# Catch game

1、 Gesture recognition instructions

The color recognition grabbing building 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 building block color HSV value 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、 Map placement

Place the block you want to grab onto the cross in the recognition area, with the colored side facing up.
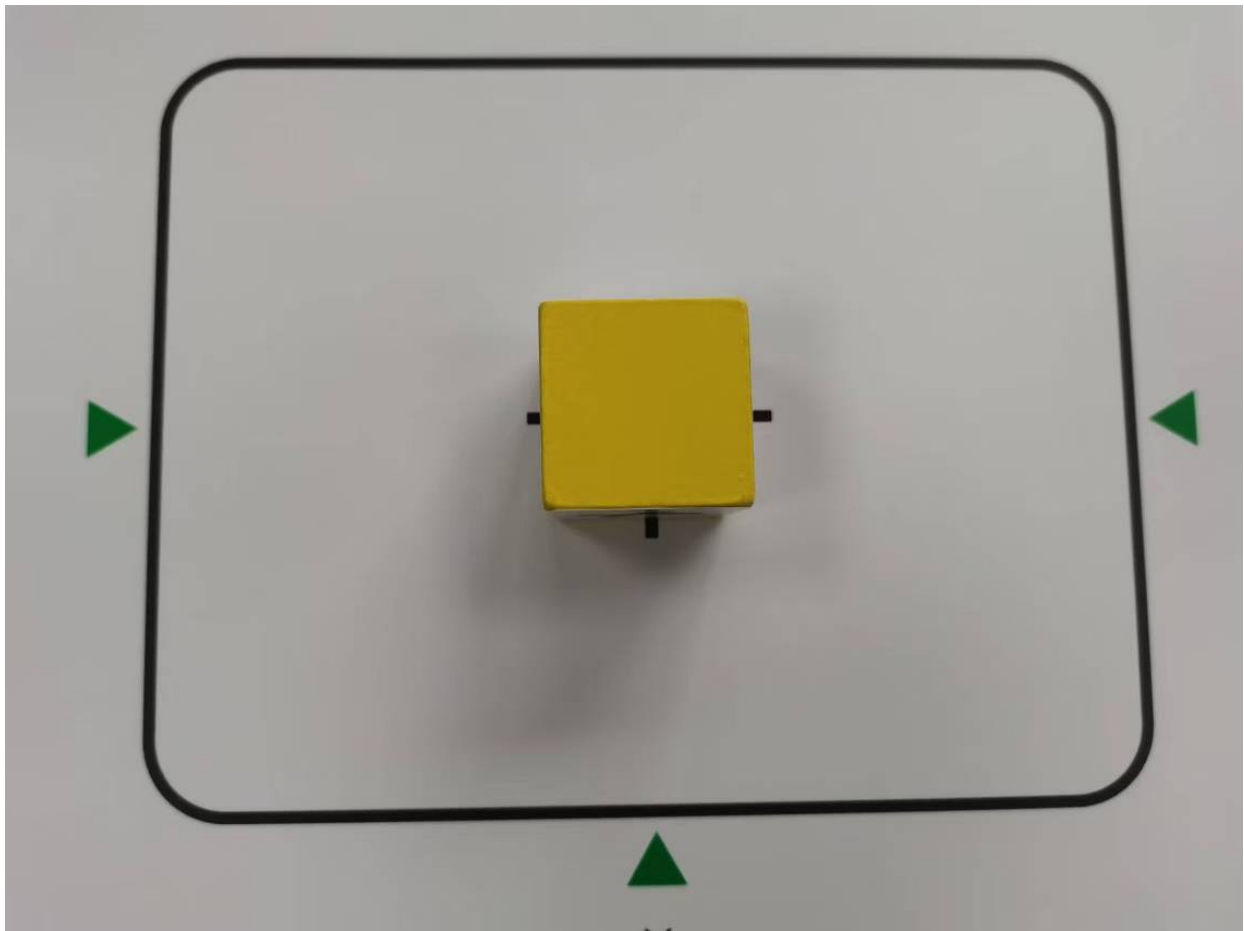
3、 About code

Code path：~/jetcobot_ws/src/jetcobot_grasp/scripts/4_put_and_grasp.py

~/jetcobot_ws/src/jetcobot_utils/src/jetcobot_utils/grasp_controller.py

Control the movement and grasping functions of the robotic arm.

```python
def grasp_run(self, color_name):
        self.graspController.goBoxCenterlayer1Pose()
        self.graspController.close_gripper(1.5)
        self.graspController.goColorOverPose()

        if color_name == 'yellow':
            self.graspController.goYellowPose()
        elif color_name == 'red':
            self.graspController.goRedPose()
        elif color_name == 'green':
            self.graspController.goGreenPose()
        elif color_name == 'blue':
            self.graspController.goBluePose()
        else:
            self.graspController.init_watch_pose()
            self.status = 'waiting'
            return
        self.graspController.open_gripper(1)
        self.graspController.rise_gripper(1)
        self.graspController.init_watch_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.

```python
# First layer of the center of the box
    def goBoxCenterlayer1Pose(self):
        coords = [220, 0, 120, -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.

```python
# color
    def goBluePose(self):
        coords = [-60, 230, 110, -175, 0, -45]
        self.go_coords(coords, 3)

    def goGreenPose(self):
        coords = [10, 230, 110, -175, 0, -45]
        self.go_coords(coords, 3)

    def goRedPose(self):
        coords = [75, 230, 110, -175, 0, -45]
```

```
        self.go_coords(coords, 3)

    def goYellowPose(self):
        coords = [140, 230, 110, -175, 0, -45]
        self.go_coords(coords, 3)
```

4、Start program

Start roscore

Open the system terminal and enter the following command. If roscore is already started, you do not need to start it again.

```
roscore
```

Start the program

Reopen a terminal and enter the following command.

```
rosrun jetcobot_grasp 4_put_and_grasp.py
```

5. Experimental results

After the program runs, the robot arm will grab the building block on the cross in the middle of the recognition area according to the recognized color, and then place it in the position of the corresponding color.

For example, if yellow is recognized, it will grab the building block on the middle cross, place it in the yellow area, and then restore the initial posture.
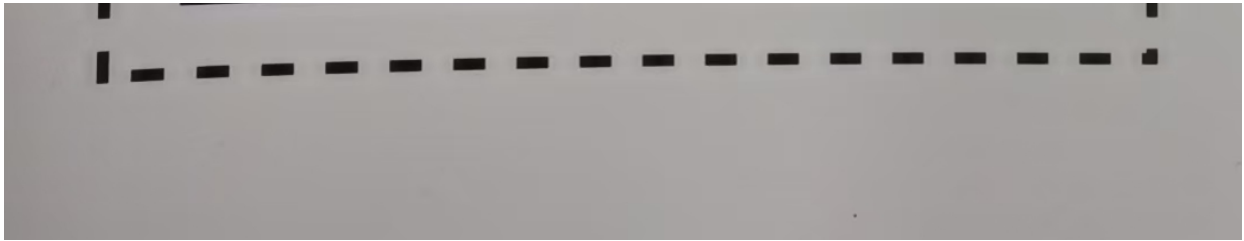
4

R

3

G

2

B

1

Before the next color recognition, you need to remove the building blocks in the color area to avoid conflicts when placing them.