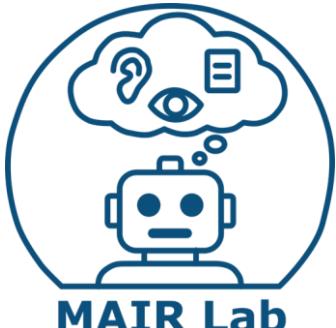


Turtlebot4 LLM

운영체제의 실제
안인규 (Inkyu An)



공유기 설정

- Nav2를 사용하기 위해서는 외부망과 연결된 인터넷이 필수
 1. Turtlebot4에 ssh로 접속
 2. ros2 run turtlebot4_setup turtlebot4_setup
 3. Wifi 설정 (오른쪽 그림 참고, 오타 조심!, 어려울 경우 TA에게 도움을 요청)



- 미래관 429호 (자율주행스튜디오)
 - **SSID:** SW_Turtlebot4_429_3(5G)
 - **PW:** swtest1!

Nav2 – Commander API

```
1 from nav2_simple_commander.robot_navigator import BasicNavigator
2 import rclpy
3
4 rclpy.init()
5 nav = BasicNavigator()
6 # ...
7 nav.setInitialPose(init_pose) → Localization의 초기 위치 지정
8 navigator.waitUntilNav2Active(localizer='controller_server') → Nav2가 online이 되길 기다림
9 # ...
10 path = nav.getPath(init_pose, goal_pose) → 경로 반환
11 smoothed_path = nav.smoothPath(path) → 부드러운 경로 생성
12 # ...
13 nav.goToPose(goal_pose) → 목적지까지 이동 명령
14 while not nav.isTaskComplete(): → Navigation task가 완료될 때 까지 기다림
15     feedback = nav.getFeedback()
16     if feedback.navigation_duration > 600:
17         nav.cancelTask()
18 # ...
19 result = nav.getResult() → 결과 확인
20 if result == TaskResult.SUCCEEDED:
21     print('Goal succeeded!')
22 elif result == TaskResult.CANCELED:
23     print('Goal was canceled!')
24 elif result == TaskResult.FAILED:
25     print('Goal failed!')
26
```

Nav2 – Commander API

```
1 from nav2_simple_commander.robot_navigator import BasicNavigator  
2 import rclpy  
3  
4 rclpy.init()  
5 nav = BasicNavigator()  
6 # ...  
7 nav.setInitialPose(init_pose) → We need to input the destination's (x, y, z)  
8 navigator.waitUntilNav2Active(localizer='controller_server') → coordinates through code  
9 # ...  
10 path = nav.getPath(init_pose, goal_pose) → Localization의 초기 위치 지정  
11 smoothed_path = nav.smoothPath(path) → Nav2가 online이 되길 기다림  
12 # ...  
13 nav.goToPose(goal_pose) → 경로 반환  
14 while not nav.isTaskComplete(): → 부드러운 경로 생성  
15     feedback = nav.getFeedback()  
16     if feedback.navigation_duration > 600:  
17         nav.cancelTask()  
18 # ...  
19 result = nav.getResult() → 목적지까지 이동 명령  
20 if result == TaskResult.SUCCEEDED:  
21     print('Goal succeeded!') → Navigation task가 완료될 때 까지 기다림  
22 elif result == TaskResult.CANCELED:  
23     print('Goal was canceled!') → 결과 확인  
24 elif result == TaskResult.FAILED:  
25     print('Goal failed!')  
26
```

Command using Natural Language

- Isn't it possible to give commands using natural language?
(e.g., Go to the toilet!)
 - In traditional navigation methods, you have to input the exact location of the toilet.



Powered by LLM

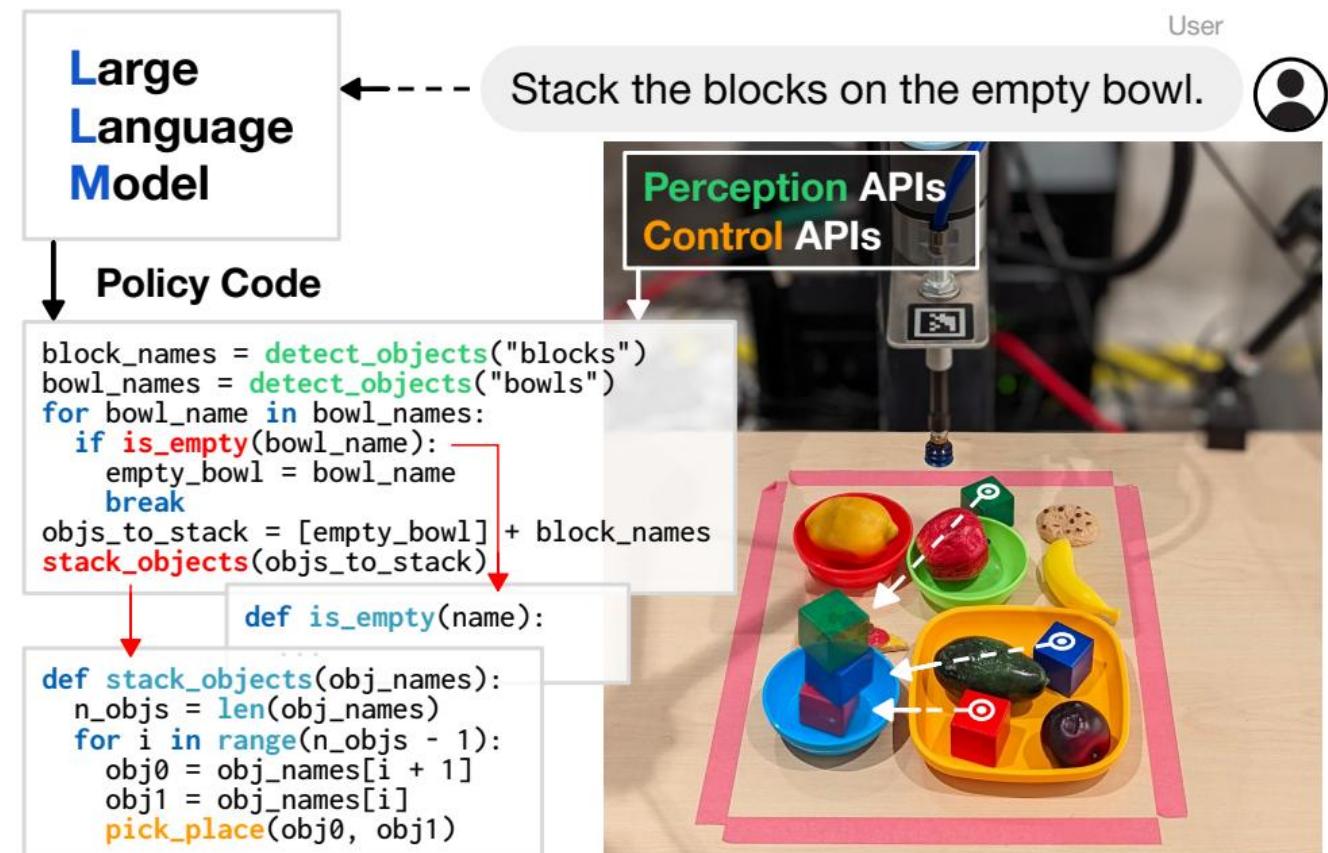
- LLMs are very powerful tools that can understand natural language.
- LLMs are also very powerful tools that can understand computer languages (e.g., Python).

Powered by LLM

- LLMs are very powerful tools that can understand natural language.
- LLMs are also very powerful tools that can understand computer languages (e.g., Python).
- What if an LLM received natural language input and generated Python code to control a robot?
 - Code as Policies: Language Model Programs for Embodied Control, Robotics at Google, arXiv 2022

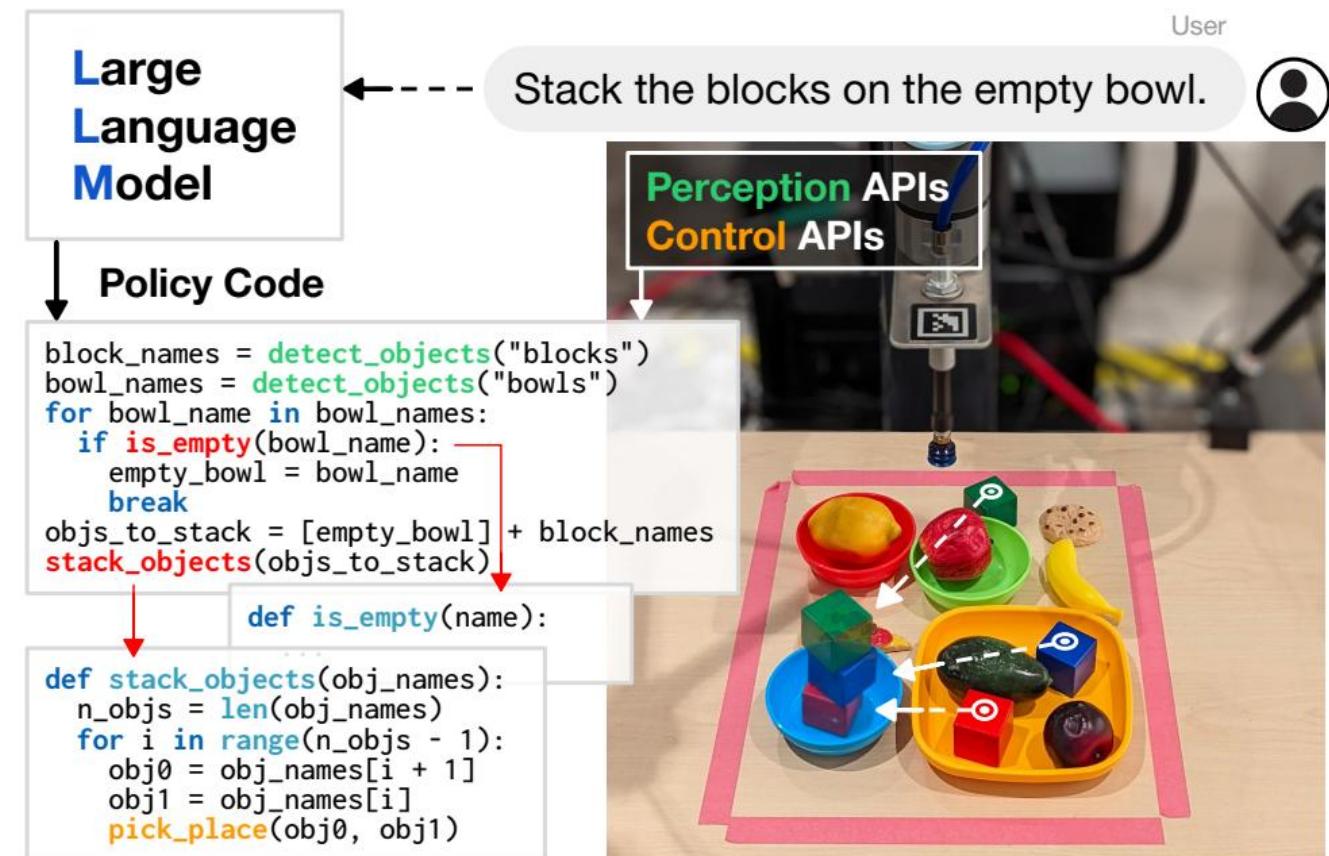
Code as Policies, Google

- Language model generated programs (LMPs)
 - Robots can use code-writing LLMs to translate natural language commands into robot policy code which process **perception** outputs, parameterize **control** primitives, recursively generate code for **undefined** function



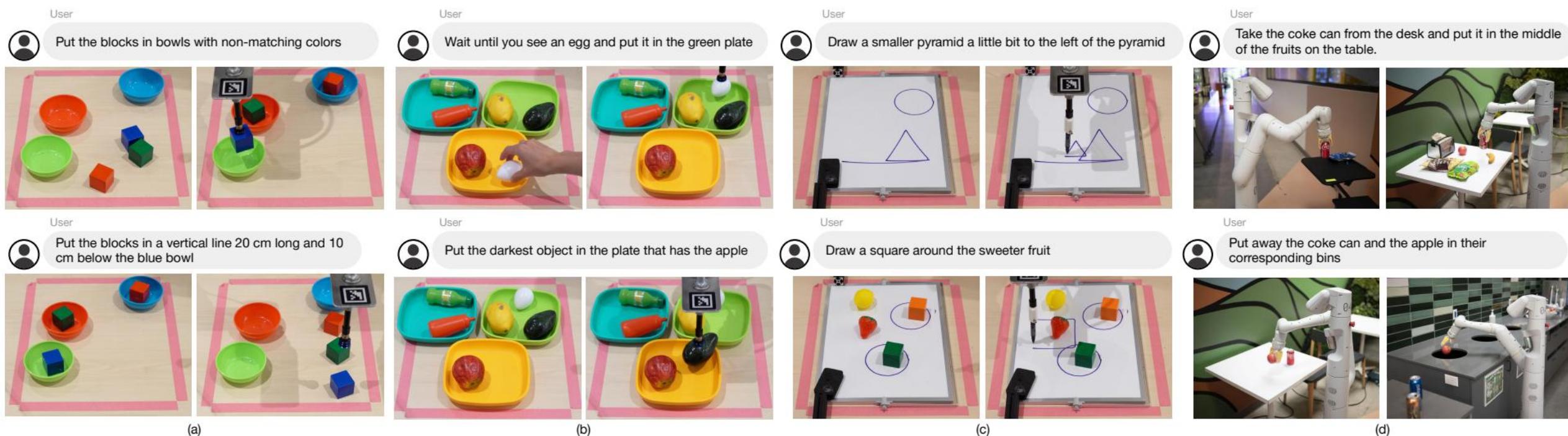
Code as Policies, Google

- Language model generated programs (LMPs)
 - Robots can use code-writing LLMs to translate natural language commands into robot policy code which process **perception** outputs, parameterize **control** primitives, recursively generate code for **undefined** function



→ Only can understand “Text” (=Prompt)!

Code as Policies, Google



- Language model generated programs (LMPs)
 - **Input:** 자연어 명령 (e.g., Put the red block on the green bowl)
 - **Output:** Python code
 - **Prompt:** Hints + Examples

Code as Policies, Google

- Prompt:
 - Hints: import statements that inform the LLM which APIs are available and type hints on how to use those APIs.
 - Example:
 - Input instructions are **green** / Generated Outputs are **highlighted**

```
# Python script
# get the variable a.
ret_val = a
# find the sum of variables a and b.
ret_val = a + b
# see if any number is divisible by 3 in a list called xs.
ret_val = any(x % 3 == 0 for x in xs)
```

Code as Policies, Google

- Prompt:
 - Hints: import statements that inform the LLM which APIs are available and type hints on how to use those APIs.
 - Example:
 - Input instructions are **green** / Generated Outputs are **highlighted**

```
import numpy as np
# move all points in pts_np toward the right.
ret_val = pts_np + [0.3, 0]
# move a pt_np toward the top.
ret_val = pt_np + [0, 0.3]
# get the left most point in pts_np.
ret_val = pts_np[np.argmin(pts_np[:, 0]), :]
# get the center of pts_np.
ret_val = np.mean(pts_np, axis=0)
# the closest point in pts_np to pt_np.
ret_val = pts_np[np.argmin(np.sum((pts_np - pt_np)**2, axis=1))]
```

Code-writing LLMs store knowledge of many popular libraries

Code as Policies, Google

- Prompt:
 - Hints: import statements that inform the LLM which APIs are available and type hints on how to use those APIs.
 - Example:
 - Input instructions are **green** / Generated Outputs are **highlighted**

```
objs = ['blue bowl', 'red block', 'red bowl', 'blue block']
# the bowls.
ret_val = ['blue bowl', 'red bowl']
# sea-colored block.
ret_val = 'blue block'
# the other block.
ret_val = 'red block'
```

Language reasoning

Code as Policies, Google

- Prompt:

- Hints: import statements that inform the LLM which APIs are available and type hints on how to use those APIs.
- Example:
 - Input instructions are green / Generated Outputs are highlighted

```
import numpy as np
from utils import get_pos, put_first_on_second
objs = ['cyan block', 'cyan bowl', 'pink bowl']
# put the cyan block in cyan bowl.
put_first_on_second('cyan block', 'cyan bowl')
objs = ['gray block', 'silver block', 'gray bowl']
# place the top most block on the gray bowl.
names = ['gray block', 'silver block']
positions = np.array([get_pos(name) for name in names])
name = names[np.argmax(positions[:,1])]
put_first_on_second(name, 'gray bowl')
objs = ['purple block', 'purple bowl']
# put the purple bowl to the left of the purple block.
target_pos = get_pos('purple block') + [-0.3, 0]
put_first_on_second('purple bowl', target_pos)
```

Full prompt

Code as Policies, Google

- Prompt:
 - Hints: import statements that inform the LLM which APIs are available and type hints on how to use those APIs.
 - Example:
 - Input instructions are **green** / Generated Outputs are **highlighted**

```
objs = ['red block', 'blue bowl', 'blue block', 'red bowl']
# blocks with area bigger than 0.2 that are left of the red bowl.
block_names = ['red block', 'blue block']
red_bowl_pos = get_pos('red bowl')
use_block_names = [name for name in block_names
                   if get_pos(name)[0] < red_bowl_pos[0]]
use_block_names = get_objs_bigger_than_area_th(use_block_names, 0.2)
ret_val = use_block_names      Undefined function
```

Code as Policies, Google

- Prompt:
 - Identify any undefined functions in the output generated by the LMP, and perform functions

```
import numpy as np
from utils import get_obj_bbox_xyxy
# define function: total = get_total(xs).
def get_total(xs):
    return np.sum(xs)
# define function: get_objs_bigger_than_area_th(obj_names, bbox_area_th).
def get_objs_bigger_than_area_th(obj_names, bbox_area_th):
    return [name for name in obj_names
            if get_obj_bbox_area(name) > bbox_area_th]
objs = ['red bowl', 'blue cube', 'green cube', 'yellow cube', 'purple cube']
# blocks with area > 0.2
block_names = [name for name in objs
               if get_obj_bbox_area(name) > 0.2]
red_bowl_pos = get_pos('red bowl')
use_block_names = [name for name in block_names
                   if get_pos(name)[0] < red_bowl_pos[0]]
use_block_names = get_objs_bigger_than_area_th(use_block_names, 0.2)
ret_val = use_block_names
```

Code as Policies, Google

- Prompt:

- Identify all functions and perform functions

```
objs = ['red bowl',  
# blocks with area  
block_names = ['red_bowl_pos = get_pos()  
use_block_names = [name for name in block_names  
                    if get_pos(name)[0] < red_bowl_pos[0]]  
use_block_names = get_objs_bigger_than_area_th(use_block_names, 0.2)  
ret_val = use_block_names
```

```
# define function: get_obj_bbox_area(obj_name).  
def get_obj_bbox_area(obj_name):  
    x1, y1, x2, y2 = get_obj_bbox_xyxy(obj_name)  
    return (x2 - x1) * (y2 - y1)
```

```
def get_total(xs):  
    return np.sum(xs)
```

```
# define function: get_objs_bigger_than_area_th(obj_names, bbox_area_th).  
def get_objs_bigger_than_area_th(obj_names, bbox_area_th):  
    return [name for name in obj_names  
            if get_obj_bbox_area(name) > bbox_area_th]
```

Turtlebot w/LLM (OpenAI)

- Installation:
 - Create an OpenAI Account (<https://auth.openai.com/create-account>) and API key (<https://platform.openai.com/account/api-keys>)
 - Verify that your OpenAI account has some credit (<https://platform.openai.com/usage>)
 - Install the OpenAI Python library via *pip install openai* (It is recommended to utilize virtual environments...)
 - Clone the code into your workspace (https://github.com/turtlebot/turtlebot4_tutorials)
 - Build it!

Turtlebot w/LLM (OpenAI)

- To run the example, first start the Gazebo simulation, specifying the 'depot' world:

```
$ ros2 launch turtlebot4_gz_bringup turtlebot4_gz.launch.py nav2:=true slam:=false localization:=true  
rviz:=true world:=depot map:=/opt/ros/jazzy/share/turtlebot4_navigation/maps/depot.yaml
```

- Open another terminal and run:

```
$ ros2 launch turtlebot4_openai_tutorials natural_language_nav_launch.py openai_api_key:=API_KEY  
parking_brake:=false
```

- Once the robot is undocked, open a third terminal and run:

```
$ ros2 topic pub --once /user_input std_msgs/msg/String "data: Dock"  
$ ros2 topic pub --once /user_input std_msgs/msg/String "data: Go to -1,0, face East"  
$ ros2 topic pub --once /user_input std_msgs/msg/String "data: Go to 5,5"  
$ ros2 topic pub --once /user_input std_msgs/msg/String "data: Move to the wooden object"  
$ ros2 topic pub --once /user_input std_msgs/msg/String "data: Navigate to the item which can hold oil"  
$ ros2 topic pub --once /user_input std_msgs/msg/String "data: Travel to the room containing a toilet"
```