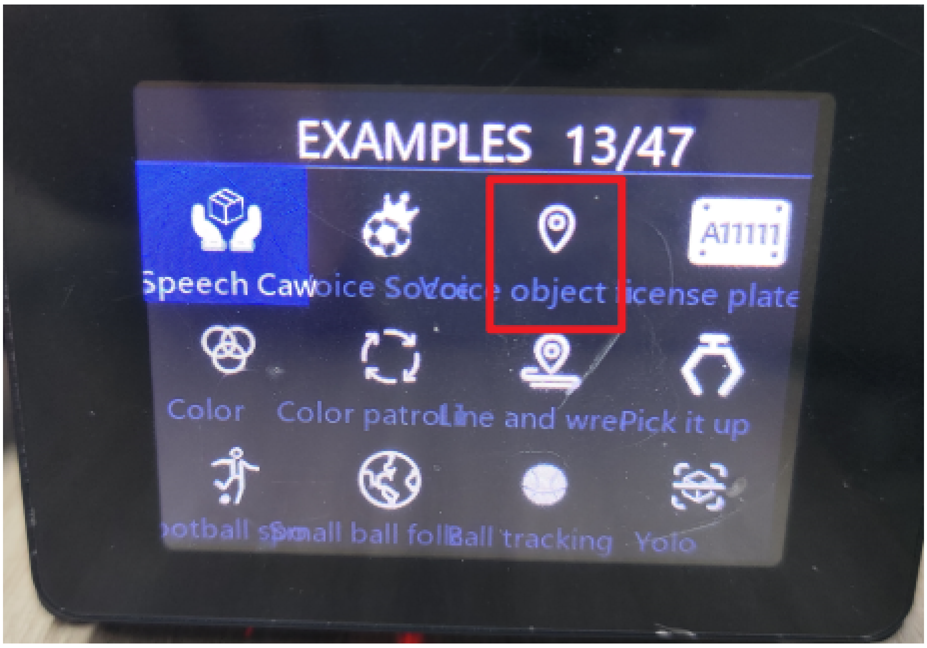
AI Voice Interaction: Object Tracking

Function Introduction

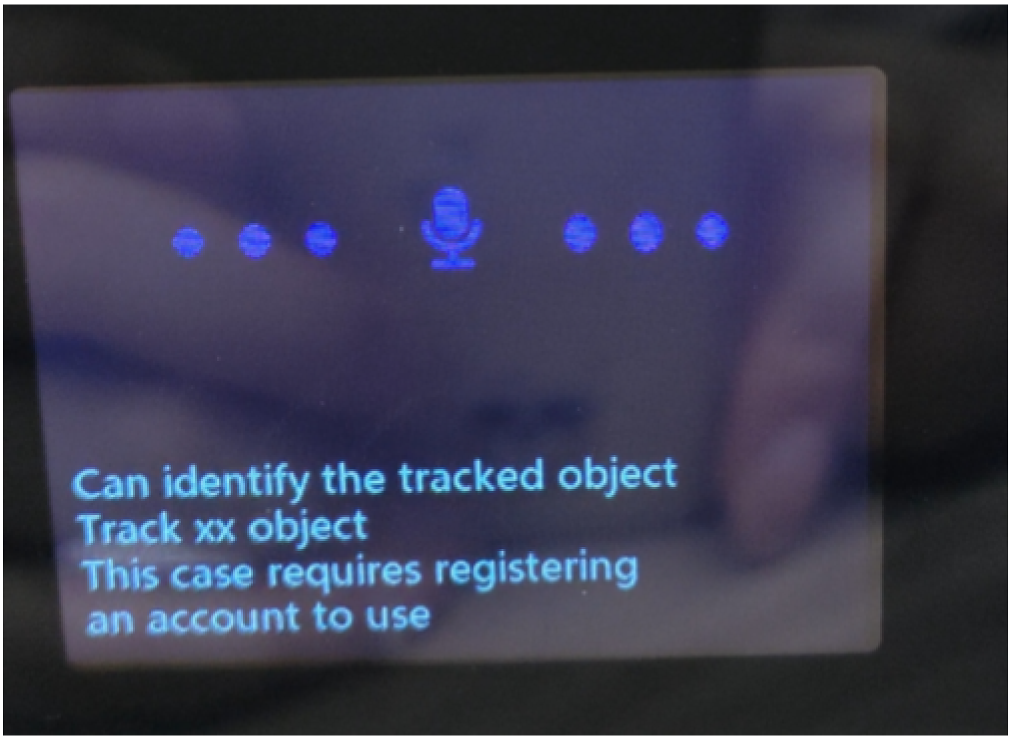
This case is a game of booting up a large program. The position of the object coordinates is obtained through the processing of a large model, so a key configuration of a large model is required.
The main large model used is: Qwen. You need to configure it in the first section before this case can be used normally.
This function requires an Internet connection to work properly

Function Experience

1. Turn on the robot dog first, press the button on the upper right of the "dog head" to enter the sample mode, and then select the voice object tracking function.

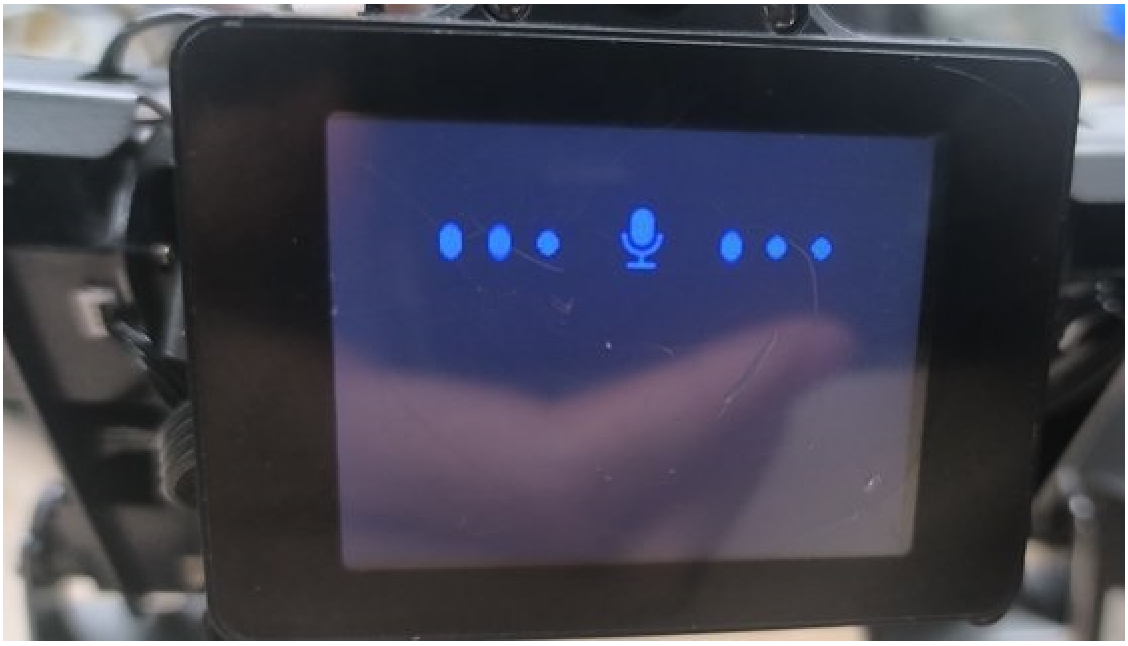


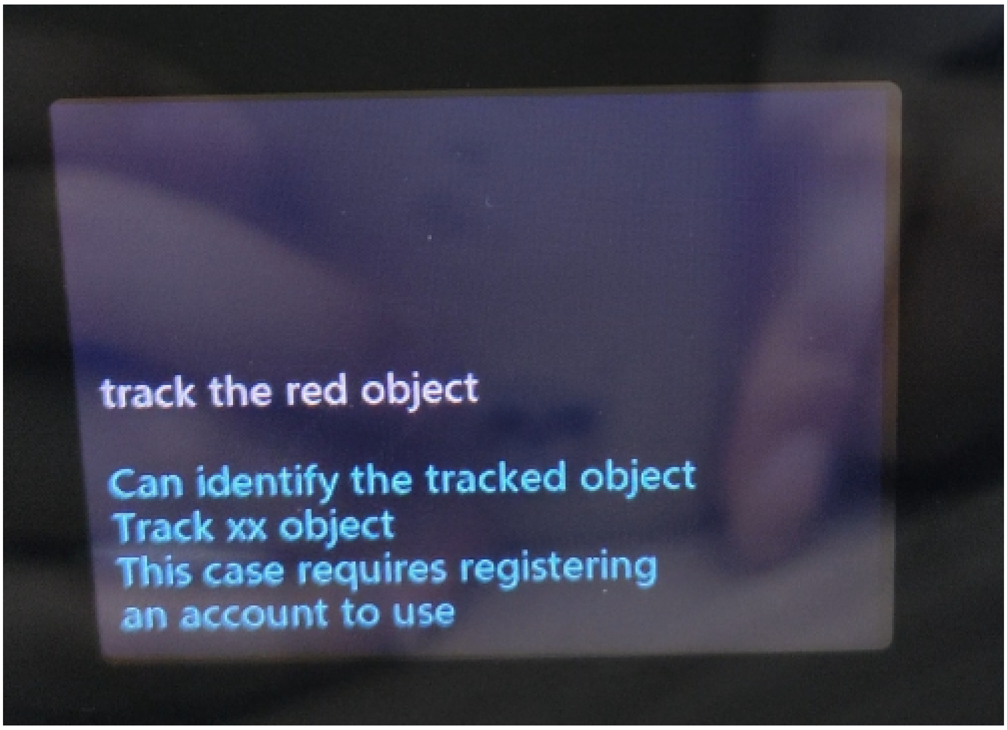
2. After entering the voice object tracking function, wake up by voice first, "lulu".



Can identify the tracked object
Track xx object
This case requires registering
an account to use

3. When you hear a ding, you can say the prompt word at the bottom of the screen.





track the red object

Can identify the tracked object
Track xx object
This case requires registering
an account to use

4. The robot dog will recognize the semantics and enter a target object tracking.

Program source code

1. First, log in to the robot dog system through VNC
2. Then enter the terminal

```
cd /home/pi/RaspberryPi-CM4-main/demos/speech_AI_food/
tree -L 1
```

3. Directory structure description
   ├── audio.py #recording file
   ├── auto_platform.py #system environment dependency
   ├── language_recognize.py #speech recognition
   ├── libnyumaya.py #speech wake-up
   ├── speech_ai_food_image.py #object tracking file
   ├── speech_AI_food.py #speech interaction main program
   ├── speech_picture.py #large model image analysis
   ├── xinghou_ImageAPI.py #image interface
   └── xinghou_tts.py #synthesized audio file

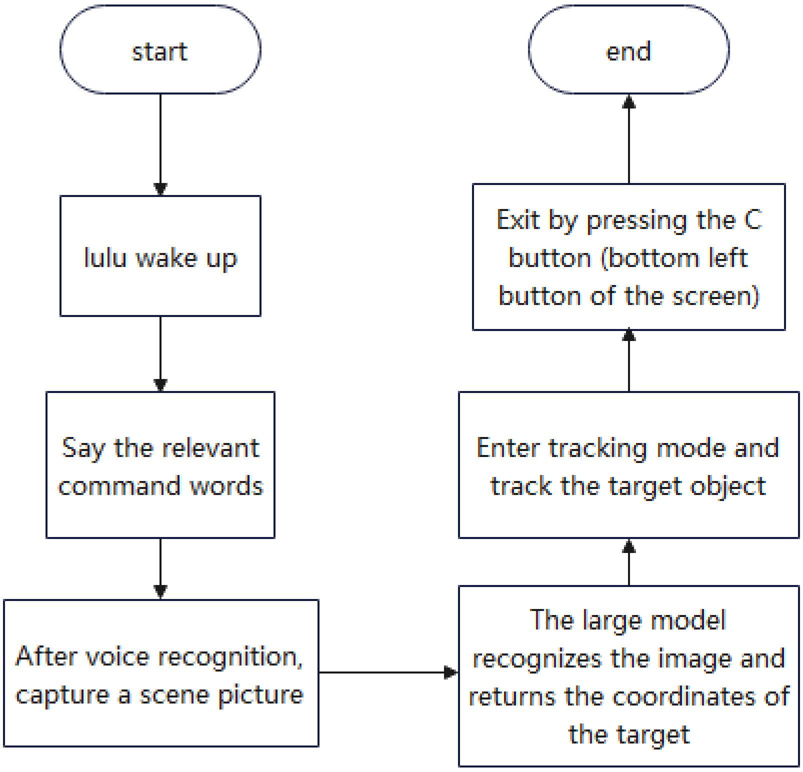
```
if net:
    while True:
        open_AI_play()
        if detect_keyword():
            clear_top()
            start_recording()
            content = test_one()

            if content != "":
                clear_top()
                speech_list = line_break(content)
                print(speech_list)

                if la == "en":
                    english_only = ''.join(char for char in speech_list if
ord(char) < 128)
                    display_text = english_only
                else:
                    display_text = speech_list

                lcd_draw_string(
                    draw,
                    10,
                    110,
                    display_text,
                    color=(255, 255, 255),
                    scale=font2,
                    mono_space=False,
                )
                display.ShowImage(splash)

                lines = len(display_text.split("\n"))
                tick = 0.3
                if lines > 6:
                    scroll_text_on_lcd(display_text, 10, 111, 6, tick)


                if not actions_AI(content):
                    continue

            if content == 0:
               break

        time.sleep(0.1)
else:
    draw_offline()
    while True:
        time.sleep(0.1)
```

The program can clearly show the general process. After waking up, say the command word, and the result will be fed back to the screen after voice recognition, and then enter the tracking target object.
Note:

1. During the tracking process, the moving speed cannot be too fast, otherwise it is easy to fail to track or misrecognize.
2. When the object to be tracked is said, the robot dog enters the tracking mode. Do not move the object to be tracked.
3. If you want to understand the tracking action, you can enter **speech_ai_food_image.py** to check it yourself.

Functional principle

The specific flowchart is as follows:


start
lulu wake up
Say the relevant command words
After voice recognition, capture a scene picture
The large model recognizes the image and returns the coordinates of the target
Enter tracking mode and track the target object
Exit by pressing the C button (bottom left button of the screen)
end