

AI Voice Interaction: Block Gripping

AI Voice Interaction: Block Gripping

[Function Introduction](#)

[Function Experience](#)

[Program source code](#)

[Functional principle](#)

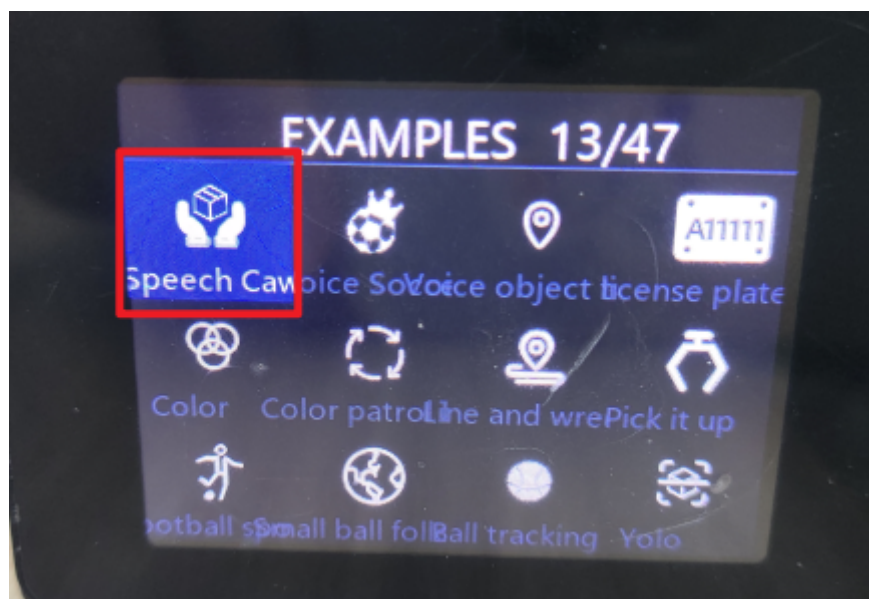
Function Introduction

This case is about how to play the startup program. You can say the color of the wooden block you need to pick up according to the words prompted on the screen, and the robot dog will pick up the wooden block of the corresponding color.

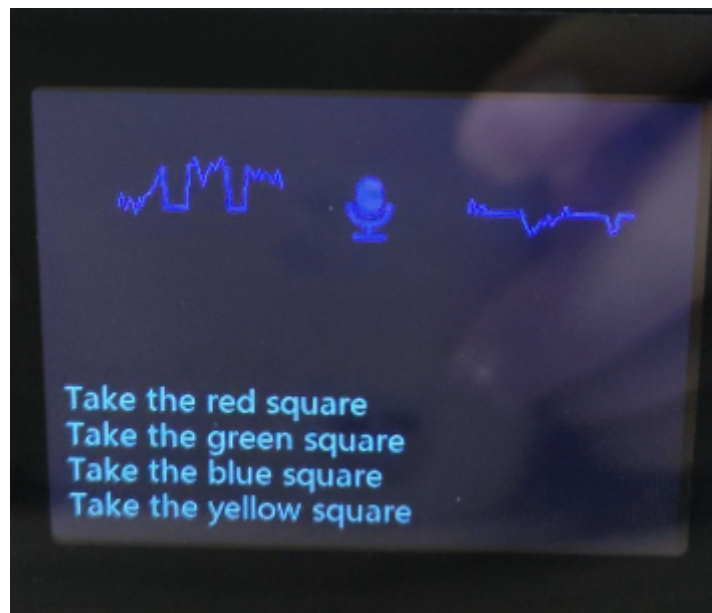
This function requires an Internet connection to work properly

Function Experience

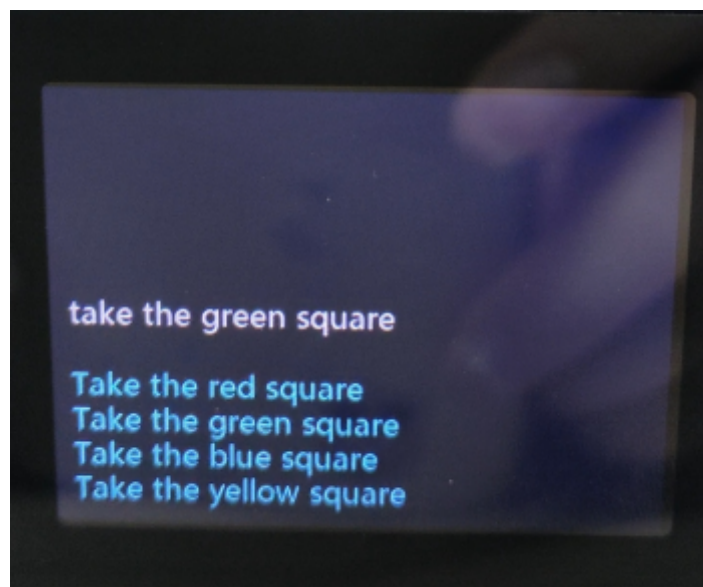
1. Turn on the robot dog first, press the button on the upper right of the "dog head" to enter the sample mode, and then select the "Speech Caw" function.



2. After entering the voice recognition function, wake up with voice first, "lulu".



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



4. The robot dog will recognize the semantics and enter a building block to grab. When the grab is successful, voice interaction is required again.

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_caw/  
tree -L 1
```

3. Directory structure description

- ├─ audio.py #Recording file
- ├─ auto_platform.py #System environment dependency
- ├─ language_recognize.py #Speech recognition
- ├─ libnymaya.py #Speech wake-up
- ├─ speach_caw.py #Clip motion interface
- ├─ speech_AI_caw.py #Speech clip interface
- ├─ 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 corresponding color clamping movement.

If you want to add more colors, you can study the function of the python file in this directory and add it yourself.

Functional principle

The specific flow chart is as follows:

