Light Up Compression Gloves

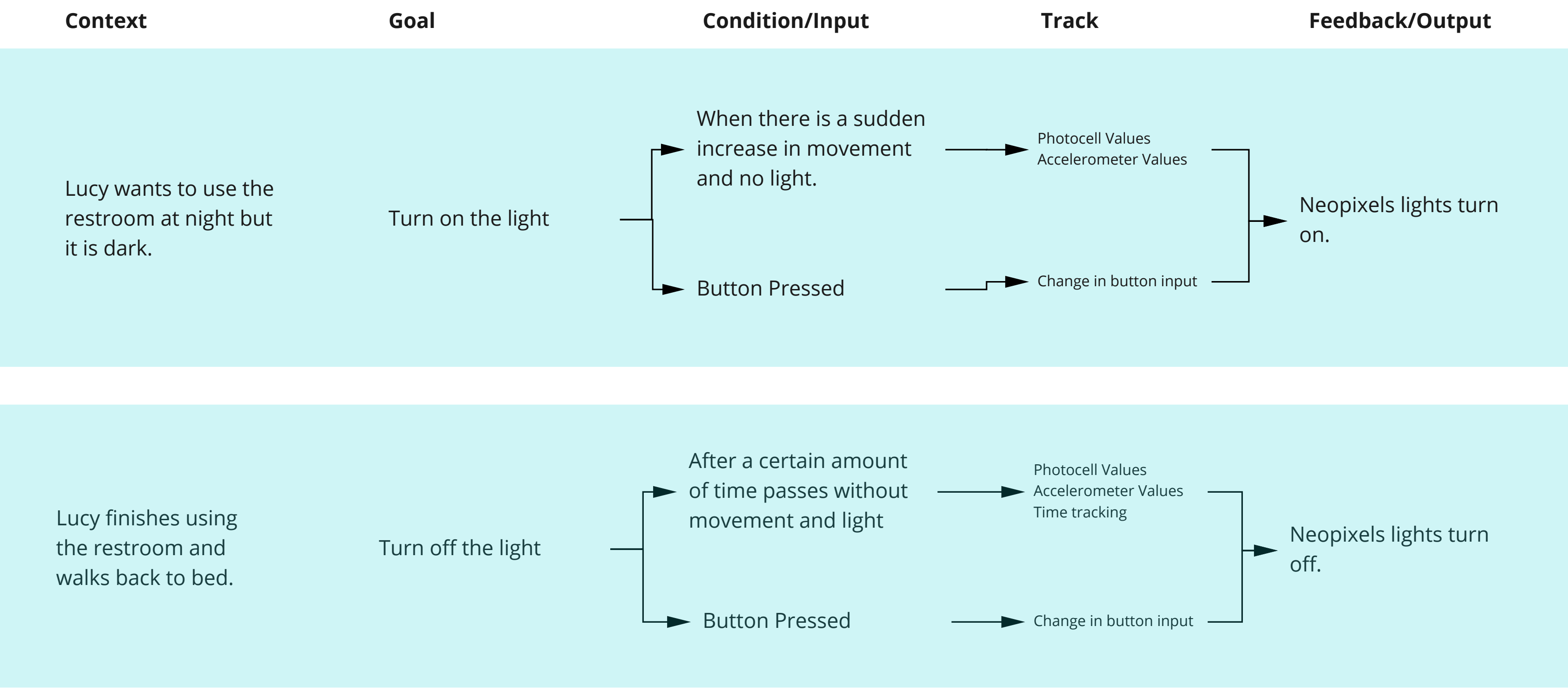
Lucas Thin | Summer 2021

Problem

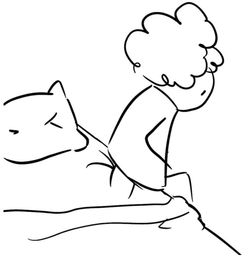
Older adults that visits the restroom at night often finds it too dark to see obstacles along the way. Most of the time the light switches also have some distance to turn it on. This could result in unnecessary falls that causes injuries.

User : Older Adults

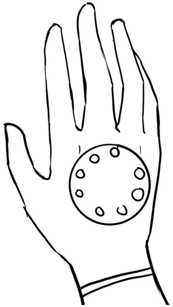
Goal : To turn the lights on when needed effortlessly.



Context	Goal	Condition/Input	Track	Feedback/Output
Lucy wants to use the restroom at night but it is dark.	Turn on the light	When there is a sudden increase in movement and no light.	Photocell Values Accelerometer Values	Neopixels lights turn on.
		Button Pressed	Change in button input	
Lucy finishes using the restroom and walks back to bed.	Turn off the light	After a certain amount of time passes without movement and light	Photocell Values Accelerometer Values Time tracking	Neopixels lights turn off.
		Button Pressed	Change in button input	



Lucy wakes up at night to use the restroom



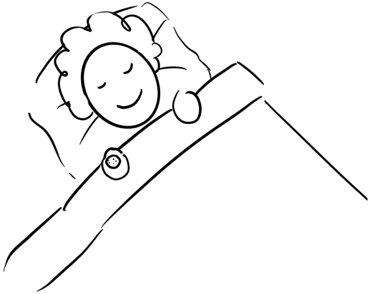
Lucy's lights detected she's active so the lights automatically turns on

With the help of the light, Lucy was able to find her walking cane.

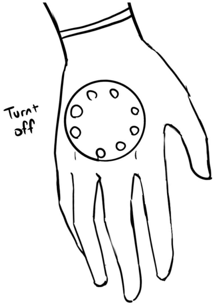
Lucy then uses the light to walk towards the bathroom.

She enters her bathroom and the bathroom's motion detecting lights turnt on. Noticing the lights, her glove light turns itself off.

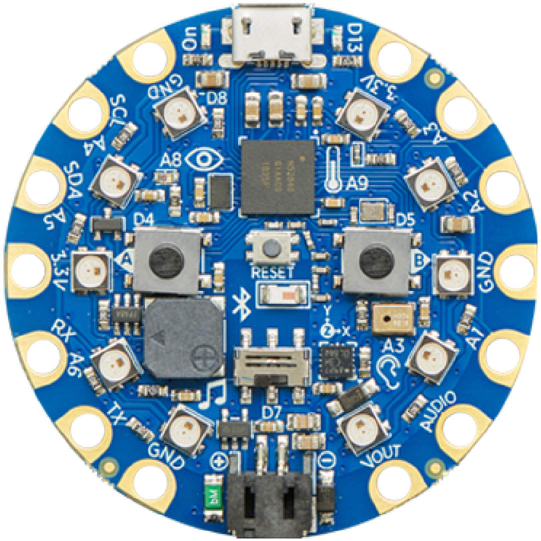
when leaving the restroom, her lights turnt itself back on.



Lucy finally reach back to bed safely.



The glove lights noticed this and turnt itself off.

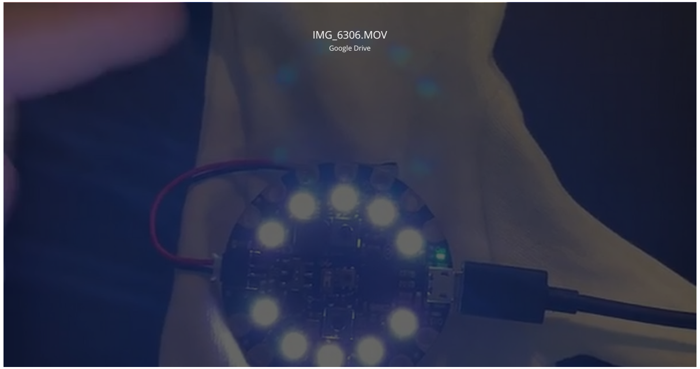


Circuit Playground Bluefruit

- 3-Axis Accelerometer
- Light Sensor
- Bluetooth
- Switch

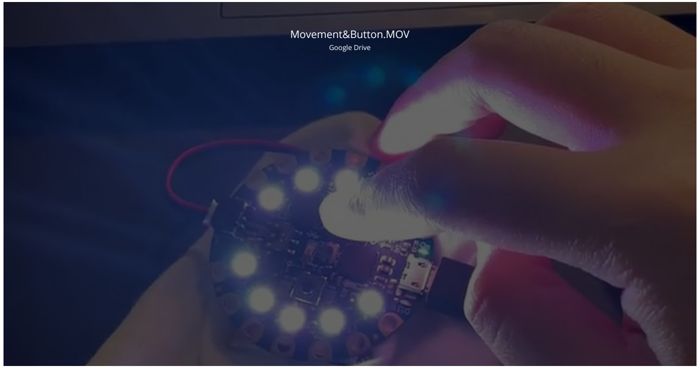
Lithium Ion Battery

Toggle Light on / off
with the light sensor

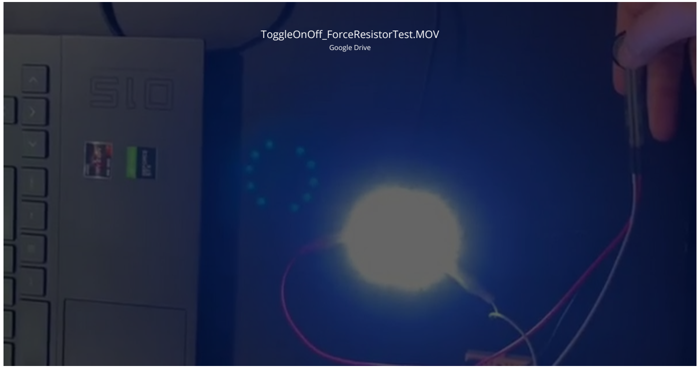


Toggle Light on / off
with the buttonA

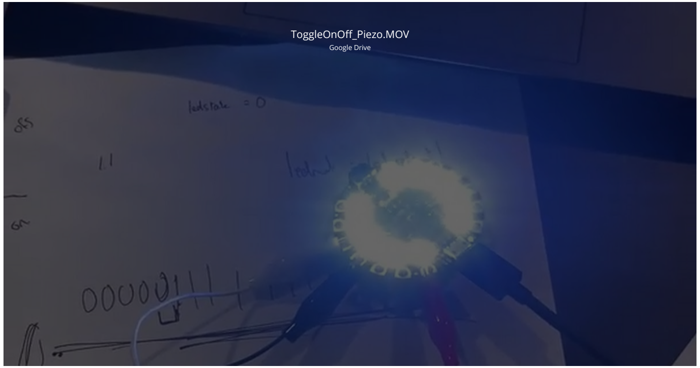
Toggle Light on / off
with just movement



Toggle Light on / off with
movement and button



Toggle Light on / off
with the force resistor



Toggle Light on / off
with the PiezoElement

Challenge

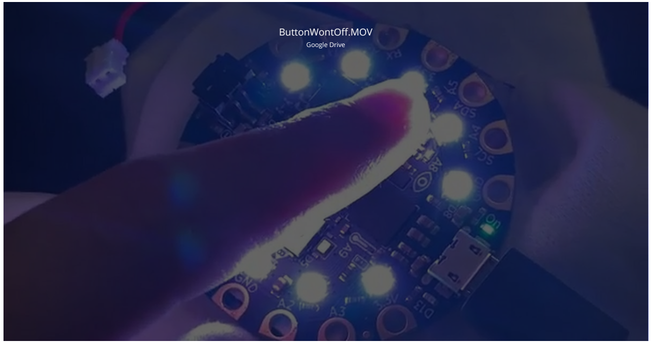
Getting two inputs to toggle the same output without clashing with each other.

Solution

- Reading inputs per interval rather than using time.sleep
- Using Timeout Timers

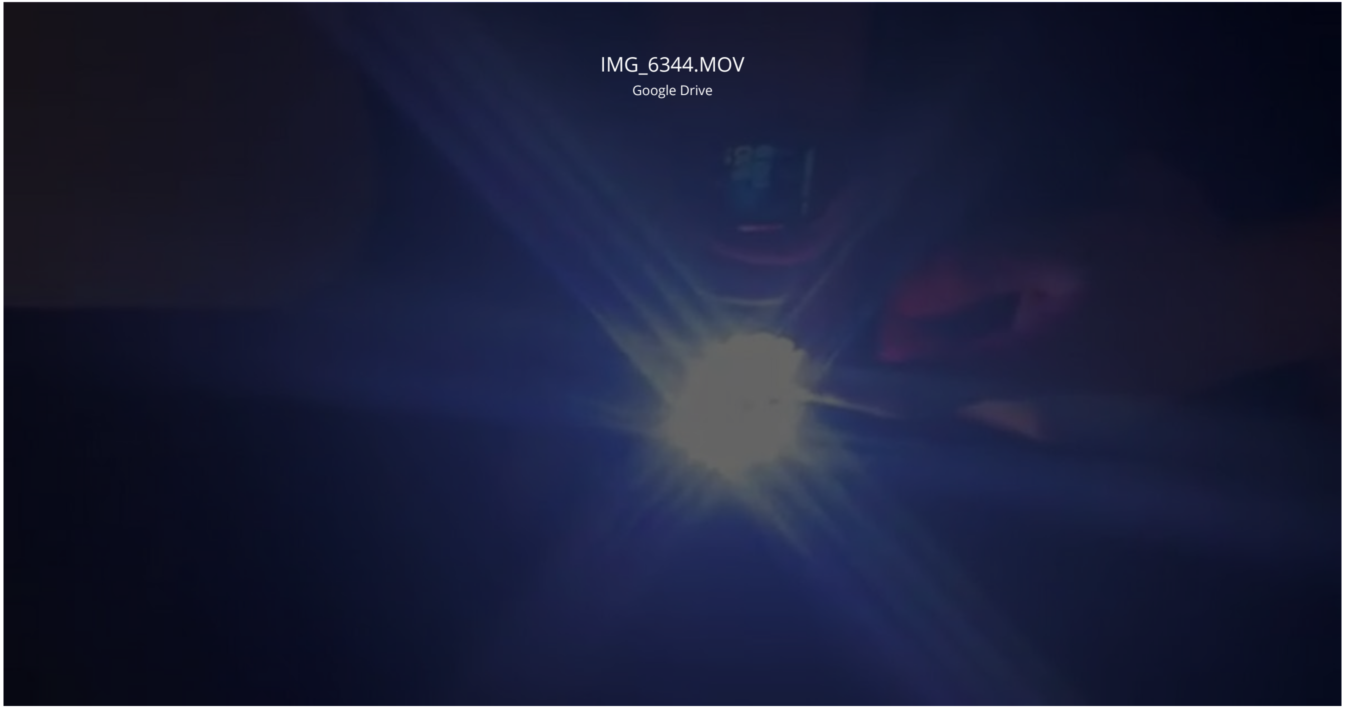
ButtonWontOn.MOV

Google Drive



ButtonWontOff.MOV

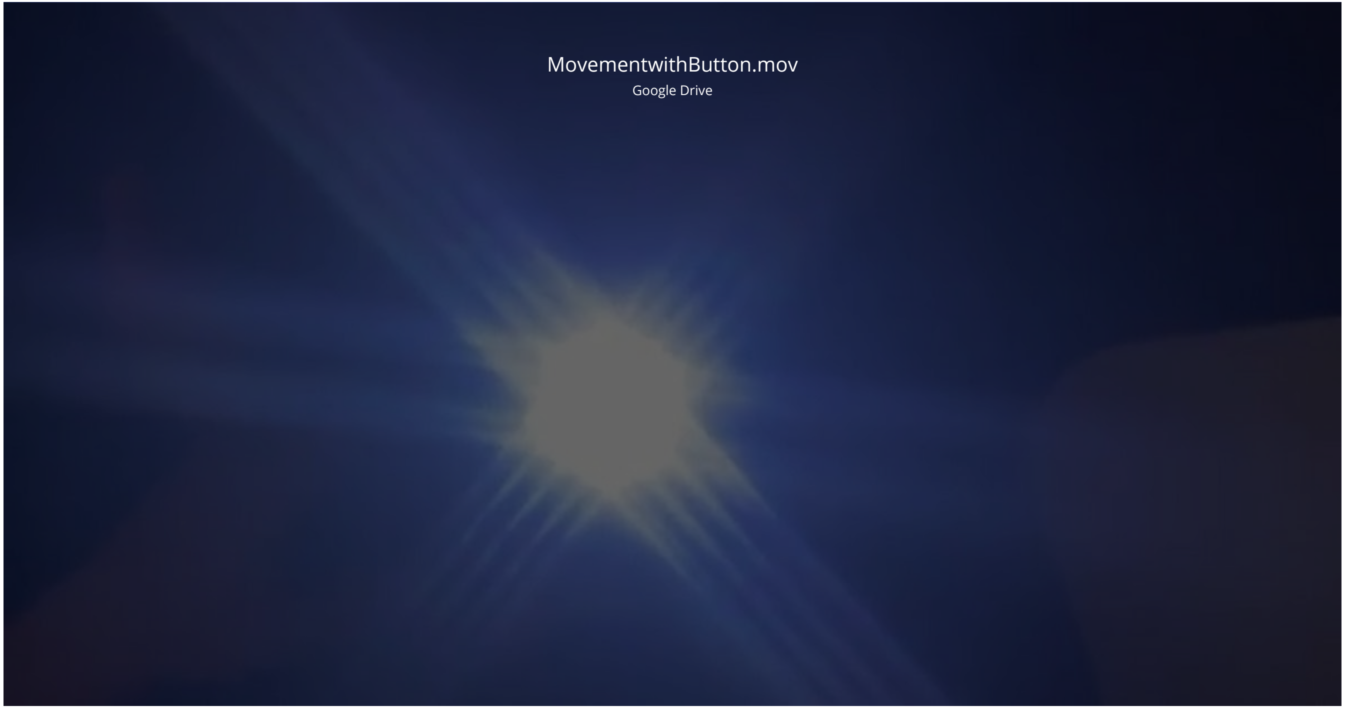
Google Drive



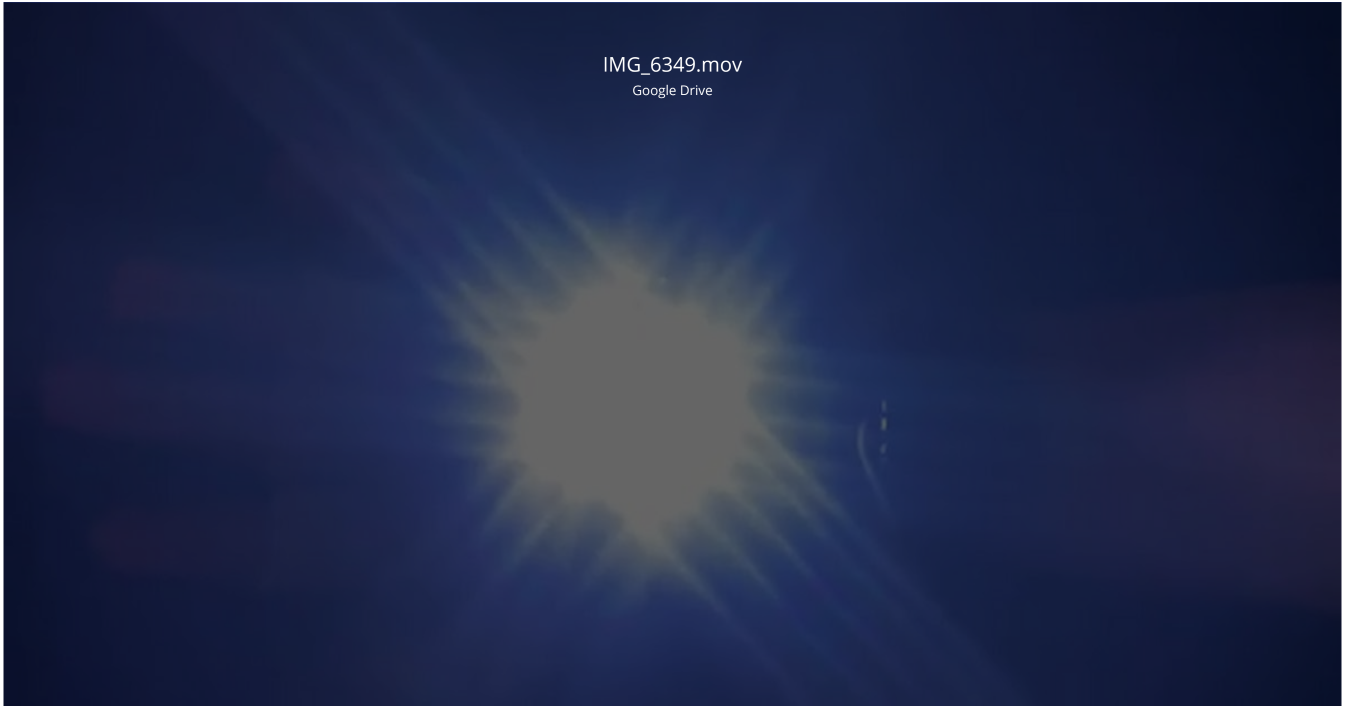
IMG_6344.MOV

Google Drive

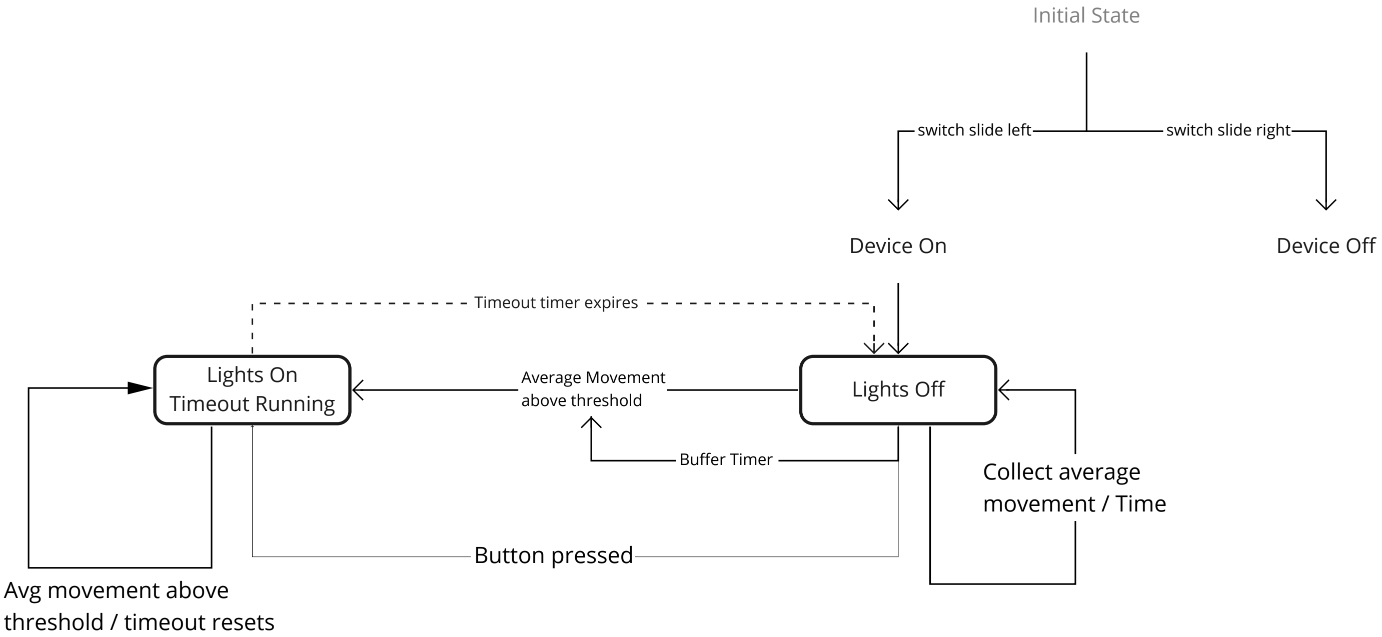
Turn on light with a button

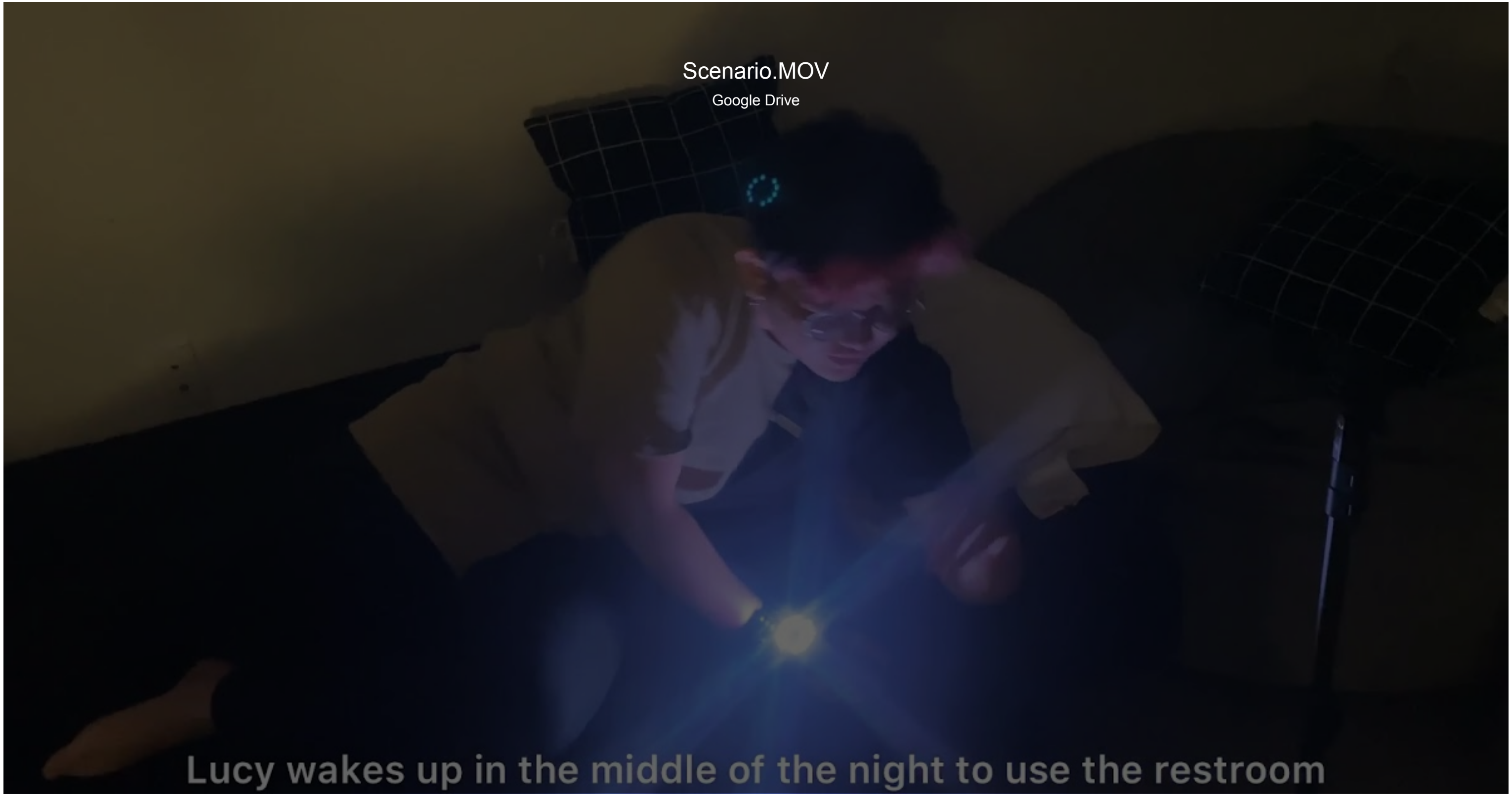


Turn on light with movement and
control with button afterwards



Toggle light off light when no movement

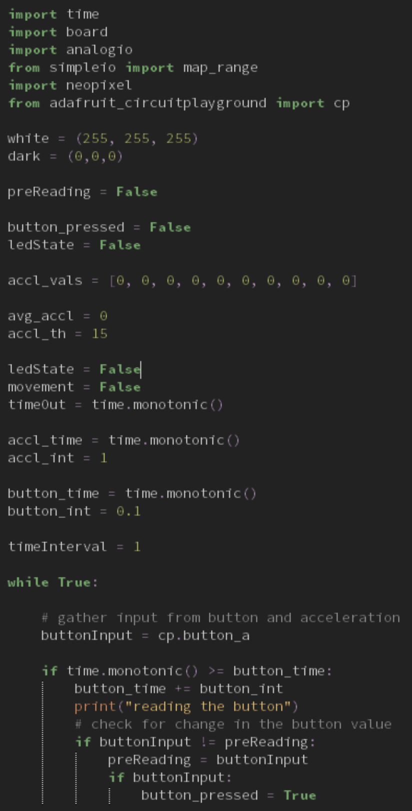
Initial State
switch slide left
switch slide right
Device On
Device Off
Timeout timer expires
Lights On
Timeout Running
Average Movement
above threshold
Lights Off
Buffer Timer
Collect average
movement / Time
Button pressed
Avg movement above
threshold / timeout resets



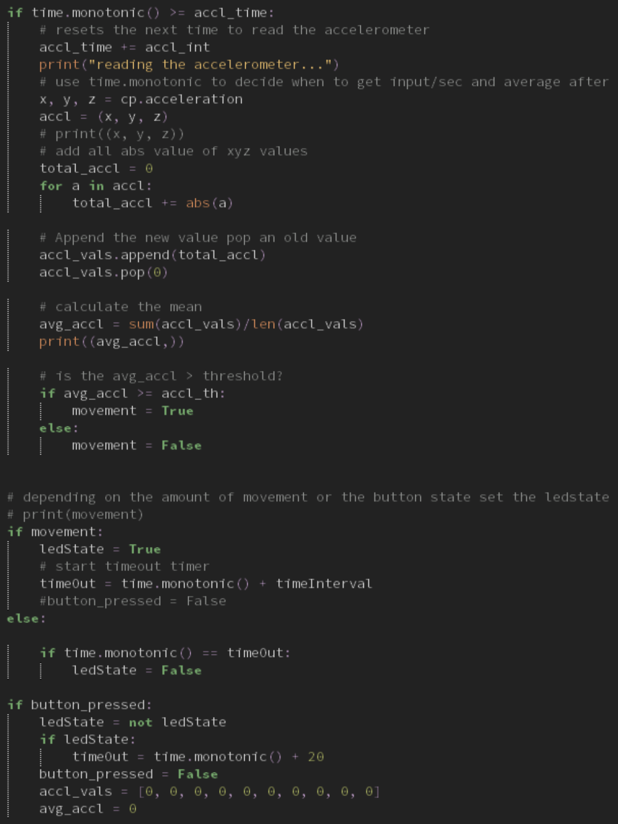
Scenario.MOV

Google Drive

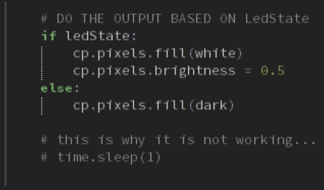
Lucy wakes up in the middle of the night to use the restroom



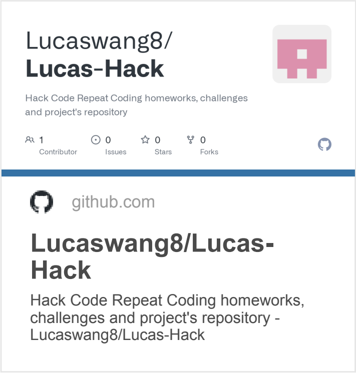
```
import time
import board
import analogio
from simpleio import map_range
import neopixel
from adafruit_circuitplayground import cp

white = (255, 255, 255)
dark = (0,0,0)

preReading = False

button_pressed = False
ledState = False

accl_vals = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

avg_accl = 0
accl_th = 15

ledState = False
movement = False
timeOut = time.monotonic()

accl_time = time.monotonic()
accl_int = 1

button_time = time.monotonic()
button_int = 0.1

timeInterval = 1

while True:

    # gather input from button and acceleration
    buttonInput = cp.button_a

    if time.monotonic() >= button_time:
        button_time += button_int
        print("reading the button")
        # check for change in the button value
        if buttonInput != preReading:
            preReading = buttonInput
            if buttonInput:
                button_pressed = True
```



```
if time.monotonic() >= accl_time:
    # resets the next time to read the accelerometer
    accl_time += accl_int
    print("reading the accelerometer...")
    # use time.monotonic to decide when to get input/sec and average after
    x, y, z = cp.acceleration
    accl = (x, y, z)
    # print((x, y, z))
    # add all abs value of xyz values
    total_accl = 0
    for a in accl:
        total_accl += abs(a)

    # Append the new value pop an old value
    accl_vals.append(total_accl)
    accl_vals.pop(0)

    # calculate the mean
    avg_accl = sum(accl_vals)/len(accl_vals)
    print((avg_accl,))

    # is the avg_accl > threshold?
    if avg_accl >= accl_th:
        movement = True
    else:
        movement = False


# depending on the amount of movement or the button state set the ledstate
# print(movement)
if movement:
    ledState = True
    # start timeout timer
    timeOut = time.monotonic() + timeInterval
    #button_pressed = False
else:

    if time.monotonic() == timeOut:
        ledState = False

if button_pressed:
    ledState = not ledState
    if ledState:
        timeOut = time.monotonic() + 20
    button_pressed = False
    accl_vals = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    avg_accl = 0
```



```
# DO THE OUTPUT BASED ON LedState
if ledState:
    cp.pixels.fill(white)
    cp.pixels.brightness = 0.5
else:
    cp.pixels.fill(dark)

# this is why it is not working...
# time.sleep(1)
```



Lucaswang8/
Lucas-Hack

Hack Code Repeat Coding homeworks, challenges and project's repository

1 Contributor 0 Issues 0 Stars 0 Forks

github.com

Lucaswang8/Lucas-Hack

Hack Code Repeat Coding homeworks, challenges and project's repository - Lucaswang8/Lucas-Hack