

ECE 476/676 - Homework #5

Graphic Display & Touch-Screen

Lights Out

Write a python program to play the game of lights out.

- The game consists of eight lights (displayed on the graphics display).
- At the start of the game, each light is either on or off at random.

Each round, the operator taps on one of the LEDs

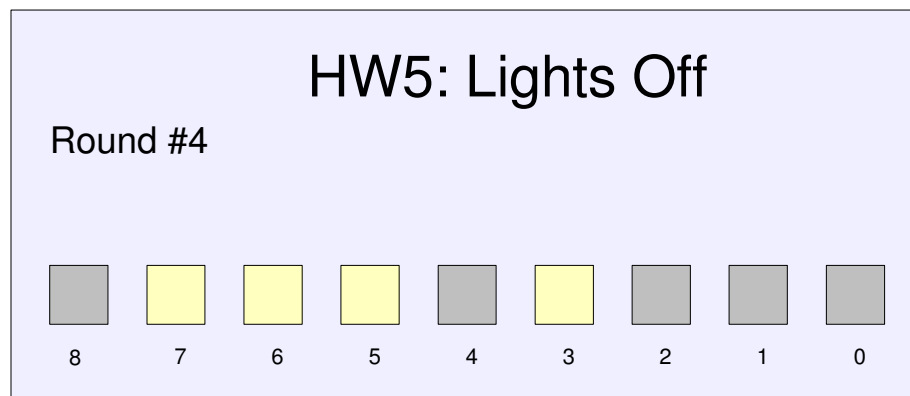
- Tapping on that LED toggles that LED as well as its two neighbors
- An ON LED becomes OFF.
- An OFF LED becomes ON

For example, in the figure below

- Tapping box #4 would toggle lights (5, 4, 3)
- Tapping box #8 would toggle lights (7, 8)
- Tapping box #1 would toggle lights (1, 0)

The goal of the game is to get all of the lights to turn off.

The LCD display could look something like this (not required - just one option)



Grading

- 10pt Generate random on/off lights at the start of each game
- 10pt Display the lights using the LCD graphics display
 - Use the shell window as an alternative (no points)
- 10pt Use the touch pad to select which LEDs to toggle
 - Use the terminal window for input as an alternative (no points)
- 10pt Toggle the selected LED and its neighbors
- 10pt Keep track and display the number of turns taken on the LCD
- 10pt Demonstrate the lights out game
 - Video preferred
 - Several photos also works

Solution:

Start with the display routine

- Display a 9-bit binary number as lights
- Orange: bit is a 1
- Black: bit is a 0

```
import LCD_16x24 as LCD
from machine import Pin
from time import sleep

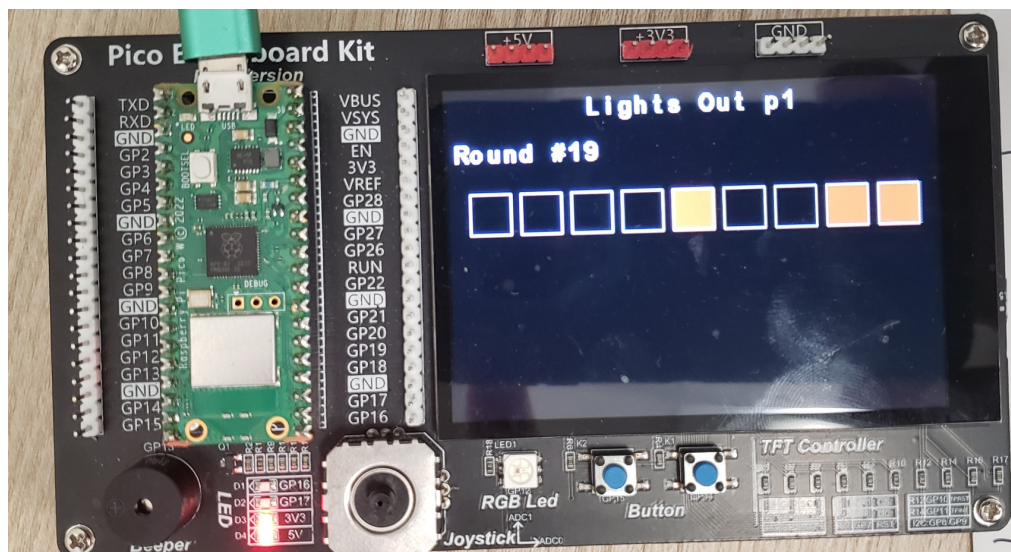
a2d0 = machine.ADC(1)

Navy = LCD.RGB(0,0,10)
Yellow = LCD.RGB(250,250,0)
White = LCD.RGB(250,250,250)
Black = LCD.RGB(0,0,0)
Orange = LCD.RGB(250,100,0)

LCD.Init()
LCD.Clear(Black)
LCD.Title('Lights Out p1', White, Black)

def Display(Round, Key):
    msg = str(Round)
    LCD.Text3('Round #' + str(Round), 5, 50, White, Black)
    for i in range(0,9):
        mask = 2 ** (8-i)
        x = 20 + 50*i
        y = 100
        if(mask & Key):
            LCD.Solid_Box(x, y, x+40, y+40, Orange)
            LCD.Box(x, y, x+40, y+40, White)
        else:
            LCD.Solid_Box(x, y, x+40, y+40, Black)
            LCD.Box(x, y, x+40, y+40, White)

for i in range(0,20):
    Display(i, i)
    sleep(0.1)
```



Lights-Out version 2

Add in a random starting position

```
def New_Game():
    LCD.Clear(Black)
    LCD.Box(1,1,479,319,White)
    LCD.Title('Lights-Out p2', White, Black)
    Key = randint(1,511)
    return(Key)
```

Read the keypad for now to test the code

```
def Read_Keypad(nx,ny):
    x = int( input('Button = ') )
    y = 3
    return(x, y)
```

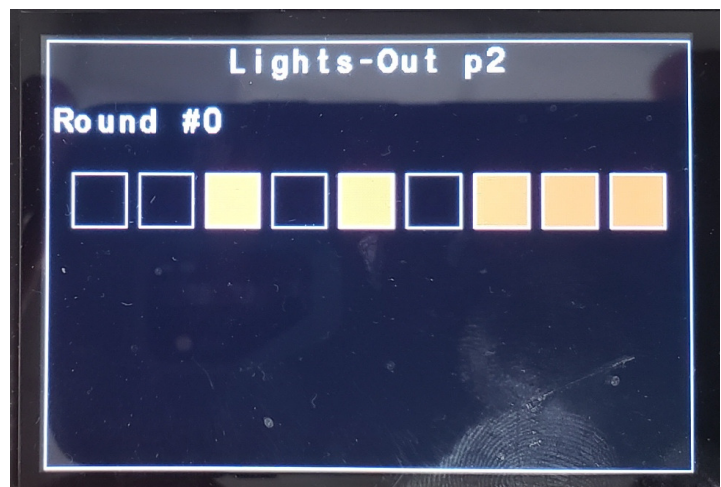
Play a game using the keyboard

```
# Main Routine
Round = 0
while(1):
    Key = New_Game()
    Display(Round, Key)
    while(Key > 0):
        [x, y] = Read_Keypad(10,10)
        if(x == 1):
            Mask = 3
        else:
            Mask = 7 << (x-2)
        Mask &= 0x1FF
        Key ^= Mask
        print('x = ', x, ' Mask = ', Mask, ' Key = ', Key)
        Round += 1
        Display(Round, Key)
    print('Completed in ', Round, ' rounds')
    LCD.Text3('Completed in ' + str(Round) + ' Rounds.', 5, 200, White, Black)
    sleep(5)
    Round = 0
```

Console Window:

```
>>> %Run -c $EDITOR_CONTENT

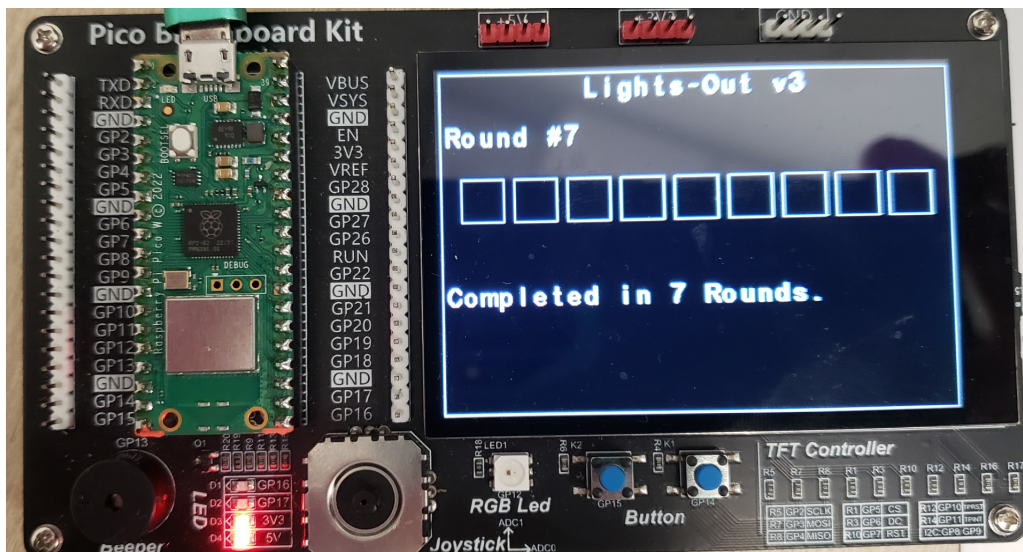
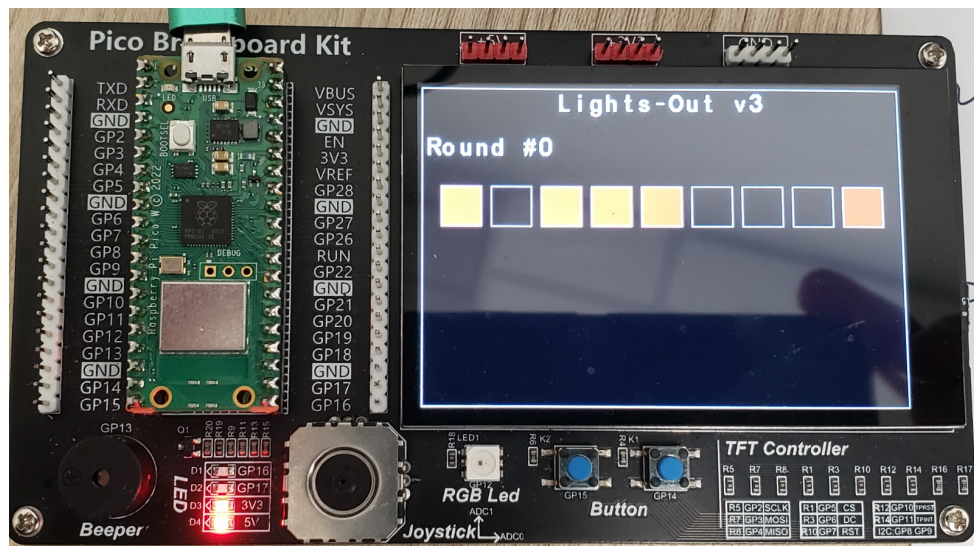
MPY: soft reboot
Button = 2
x = 2 Mask = 7 Key = 80
Button = 6
x = 6 Mask = 112 Key = 32
Button = 5
x = 5 Mask = 56 Key = 24
Button = 4
x = 4 Mask = 28 Key = 4
Button = 1
x = 1 Mask = 3 Key = 7
Button = 2
x = 2 Mask = 7 Key = 0
Completed in 6 rounds
```



Lights-Out Version 3

Replace the keyboard entry with a touch screen entry

```
def Read_Keypad(x, y):
    Sx = 480//x
    Sy = 320//y
    num_points, points_data = touch.read_points()
    while(num_points == 0):
        num_points, points_data = touch.read_points()
        sleep_ms(30)
    tx = points_data[0][0] // Sx
    ty = points_data[0][1] // Sy
    while(num_points > 0):
        num_points, points_data = touch.read_points()
        sleep_ms(30)
    return(x-tx, ty)
```



Final Code

```
from machine import Pin, I2C
from time import sleep, sleep_ms
from random import randint
import LCD_16x24 as LCD
from gt911 import GT911

# Touch Screen Initialization
rst_pin = Pin(10, Pin.OUT)
irq_pin = Pin(11)
sda_pin = Pin(8)
scl_pin = Pin(9)
touch = GT911(
    I2C(0, scl=scl_pin, sda=sda_pin, freq=100_000),
    rst_pin,
    irq_pin
)

touch.init(touch_points=1, refresh_rate=50)

def Read_Keypad(x, y):
    Sx = 480//x
    Sy = 320//y
    num_points, points_data = touch.read_points()
    while(num_points == 0):
        num_points, points_data = touch.read_points()
        sleep_ms(30)
    tx = points_data[0][0] // Sx
    ty = points_data[0][1] // Sy
    while(num_points > 0):
        num_points, points_data = touch.read_points()
        sleep_ms(30)
    return(x-tx, ty)

# LCD initialization

LCD.Init()
Black = LCD.RGB(0,0,0)
White = LCD.RGB(250,250,250)
Grey = LCD.RGB(50,50,50)
Red = LCD.RGB(250,0,0)
Orange = LCD.RGB(250,100,0)
Yellow = LCD.RGB(250,250,0)
Green = LCD.RGB(50,250,0)
Blue = LCD.RGB(0, 50, 250)

def New_Game():
    LCD.Clear(Black)
    LCD.Box(1,1,479,319,White)
    LCD.Title('Lights-Out v3', White, Black)
    Key = randint(1,511)
    return(Key)

def Display(Round, Key):
    msg = str(Round)
    LCD.Text3('Round #' + str(Round), 5, 50, White, Black)
    for i in range(0,9):
        mask = 2 ** (8-i)
        x = 20 + 50*i
        y = 100
        if(mask & Key):
            LCD.Solid_Box(x, y, x+40, y+40, Orange)
            LCD.Box(x, y, x+40, y+40, White)
        else:
            LCD.Solid_Box(x, y, x+40, y+40, Black)
            LCD.Box(x, y, x+40, y+40, White)
```

```

# Main Routine
Round = 0
while(1):
    Key = New_Game()
    Display(Round, Key)
    while(Key > 0):
        [x, y] = Read_Keypad(9,6)
        if(x == 1):
            Mask = 3
        else:
            Mask = 7 << (x-2)
        Mask &= 0x1FF
        Key ^= Mask
        print('x = ', x, ' Mask = ', Mask, ' Key = ', Key)
        Round += 1
        Display(Round, Key)
    print('Completed in ', Round, ' rounds')
    LCD.Text3('Completed in ' + str(Round) + ' Rounds.', 5, 200, White, Black)
    sleep(5)
    Round = 0

```