Wiktor Nosarzewski
podane numery odnoszą się do numerów, pod którymi zapisałem w hubie poniższe programy.

-------------------------------MELODIA TEST nr.0

```
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

hub = PrimeHub()
app = App()

while True:
    hub.speaker.beep(60, 0.5)
    hub.speaker.beep(67, 0.5)
    hub.speaker.beep(60, 0.5)
```

------------------------------HELP! nr.1

```
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

hub = PrimeHub()
app = App()
color_sensor = ColorSensor('B')
# This is story #1: Kiki is going for a walk. She's outside and
happy, until...
while True:
    hub.light_matrix.write('Help!')
    hub.left_button.wait_until_pressed()
    color_sensor.wait_until_color('blue')
    app.play_sound('Traffic')
    color_sensor.wait_until_color('yellow')
    app.play_sound('Ring Tone')
    color_sensor.wait_until_color('green')
    app.play_sound('Dog Bark 1')
    app.play_sound('Dog Bark 1')
    #This is story #2.
    hub.right_button.wait_until_pressed()
    color_sensor.wait_until_color('blue')
    app.play_sound('Door Knock')
    color_sensor.wait_until_color('yellow')
    app.play_sound('Glass Breaking')
    color_sensor.wait_until_color('green')
    app.play_sound('Dog Bark 3')
```

------------------------------Hopper Race nr.2

```
from spike import PrimeHub, MotorPair
from spike.control import wait_for_seconds

hub = PrimeHub()
hopper_motors = MotorPair('E', 'F')

hopper_motors.set_default_speed(50)
hub.light_matrix.write('Hopper Race')
hub.light_matrix.write('3')
wait_for_seconds(1)
hub.light_matrix.write('2')
wait_for_seconds(1)
hub.light_matrix.write('1')
wait_for_seconds(1)
hub.light_matrix.show_image('CHESSBOARD')
# Adjust this to change the distance your Hopper will move.
# --------------v
hopper_motors.move(10, 'seconds')
```

--------------------------------Super Cleanup nr.3

```
from spike import ForceSensor, Motor

force_sensor = ForceSensor('E')
grabber_motor = Motor('A')

hub.light_matrix.write('Super Cleanup')

while True:
    force_sensor.wait_until_pressed()
    grabber_motor.set_stall_detection(False)
    grabber_motor.start(-75)
    force_sensor.wait_until_released()
    grabber_motor.set_stall_detection(True)
    grabber_motor.start(75)
```

-----------------------------Broken nr.4

```
from spike import PrimeHub, Motor
from spike.control import wait_for_seconds

hub = PrimeHub()
x_motor = Motor('A')
y_motor = Motor('C')

hub.light_matrix.write('Broken')

hub.left_button.wait_until_pressed()

x_motor.set_default_speed(-100)
x_motor.run_for_seconds(1.5)
wait_for_seconds(1)

# These 4 blocks should 'cut' a square.

x_motor.set_default_speed(100)
y_motor.set_default_speed(100)
x_motor.run_for_degrees(400)
y_motor.run_for_degrees(575)
x_motor.run_for_degrees(-400)
y_motor.run_for_degrees(-575)
hub.right_button.wait_until_pressed()
x_motor.set_default_speed(100)
x_motor.run_for_seconds(1.5)
wait_for_seconds(1)

# These 4 blocks should 'cut' a rectangle.

x_motor.run_for_degrees(-60)
x_motor.run_for_degrees(-400)
y_motor.run_for_degrees(-800)
x_motor.run_for_degrees(400)
y_motor.run_for_degrees(800)
```

--Design for Someone nr.5

```
from spike import PrimeHub, Motor, ForceSensor
from spike.control import wait_for_seconds

hub = PrimeHub()

motor_a = Motor('A')
motor_e = Motor('E')

force_sensor = ForceSensor('B')


motor_a.set_default_speed(100)
motor_e.set_default_speed(-100)
motor_a.set_stall_detection(False)
motor_e.set_stall_detection(False)
motor_a.set_stop_action('hold')
motor_e.set_stop_action('hold')
motor_a.run_to_position(0)
hub.speaker.beep(60)
hub.speaker.beep(72)
hub.light_matrix.write('Design for Someone')
# make the prothesis grab onto someones arm
motor_a.run_for_seconds(1)
motor_e.run_for_seconds(1)
while True:
    if hub.right_button.was_pressed():
        # make the prothesis let go
        motor_a.run_to_position(0)
        motor_e.run_to_position(0)
        break

    if force_sensor.get_force_newton() > 5:
        hub.light_matrix.show_image('SQUARE')
    else:
        hub.light_matrix.off()

    wait_for_seconds(0.01)
```

--koniec części pierwszej---

pozostałe części poniżej

-------------------------Place your Order [6]

```
from spike import PrimeHub, App, ColorSensor, DistanceSensor, Motor
from spike.control import wait_for_seconds

hub = PrimeHub()
app = App()

distance_sensor = DistanceSensor('C')
color_sensor = ColorSensor('D')
arm_motor = Motor('A')
base_motor = Motor('F')
arm_motor.set_default_speed(50)
base_motor.set_default_speed(50)
arm_motor.run_to_position(350)
base_motor.run_to_position(350)
app.start_sound('Connect')
distance_sensor.light_up_all()
for x in range(10):
    hub.light_matrix.show_image('HEART')
    wait_for_seconds(0.5)
    hub.light_matrix.show_image('HEART_SMALL')
    wait_for_seconds(0.5)
hub.light_matrix.show_image('HEART')

while True:
    color_sensor.wait_until_color('violet')
    arm_motor.run_for_degrees(30)
    arm_motor.run_for_degrees(-60)
    arm_motor.run_for_degrees(60)
    arm_motor.run_for_degrees(-30)
    app.start_sound('Connect')
    hub.light_matrix.show_image('HEART')
```

------------------Out of Order [7]

```
from spike import PrimeHub, DistanceSensor, Motor, MotorPair
from spike.control import wait_for_seconds

hub = PrimeHub()
```

```
distance_sensor = DistanceSensor('B')
drive_motors = MotorPair('A', 'E')
small_wheel_motor = Motor('C')
small_wheel_motor.set_default_speed(100)
drive_motors.set_default_speed(50)

hub.left_button.wait_until_pressed()
#This is one way of debugging the first program.
small_wheel_motor.run_to_position(0)
drive_motors.start()
# adjust the value here ----------------------v
distance_sensor.wait_for_distance_closer_than(15,
DistanceSensor.CM)
drive_motors.stop()hub.right_button.wait_until_pressed()
#This is one way of debugging the second program.
small_wheel_motor.run_to_position(0)
drive_motors.start()
# adjust the value here ----------------------v
distance_sensor.wait_for_distance_closer_than(15,
DistanceSensor.CM)
drive_motors.stop()
# adjust the value here ----------v
small_wheel_motor.run_to_position(20)
wait_for_seconds(1)
drive_motors.move(-50, DistanceSensor.CM)
drive_motors.stop()
small_wheel_motor.run_to_position(0)
wait_for_seconds(1)
# adjust the value here
# ----------------v
drive_motors.move(50, DistanceSensor.CM)
drive_motors.stop()
```

---------------------------Track your Packages [8]

```
from spike import PrimeHub, Motor
from spike.control import wait_for_seconds

hub = PrimeHub()
horizontal_motor = Motor('A')
vertical_motor = Motor('C')
horizontal_motor.set_default_speed(75)
```

```
vertical_motor.set_default_speed(75)
# This program will track your package on map #1
hub.left_button.wait_until_pressed()
horizontal_motor.run_for_seconds(1)
wait_for_seconds(1)
vertical_motor.run_for_degrees(475)
horizontal_motor.run_for_degrees(-545)
vertical_motor.run_for_degrees(950)
horizontal_motor.run_for_degrees(550)
vertical_motor.run_for_degrees(380)
# run both motors at same time to move diagonally
vertical_motor.start(speed=75)
horizontal_motor.run_for_degrees(-540, speed=50)
vertical_motor.stop()
vertical_motor.run_for_degrees(175)
```

--------------------------------Keep it safe [9]

```
from spike import PrimeHub, Motor, LightMatrix
from spike.control import wait_for_seconds, wait_until
from spike.operator import greater_than

hub = PrimeHub()
lock_motor = Motor('C')
dial_motor = Motor('B')
lock_motor.set_default_speed(50)
hub.speaker.beep(60)
hub.speaker.beep(72)

# This locks the door.
dial_motor.set_stop_action('coast')
dial_motor.run_to_position(0)
dial_motor.set_degrees_counted(0)
hub.light_matrix.show_image('NO')

# This unlocks the door when the Left Button on the Hub is pressed.
hub.left_button.wait_until_pressed()
hub.speaker.beep(72)
wait_until(dial_motor.get_degrees_counted, greater_than, 180)
hub.speaker.beep(60)
lock_motor.run_for_seconds(1)
```

```
hub.light_matrix.show_image('NO')
wait_for_seconds(2)
hub.light_matrix.show_image('YES')
wait_for_seconds(5)
```

----------------------------Keep it really safe! [10]

```
from spike import PrimeHub, App, Motor
from spike.control import Timer, wait_for_seconds

hub = PrimeHub()
app = App()

dial = Motor('B')
lock = Motor('C')
dial_cover = Motor('E')

timer = Timer()
dial.set_default_speed(75)
lock.set_default_speed(75)
dial_cover.set_default_speed(75)

def unlock():

while not hub.left_button.is_pressed() and
dial.get_degrees_counted() < 180:
    hub.speaker.beep(60)
    dial_cover.run_for_degrees(15)
    wait_for_seconds(0.8)
    if timer.now() > 5:
        app.play_sound('Bonk')
        return
    hub.light_matrix.show_image('NO')
    wait_for_seconds(2)
    hub.light_matrix.show_image('YES')
    dial_cover.run_to_position(0)
    lock.run_for_seconds(1)
    app.play_sound('Wand')
    wait_for_seconds(5)

# This locks the door and starts the extra protection mechanism.
```

```
hub.speaker.beep(60)
hub.speaker.beep(72)
lock.run_for_seconds(-1)
dial.run_to_position(0)
dial_cover.run_to_position(0)
dial.set_degrees_counted(0)
dial.set_stop_action('coast')
hub.light_matrix.show_image('NO')
timer.reset()
unlock()
```

--------------------------------Automate it! [11]

```
from spike import App, Motor, ColorSensor
from spike.control import wait_for_seconds

app = App()
base_motor = Motor('A')
arm_motor = Motor('F')
color_sensor = ColorSensor('D')
base_motor.set_default_speed(25)
arm_motor.set_default_speed(25)

def check_color():
 #This will check the color of the package.
    arm_motor.run_to_position(235)
    wait_for_seconds(4)
    if color_sensor.get_color() == 'violet':
        base_motor.run_to_position(0)
        arm_motor.run_to_position(25)
        app.play_sound('Triumph')
        arm_motor.run_to_position(240)
    else:
        app.play_sound('Oops')
        arm_motor.run_to_position(25)
        for x in range(3):
            arm_motor.run_for_degrees(-100, speed=100)
            arm_motor.run_for_degrees(100, speed=100)

# This powers up the robot and makes it grab one package from
```

```
each side.
base_motor.run_to_position(0)
arm_motor.run_to_position(240)
base_motor.run_to_position(90)
arm_motor.run_to_position(25)
check_color()
base_motor.run_to_position(0)
arm_motor.run_to_position(240)
base_motor.run_to_position(270)
arm_motor.run_to_position(25)
check_color()
base_motor.run_to_position(0)
arm_motor.run_to_position(240)
```

---Break Dance [12]

```
from spike import PrimeHub, Motor, ColorSensor
from spike.control import wait_for_seconds

hub = PrimeHub()
leg_motor = Motor('F')
arm_motor = Motor('B')
color_sensor = ColorSensor('D')
leg_motor.set_default_speed(-80)
arm_motor.set_default_speed(-80)
leg_motor.run_to_position(0)
arm_motor.run_to_position(0)
wait_for_seconds(1)
for x in range(10):
    hub.light_matrix.write("1")
    leg_motor.start()
    arm_motor.run_for_rotations(1)
    leg_motor.stop()
    wait_for_seconds(0.45)
    hub.light_matrix.write("2")
    leg_motor.start()
    arm_motor.run_for_rotations(1)
    leg_motor.stop()
    wait_for_seconds(0.45)
    hub.light_matrix.write("3")
    leg_motor.start()
    arm_motor.run_for_rotations(1)
    leg_motor.stop()
    wait_for_seconds(0.45)
```

---Repeat 5 Times [13]

```
from spike import PrimeHub, App, Motor
from spike.control import wait_until, wait_for_seconds
from spike.operator import equal_to

hub = PrimeHub()
app = App()

left_leg_motor = Motor('B')
right_leg_motor = Motor('F')
left_leg_motor.set_default_speed(50)
right_leg_motor.set_default_speed(-50)
left_leg_motor.start()
right_leg_motor.start()
wait_until(hub.motion_sensor.get_orientation, equal_to, 'leftside')
right_leg_motor.stop()
left_leg_motor.stop()
app.play_sound('Sport Whistle 1')
for count in range(5):
    left_leg_motor.set_default_speed(-50)
    right_leg_motor.set_default_speed(50)
    left_leg_motor.start()
    right_leg_motor.start()
    wait_until(hub.motion_sensor.get_orientation, equal_to, 'front')
    right_leg_motor.stop()
    left_leg_motor.stop()
    app.start_sound('Male Jump 1')
    hub.light_matrix.write(count + 1)
    wait_for_seconds(0.5)
    left_leg_motor.set_default_speed(50)
    right_leg_motor.set_default_speed(-50)
    left_leg_motor.start()
    right_leg_motor.start()
    wait_until(hub.motion_sensor.get_orientation, equal_to,
'leftside')
    right_leg_motor.stop()
    left_leg_motor.stop()
    wait_for_seconds(0.5)

app.play_sound('Sport Whistle 2')
```

---Rain or shine? [14]

```
from spike import PrimeHub, App, Motor
from spike.control import wait_for_seconds

hub = PrimeHub()
app = App()
umbrella_motor = Motor("B")
glasses_motor = Motor("F")

#adjust weather here: (sunny or rainy)
YOUR_LOCAL_FORECAST = "sunny"

umbrella_motor.set_default_speed(100)
glasses_motor.set_default_speed(100)
# This gets the robot in the correct starting position.
umbrella_motor.run_to_position(45)
glasses_motor.run_to_position(300)
hub.speaker.beep(60, seconds=0.1)
hub.speaker.beep(72, seconds=0.1)

if YOUR_LOCAL_FORECAST == "sunny":
# if sunny, then put on sunglasses
    glasses_motor.run_to_position(0)
    hub.light_matrix.show_image("SQUARE")
    wait_for_seconds(2)
    glasses_motor.run_to_position(300)
elif YOUR_LOCAL_FORECAST == "rainy":
# or if rainy, lift umbrella.
    umbrella_motor.run_to_position(340)
    app.play_sound("Rain")
    umbrella_motor.run_to_position(45)
else:
# otherwise show this
    hub.light_matrix.show_image("NO")
```

----------------------------Wind Speed [15]

```
from spike import App, Motor
from spike.control import wait_for_seconds

tilt_motor = Motor("A")

WIND_SPEED_FORECAST = 8

tilt_motor.set_default_speed(20)
tilt_motor.run_to_position(5)

if WIND_SPEED_FORECAST < 5.5:
    tilt_motor.run_for_degrees(30)
    wait_for_seconds(1)
    tilt_motor.run_for_degrees(-30)
else:
    tilt_motor.run_for_degrees(60)
    wait_for_seconds(1)
    tilt_motor.run_for_degrees(-60)
```

------------------------------------Veggie Love [16]

```
from spike import PrimeHub, App, Motor

hub = PrimeHub()
app = App()

pointer_motor = Motor("E")
pointer_motor.set_default_speed(-50)

WEEK_RAIN = 50
ROTATION = 0

hub.left_button.wait_until_pressed()
pointer_motor.run_for_seconds(2)
pointer_motor.set_degrees_counted(0)
pointer_motor.set_default_speed(50)
pointer_motor.run_for_seconds(2)
hub.light_matrix.write(abs(pointer_motor.get_degrees_counted()))
rotation = int(week_rain *
abs(pointer_motor.get_degrees_counted()) / 60)
print(ROTATION)
```

```
hub.right_button.wait_until_pressed()
pointer_motor.set_degrees_counted(0)
pointer_motor.set_default_speed(-50)
pointer_motor.run_for_degrees(ROTATION)
hub.light_matrix.write(WEEK_RAIN)
print(WEEK_RAIN)
```

--------------------------------Brain Game [17]

```
from spike import PrimeHub, App, Motor, ColorSensor
from spike.control import wait_for_seconds

hub = PrimeHub()
app = App()

mouth_motor = Motor('A')
color_sensor = ColorSensor('B')

candy1 = []
candy2 = []

while True:
    hub.left_button.wait_until_pressed()
# This makes the Game Master eat the candy stick then read and
record its sequence of colors in the list called "candy1".
    hub.light_matrix.off()
    candy1.clear()
    mouth_motor.set_default_speed(-50)
    mouth_motor.run_for_seconds(2)
    app.play_sound('Bite')
    app.play_sound('Bite')
    for x in range(5):
        candy1.append(color_sensor.get_color())
        wait_for_seconds(1)
        mouth_motor.set_default_speed(50)
        mouth_motor.run_for_degrees(95)
        wait_for_seconds(1)
    hub.right_button.wait_until_pressed()
 # This makes the Game Master eat the candy stick then read and
record its sequence of colors in the list called "candy2".
    candy2.clear()
    mouth_motor.set_default_speed(-50)
    mouth_motor.run_for_seconds(2)
    app.play_sound('Bite')
    app.play_sound('Bite')

    for x in range(5):
        candy2.append(color_sensor.get_color())
        wait_for_seconds(1)
        mouth_motor.set_default_speed(50)
        mouth_motor.run_for_degrees(95)
        wait_for_seconds(1)
    # Light up the position of the red bricks if it is in the same
position in both of the candy sticks.
    candy1_red_index = candy1.index('red')
    candy2_red_index = candy2.index('red')
    for x in range(5):
        print(candy1[x])
    if candy1_red_index == candy2_red_index:
        for x in range(5):
            hub.light_matrix.set_pixel(x, candy1_red_index)
        app.play_sound('Win')
    else:
        app.play_sound('Oops')
```

-----------------------------------The Coach [18]

```
from spike import Motor
from spike.control import Timer, wait_for_seconds

left_leg_motor = Motor('F')
right_leg_motor = Motor('B')

timer = Timer()

left_leg_motor.run_to_position(0)
right_leg_motor.run_to_position(0)

while True:
    while timer.now() < 5:
        left_leg_motor.start_at_power(-80)
        right_leg_motor.start_at_power(80)
        wait_for_seconds(0.1)
        left_leg_motor.start_at_power(80)
        right_leg_motor.start_at_power(-80)
        wait_for_seconds(0.1)
```

---KONIEC CZĘŚCI TRZECIEJ

by Wiktor Nosarzewski