from spike import PrimeHub, LightMatrix, Button, StatusLight, ForceSensor, MotionSensor, Speaker, ColorSensor, App, DistanceSensor, Motor, MotorPair

from spike.control import wait\_for\_seconds, wait\_until, Timer

from math import \*

import random

hub = PrimeHub()

motor1 = Motor('E')

motor2 = Motor('D')

motorLock = Motor('F')

timer = Timer()

motorLock.set\_default\_speed(10)

motorLock.run\_to\_position(0)

matrix = [[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0]]

def show\_matrix(matrix):

for i in range(len(matrix)):

for j in range(len(matrix[0])):

if matrix[i][j] == 1:

hub.light\_matrix.set\_pixel(j, i, 100)

else:

hub.light\_matrix.set\_pixel(j, i, 0)

def motor1\_position():

position = motor1.get\_position()

if 12 > position or position > 349:

matrix[0][2] = 1

else:

matrix[0][2] = 0

if 33 > position and position > 11:

matrix[0][3] = 1

else:

matrix[0][3] = 0

if 56 > position and position > 32:

matrix[0][4] = 1

else:

matrix[0][4] = 0

if 77 > position and position > 55:

matrix[1][4] = 1

else:

matrix[1][4] = 0

if 101 > position and position > 76:

matrix[2][4] = 1

else:

matrix[2][4] = 0

if 123 > position and position > 100:

matrix[3][4] = 1

else:

matrix[3][4] = 0

if 144 > position and position > 122:

matrix[4][4] = 1

else:

matrix[4][4] = 0

if 168 > position and position > 143:

matrix[4][3] = 1

else:

matrix[4][3] = 0

if 192 > position and position > 167:

matrix[4][2] = 1

else:

matrix[4][2] = 0

if 213 > position and position > 191:

matrix[4][1] = 1

else:

matrix[4][1] = 0

if 236 > position and position > 212:

matrix[4][0] = 1

else:

matrix[4][0] = 0

if 259 > position and position > 235:

matrix[3][0] = 1

else:

matrix[3][0] = 0

if 282 > position and position > 258:

matrix[2][0] = 1

else:

matrix[2][0] = 0

if 303 > position and position > 281:

matrix[1][0] = 1

else:

matrix[1][0] = 0

if 325 > position and position > 302:

matrix[0][0] = 1

else:

matrix[0][0] = 0

if 350 > position and position > 325:

matrix[0][1] = 1

else:

matrix[0][1] = 0

def motor2\_position():

position = motor2.get\_position()

if 23 > position or position > 338:

matrix[1][2] = 1

else:

matrix[1][2] = 0

if 48 > position and position > 23:

matrix[1][3] = 1

else:

matrix[1][3] = 0

if 113 > position and position > 47:

matrix[2][3] = 1

else:

matrix[2][3] = 0

if 158 > position and position > 112:

matrix[3][3] = 1

else:

matrix[3][3] = 0

if 203 > position and position > 157:

matrix[3][2] = 1

else:

matrix[3][2] = 0

if 248 > position and position > 203:

matrix[3][1] = 1

else:

matrix[3][1] = 0

if 293 > position and position > 247:

matrix[2][1] = 1

else:

matrix[2][1] = 0

if 339 > position and position > 292:

matrix[1][1] = 1

else:

matrix[1][1] = 0

def check\_for\_unlock():

if matrix[4][1] == 1 and matrix[3][2] == 1:

if timer.now() > 2:

return True

else:

timer.reset()

return False

def light\_matrix\_open\_lock():

matrix = [[0,1,1,1,0],

[0,1,0,1,0],

[0,0,0,1,0],

[1,1,1,1,1],

[1,1,1,1,1]]

show\_matrix(matrix)

wait\_for\_seconds(3)

def light\_matrix\_closed\_lock():

matrix = [[0,1,1,1,0],

[0,1,0,1,0],

[0,1,0,1,0],

[1,1,1,1,1],

[1,1,1,1,1]]

show\_matrix(matrix)

wait\_for\_seconds(3)

def open\_chest():

motorLock.run\_to\_position(0)

light\_matrix\_open\_lock()

def close\_motors():

motor1.set\_default\_speed(80)

motor2.set\_default\_speed(80)

motor1.run\_to\_position(random.randint(0, 359))

motor2.run\_to\_position(random.randint(0, 359))

motor1.set\_default\_speed(-80)

motor2.set\_default\_speed(-80)

motor1.run\_to\_position(random.randint(0, 359))

motor2.run\_to\_position(random.randint(0, 359))

motor1.set\_default\_speed(80)

motor2.set\_default\_speed(80)

motor1.run\_to\_position(random.randint(0, 359))

motor2.run\_to\_position(random.randint(0, 359))

def close\_chest():

light\_matrix\_closed\_lock()

show\_matrix(matrix)

motorLock.run\_to\_position(270)

close\_motors()

while True:

motor1\_position()

motor2\_position()

show\_matrix(matrix)

if check\_for\_unlock():

break

open\_chest()

while True:

close\_chest()

hub.left\_button.wait\_until\_pressed()