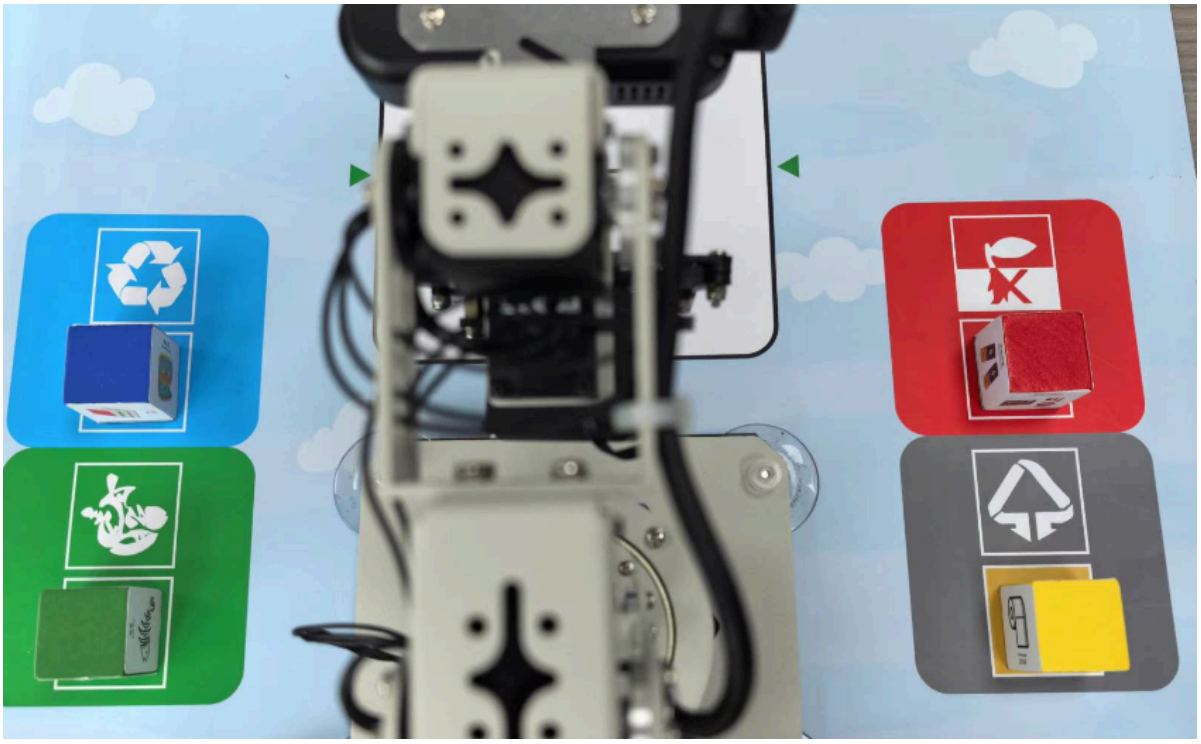
Stacking Humans

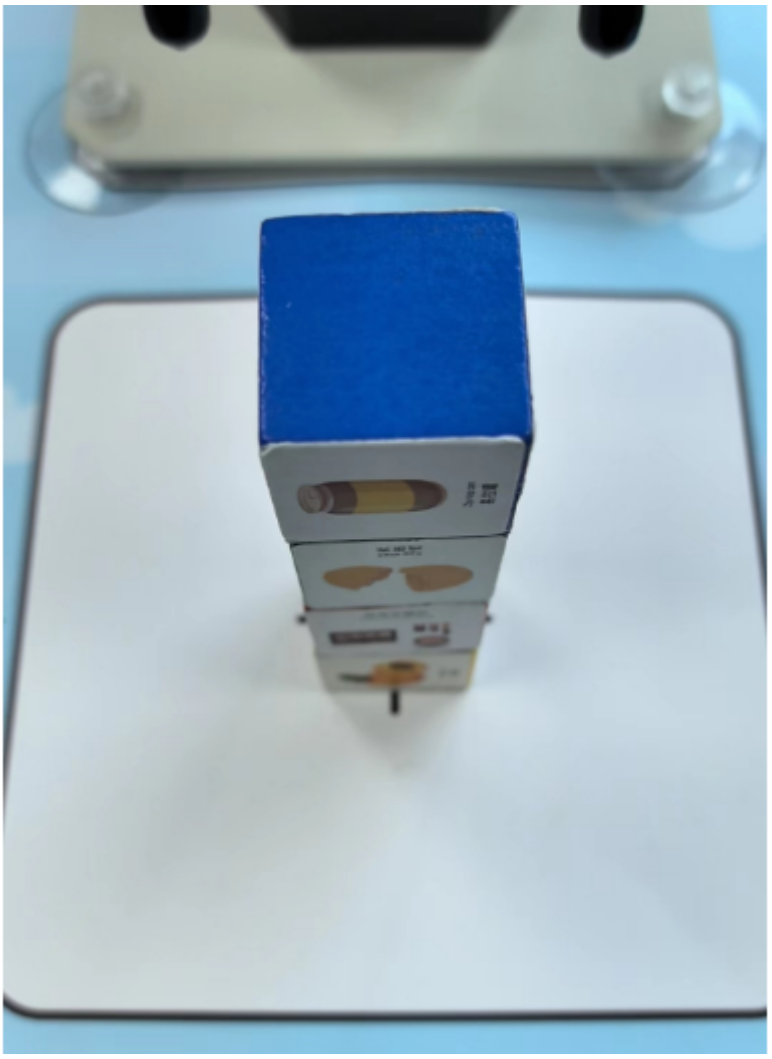
1. Introduction to the game

The purpose of this experiment is exactly the opposite of the last lesson's "Nature Porter". It is to pick up the building blocks from different sides in the order of yellow, red, green, and blue and stack them in the middle cross area.

The way to place the building blocks is as shown in the figure below:



After executing the code, the robot arm will stack the building blocks, and the final effect is as shown in the figure below:



2. Code content

Code path:

```
~/dofbot_pro/dofbot_ctrl/scripts/11.heap_up.ipynb
```

```
#!/usr/bin/env python3
#coding=utf-8
import time
from Arm_Lib import Arm_Device
from dofbot_utils.robot_controller import Robot_Controller

# Create a robot object
Arm = Arm_Device()
time.sleep(.1)
robot = Robot_Controller()
```

```
# Define the function of clamping blocks, enable=1: clamp, =0: release
def arm_clamp_block(enable):
if enable == 0:
Arm.Arm_serial_servo_write(6, 60, 400)
else:
Arm.Arm_serial_servo_write(6, 135, 400)
time.sleep(.5)
```

```
# Define the function of moving the robot arm, and control the movement of servos
1-5 at the same time, p=[S1,S2,S3,S4,S5]
def arm_move(p, s_time = 500):
for i in range(5):
id = i + 1
if id == 5:
time.sleep(.1)
Arm.Arm_serial_servo_write(id, p[i], int(s_time*1.2))
 elif id == 1 :
 Arm.Arm_serial_servo_write(id, p[i], int(3*s_time/4))

 else:
 Arm.Arm_serial_servo_write(id, p[i], int(s_time))
 time.sleep(.01)
 time.sleep(s_time/1000)
```

```
# Define variable parameters at different locations
p_mould = robot.P_LOOK_AT
p_top = robot.P_TOP
p_layer_4 = robot.P_CENTER_HEAP_L4
p_layer_3 = robot.P_CENTER_HEAP_L3
p_layer_2 = robot.P_CENTER_HEAP_L2
p_layer_1 = robot.P_CENTER_HEAP_L1
p_Yellow = robot.P_YELLOW
p_Red = robot.P_RED
p_Green = robot.P_GREEN
p_Blue = robot.P_BLUE
```

```
# Move the robot arm to a position ready for grabbing
arm_clamp_block(0)
arm_move(p_mould, 1000)
time.sleep(1)
```

```
# Clamp the blocks in the yellow area and stack them to the bottom in the middle.
arm_move(p_top, 1000)
arm_move(p_Yellow, 1000)
arm_clamp_block(1)
Arm.Arm_serial_servo_write(2, 90, 1000)
time.sleep(1)
arm_move(p_top, 1000)
arm_move(p_layer_1, 1000)
arm_clamp_block(0)
time.sleep(.1)
arm_move(p_mould, 1100)
```

```
# Clamp the blocks in the red area and stack them to the second layer in the
middle.
arm_move(p_top, 1000)
arm_move(p_Red, 1000)
arm_clamp_block(1)
Arm.Arm_serial_servo_write(2, 90, 1000)
time.sleep(1)
arm_move(p_top, 1000)
arm_move(p_layer_2, 1000)
arm_clamp_block(0)
time.sleep(.1)
arm_move(p_mould, 1100)
```

```
# Clamp the blocks in the green area and stack them to the third layer in the
middle.
arm_move(p_top, 1000)
arm_move(p_Green, 1000)
arm_clamp_block(1)
Arm.Arm_serial_servo_write(2, 90, 1000)
time.sleep(1)
arm_move(p_top, 1000)
arm_move(p_layer_3, 1000)
arm_clamp_block(0)
time.sleep(.2)
arm_move(p_mould, 1100)
```

```
# Clamp the blocks in the blue area and stack them to the fourth layer in the
middle.
arm_move(p_top, 1000)
arm_move(p_Blue, 1000)
arm_clamp_block(1)
Arm.Arm_serial_servo_write(2, 90, 1000)
time.sleep(1)
arm_move(p_top, 1000)
arm_move(p_layer_4, 1000)
arm_clamp_block(0)
time.sleep(.2)
arm_move(p_mould, 1100)
```

```
del Arm # Release the Arm object
```

Open the program file from jupyter lab and click the Run the entire notebook button on the jupyter lab toolbar to see the effect of the robot arm stacking function.