Team 1 – Task 1
(Computer vision, Navigation)

Akenat Loawkitpanich 6210552706

Nattanon Manotrairat 6210554661

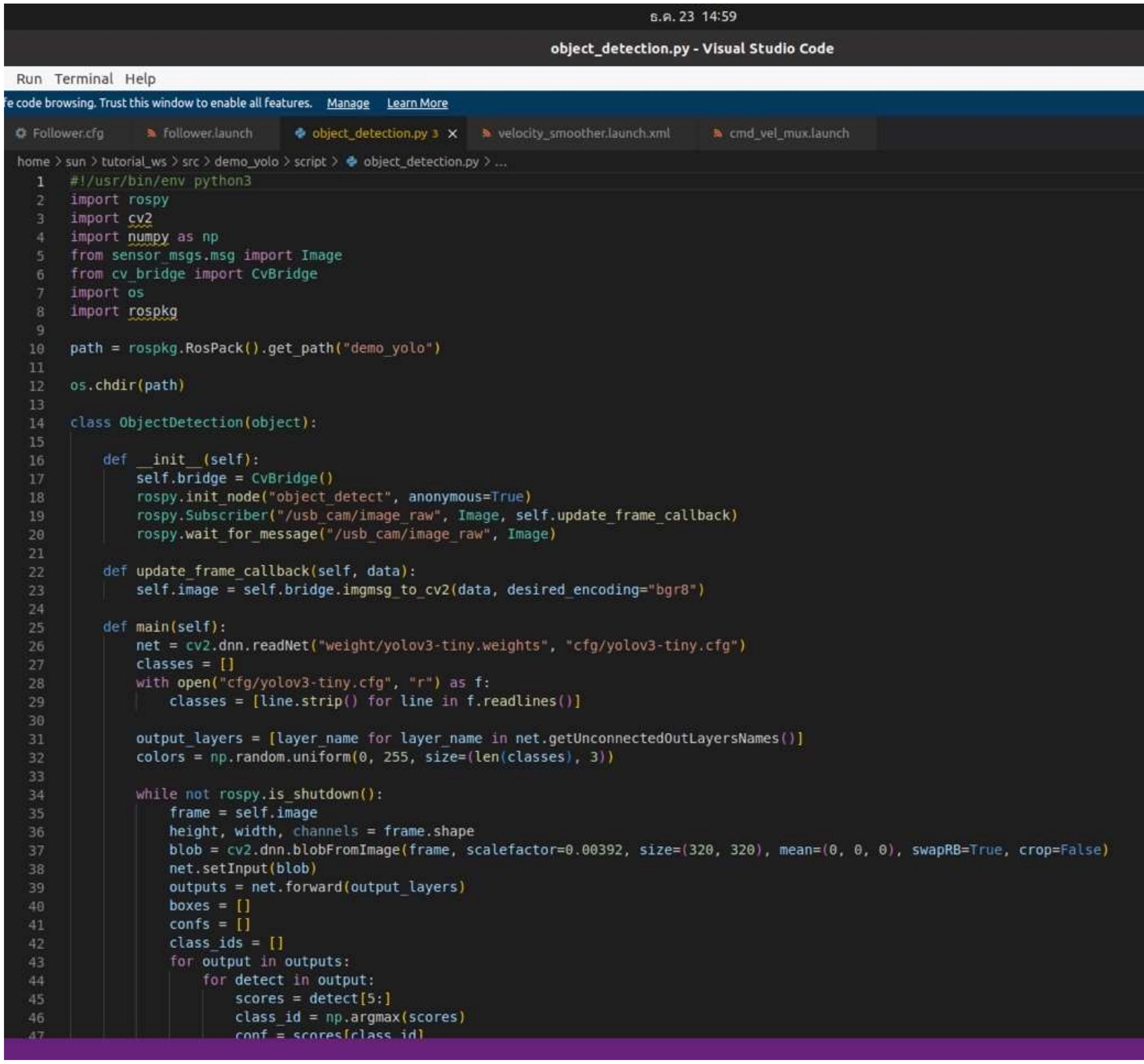
Objectives

- Robot can detect a person
- Navigate to a person (skipped)
- Detect a raised hand
- Avoid obstacle (skipped)

Task 1.1

for person detection

- Using yolo_v4
- Weight: https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov3.weights


ธ.ค. 23 14:59
object_detection.py - Visual Studio Code
Run Terminal Help
e code browsing. Trust this window to enable all features. Manage Learn More
Follower.cfg
follower.launch
object_detection.py 3
velocity_smoother.launch.xml
cmd_vel_mux.launch
home > sun > tutorial_ws > src > demo_yolo > script > object_detection.py > ...

```python
#!/usr/bin/env python3
import rospy
import cv2
import numpy as np
from sensor_msgs.msg import Image
from cv_bridge import CvBridge
import os
import rospkg

path = rospkg.RosPack().get_path("demo_yolo")

os.chdir(path)

class ObjectDetection(object):

    def __init__(self):
        self.bridge = CvBridge()
        rospy.init_node("object_detect", anonymous=True)
        rospy.Subscriber("/usb_cam/image_raw", Image, self.update_frame_callback)
        rospy.wait_for_message("/usb_cam/image_raw", Image)

    def update_frame_callback(self, data):
        self.image = self.bridge.imgmsg_to_cv2(data, desired_encoding="bgr8")

    def main(self):
        net = cv2.dnn.readNet("weight/yolov3-tiny.weights", "cfg/yolov3-tiny.cfg")
        classes = []
        with open("cfg/yolov3-tiny.cfg", "r") as f:
            classes = [line.strip() for line in f.readlines()]

        output_layers = [layer_name for layer_name in net.getUnconnectedOutLayersNames()]
        colors = np.random.uniform(0, 255, size=(len(classes), 3))

        while not rospy.is_shutdown():
            frame = self.image
            height, width, channels = frame.shape
            blob = cv2.dnn.blobFromImage(frame, scalefactor=0.00392, size=(320, 320), mean=(0, 0, 0), swapRB=True, crop=False)
            net.setInput(blob)
            outputs = net.forward(output_layers)
            boxes = []
            confs = []
            class_ids = []
            for output in outputs:
                for detect in output:
                    scores = detect[5:]
                    class_id = np.argmax(scores)
                    conf = scores[class_id]
```

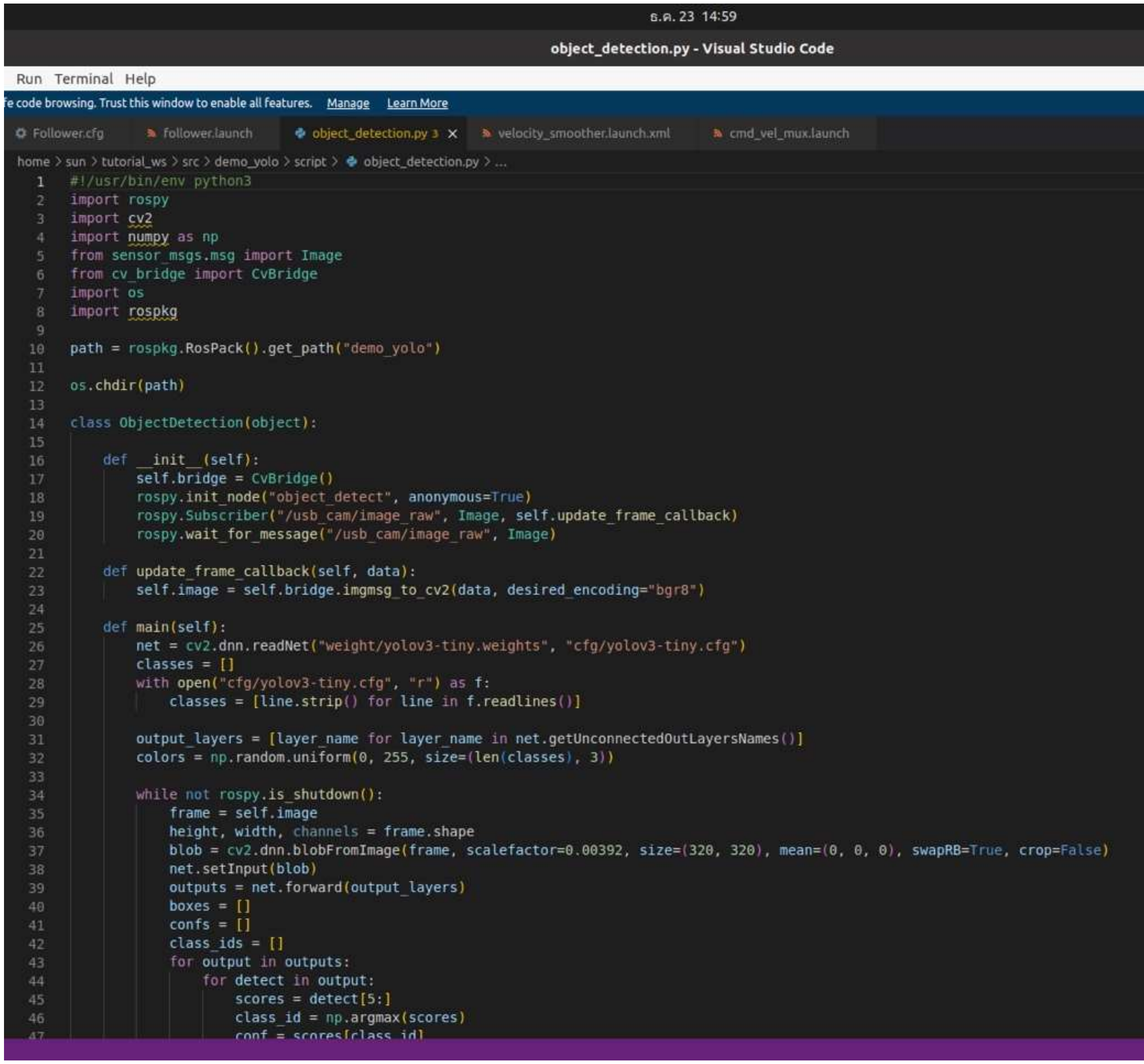
Task 1.1

for person detection

- declare the path for ros package and change to that directory

In ObjectDetection function:

- Change the input to use the Kinect
- Then importing the weight to detect a person

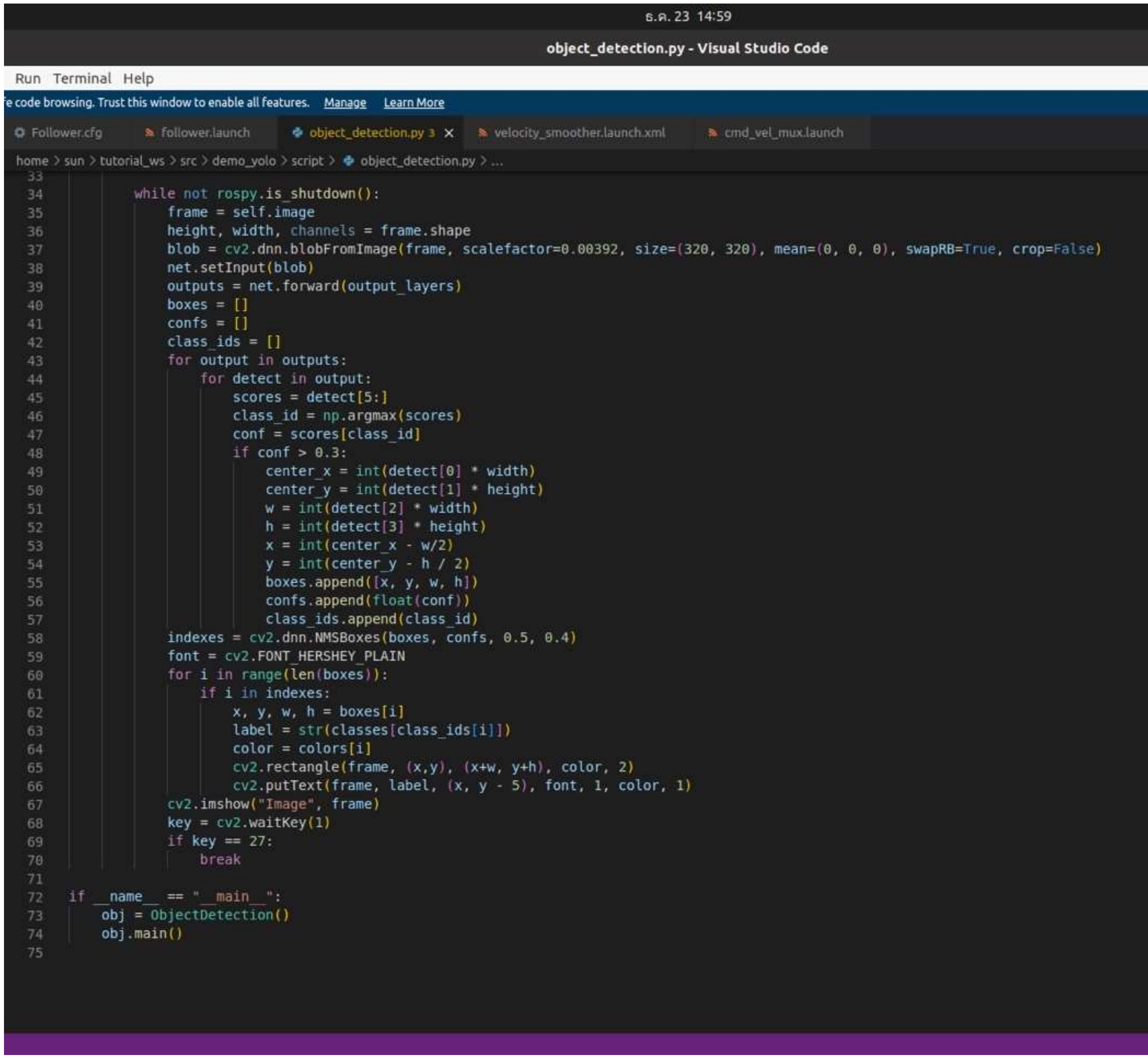

ธ.ค. 23 14:59
object_detection.py - Visual Studio Code
Run Terminal Help
fe code browsing. Trust this window to enable all features. Manage Learn More
Follower.cfg
follower.launch
object_detection.py 3
velocity_smoother.launch.xml
cmd_vel_mux.launch
home > sun > tutorial_ws > src > demo_yolo > script > object_detection.py > ...

```python
#!/usr/bin/env python3
import rospy
import cv2
import numpy as np
from sensor_msgs.msg import Image
from cv_bridge import CvBridge
import os
import rospkg

path = rospkg.RosPack().get_path("demo_yolo")

os.chdir(path)

class ObjectDetection(object):

    def __init__(self):
        self.bridge = CvBridge()
        rospy.init_node("object_detect", anonymous=True)
        rospy.Subscriber("/usb_cam/image_raw", Image, self.update_frame_callback)
        rospy.wait_for_message("/usb_cam/image_raw", Image)

    def update_frame_callback(self, data):
        self.image = self.bridge.imgmsg_to_cv2(data, desired_encoding="bgr8")

    def main(self):
        net = cv2.dnn.readNet("weight/yolov3-tiny.weights", "cfg/yolov3-tiny.cfg")
        classes = []
        with open("cfg/yolov3-tiny.cfg", "r") as f:
            classes = [line.strip() for line in f.readlines()]

        output_layers = [layer_name for layer_name in net.getUnconnectedOutLayersNames()]
        colors = np.random.uniform(0, 255, size=(len(classes), 3))

        while not rospy.is_shutdown():
            frame = self.image
            height, width, channels = frame.shape
            blob = cv2.dnn.blobFromImage(frame, scalefactor=0.00392, size=(320, 320), mean=(0, 0, 0), swapRB=True, crop=False)
            net.setInput(blob)
            outputs = net.forward(output_layers)
            boxes = []
            confs = []
            class_ids = []
            for output in outputs:
                for detect in output:
                    scores = detect[5:]
                    class_id = np.argmax(scores)
                    conf = scores[class_id]
```

Task 1.1

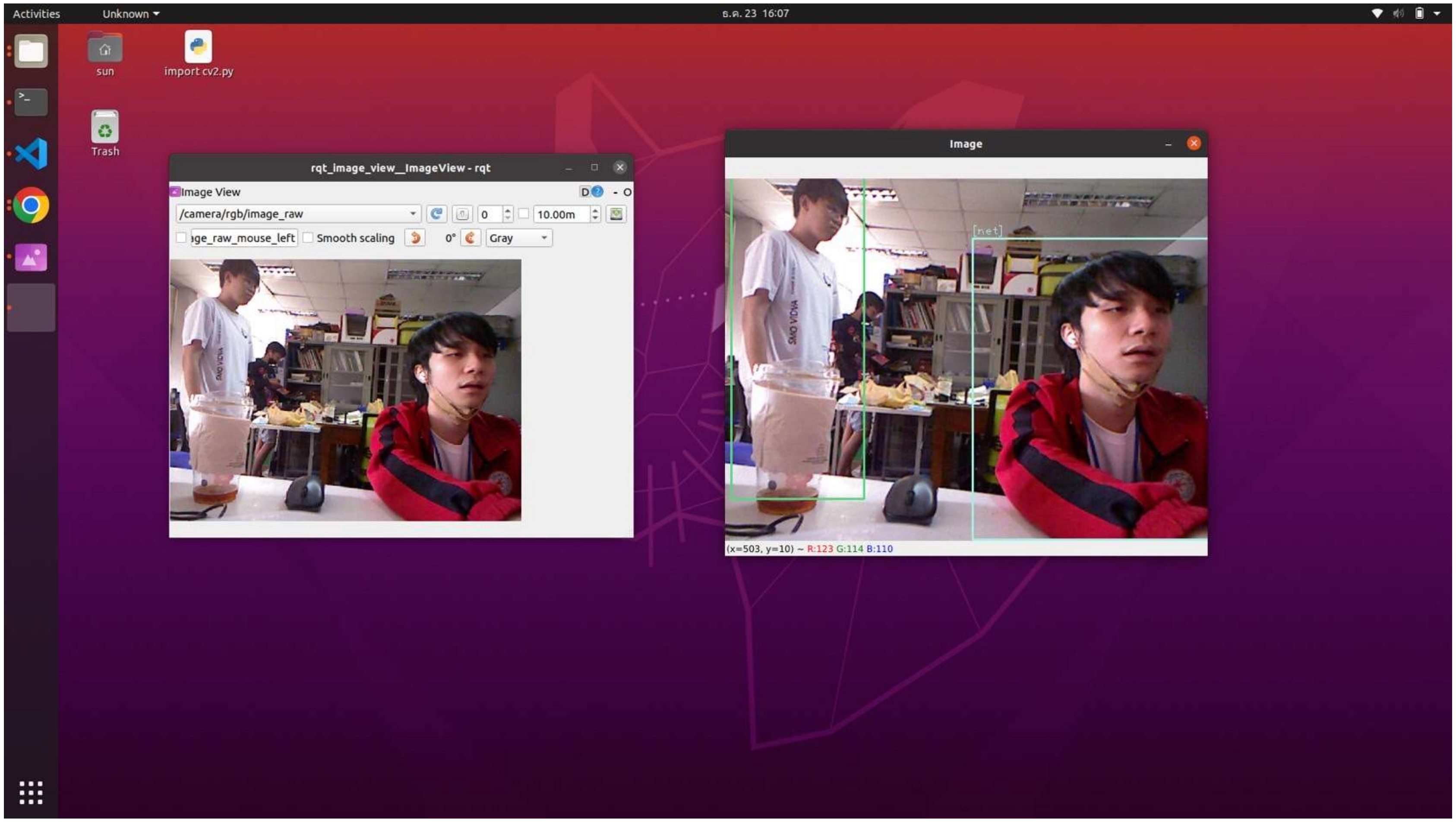
for person detection

- The code below is just for the output to show up as a box with different colour for different persons
  (the colour is random)
- Then call a command in main


ธ.ค. 23 14:59
object_detection.py - Visual Studio Code
Run Terminal Help
e code browsing. Trust this window to enable all features. Manage Learn More
Follower.cfg
follower.launch
object_detection.py 3
velocity_smoother.launch.xml
cmd_vel_mux.launch
home > sun > tutorial_ws > src > demo_yolo > script > object_detection.py > ...

```python
        while not rospy.is_shutdown():
            frame = self.image
            height, width, channels = frame.shape
            blob = cv2.dnn.blobFromImage(frame, scalefactor=0.00392, size=(320, 320), mean=(0, 0, 0), swapRB=True, crop=False)
            net.setInput(blob)
            outputs = net.forward(output_layers)
            boxes = []
            confs = []
            class_ids = []
            for output in outputs:
                for detect in output:
                    scores = detect[5:]
                    class_id = np.argmax(scores)
                    conf = scores[class_id]
                    if conf > 0.3:
                        center_x = int(detect[0] * width)
                        center_y = int(detect[1] * height)
                        w = int(detect[2] * width)
                        h = int(detect[3] * height)
                        x = int(center_x - w/2)
                        y = int(center_y - h / 2)
                        boxes.append([x, y, w, h])
                        confs.append(float(conf))
                        class_ids.append(class_id)
            indexes = cv2.dnn.NMSBoxes(boxes, confs, 0.5, 0.4)
            font = cv2.FONT_HERSHEY_PLAIN
            for i in range(len(boxes)):
                if i in indexes:
                    x, y, w, h = boxes[i]
                    label = str(classes[class_ids[i]])
                    color = colors[i]
                    cv2.rectangle(frame, (x,y), (x+w, y+h), color, 2)
                    cv2.putText(frame, label, (x, y - 5), font, 1, color, 1)
            cv2.imshow("Image", frame)
            key = cv2.waitKey(1)
            if key == 27:
                break

if __name__ == "__main__":
    obj = ObjectDetection()
    obj.main()
```

Person Tracking

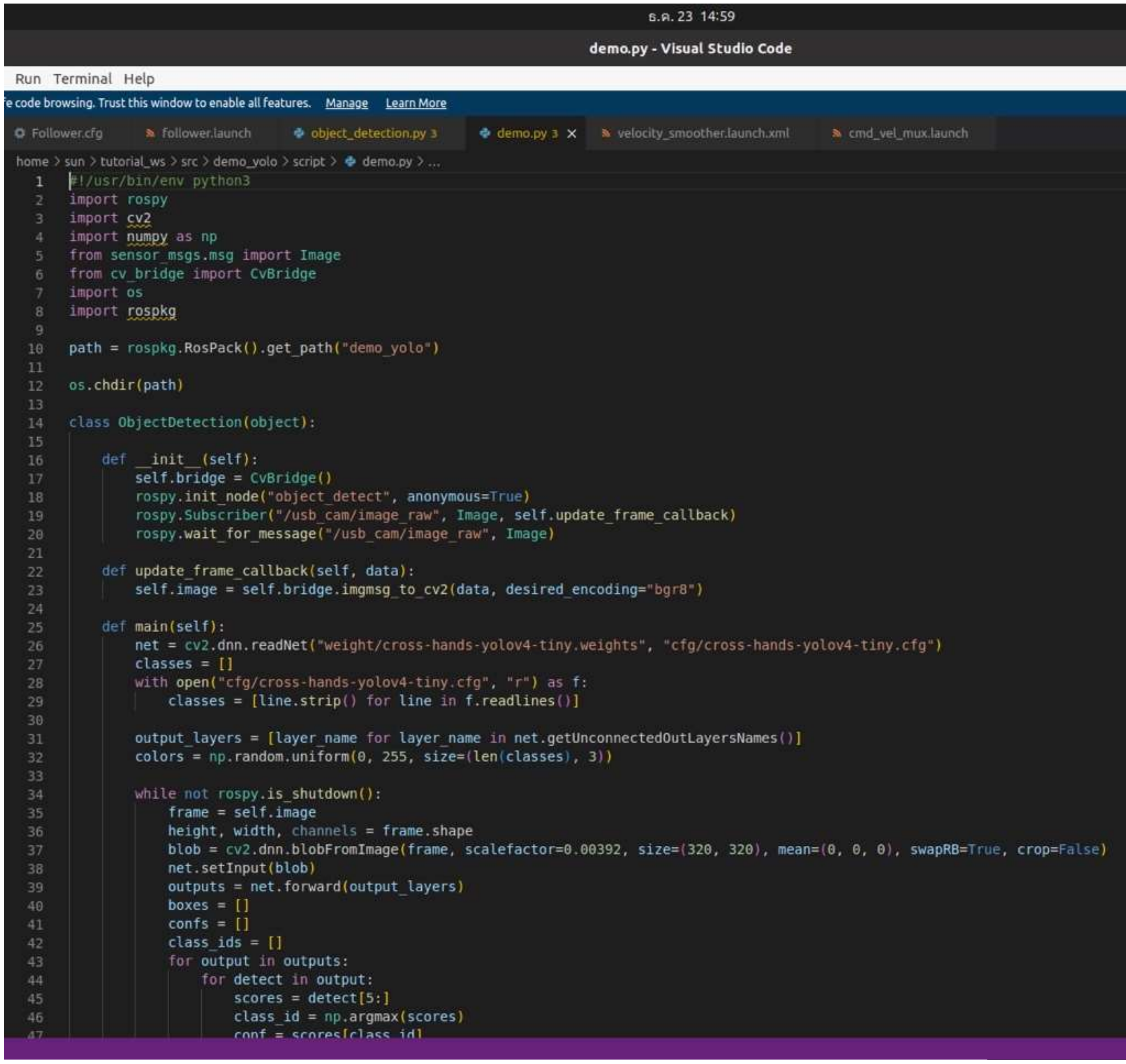


Task 1.3

for hands detection

- Weight & original code: https://github.com/cansik/yolo-hand-detection

- Modified from the original code to get hands by changing path and config to hands model

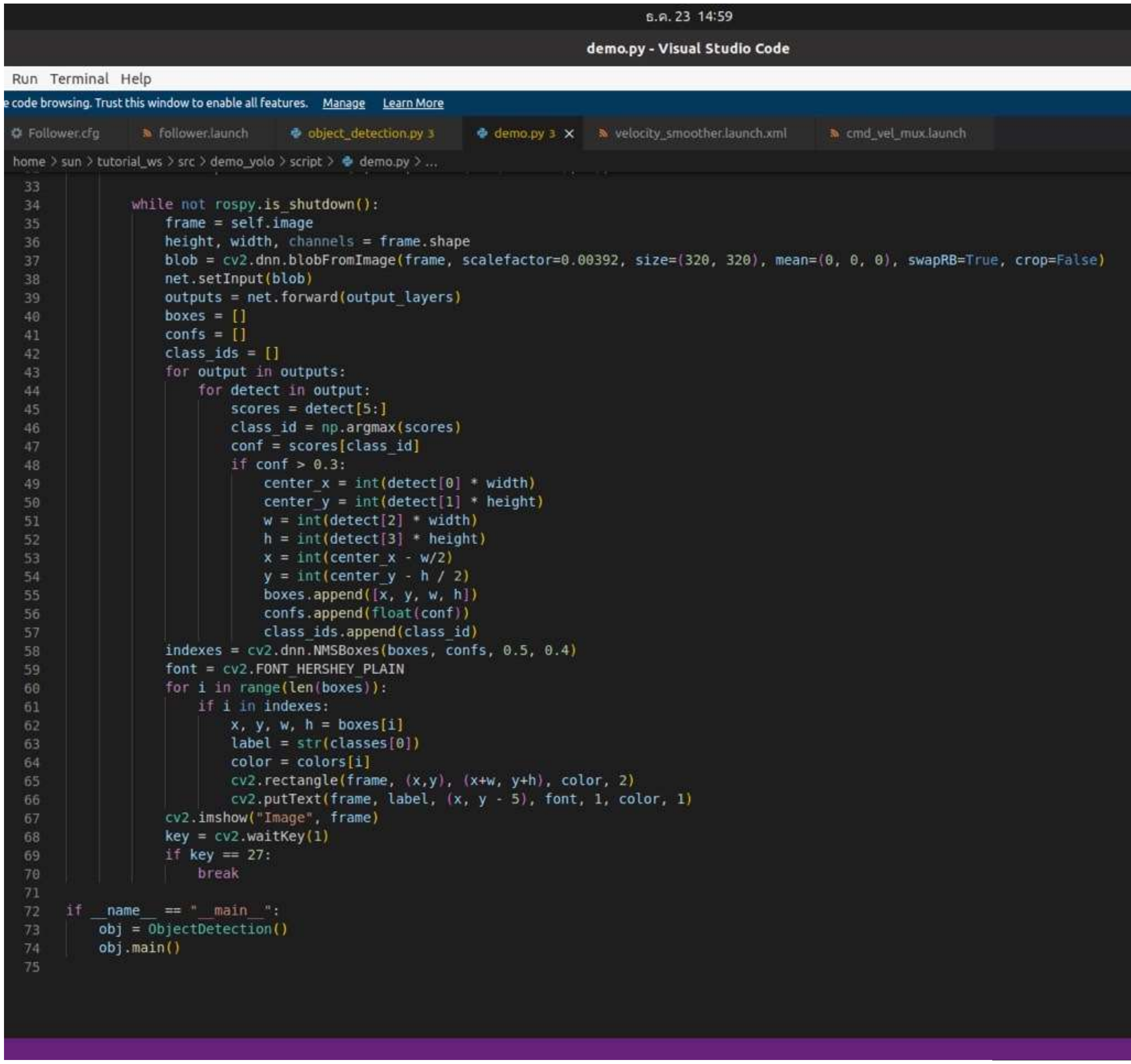

ธ.ค. 23 14:59
demo.py - Visual Studio Code
Run Terminal Help
e code browsing. Trust this window to enable all features. Manage Learn More
Follower.cfg
follower.launch
object_detection.py 3
demo.py 3
velocity_smoother.launch.xml
cmd_vel_mux.launch
home > sun > tutorial_ws > src > demo_yolo > script > demo.py > ...

```python
#!/usr/bin/env python3
import rospy
import cv2
import numpy as np
from sensor_msgs.msg import Image
from cv_bridge import CvBridge
import os
import rospkg

path = rospkg.RosPack().get_path("demo_yolo")

os.chdir(path)

class ObjectDetection(object):

    def __init__(self):
        self.bridge = CvBridge()
        rospy.init_node("object_detect", anonymous=True)
        rospy.Subscriber("/usb_cam/image_raw", Image, self.update_frame_callback)
        rospy.wait_for_message("/usb_cam/image_raw", Image)

    def update_frame_callback(self, data):
        self.image = self.bridge.imgmsg_to_cv2(data, desired_encoding="bgr8")

    def main(self):
        net = cv2.dnn.readNet("weight/cross-hands-yolov4-tiny.weights", "cfg/cross-hands-yolov4-tiny.cfg")
        classes = []
        with open("cfg/cross-hands-yolov4-tiny.cfg", "r") as f:
            classes = [line.strip() for line in f.readlines()]

        output_layers = [layer_name for layer_name in net.getUnconnectedOutLayersNames()]
        colors = np.random.uniform(0, 255, size=(len(classes), 3))

        while not rospy.is_shutdown():
            frame = self.image
            height, width, channels = frame.shape
            blob = cv2.dnn.blobFromImage(frame, scalefactor=0.00392, size=(320, 320), mean=(0, 0, 0), swapRB=True, crop=False)
            net.setInput(blob)
            outputs = net.forward(output_layers)
            boxes = []
            confs = []
            class_ids = []
            for output in outputs:
                for detect in output:
                    scores = detect[5:]
                    class_id = np.argmax(scores)
                    conf = scores[class_id]
```

Task 1.3

for hands detection

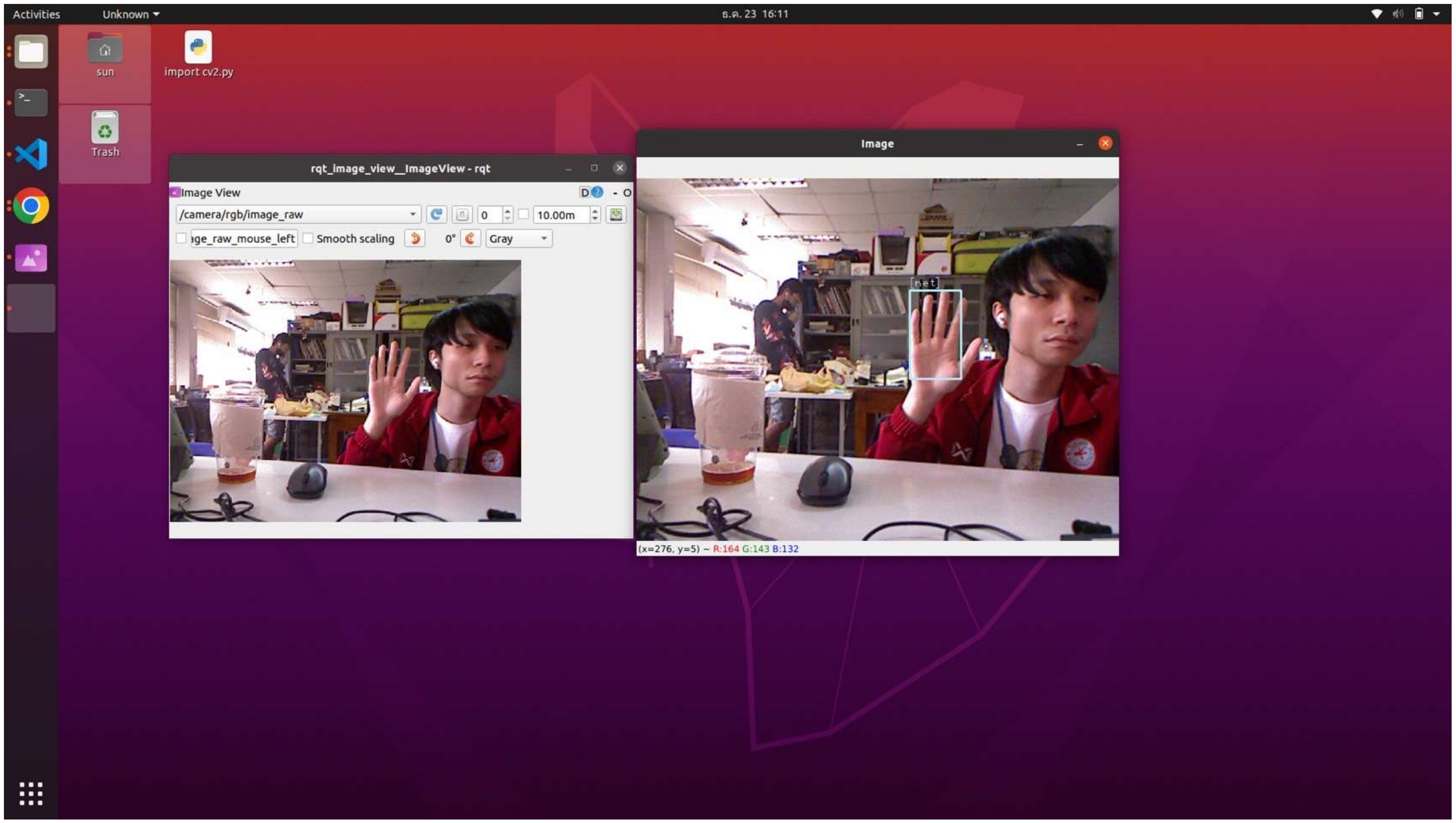
- The rest is the same as in task 1.1


ธ.ค. 23 14:59
demo.py - Visual Studio Code
Run Terminal Help
Follower.cfg
follower.launch
object_detection.py 3
demo.py 3
velocity_smoother.launch.xml
cmd_vel_mux.launch
home > sun > tutorial_ws > src > demo_yolo > script > demo.py > ...

e code browsing. Trust this window to enable all features. Manage Learn More

```python
        while not rospy.is_shutdown():
            frame = self.image
            height, width, channels = frame.shape
            blob = cv2.dnn.blobFromImage(frame, scalefactor=0.00392, size=(320, 320), mean=(0, 0, 0), swapRB=True, crop=False)
            net.setInput(blob)
            outputs = net.forward(output_layers)
            boxes = []
            confs = []
            class_ids = []
            for output in outputs:
                for detect in output:
                    scores = detect[5:]
                    class_id = np.argmax(scores)
                    conf = scores[class_id]
                    if conf > 0.3:
                        center_x = int(detect[0] * width)
                        center_y = int(detect[1] * height)
                        w = int(detect[2] * width)
                        h = int(detect[3] * height)
                        x = int(center_x - w/2)
                        y = int(center_y - h / 2)
                        boxes.append([x, y, w, h])
                        confs.append(float(conf))
                        class_ids.append(class_id)
            indexes = cv2.dnn.NMSBoxes(boxes, confs, 0.5, 0.4)
            font = cv2.FONT_HERSHEY_PLAIN
            for i in range(len(boxes)):
                if i in indexes:
                    x, y, w, h = boxes[i]
                    label = str(classes[0])
                    color = colors[i]
                    cv2.rectangle(frame, (x,y), (x+w, y+h), color, 2)
                    cv2.putText(frame, label, (x, y - 5), font, 1, color, 1)
            cv2.imshow("Image", frame)
            key = cv2.waitKey(1)
            if key == 27:
                break

if __name__ == "__main__":
    obj = ObjectDetection()
    obj.main()
```

Hand Tracking

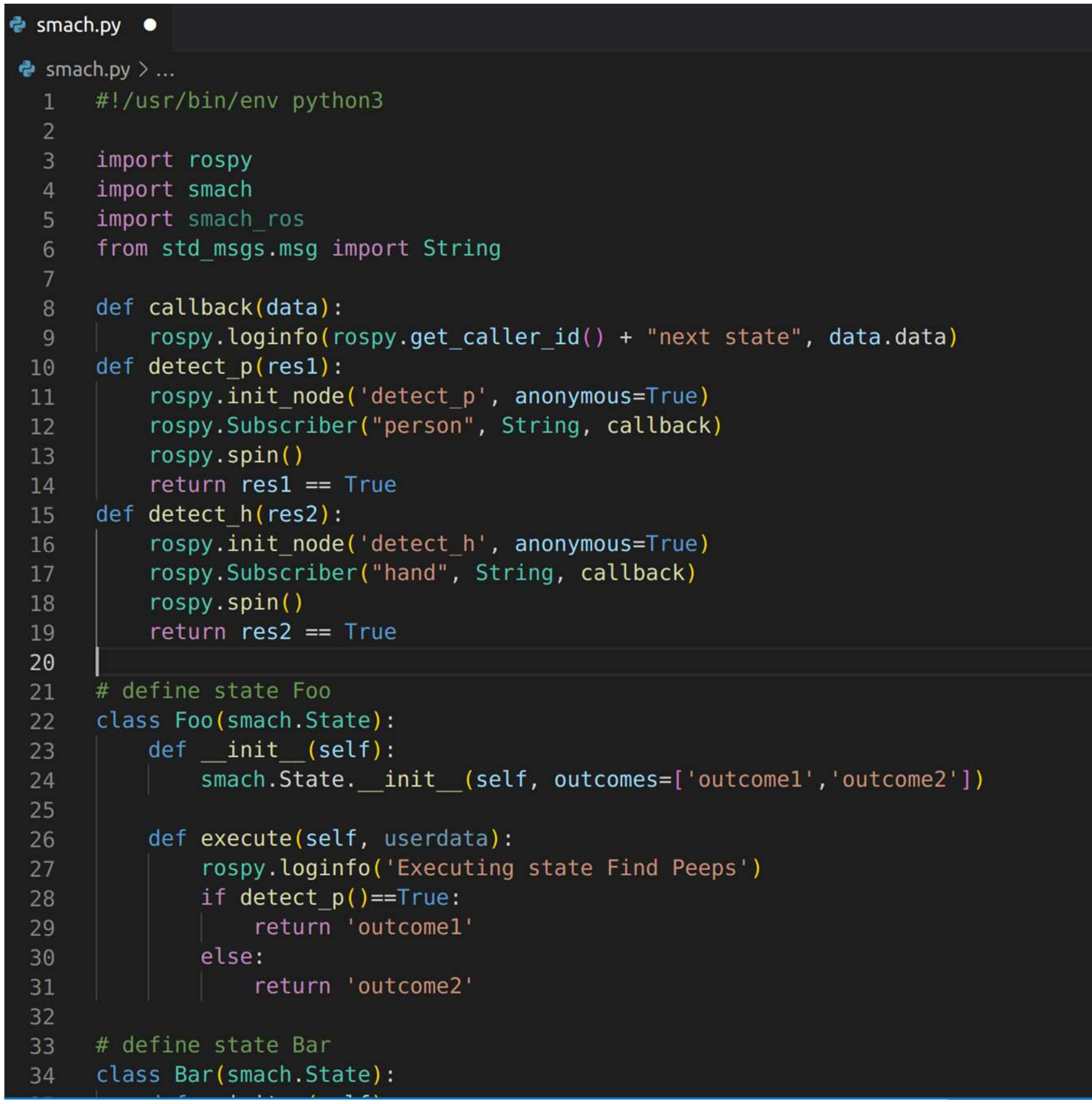


State Machine

(Not working)

In theory:

- Get string confirming result
- If person is detected (move on to detect hand)
- If hand is detected (nav command run that included with obstacle avoidance)



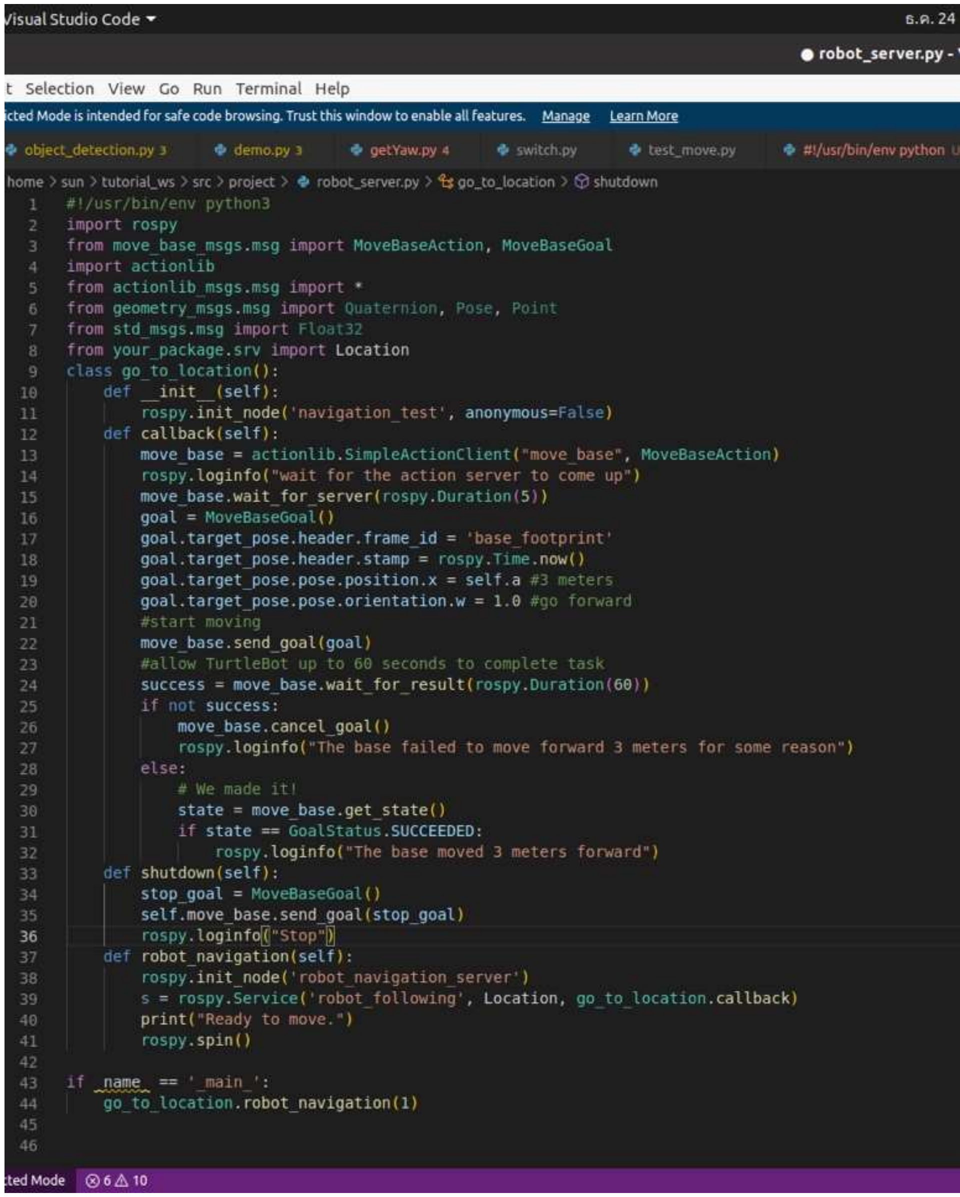
smach.py

smach.py > ...

```python
#!/usr/bin/env python3

import rospy
import smach
import smach_ros
from std_msgs.msg import String

def callback(data):
    rospy.loginfo(rospy.get_caller_id() + "next state", data.data)
def detect_p(res1):
    rospy.init_node('detect_p', anonymous=True)
    rospy.Subscriber("person", String, callback)
    rospy.spin()
    return res1 == True
def detect_h(res2):
    rospy.init_node('detect_h', anonymous=True)
    rospy.Subscriber("hand", String, callback)
    rospy.spin()
    return res2 == True

# define state Foo
class Foo(smach.State):
    def __init__(self):
        smach.State.__init__(self, outcomes=['outcome1','outcome2'])

    def execute(self, userdata):
        rospy.loginfo('Executing state Find Peeps')
        if detect_p()==True:
            return 'outcome1'
        else:
            return 'outcome2'

# define state Bar
class Bar(smach.State):
```

Skipped Tasks

- GoToLocation code modification
- The obstacle avoidance is in gotolocation


Visual Studio Code
ธ.ค. 24
● robot_server.py -
t Selection View Go Run Terminal Help
cted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
object_detection.py 3
demo.py 3
getYaw.py 4
switch.py
test_move.py
#!/usr/bin/env python
home > sun > tutorial_ws > src > project > robot_server.py > go_to_location > shutdown

```python
#!/usr/bin/env python3
import rospy
from move_base_msgs.msg import MoveBaseAction, MoveBaseGoal
import actionlib
from actionlib_msgs.msg import *
from geometry_msgs.msg import Quaternion, Pose, Point
from std_msgs.msg import Float32
from your_package.srv import Location
class go_to_location():
    def __init__(self):
        rospy.init_node('navigation_test', anonymous=False)
    def callback(self):
        move_base = actionlib.SimpleActionClient("move_base", MoveBaseAction)
        rospy.loginfo("wait for the action server to come up")
        move_base.wait_for_server(rospy.Duration(5))
        goal = MoveBaseGoal()
        goal.target_pose.header.frame_id = 'base_footprint'
        goal.target_pose.header.stamp = rospy.Time.now()
        goal.target_pose.pose.position.x = self.a #3 meters
        goal.target_pose.pose.orientation.w = 1.0 #go forward
        #start moving
        move_base.send_goal(goal)
        #allow TurtleBot up to 60 seconds to complete task
        success = move_base.wait_for_result(rospy.Duration(60))
        if not success:
            move_base.cancel_goal()
            rospy.loginfo("The base failed to move forward 3 meters for some reason")
        else:
            # We made it!
            state = move_base.get_state()
            if state == GoalStatus.SUCCEEDED:
                rospy.loginfo("The base moved 3 meters forward")
    def shutdown(self):
        stop_goal = MoveBaseGoal()
        self.move_base.send_goal(stop_goal)
        rospy.loginfo("Stop")
    def robot_navigation(self):
        rospy.init_node('robot_navigation_server')
        s = rospy.Service('robot_following', Location, go_to_location.callback)
        print("Ready to move.")
        rospy.spin()

if __name__ == '__main__':
    go_to_location.robot_navigation(1)
```