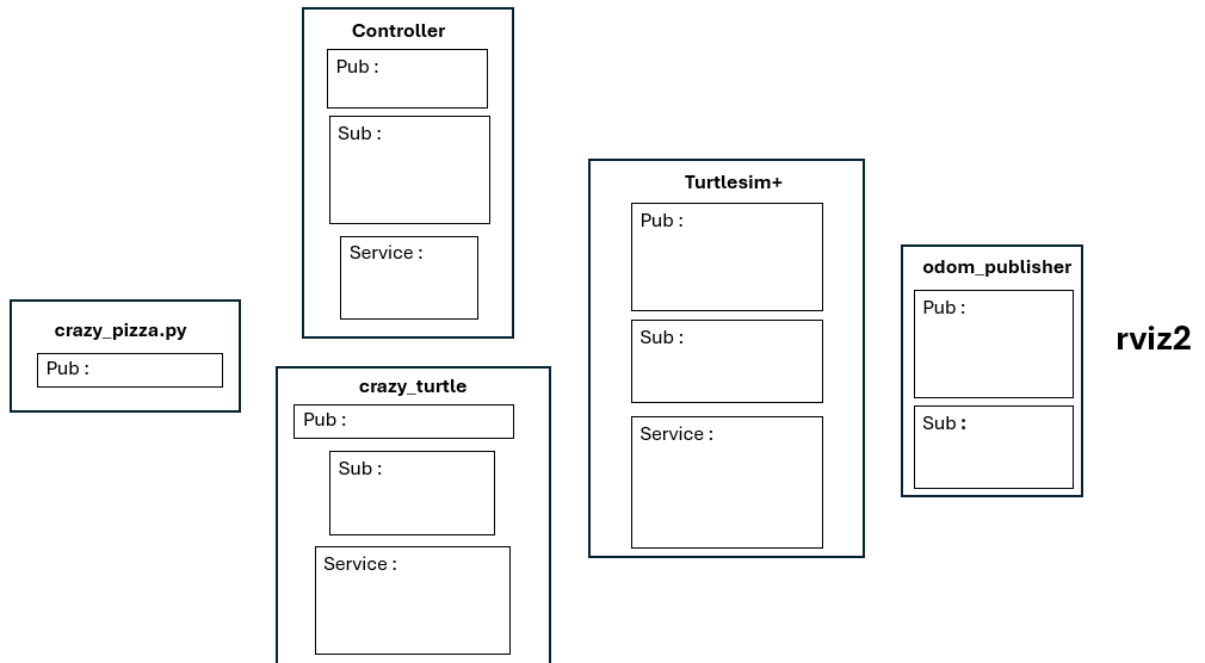


FUN2: เค้าตัวมเตี้ยม (5 คะแนน)

ภารกิจ

1. เติมแผนภาพและโยงเส้นการส่งข้อมูลเพื่ออธิบายระบบให้สมบูรณ์ (0.5 คะแนน)



2. สร้าง package ที่มีชื่อว่า turtle_bringup (0.5 คะแนน)
3. ภายใน turtle_bringup จะประกอบด้วย 3 node ซึ่งแต่ละ node มีหน้าที่ดังนี้
 1. controller (1.4 คะแนน)

node นี้มีหน้าที่ควบคุม turtle1 ให้ไปยังจุดที่กำหนดได้

 - Node สามารถ spawn pizza บนจุดที่คลิกใน GUI (0.2 คะแนน)
 - Node สามารถ spawn pizza บนจุดที่คลิกใน RVIZ2 (0.2 คะแนน)
 - Node สามารถควบคุม turtle1 ไปยังบริเวณจุดที่ spawn pizza และกิน pizza ภายใน 3 วินาทีโดยที่เมื่อ turtle1 ยังกินของเก่าไม่เสร็จจะไม่สามารถสั่งให้ไปกินอันถัดไปได้ (0.8 คะแนน)

หมายเหตุ: หากเค้าสามารถไปกิน pizza ได้แต่ผัดเผ็ดเนื้อไข่อัดหรือเกิดปัญหาขึ้นระหว่างการไปกิน pizza ข้อนี้จะได้ 0.2 คะแนน เท่านั้น

 - สามารถ run node นี้ผ่าน terminal โดยใช้คำสั่งของ ros2 (0.2 คะแนน)
 2. crazy_turtle (1.2 คะแนน)

node นี้มีหน้าที่ควบคุม turtle2 ให้ไปในจุดที่ได้จากการสุ่มจาก crazy_pizza.py

 - Node สามารถ spawn เค้าที่มีชื่อว่า turtle2 ได้ (0.2 คะแนน)

- Node สามารถรับค่าจาก topic ที่ได้รับจาก crazy_pizza.py ได้และใช้ค่านั้นในการ spawn pizza ในตำแหน่งที่กำหนด (0.3 คะแนน)
- turtle2 สามารถเคลื่อนที่ไปยังตำแหน่งของ pizza ที่ spawn ขึ้นมาได้สำเร็จและกิน pizza (0.5 คะแนน)
- หมายเหตุ** หาก turtle2 สามารถเคลื่อนที่ไปถึง pizza แต่ไม่สามารถกินได้ หรือเกิดความผิดพลาดอื่นๆ ที่ทำให้เหลือ pizza อยู่บนหน้าจอ (0.2 คะแนน)
- สามารถ run node นี้ผ่าน terminal โดยใช้คำสั่งของ ros2 (0.2 คะแนน)

3. odom_publisher (1.4 คะแนน)

- สามารถสร้าง Subscriber สำหรับรับข้อมูลตำแหน่งของ turtle1 และ turtle2 จาก topic /turtle1/pose และ /turtle2/pose ได้ (0.2 คะแนน)
 - Map default grid ที่มีขนาด (10x10m) ให้สามารถ interface ตรงกับ turtlesim+ GUI ได้ (0.2 คะแนน)
 - สามารถสร้าง Publisher สำหรับ Publish ข้อมูล Odometry ของ turtle1 และ turtle2 บน topic /odom1 และ /odom2 ได้ถูกต้อง (0.2 คะแนน)
 - สามารถสร้างและ Broadcast Transform ระหว่าง odom และ frame ของ turtle แต่ละตัว (turtle1 และ turtle2) ได้อย่างถูกต้อง. (0.2 คะแนน)
 - เขียนฟังก์ชันสำหรับ publish ข้อมูล odometry ทั้ง /odom1, /odom2 รวมถึง /tf ของแต่ละตัว 2 ตัว (0.2 คะแนน) ยกตัวอย่างการเขียนฟังก์ชัน
- ```
def example_pub(self, msg, turtle_name, child_frame_id):
```
- โค้ดสามารถทำงานร่วมกับ config fun2.rviz ได้ (0.2 คะแนน)
  - สามารถ run node นี้ผ่าน terminal โดยใช้คำสั่งของ ros2 (0.2 คะแนน)

หมายเหตุ:

- นักศึกษามีเวลาในการทำแบบฝึกหัดนี้สูงสุด 2 ชั่วโมง
- นักศึกษาสามารถใช้ internet ได้แต่ไม่อนุญาตให้ใช้ generative AI
- ในระหว่างทำแบบฝึกหัดไม่อนุญาตให้ถามคำถามหรือคุยกับเพื่อนได้ในทุกช่องทาง
- หากทำผิดเงื่อนไขดังกล่าว**จะถือว่าเป็นการทุจริต**คะแนนรอบนี้จะถูกปรับเป็น 0 ทันที