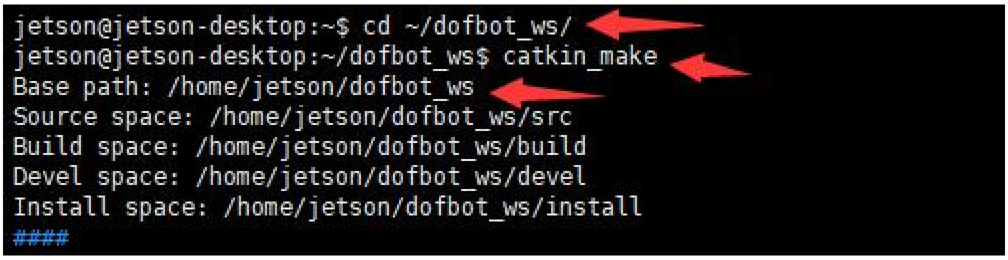
Path: dofbot_ws/src/dofbot_snake_follow/scripts

Before starting catch game function, you must start the reverse solution server and keep running.

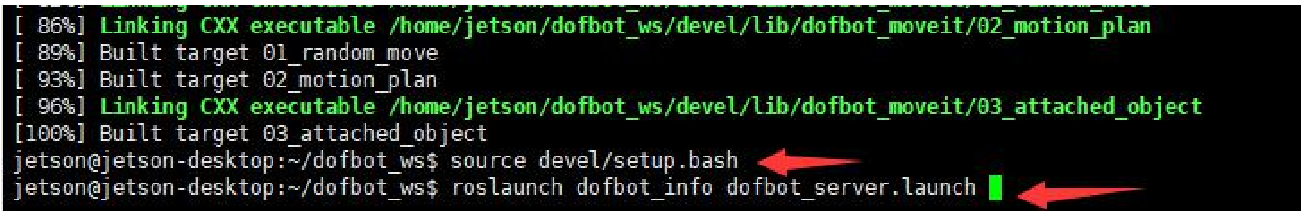
Input following command:

```
cd ~/dofbot_ws/    # Entering Workspace
catkin_make      # compile
source devel/setup.bash    # Update the system environment
roslaunch dofbot_info dofbot_server.launch    # Start the server terminal node
```

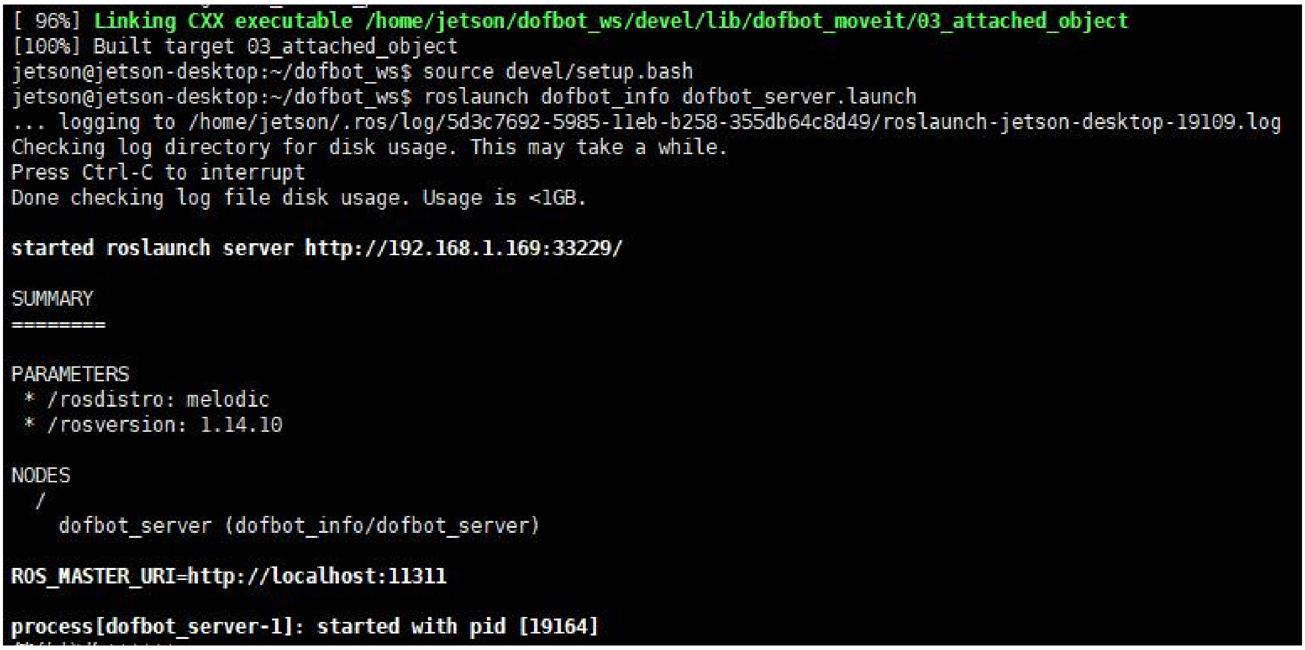
As shown below.



```
jetson@jetson-desktop:~$ cd ~/dofbot_ws/
jetson@jetson-desktop:~/dofbot_ws$ catkin_make
Base path: /home/jetson/dofbot_ws
Source space: /home/jetson/dofbot_ws/src
Build space: /home/jetson/dofbot_ws/build
Devel space: /home/jetson/dofbot_ws/devel
Install space: /home/jetson/dofbot_ws/install
####
```



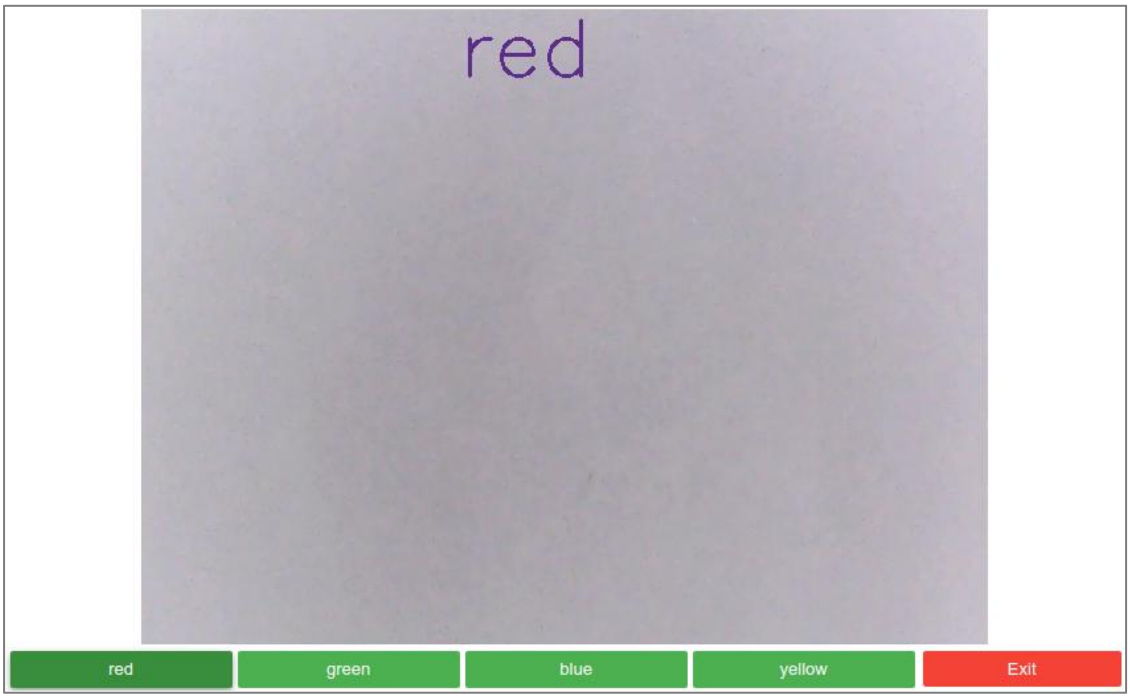
```
[ 86%] Linking CXX executable /home/jetson/dofbot_ws/devel/lib/dofbot_moveit/02_motion_plan
[ 89%] Built target 01_random_move
[ 93%] Built target 02_motion_plan
[ 96%] Linking CXX executable /home/jetson/dofbot_ws/devel/lib/dofbot_moveit/03_attached_object
[100%] Built target 03_attached_object
jetson@jetson-desktop:~/dofbot_ws$ source devel/setup.bash
jetson@jetson-desktop:~/dofbot_ws$ roslaunch dofbot_info dofbot_server.launch
```



```
[ 96%] Linking CXX executable /home/jetson/dofbot_ws/devel/lib/dofbot_moveit/03_attached_object
[100%] Built target 03_attached_object
jetson@jetson-desktop:~/dofbot_ws$ source devel/setup.bash
jetson@jetson-desktop:~/dofbot_ws$ roslaunch dofbot_info dofbot_server.launch
... logging to /home/jetson/.ros/log/5d3c7692-5985-11eb-b258-355db64c8d49/roslaunch-jetson-desktop-19109.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://192.168.1.169:33229/

SUMMARY
========

PARAMETERS
 * /rosdistro: melodic
 * /rosversion: 1.14.10

NODES
  /
    dofbot_server (dofbot_info/dofbot_server)

ROS_MASTER_URI=http://localhost:11311

process[dofbot_server-1]: started with pid [19164]
```

1. Experimental phenomena

1.1 After running the program, you can see the interface as shown below. Click the color button below to select a color, and the video screen will display the color you selected.



1.2 When the color(you choose) block appears in the video, the robotic arm will calculate the position of the block.

1.3 When the area of the block in the video becomes smaller, the robotic arm will drive forward until the front end completes the grasping action. When the area of the block in the video becomes larger, the robotic arm will move back and shake its head.

2. About code

2.1 Using polygonal approximation method to obtain the area of color objects in the field of view

```
for i, cnt in enumerate(contours):

    mm = cv.moments(cnt)
    if mm['m00'] == 0:
    continue
    cx = mm['m10'] / mm['m00']
    cy = mm['m01'] / mm['m00']
    area = cv.contourArea(cnt)
```

2.2 Get the current position and posture through positive solutions

```
def get_Posture(self):
    self.read_joint()
    self.client.wait_for_service()
    request = kinemaricsRequest()
    ... ...
```

2.3 Estimate location based on area of block in video

```
distance = 27.05 * math.pow(area, -0.51) - 0.2
```

```
target_dist = distance + self.Posture[1]
```

2.4 Find the angle that each servo needs to turn

```
def snake_run(self, point_y):
        pass
```

2.5 Inverse design

```
///////////////    Catch game    ///////////////
if (request.tar_z >= 0.2) {
        x=request.tar_x;
        y=request.tar_y;
        z=request.tar_z;
        Roll= -90;
}
```

More detail: dofbot_ws/src/dofbot_info/src/dofbot_server.cpp