Multimodal Large Model+Color Blocks Back In Place（Voice Version)

Before running the function, you need to close the App and large programs. For the closing method, refer to [4. Preparation] - [1. Manage APP control services].

1. Function Description

After the program runs, wake up the voice module and input the command to remember the current positions of color blocks. After the program completes recording, scramble the positions of the color blocks. Then wake up the voice module again and input the command to end the current recording task. Then wake up the voice module again and input the color block return command. After thinking, the large model will plan an action list, and the program will control the robotic arm to complete the color block return task according to the action functions in the action list.

2. Startup

Users with Jetson-Nano mainboard version need to enter the docker container first and then input the following command. Users with Orin mainboard can directly open the terminal and input the following command:

```
ros2 launch largemodel largemodel_control.launch.py
```

After waking up the voice module, input by voice:

```
Remember the current positions of the color blocks
```

After recording the positions of the color blocks, wake up the voice module again and input by voice:

```
End the current recording task
```

Then you can scramble the positions of the color blocks. After scrambling, wake up the voice module and input by voice:

```
Put the blocks on the desktop back to their original positions
```

After the buzzer beeps once, it will start gripping the color blocks and returning them to their original positions according to the order recorded in the program. If there are other color blocks on top of the current color block that needs to be returned, it will first remove the color blocks on top, then grip and return to the original position.

3. Task Planning

3.1. Record Current Positions of Color Blocks

1. Call `seewhat()` to observe the environment;
2. Call `seewhat()` again to observe the environment, then determine which color block is the top color block;
3. If there are two or more piles of color blocks in the environment, based on whether they are individual color blocks, choose to execute the following content, calculating each pile of color blocks. Here only one action function can be output at a time:
    - If it is an individual color block, pass the outer bounding box coordinates of the color block to the function `compute_pose(x1, y1, x2, y2, 'name')`, where `name` represents the color of the color block, with values `red`, `blue`, `green` or `yellow`;
    - If it is not an individual color block, pass the outer bounding box coordinates of the top color block to the function `compute_pose_order(x1, y1, x2, y2, 'name', 'order')`, where `name` represents the color of the color block, with values `red`, `blue`, `green` or `yellow`, `order` represents the arrangement order of color blocks from top to bottom, with the value being a string composed of the first letters of each layer's color from top to bottom. For example, if the arrangement order of color blocks from top to bottom is: red, blue, green, yellow, then the value of `order` is `rbgy`, and so on.
4. If there is only one pile of color blocks, pass the outer bounding box coordinates of the top color block to the function `compute_pose_order(x1, y1, x2, y2, 'name', 'order')`, where `name` represents the color of the color block, with values `red`, `blue`, `green` or `yellow`, `order` represents the arrangement order of color blocks from top to bottom, with the value being a string composed of the first letters of each layer's color from top to bottom. For example, if the arrangement order of color blocks from top to bottom is: red, blue, green, yellow, then the value of `order` is `rbgy`, and so on.

3.2. Color Block Return

1. First, call `color_back_to_orin()` function to control color blocks to return to original positions;
2. Call `check_remove(color)` function to check if the target color block exists in the removal list (where `color` is the color of the target color block, with values `red`, `blue`, `green` or `yellow`);
3. If the target color block exists in the removal list, perform the following operations:
   (1) Call `grasp_from_rm_list(color)` function to grip the target color block from the removal list (meaning of `color` parameter is the same as above);
   (2) Pass the color of the target color block as a parameter to the `return_to_orin(color)` function, for example when the target color block is red, call `return_to_orin(red)` (range of `color` parameter is the same as above);
4. If the target color block does not exist in the removal list, the color block needs to be returned to its original position according to the following steps (only output one action function at a time):
   (1) Call `seewhat()` function to find the current target color block that needs to be returned;
   (2) Call `seewhat()` function to determine if there are other objects on top of the current target color block that needs to be returned;
   (3) If there are no other color blocks on top of the current target color block that needs to be returned, perform the following steps:
   ① Call `grasp_obj(x1, y1, x2, y2)` function to grip the current target color block that needs to be returned (where `(x1, y1, x2, y2)` are the top surface bounding box coordinates of the current target color block that needs to be returned);

② Pass the color of the target color block as a parameter to the `return_to_orin(color)` function (range of `color` parameter is the same as above);

(4) If there are other color blocks on top of the current target color block that needs to be returned, perform the following steps:

① Call `remove_obj(x1, y1, x2, y2, color)` function to remove this color block (where `(x1, y1, x2, y2)` are the top surface bounding box coordinates of this color block, `color` is the color of this color block, with values `red`, `blue`, `green` or `yellow`);

② Call `seewhat()` function to observe the environment;

③ If there are no other color blocks on top of the current target color block that needs to be returned, call `grasp_obj(x1, y1, x2, y2)` function to grip the target color block (where `(x1, y1, x2, y2)` are the top surface bounding box coordinates of the target color block);

④ Pass the color of the target color block as a parameter to the `return_to_orin(color)` function (range of `color` parameter is the same as above)

4. Core Code Analysis

You can refer to the content in **4. Core Code Analysis** from tutorial [17. AI Model - Text Version] - [Multimodal Large Model+Color Blocks Back In Place]. The voice version and text version have the same action functions, only the task command input method is different.