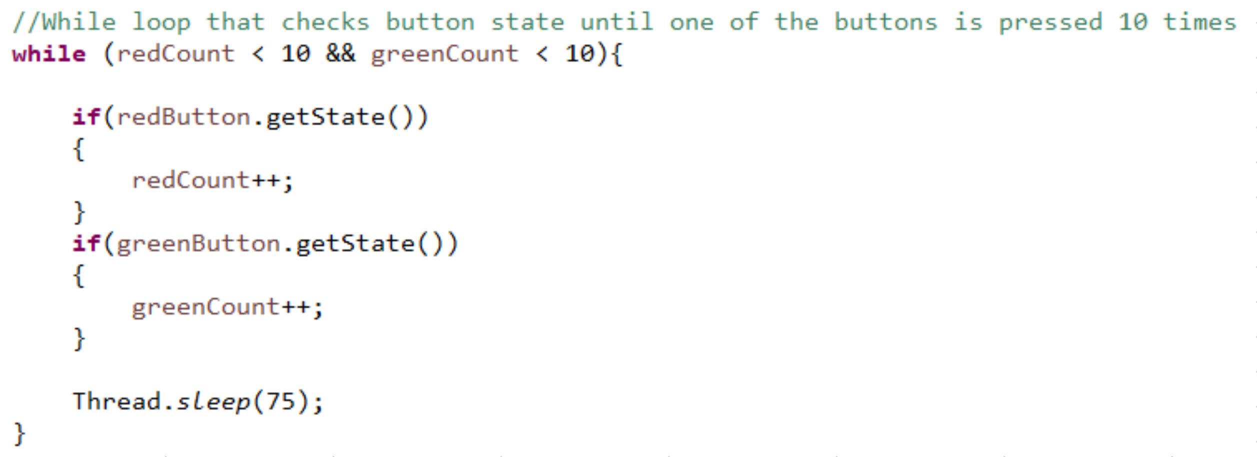
```
greenLED.open(1000);
```

Used a for loop to display a countdown starting from 3. Once the countdown reaches 1 it will display Start.



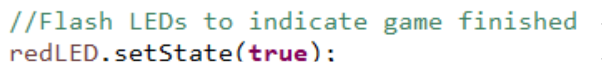
```
//Countdown to start using for loop
for (int i = 3; i > 0; i--)
{
    System.out.println(i);
    Thread.sleep(1000);
    if (i == 1)
    {
        System.out.println("Start!");

    }
}
```

Used a while loop to continue looping while the red and green count are under 10. If the red button state is true, it will add 1 to the red count. If the green button state is true, it will add 1 to the green count. The code will then stop running for 75 miliseconds before continuing.

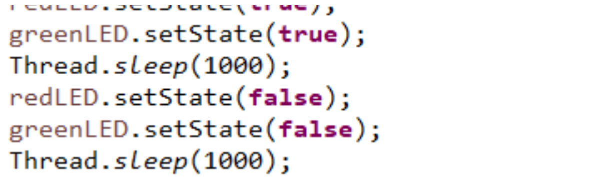


```
//While loop that checks button state until one of the buttons is pressed 10 times
while (redCount < 10 && greenCount < 10){

    if(redButton.getState())
    {
        redCount++;
    }
    if(greenButton.getState())
    {
        greenCount++;
    }

    Thread.sleep(75);
}
```

This code of code flashes both the red and green LEDs once. Red and green LED are set as true to turn it on, then the code will stop running for 1 second using Thread.sleep, leaving the LEDs on. Both the LEDs are then set as false to turn off the lights, waiting 1 second before moving on to the next part of the code.

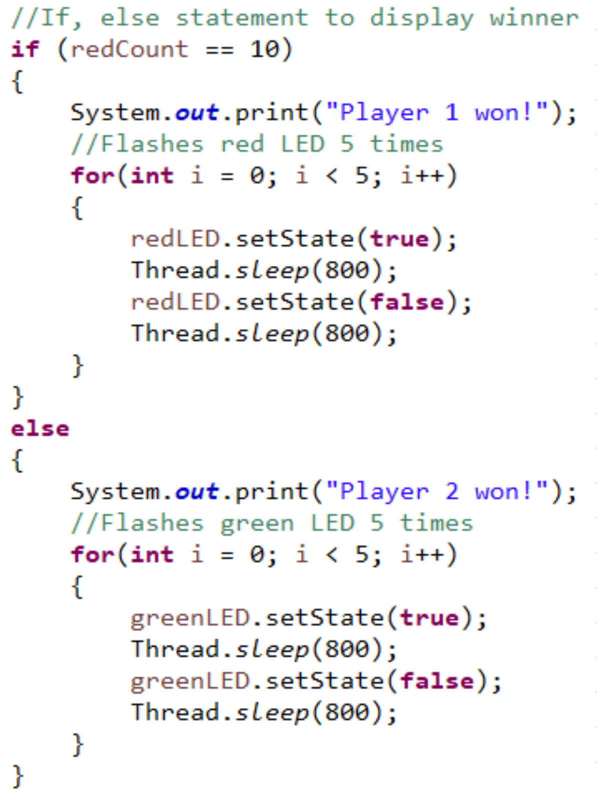


```
//Flash LEDs to indicate game finished
redLED.setState(true);
```



```java
redLED.setState(true);
greenLED.setState(true);
Thread.sleep(1000);
redLED.setState(false);
greenLED.setState(false);
Thread.sleep(1000);
```

Used an if statement to display winner. If the red count is equal to 10, it will print out that Player 1 has won. I then used a for loop to flash the red LED 5 times by setting its state as true then leaving it on for 800 miliseconds using Thread.sleep before turning off the LED by setting its state as false. Repeated above for Player 2 using an else statement and flashing the green LED.



```java
//If, else statement to display winner
if (redCount == 10)
{
    System.out.print("Player 1 won!");
    //Flashes red LED 5 times
    for(int i = 0; i < 5; i++)
    {
        redLED.setState(true);
        Thread.sleep(800);
        redLED.setState(false);
        Thread.sleep(800);
    }
}
else
{
    System.out.print("Player 2 won!");
    //Flashes green LED 5 times
    for(int i = 0; i < 5; i++)
    {
        greenLED.setState(true);
        Thread.sleep(800);
        greenLED.setState(false);
        Thread.sleep(800);
    }
}
```