2021

**Transport and Telecommunication Institute**



**Программирование функций-инвентов NAO   
с помощью модуля eventmanager на Python.**

Николай Котцов

[**1.** **Задание** 2](#_Toc73372980)

[**2.** **Реализация** 2](#_Toc73372981)

[ Основной блок программы 2](#_Toc73372982)

[o Блок подключения библиотек \ модулей 2](#_Toc73372983)

[o Блок стартовых настроек NAO перед работой основной части 2](#_Toc73372984)

[o Блок объявления обработчика ивентов и добавления их в обработчик 2](#_Toc73372985)

[o Запуск обработчика и ожидание окончания его работы 2](#_Toc73372986)

[ Дополнительный блок 2](#_Toc73372987)

[o Блок объявления функций \ ивентов 2](#_Toc73372988)

[o Блок объявления функций – предикатов, которые возвращают True \ False 2](#_Toc73372989)

[ Блок запуска программы 2](#_Toc73372990)

[o Запуск с заданными аргументами(IP, PORT) 2](#_Toc73372991)

[**3.** **Дополнительно** 3](#_Toc73372992)

1. **Задание**

Создать программу, которая позволяет гулять с NAO. А именно, при наклоне руки прямо, он идет прямо. При наклоне в сторону, он поворачивает. После чего загрузить программу в самого NAO и добавить возможность вывоза при помощи голоса.

1. **Реализация**

Для реализации данного задания был разработан модуль eventmanager, подробнее о нем можно почитать в документации о нем. Данный модуль позволяет упростить работу с потоками, упростить обработку и вызов ивентов.

Код проекта прогулки с NAO, является наглядным примером использования данного модуля. В будущих проектах функционал самого модуля может быть дополнен или обновлен. В таком случае будет обновляться документация по модулю eventmanager. Также всю информацию по установке, также можно будет найти там же.

**По коду программы:**

* Основной блок программы
  + Блок подключения библиотек \ модулей
    - Строки 3 – 12
  + Блок стартовых настроек NAO перед работой основной части
    - Строки 18 – 54
  + Блок объявления обработчика ивентов и добавления их в обработчик
    - Строки 59 – 108
  + Запуск обработчика и ожидание окончания его работы
    - Строки 111 – 114
* Дополнительный блок
  + Блок объявления функций \ ивентов
    - Строки 118 – 134
  + Блок объявления функций – предикатов, которые возвращают True \ False
    - Строки 137 – 168
* Блок запуска программы
  + Запуск с заданными аргументами(IP, PORT)
    - Строка 170

1. **Дополнительно**

**Код программы**

# -\*- encoding: UTF-8 -\*-

**#3**

import time

import threading

import random

from threading import Thread

from eventmanager import \*

from naoqi import ALProxy

from naoqi import ALBroker

from naoqi import ALModule

**#12**

def eventRuka(robot\_IP, robot\_PORT=9559):

**#18**

    global tts, memory, motion, alife, touch, posture

    tts = ALProxy("ALTextToSpeech", robot\_IP, robot\_PORT)

    memory = ALProxy("ALMemory",robot\_IP,robot\_PORT)

    motion = ALProxy("ALMotion", robot\_IP, robot\_PORT)

    posture = ALProxy("ALRobotPosture", robot\_IP, robot\_PORT)

    alife = ALProxy("ALAutonomousLife", robot\_IP, robot\_PORT)

    touch = ALProxy("ALTouch", robot\_IP, robot\_PORT)

    global phrases      #   фразы при движении

    phrases = {

        "general": ["what a great day", "and you are good at copying me human", "good", "I'm not just some kind of robot as you might think, I'm something more"],

        "general\_move": ["ok let's go here", "I'm tired, let's stand", "lead me", "you walk well, for human of course. ha ha ha" , "not tired yet, human?", "lead me flesh bag", "lead me hairless ape", "don't hurt my hand"],

        "forward": ["wait i'm tired", "ok let's go forward", "text"],

        "backward": ["text", "ok let's go backward", "text"],

        "turn\_left": ["I got dizzy", "ok let's go left", "text"],

        "turn\_right": ["I got dizzy", "ok let's go right", "text"],

        "forward\_left": ["Is it just me or are we walking in circles?", "are we walking in circles?", "I'm having fun with you, but we go in circles", "Port!"],

        "forward\_right": ["Is it just me or are we walking in circles?", "are we walking in circles?", "I'm having fun with you, but we go in circles", "Starboard!"],

        "backward\_left": ["Is it just me or are we walking in circles?", "are we walking in circles?", "I'm having fun with you, but we go in circles"],

        "backward\_right": ["Is it just me or are we walking in circles?", "are we walking in circles?", "I'm having fun with you, but we go in circles"],

        "stop": ["why are we standing?", "can we go already?", "tired of just standing still", "come on, take my hand and go for a walk", "and we stand again", "sounds of falling asleep NAO"]

    }

    alife.setState("disabled")

    motion.stopMove()

    motion.wakeUp()

    motion.setMoveArmsEnabled(True, False) #(left,right)

    tts.post.say("Take my hand and let's go for a walk")

    motion.post.setAngles("RShoulderPitch", -1.5, 0.4)

    motion.post.setAngles("RShoulderRoll", 0, 0.4)

    motion.openHand("RHand")

    global loop

    loop = True

**#54**

**#59**

    global eventHandler

    #   Создание экземпляра класса Eventloop

    eventHandler = Eventloop()

    #   eventHandler.addGroup(EventGroup)

    #   eventHandler.addEvent(Event(lambda: Предикат(проверка), Функция, Аргументы, single\_use=False, threadable=True, cooldown=0))

    #   Создание групп ивентов

    generalGroup = EventGroup()

    motionGroup = EventGroup()

    ttsGroup = EventGroup(cooldown = 5)

    #   generalGroup.addEvent(Event(lambda: Предикат(проверка), Функция, Аргументы, single\_use=False, threadable=True, cooldown=0))

    #   Создание и добавление ивентов на обработку (по стандарту внутри уже создана группа для ивентов, не требующих определенную группу)

    eventHandler.addEvent(Event( naoExit, [], binaryPredicate(lambda: touch.getStatus()[8][1], True, False), single\_use=True))

    eventHandler.addEvent(Event( touchMyHand, [], binaryPredicate(lambda: touch.getStatus()[18][1], True, False), single\_use=True, threadable=True))

    #   Создание и обьединение ивентов в группы

    motionGroup.addEvent(Event( motion.moveTo, [10,0,0], getForward , threadable=True))

    ttsGroup.addEvent(Event( walkPhrases, ["forward"], getForward , threadable=True, cooldown=10))

    motionGroup.addEvent(Event( motion.moveTo, [-10,0,0], getBackward, threadable=True))

    ttsGroup.addEvent(Event( walkPhrases, ["backward"], getBackward , threadable=True, cooldown=10))

    motionGroup.addEvent(Event( motion.moveTo, [0,0,0.78], getLeft, threadable=True))

    ttsGroup.addEvent(Event( walkPhrases, ["turn\_left"], getLeft , threadable=True, cooldown=10))

    motionGroup.addEvent(Event( motion.moveTo, [0,0,-0.78], getRight, threadable=True))

    ttsGroup.addEvent(Event( walkPhrases, ["turn\_right"], getRight , threadable=True, cooldown=10))

    motionGroup.addEvent(Event( motion.moveToward, [1,0,0.5,[["Frequency", 0.1]]], getForwardLeft, threadable=True))

    ttsGroup.addEvent(Event( walkPhrases, ["forward\_left"], getForwardLeft , threadable=True, cooldown=10))

    motionGroup.addEvent(Event( motion.moveToward, [1,0,-0.5,[["Frequency", 0.1]]], getForwardRight, threadable=True))

    ttsGroup.addEvent(Event( walkPhrases, ["forward\_right"], getForwardRight , threadable=True, cooldown=10))

    motionGroup.addEvent(Event( motion.moveToward, [-1,0,-0.5,[["Frequency", 0.1]]], getBackwardLeft, threadable=True))

    ttsGroup.addEvent(Event( walkPhrases, ["backward\_left"], getBackwardLeft , threadable=True, cooldown=10))

    motionGroup.addEvent(Event( motion.moveToward, [-1,0,0.5,[["Frequency", 0.1]]], getBackwardright, threadable=True))

    ttsGroup.addEvent(Event( walkPhrases, ["backward\_right"], getBackwardright , threadable=True, cooldown=10))

    motionGroup.addEvent(Event( motion.stopMove, [], getStopMove))

    ttsGroup.addEvent(Event( walkPhrases, ["stop"], getStopMove , threadable=True, cooldown=20))

    #   Добавление групп в обработчик ивентов

    eventHandler.addEventGroup(motionGroup)

    eventHandler.addEventGroup(ttsGroup)

**#108**

**#111**

    #   Запуск обработчика ивентов

    eventHandler.start()

    #   Ожидание обработчика ивентов

    eventHandler.join()

    alife.setState("interactive")

**#114**

**#118**

def walkPhrases(state):

    tts.say(random.choice(phrases["general"] + phrases[state] + [i for i in phrases["general\_move"] if state != "stop"]))

def touchMyHand():

    motion.post.stiffnessInterpolation("RShoulderPitch",0.0,1.0)

    motion.post.stiffnessInterpolation("RShoulderRoll",0.0,1.0)

    motion.post.stiffnessInterpolation("RWristYaw",0.0,1.0)

    motion.closeHand("RHand")

    tts.say("ok, lead me")

def naoExit():

    global loop

    eventHandler.stop()

    loop=False

    motion.stopMove()

    posture.goToPosture("Stand",0.4)

**#134**

**#137**

def getRShoulderPitch():

    return motion.getAngles("RShoulderPitch",False)[0]

def getRShoulderRoll():

    return motion.getAngles("RShoulderRoll",False)[0]

def getForward():

    return -0.85 < getRShoulderPitch() and (-0.2 <= getRShoulderRoll() <= 0.1) and not getRShoulderPitch()>0.2 and loop

def getForwardLeft():

    return -0.85 < getRShoulderPitch() and 0.1 < getRShoulderRoll()  and not getRShoulderPitch()>0.2 and loop

def getForwardRight():

    return -0.85 < getRShoulderPitch() and getRShoulderRoll() < -0.2  and not getRShoulderPitch()>0.2 and loop

def getBackward():

    return getRShoulderPitch() < -1.75 and (-0.2 <= getRShoulderRoll() <= 0.1) and not getRShoulderPitch()>0.2 and loop

def getBackwardLeft():

    return getRShoulderPitch() < -1.75 and 0.1 < getRShoulderRoll()  and not getRShoulderPitch()>0.2 and loop

def getBackwardright():

    return getRShoulderPitch() < -1.75 and getRShoulderRoll() < -0.2  and not getRShoulderPitch()>0.2 and loop

def getLeft():

    return (-1.75 <= getRShoulderPitch() <= -0.85) and 0.1 < getRShoulderRoll() and not getRShoulderPitch()>0.2 and loop

def getRight():

    return (-1.75 <= getRShoulderPitch() <= -0.85) and getRShoulderRoll() < -0.2 and not getRShoulderPitch()>0.2 and loop

def getStopMove():

    return not(getForward() or getForwardLeft() or getForwardRight() or getBackward() or getBackwardLeft() or getBackwardright() or getLeft() or getRight())

**#168**

**#170**

eventRuka("192.168.252.226", 9559)

# eventRuka("NaoThree.local", 9559)