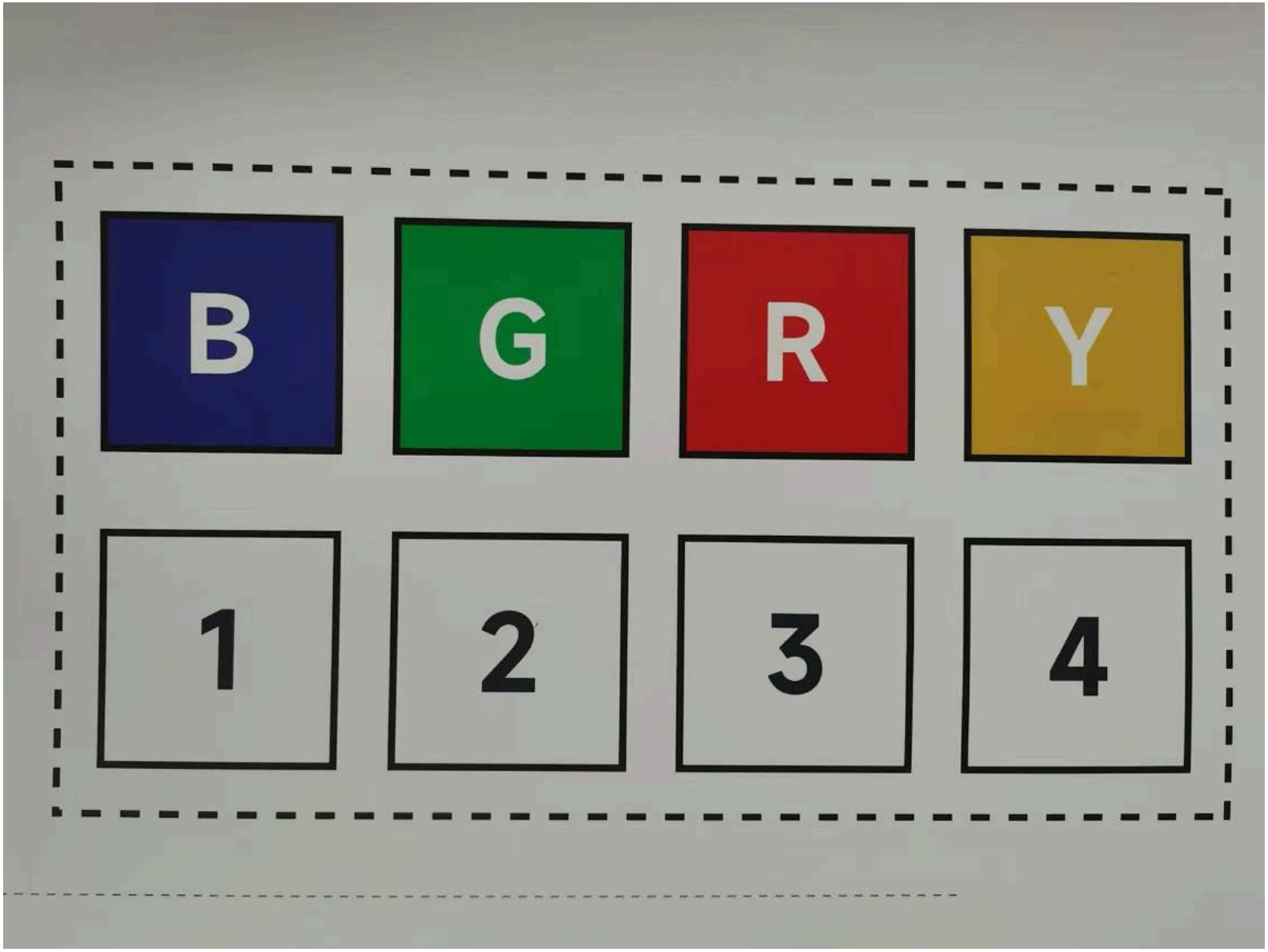
Color recognition grab blocks

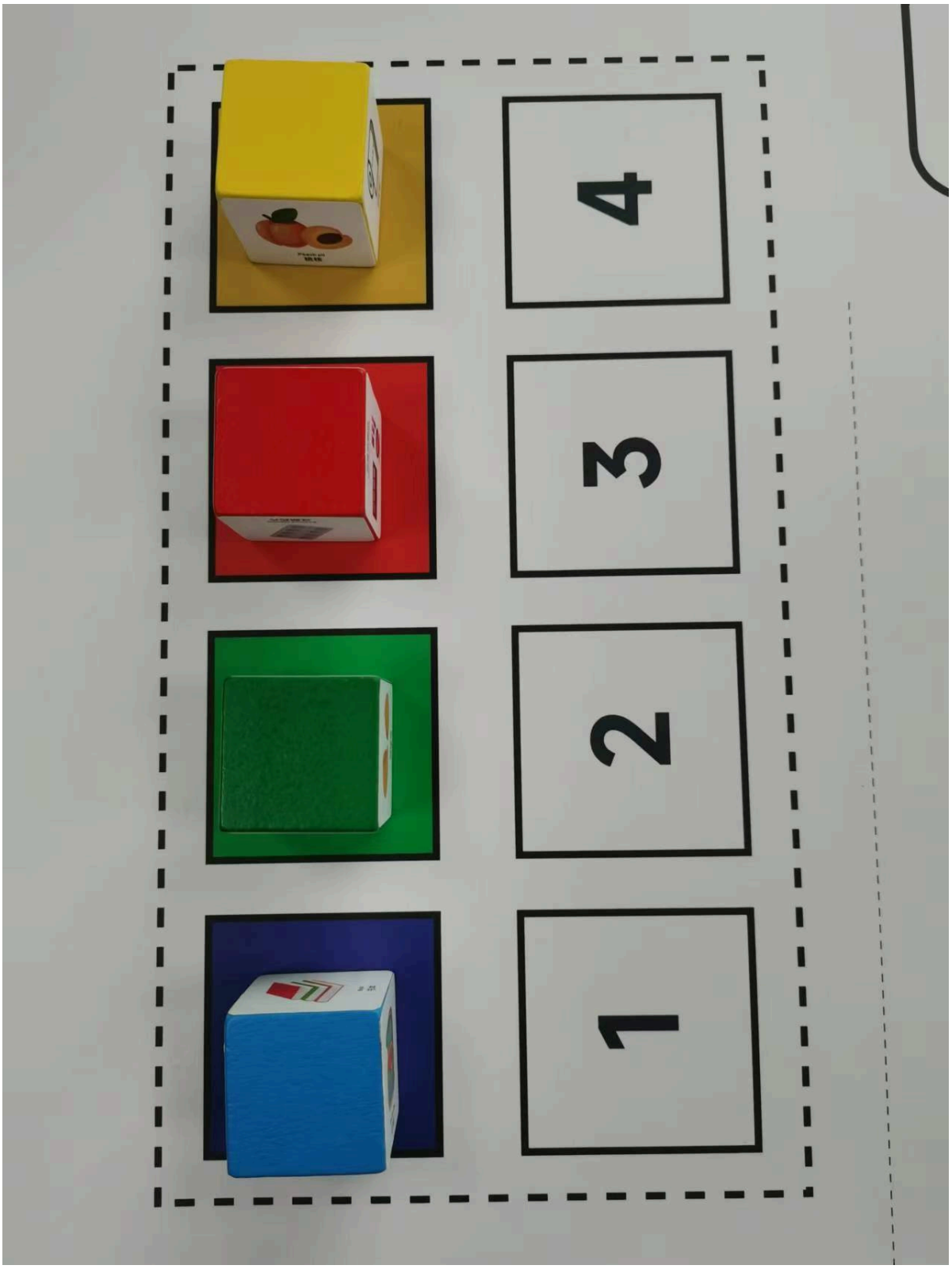
1. Color recognition instructions

The color recognition grab blocks use the HSV color recognition function. The path where the HSV color calibration file is saved is ~/jetcobot_ws/src/jetcobot_color_identify/scripts/HSV_config.txt. If the color recognition is not accurate enough, please recalibrate the HSV value of the block color according to the [Color Threshold Adjustment Color Block Calibration] course. After the calibration operation is completed, it will be automatically saved to the HSV_config file. Rerun the program without additional code modification.

2. Experimental placement

Place the blocks in the four color areas of red, green, blue and yellow. The yellow block is placed in the Y area, the red block is placed in the R area, the green block is placed in the G area, and the blue block is placed in the B area.


B
G
R
Y
1
2
3
4



3. Important code explanation

Code path: ~/jetcobot_ws/src/jetcobot_grasp/jetcobot_grasp/2_color_recognition_grasp.py

~/jetcobot_ws/src/jetcobot_grasp/jetcobot_grasp/grasp_controller.py

Control the movement and grasping function of the robot arm.

```
def grasp_run(self, color_name):
        self.graspController.ctrl_nod()
        if color_name == 'yellow':
            self.graspController.goYellowfixedPose()
```

```
        elif color_name == 'red':
            self.graspController.goRedfixedPose()
        elif color_name == 'green':
            self.graspController.goGreenfixedPose()
        elif color_name == 'blue':
            self.graspController.goBluefixedPose()
        else:
            return
        self.graspController.drop_gripper(1.5)
        self.graspController.close_gripper(1)
        self.graspController.goColorOverPose()
        self.graspController.goBoxCenterlayer1Pose()
        self.graspController.open_gripper(1)
        self.graspController.rise_gripper(1)
        self.graspController.init_pose()
        self.status = 'waiting'
```

The position coordinates corresponding to the color area. If the clamping position coordinates are inaccurate, you can modify this coordinate value appropriately.

```
# Color fixed point grab position
    def goYellowfixedPose(self):
        coords = [140, 250, 160, -175, 0, -45]
        self.go_coords(coords, 3)

    def goRedfixedPose(self):
        coords = [75, 250, 160, -175, 0, -45]
        self.go_coords(coords, 3)

    def goGreenfixedPose(self):
        coords = [10, 250, 160, -175, 0, -45]
        self.go_coords(coords, 3)

    def goBluefixedPose(self):
        coords = [-60, 250, 160, -175, 0, -45]
        self.go_coords(coords, 3)
```

The coordinate value of the placement position. If the placement position coordinate is inaccurate, you can modify this coordinate value appropriately.

```
# The first layer of the box center
    def goBoxCenterlayer1Pose(self):
        coords = [220, 0, 120, -175, 0, -45]
        self.go_coords(coords, 3)
```

4. Start the program

Start the program

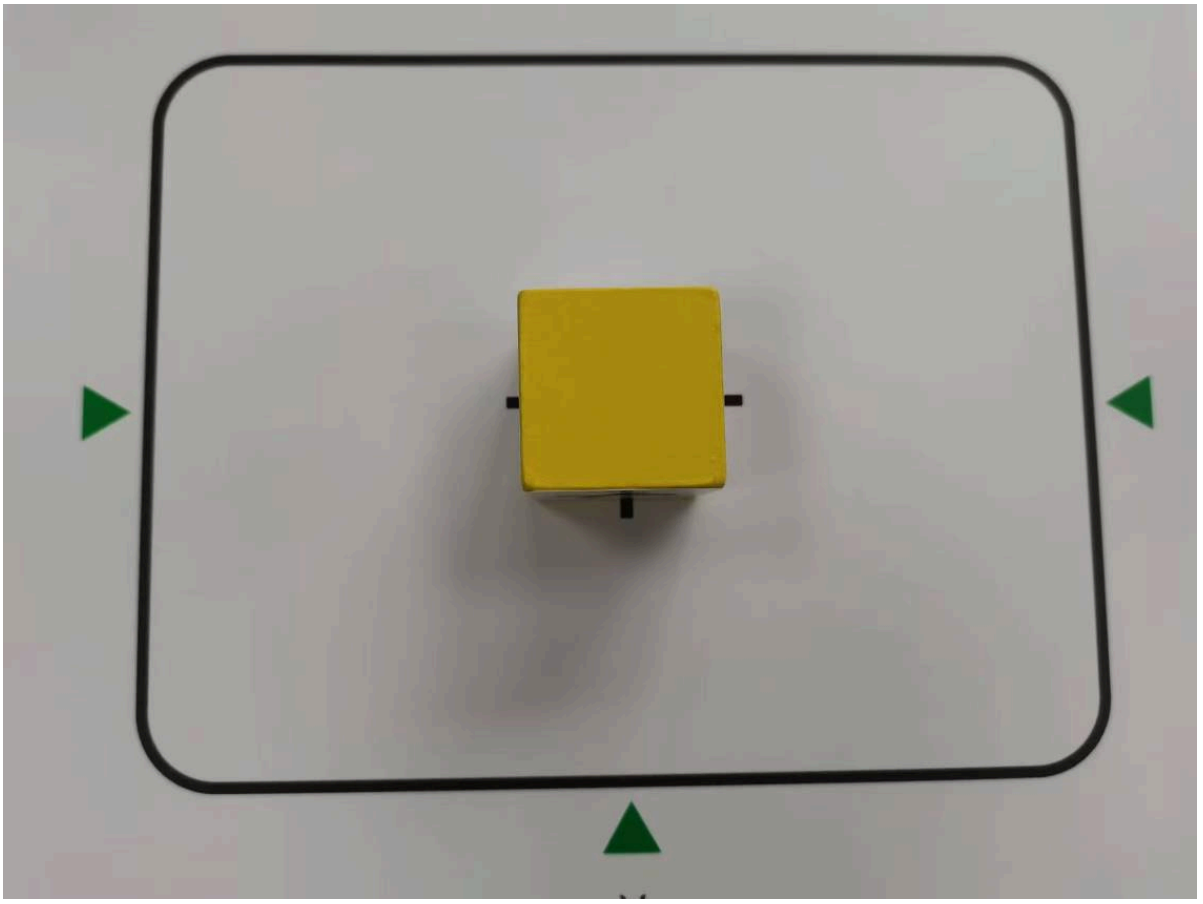
Reopen a terminal and enter the following command.

```
ros2 run jetcobot_grasp 2_color_recognition_grasp
```

5. Experimental results

After the program runs, the robot arm will grab the building blocks in the corresponding color according to the recognized color and put them in the middle area.

For example, if yellow is recognized, it will grab the building blocks in the yellow position, put them in the middle, and then restore the initial position.



Before the next color recognition, you need to remove the block that was clamped in the middle last time, otherwise there will be a conflict.