

FUN3: เต้าหัดเดิน (5 คะแนน)

ใน FUN 2 นักศึกษาได้มีการสร้าง node สำหรับการกิน pizza 3 node ซึ่งแต่ละ node มีหน้าที่ดังนี้

1. **controller node** – ควบคุมตำแหน่งของ /turtle1 ให้สามารถไปกิน pizza จากตำแหน่งที่กำหนดไว้ด้วย turtlesim+ GUI หรือ /goal_pose ของ rviz2
2. **crazy_turtle node** - สร้างและควบคุมตำแหน่งของ /turtle2 ให้สามารถไปกิน pizza จากตำแหน่งที่ได้มาจากการสุ่มของ crazy_pizza.py
3. **odom_publisher node** - ทำการ publisher /tf และ /odom ของ /turtle1 และ /turtle2

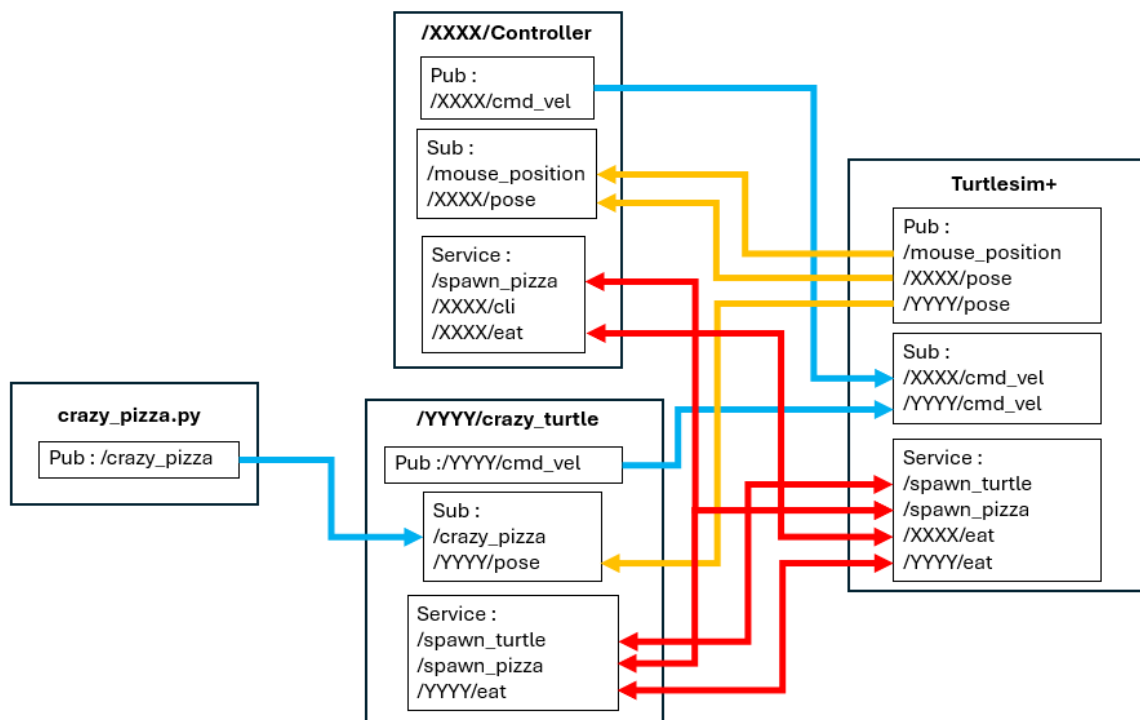
หมายเหตุ: ใน FUN 3 จะไม่มีการใช้ odom_publisher โดยจะแสดงผลและสั่งคำสั่งผ่าน GUI เท่านั้น

ภารกิจ

1. การสร้าง service server และ ros2 parameter (1.3 คะแนน)
 1. สร้าง package สำหรับ custom service ที่มีชื่อว่า controller_interfaces (0.1 คะแนน)
 2. โดยใน controller_interfaces สร้าง custom service 2 ตัวได้แก่
 - SetTarget – service นี้จะใช้เพื่อกำหนด target ให้กับเต้าหัดเพื่อใช้ควบคุมไปกับ /mouse_position โดย request จะมีตัวแปร 1 ตัวที่มีชื่อว่า target เป็น msg ชนิด geometry_msgs/Point (0.1 คะแนน)
 - SetParam – service นี้จะใช้เพื่อกำหนด controller gain ให้กับ controller โดย request จะมีตัวแปร 2 ตัวที่มีชื่อว่า kp_linear และ kp_angular ทั้ง 2 ตัวเป็น msg ชนิด std_msgs/Float64 (0.1 คะแนน)
 - สามารถยืนยันความสำเร็จในการสร้าง custom service ทั้ง 2 ผ่าน CLI (0.2 คะแนน)
 3. ใช้ ros2 parameter สำหรับการปรับความถี่ control loop ของ controller node โดยกำหนดให้ parameter มีชื่อว่า frequency โดยให้กำหนดค่าเริ่มต้นเป็น 10 Hz (0.1 คะแนน)
 4. ใช้ ros2 parameter สำหรับการปรับความถี่ control loop ของ crazy_turtle node โดยกำหนดให้ parameter มีชื่อว่า frequency โดยให้กำหนดค่าเริ่มต้นเป็น 10 Hz (0.1 คะแนน)
 5. สามารถกำหนด target ให้กับ controller node โดยใช้ SetTarget service ผ่าน CLI ควบคู่ไปกับการกำหนด target ผ่าน /mouse_position โดยกำหนดให้ service ชื่อ /cli (0.2 คะแนน)
 6. สามารถกำหนด controller gain ให้กับ controller node โดยใช้ SetParam service ผ่าน CLI (0.2 คะแนน)
 7. สามารถกำหนด controller gain ให้กับ crazy_turtle node โดยใช้ SetParam service ผ่าน CLI (0.2 คะแนน)
2. การสร้าง basic launch file (1.2 คะแนน)
 1. สร้าง folder ที่มีชื่อว่า launch ใน package turtle_bringup (0.1 คะแนน)

2. สร้าง file turtle_bringup.launch.py ใน folder launch ที่สามารถ run node ได้ดังนี้
 - สามารถ run turtlesim+ node ได้ (0.2 คะแนน)
 - สามารถ run controller node ได้ (0.2 คะแนน)
 - สามารถ run crazy_pizza node ได้ (0.2 คะแนน)
 - สามารถ run crazy_turtle node ได้ (0.2 คะแนน)
3. กำหนด frequency ของ controller node โดยใช้ ros2 parameter ผ่าน launch file โดยกำหนดให้ frequency มีค่าเท่ากับ 100 Hz (0.15 คะแนน)
4. กำหนด frequency ของ crazy_turtle node โดยใช้ ros2 parameter ผ่าน launch file โดยกำหนดให้ frequency มีค่าเท่ากับ 100 Hz (0.15 คะแนน)
3. การจัดการ project ด้วย launch file และ namespace (2.5 คะแนน)

โครงสร้างของ project



1. ให้ทำการแก้ไข service และ topic ของ controller node ให้สามารถกำหนด namespace ได้ตามโครงสร้างของ project ที่กำหนด โดย XXXX จะเป็นชื่อของ namespace (1.0 คะแนน)
หมายเหตุ: ชื่อของ namespace จะถูกกำหนดไว้ในตอนที่ทำการตรวจ ดังนั้นนักศึกษาสามารถกำหนด namespace เองได้ในระหว่างการทำแบบฝึกหัด
2. ให้ทำการแก้ไข service และ topic ของ crazy_turtle node ให้สามารถกำหนด namespace ได้ตามโครงสร้างของ project ที่กำหนด โดย YYYY จะเป็นชื่อของ namespace (1.0 คะแนน)
หมายเหตุ: ชื่อของ namespace จะถูกกำหนดไว้ในตอนที่ทำการตรวจ ดังนั้นนักศึกษาสามารถกำหนด namespace เองได้ในระหว่างการทำแบบฝึกหัด

3. ให้ทำการ kill เต่าที่ชื่อ /turtle1 ผ่าน launch file (0.1 คะแนน)
4. ให้ทำการ spawn เต่าที่ชื่อ /XXXX ผ่าน launch file โดย XXXX จะเป็นชื่อของ namespace (0.1 คะแนน)
5. ให้ทำการ spawn เต่าที่ชื่อ /YYYY ผ่าน launch file โดย XXXX จะเป็นชื่อของ namespace (0.1 คะแนน)
6. ให้ทำการกำหนด namespace ให้กับ controller node ผ่าน launch file ตามโครงสร้างของ project ที่กำหนด (0.1 คะแนน)
7. ให้ทำการกำหนด namespace ให้กับ crazy_turtle node ผ่าน launch file ตามโครงสร้างของ project ที่กำหนด (0.1 คะแนน)

หมายเหตุ:

- นักศึกษามีเวลาในการทำแบบฝึกหัดนี้สูงสุด 1 ชั่วโมง 30 นาที
- นักศึกษาสามารถใช้ internet ได้แต่ไม่อนุญาตให้ใช้ generative AI
- ในระหว่างทำแบบฝึกหัดไม่อนุญาตให้ถามถามหรือคุยกับเพื่อนได้ในทุกช่องทาง
- หากทำผิดเงื่อนไขดังกล่าว**จะถือว่าเป็นการทุจริต**คะแนนรอบนี้จะถูกปรับเป็น 0 ทันที
- อนุญาตให้สอบถาม TA ที่อยู่ภายในห้องได้