

让科技便捷生活

Pepper

Terminal Multifunctional Service Robot project document

Yuzheng Hu, Youran Xu, Jiayi Chen

Senior Team

From

High School Affiliated to Zhejiang Normal University

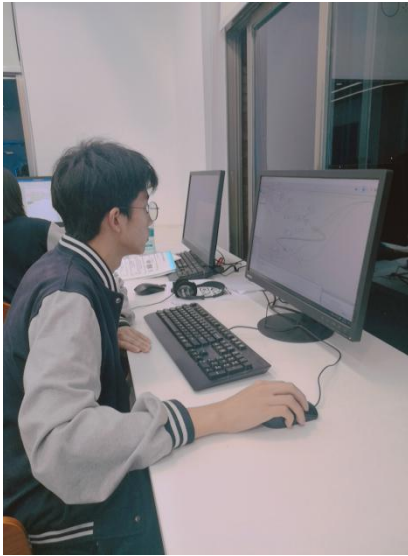


CONTENTS

目录

- 一、 Our Team
- 二、 Project Background
- 三、 Production Introduction
- 四、 Function Introduction
- 五、 Techs
- 六、 Conclusion and Logs

Our EXCELLENT Team



Yuzheng Hu Leader

Program Development:

Luggage Care

Self-help Order

Time Remind

Designer

Function design, Document, Video

Designer

PowerPoint, Flow Chart, Interface

Youran Xu



Jiayi Chen



Background

~Our Role, Tang, who go to Beijing to for university by plane alone



1

He wanted to see the souvenirs in the shop, but he could hardly walk with a heavy schoolbag on his back. Finally, he found the baggage management office in the corner of the terminal.



2

He was relieved at last. After relaxing, he felt hungry. However, where is the restaurant? He has no strength to look for it. I really hope the delicious food can be sent directly to me.



3

When he finished reading a novel with his mobile phone, he suddenly felt excited. For a moment, he didn't see the time. How long will he have to board? Won't he miss the plane.



4

Fortunately, he got on the plane and thought to himself: I really hope the terminal can have better and more comfortable service. It makes my way to the University easier.

To be continued . . .

创作初衷

So, the smart and intelligent research team (kochab) in High School Affiliated to Zhejiang Normal University heard the voice from Tang. We racked our brains to design this terminal service robot.

Production Introduction

- ◆ Luggage Care
- ◆ Self-help Order
- ◆ Time Remind

The terminal pepper intelligent service robot developed by the team is a comprehensive service robot suitable for the terminal. It has the functions of human-computer interaction and image processing. As a new service mode, it makes the service more detailed, deeper and more in line with the psychology and needs of passengers.

Function Introduction



A few years later, Tang's parents went to Beijing to attend Tang's graduation ceremony. They met the pepper intelligent service robot designed by the (kochab) team for the terminal. They found that a long time ago, when Tang took the opportunity to complain to them about the bad experience, they all solved it after meeting pepper.



When the legs and feet are inconvenient, they only need to order their meals by themselves in their seats, and after waiting for a while, they can enjoy the steaming food.



They can get out of heavy luggage.



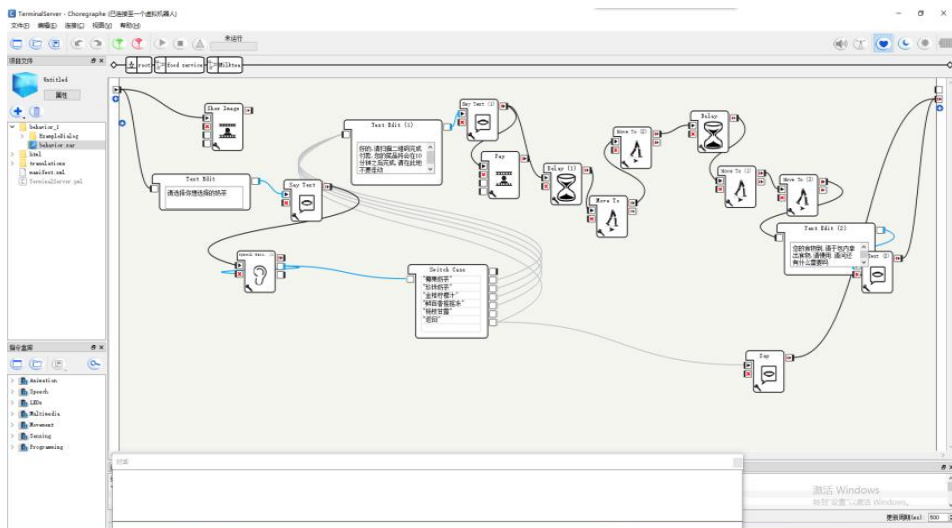
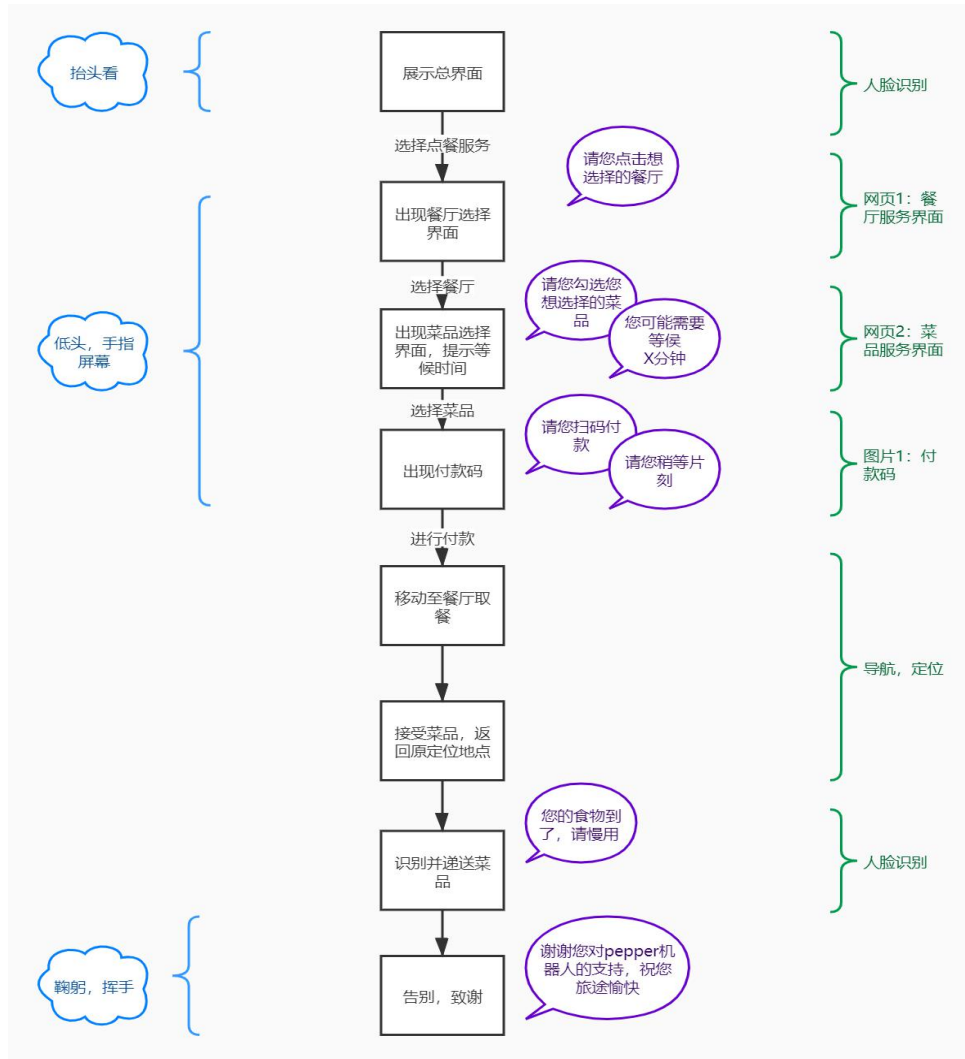
They don't have to worry about missing the boarding time because they can't skillfully use their mobile phones to check the plane information.



可爱的 Pepper
准备着为您服务

1、Self-help Order

Flow Chart



Q: Pepper, I'm so hungry, but I don't want to buy it. Can you order for me here and deliver the food to me!

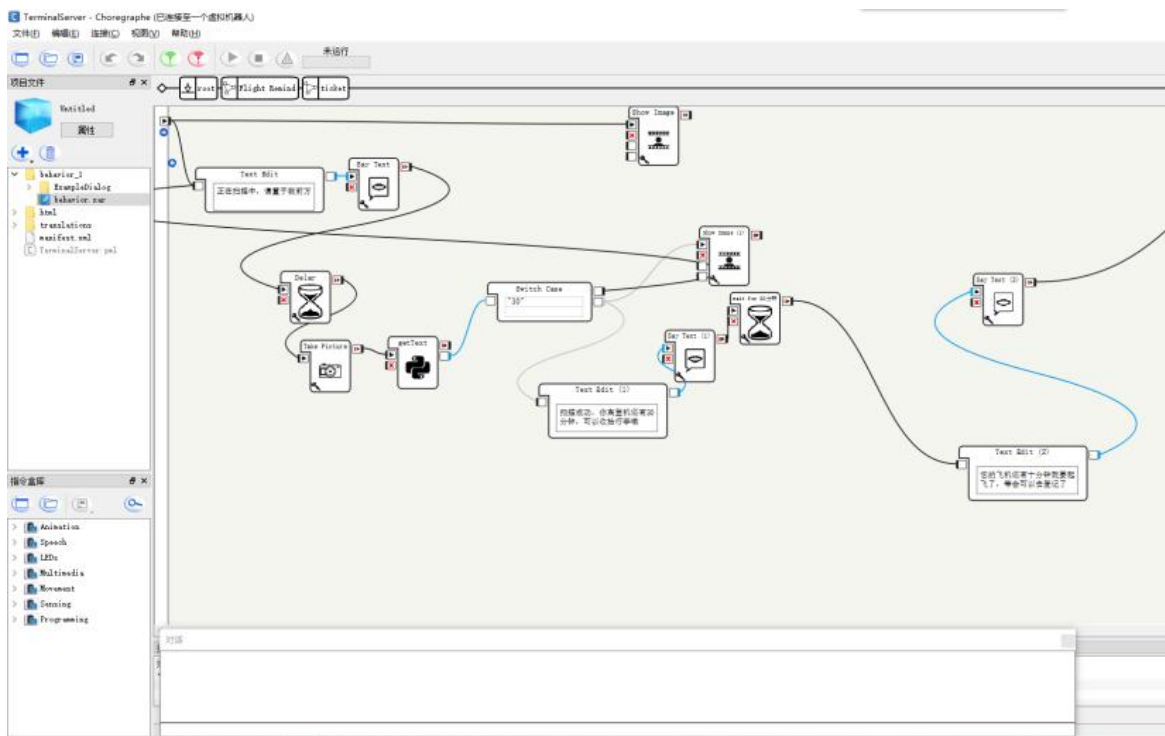
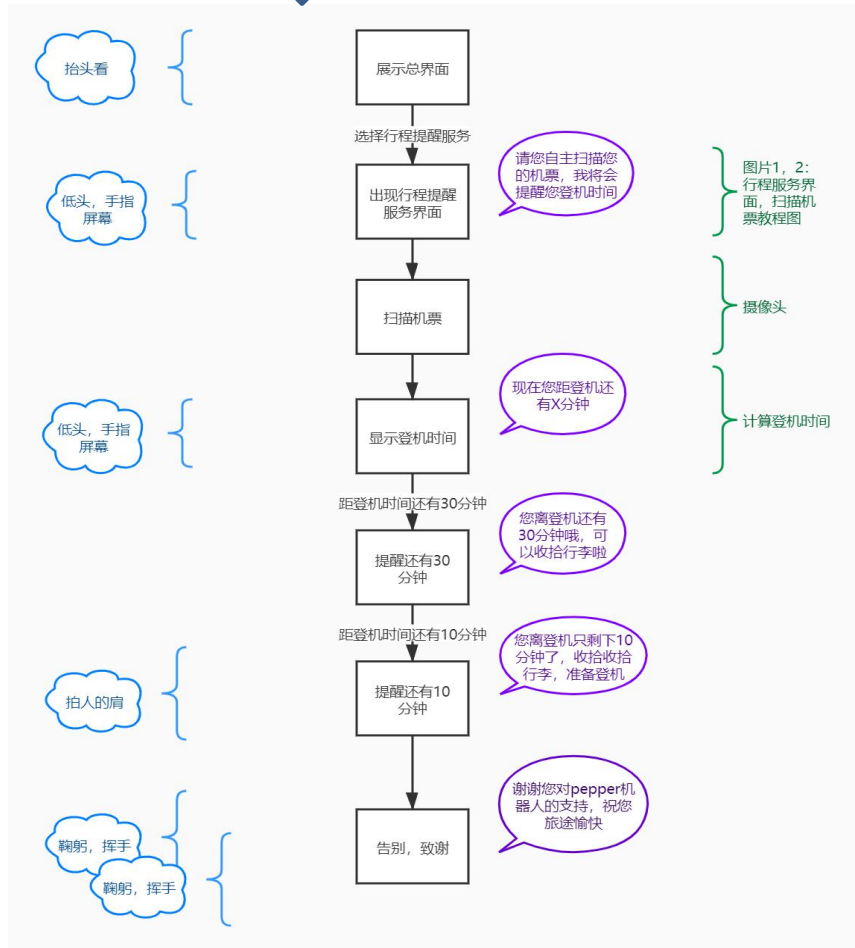
——Of course

Self-help Order

Reduce the total time of dining in the terminal, and solve the problem of mobility inconvenience caused by the large area of the terminal.



2、Time Remind Flow Chart

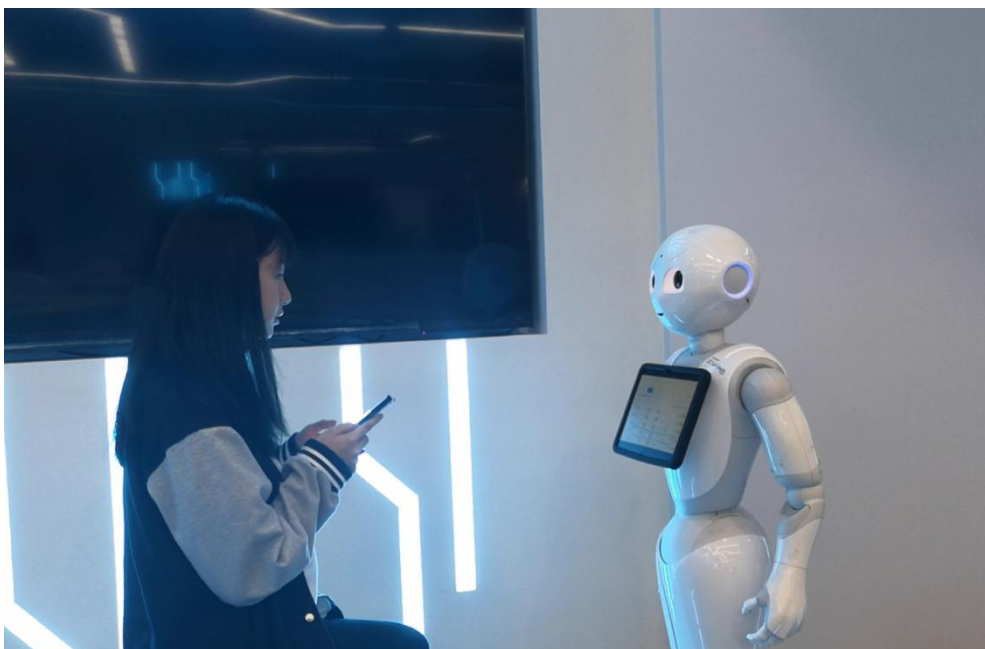


Q: I'm afraid I'll miss the boarding time, pepper. Can you tell me how long it's going to take?

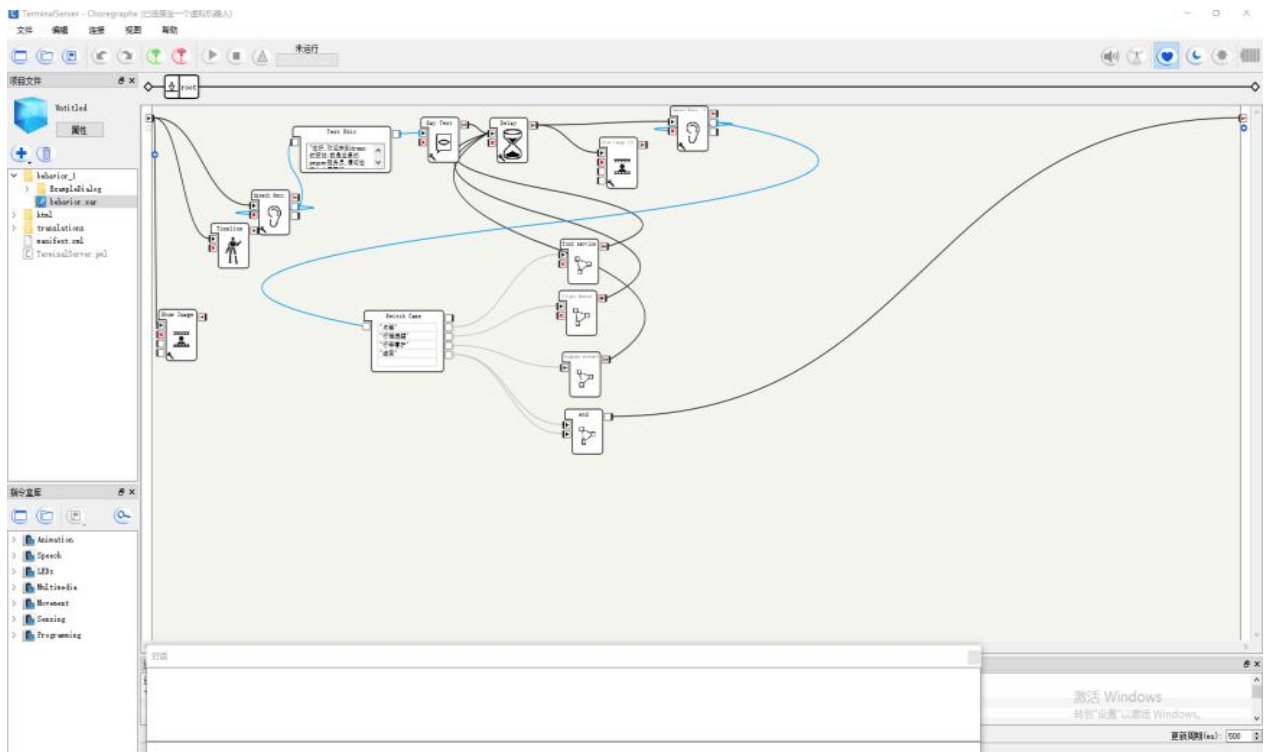
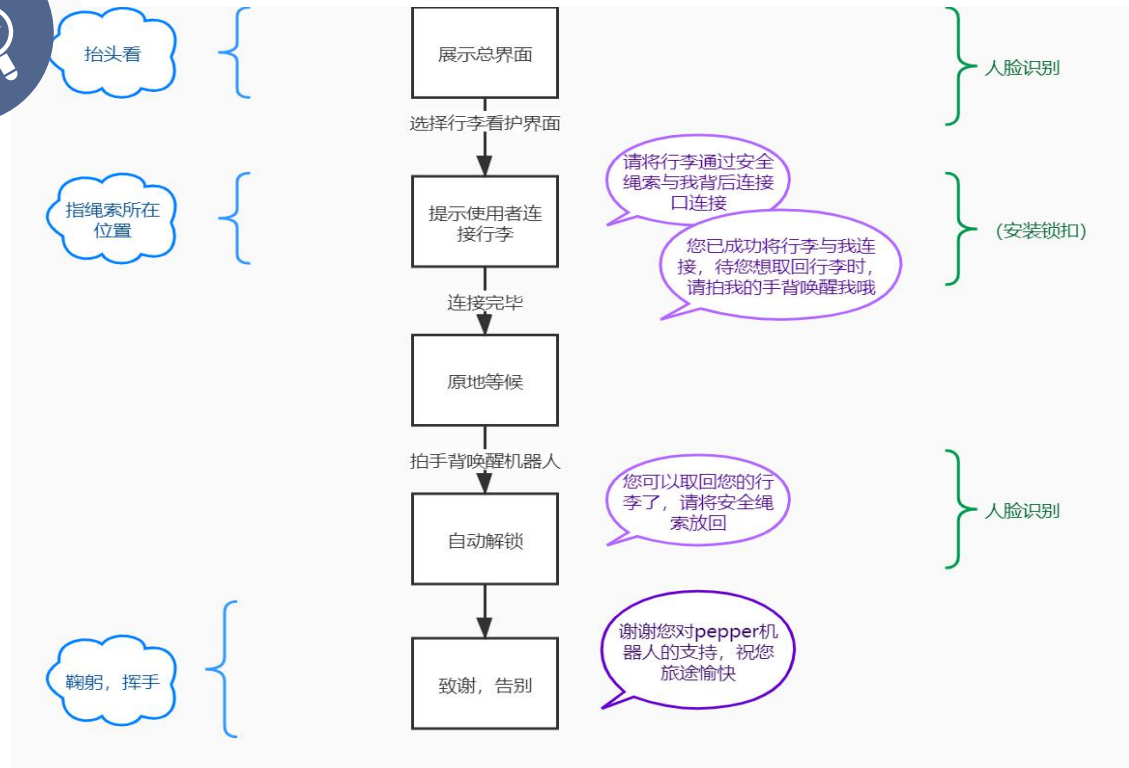
——Of course, I can be with you. I will remind you immediately in 30 minutes or 10 minutes.。

Time Remind

To solve the problem of space and time inconvenience in terminal air ticket inquiry.



3、Luggage Care Flow Chart



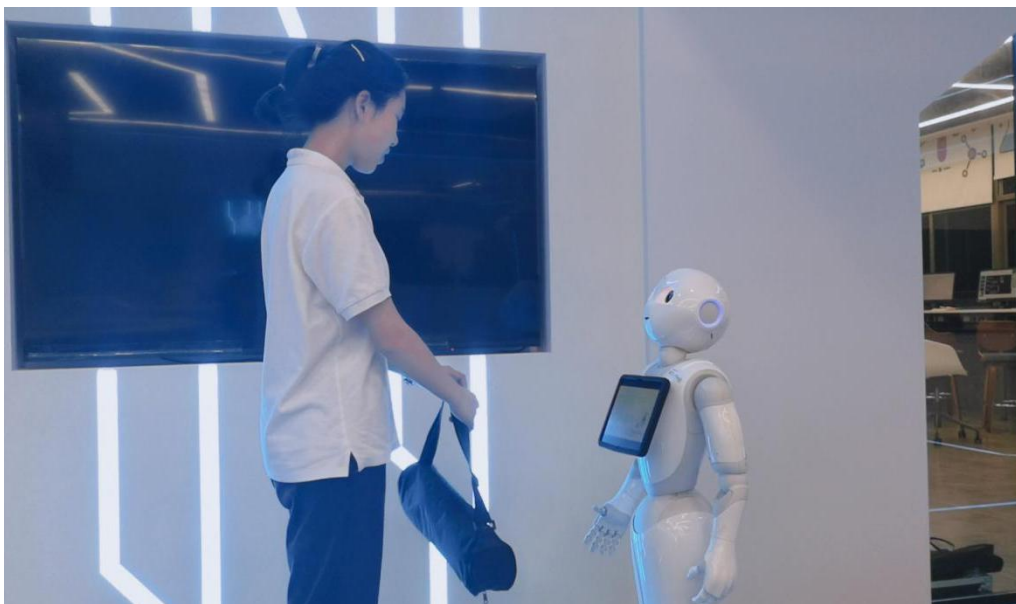
Q: Pepper, I want to go shopping, but my schoolbag makes it hard for me to go shopping happily,

Can you take care of it for me?

——Sure. Can you put your schoolbag in my hand? Then you can go shopping!

Luggage Care

Solve the problem of inconvenient activities in the terminal due to the baggage can't get away.



Ticket scanning + ticket recognition

The QR code is placed in front of the camera, and pepper automatically scans and identifies it through image processing technology, and then the flight information is displayed on the screen.

```

1 import time
2 class MyClass(GeneratedClass):
3     def __init__(self):
4         GeneratedClass.__init__(self, False)
5         self.resolutionMap = {
6             '160 x 120': 0,
7             '320 x 240': 1,
8             '640 x 480': 2,
9             '1280 x 960': 3
10        }
11        self.cameraMap = {
12            'Top': 0,
13            'Bottom': 1
14        }
15        self.recordFolder = "/home/nao/recordings/cameras/"
16
17    def onLoad(self):
18        self.bIsRunning = False
19        try:
20            self.photoCapture = ALProxy("ALPhotoCapture")
21        except Exception as e:
22            self.photoCapture = None
23            self.logger.error(e)
24
25    def onUnload(self):
26        pass
27
28    def onInput_onStart(self):
29        if self.bIsRunning:
30            return
31        self.bIsRunning = True
32        resolution = self.resolutionMap[self.getParameter("Resolution")]
33        cameraID = self.cameraMap[self.getParameter("Camera")]
34        fileName = self.getParameter("File Name")
35        if self.photoCapture:
36            self.photoCapture.setResolution(resolution)
37            self.photoCapture.setCameraID(cameraID)
38            self.photoCapture.setPictureFormat("jpg")
39            self.photoCapture.takePicture(self.recordFolder, fileName)
40        self.bIsRunning = False
41        self.onStopped()

```

```

1 import requests
2 import base64
3 import json
4 from urllib import urlopen
5 from urllib import Request
6 from urllib import URLError
7 from urllib import urlencode
8 import urllib
9 import urllib2
10 class MyClass(GeneratedClass):
11     def __init__(self):
12         GeneratedClass.__init__(self)
13
14     def onLoad(self):
15         #put initialization code here
16         pass
17
18     def onUnload(self):
19         #put clean-up code here
20         pass
21
22     def onInput_onStart(self):
23         #self.onStopped() #activate the output of the box
24         url = "http://bdjgcode.api.bdykx.com/ocr/qrcode"
25         headers = {
26             'Content-Type': 'application/x-www-form-urlencoded',
27             'X-Box-Signature': 'AppCode/363ca3b360a7e50836ad16c47fe8da',
28             'charset': 'unicode'
29         }
30         f = open("/data/home/nao/recordings/cameras/image.jpg", 'rb')
31         img = base64.b64encode(f.read())
32         formData = {
33             "imageType": "0",
34             "image": img
35         }
36         data = urllib.urlencode(formData)
37         req = urllib2.Request(url, data=data, headers=headers)
38         res = urllib2.urlopen(req)
39         result=res.read()
40         topresult=result.find("text":{"")+9
41         bot=result.find("}")
42         result=result[top:bot]
43         self.onGetText(result)
44
45     def onInput_onStop(self):
46         self.onUnload() #it is recommended to reuse the clean-up as the box is stopped
47         self.onStopped() #activate the output of the box

```

Wake up Robot

```

1 class MyClass(GeneratedClass):
2     def __init__(self):
3         GeneratedClass.__init__(self, False)
4
5     def onLoad(self):
6         #- puts code for box initialization here
7         pass
8
9     def onUnload(self):
10        #- puts code for box cleanup here
11        pass
12
13    def onInput_onStart(self, p):
14        if(p > 0):
15            self.onStopped() #- activate output of the box
16
17    def onInput_onStop(self):
18        self.onUnload() #- it is usually a good idea to call onUnload of this box in a onStop
19        method, as the code written in onUnload is used to finish the working of the box as well
20        pass

```

A uniform electric field is applied to the four vertices RT, Rb, LT and lb of the touch screen to make the lower layer (indium oxide) ITO glass covered with a uniform voltage. The upper layer is a receiving signal device. When the pen or finger presses any point on the surface, the controller detects the change of resistance at the finger pressing position, and then changes the coordinates.

Directional movement + obstacle handling

```
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self, False)
        self.motion = AlProxy("AlMotion")
        self.positionErrorThresholdPos = 0.01
        self.positionErrorThresholdAng = 0.03

    def onLoad(self):
        pass

    def onUnload(self):
        self.motion.moveTo(0.0, 0.0, 0.0)

    def onInput_onStart(self):
        import almath
        # The command position estimation will be set to the sensor position
        # when the robot starts moving, so we use sensors first and commands later.
        initPosition = almath.Pose2D(self.motion.getRobotPosition(True))
        targetDistance = almath.Pose2D(self.getParameter("Distance X (m)"),
            self.getParameter("Distance Y (m)"),
            self.getParameter("Theta (deg)") * almath.PI / 180)
        expectedEndPosition = initPosition + targetDistance
        enableArms = self.getParameter("Arms movement enabled")
        self.motion.setMoveArmsEnabled(enableArms, enableArms)
        self.motion.moveTo(self.getParameter("Distance X (m)"),
            self.getParameter("Distance Y (m)"),
            self.getParameter("Theta (deg)") * almath.PI / 180)

        # The move is finished so output
        realEndPosition = almath.Pose2D(self.motion.getRobotPosition(False))
        positionError = realEndPosition.diff(expectedEndPosition)
        positionError.theta = almath.modulo2PI(positionError.theta)
        if (abs(positionError.x) < self.positionErrorThresholdPos
            and abs(positionError.y) < self.positionErrorThresholdPos
            and abs(positionError.theta) < self.positionErrorThresholdAng):
            self.onArrivedAtDestination()
        else:
            self.onStoppedBeforeArriving(positionError.toVector())

    def onInput_onStop(self):
        self.onUnload()
```

The move to command box can control the pepper to move in a straight line on the ground and stop when encountering obstacles.

In parameter setting, the X coordinate, y coordinate, turning angle and moving distance can be set.

Payment code identification

```
import requests
import base64
import json
from urllib2 import urlopen
from urllib2 import Request
from urllib2 import URLError
from urllib import urlencode
import urllib
import urllib2

class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)

    def onLoad(self):
        #put initialization code here
        pass

    def onUnload(self):
        #put clean-up code here
        pass

    def onInput_onStart(self):
        #self.onStopped() #activate the output of the box
        url = "http://bodygroove.api.bodymkt.com/ocr/qrcode"
        headers = {
            'Content-Type': 'application/x-www-form-urlencoded',
            'X-Boxe-Signature': 'AppCode/365ca9b860a74908963a61ec847fe5de',
            'charset': 'unicode'
        }
        f = open('/data/home/nao/recordings/cameras/image.jpg', 'rb')
        img = base64.b64encode(f.read())
        formdata = {
            "imageType": "0",
            "image": img
        }
        data = urllib.urlencode(formdata)
        req = urllib2.Request(url, data=data, headers=headers)
        res = urllib2.urlopen(req)
        result=res.read()
        top=result.find('"text":')
        bot=result.find('"");')
        result=result[top:bot]
        self.onGetText(result)

    def onInput_onStop(self):
        self.onUnload() #it is recommended to reuse the clean-up as the box is stopped
        self.onStopped() #activate the output of the box
```

Through the robot's head infrared scanning of the two-dimensional code in the passenger's mobile phone, and use the collected two-dimensional code image for processing. Finally, the passenger can pay.

With the Pepper Terminal Multifunctional Service Robot

——You will experience the intelligent voice ordering and pepper service in the terminal.

——You will ask pepper to take care of your luggage in the terminal so that you can enjoy the activities easily.

——You will check the boarding time of the ticket through pepper in the terminal building, and enjoy the reminder from pepper at each time point.

Pepper's service at the airport can make every ordinary person feel the change of life brought by science and technology.

Log

日期	2020.10.6	2020.12.26	2021.4.13	2021.4.20	2021.4.21	2021.4.29	2021.5.4
研发任务	第一次参加科创活动，开始接触Pepper机器人的学习。	开始学习基础Python语言。	组成“Robotcup”比赛团队，开始策划并制作“航站楼Pepper智能服务机器人”。	策划组开始制作流程图，技术组开始进行相关功能尝试。	策划组拓展流程图，细化任务。	策划组进行界面设计。	进行项目说明书和PPT的初步制作。
日期	2021.5.6	2021.5.11	2021.5.13	2021.5.18	2021.5.19		
研发任务	技术组进行自助点餐、行李托管、机票查询服务的初步测试。	策划组与技术组交流合作，完成项目说明书与PPT的初稿。	策划组与技术组合作，首次尝试视频拍摄。	与技术组交流完成核心技术的编写。	完成PPT和项目说明书的完整版，进行正式版视频拍摄。		

THANKS

让科技便捷生活

一切都将过去，而科技将会永存