

EXAM/HW 1: เต่าหรรษา+ (15 คะแนน)

Objective

- เพื่อศึกษาการทำงาน ROS2 ในหัวข้อ Topics, Service, Parameter, Namespace, Launch File
- เพื่อให้ นักศึกษาสามารถออกแบบสถาปัตยกรรมของโปรแกรมได้อย่างเหมาะสมและทำงานได้
- เพื่อให้ นักศึกษาใช้การทำงานของ ROS2 มาเป็น Framework สำหรับการทำงานร่วมกัน

Mission

สร้าง workspace ที่ประกอบไปด้วย Packages สำหรับภารกิจ เต่าหรรษา+ โดยภารกิจนี้จะประกอบไปด้วย 2 ส่วนหลักโดยแบ่งเป็นเต่า teleop และฝูงเต่า copy ซึ่งในส่วนแรก เต่า teleop มีหน้าที่ในการรับคำสั่งจาก keyboard เพื่อควบคุมการทำงานต่างๆ โดยคำสั่งที่รับมาจากคีย์บอร์ดจะมีดังนี้

1. move - รับคำสั่งจากแป้นพิมพ์ (เช่น asdw) เพื่อควบคุมทิศทางการเคลื่อนที่ของเต่า
2. spawn_pizza - เมื่อกดคีย์เฉพาะ โปรแกรมจะทิ้งพิซซ่าในตำแหน่งที่เต่าตั้งอยู่
3. save - เมื่อกดคีย์เฉพาะ เต่าจะ “บันทึกตำแหน่ง” ของพิซซ่าทั้งหมดที่ยังไม่ถูกบันทึกไว้ในไฟล์ .yaml
4. clear - เมื่อกดคีย์เฉพาะ เต่าจะทำการ “ลบทิ้ง” พืชชาทั้งหมดที่ยังไม่ถูกบันทึกไว้ในไฟล์ .yaml

หมายเหตุเต่า teleop

1. จำนวน pizza ที่สามารถ spawn ได้จะมีจำนวน n pizza ซึ่งสามารถกำหนดค่าได้ผ่าน rqt
2. สามารถปรับค่า controller gain ได้ผ่าน rqt
3. เมื่อ save ครบ 4 ครั้งแล้วโปรแกรมจะไม่สามารถ save ตำแหน่งใหม่ได้อีกจนกว่าจะทำการ reset
4. เต่า teleop ต้องสามารถกำหนด namespace ได้

และในส่วนที่ 2 ส่วนของฝูงเต่า copy เต่าฝูงนี้มีหน้าที่ในการ spawn pizza ให้เหมือนกับตำแหน่งที่ save ไว้ในไฟล์ .yaml โดยฝูงเต่า copy จะเริ่มทำงานอัตโนมัติเมื่อ เต่า teleop บันทึกตำแหน่งครบ 4 ครั้ง ซึ่งขั้นตอนการทำงานของฝูงเต่า copy มีดังต่อไปนี้

1. ในการทำงานเริ่มต้น จะต้อง Spawn หุ่นยนต์เต่าทั้ง 4 ตัวจากการเริ่มต้นที่มุมล่างซ้าย
2. หุ่นยนต์เต่าตัวที่ 1 ชื่อ Foxy มีหน้าที่วิ่งไปทิ้ง Pizza ตามชุดตำแหน่งที่ได้จากการ save ครั้งที่ 1
3. หุ่นยนต์เต่าตัวที่ 2 ชื่อ Noetic มีหน้าที่วิ่งไปทิ้ง Pizza ตามชุดตำแหน่งที่ได้จากการ save ครั้งที่ 2
4. หุ่นยนต์เต่าตัวที่ 3 ชื่อ Humble มีหน้าที่วิ่งไปทิ้ง Pizza ตามชุดตำแหน่งที่ได้จากการ save ครั้งที่ 3
5. หุ่นยนต์เต่าตัวที่ 4 ชื่อ Iron มีหน้าที่วิ่งไปทิ้ง Pizza ตามชุดตำแหน่งที่ได้จากการ save ครั้งที่ 4
6. เมื่อหุ่นยนต์แต่ละตัววาง pizza ตามชุดตำแหน่งของตัวเองเสร็จแล้วให้หยุดรอจนกว่าทุกตัวจะทำได้สำเร็จ จากนั้นให้หุ่นยนต์ทุกตัววิ่งไปที่ตำแหน่งบนขวา

NOTE

1. นักศึกษาต้องสร้าง Launch สำหรับ Run node ทั้งหมดยกเว้น node สำหรับรับคำสั่งจากคีย์บอร์ด
2. ถ้านักศึกษาใช้วิธีการ Hard code ในการแก้ปัญหาก็จะถูกต้อง (ถึงแม้ใช้งานได้ก็ตาม)
3. นักศึกษาจะต้องออกแบบสถาปัตยกรรมของโปรแกรมว่าประกอบไปด้วย package อะไรบ้างในแต่ละ package ประกอบด้วย node อะไรบ้าง ในแต่ละ node ประกอบด้วย Topic/Service อะไรบ้าง และแต่ละส่วนมีการเชื่อมโยงกันอย่างไร พร้อมเขียนอธิบายใน Document (Readme.md)
4. นักศึกษาจะต้องใช้ namespace ในการจัดการ project ให้เป็นประโยชน์ ยกตัวอย่างเช่นการใช้ controller เพียง 1 node ในการควบคุมเตา teleop และเตา copy
5. นักศึกษาจะต้องใช้ ros parameter, service ให้เกิดประโยชน์และเหมาะสม

Bonus (ปรับคะแนน FUN ที่ได้น้อยสุดให้ได้เต็มสามารถเก็บไว้ใช้ในอนาคตได้)

เมื่อฝูงเตา copy ทั้ง 4 ตัวเคลื่อนที่ไปอยู่มุมบนขวาแล้ว จะ spawn เตามีชื่อว่า “Melodic” ขึ้นมาบนฝั่งของฝูงเตา copy จากนั้นให้ Melodic ทำการไล่กิน pizza และเตาทั้ง 4 ตัว (หากหน้าจอฟังฝูงเตา copy ไม่เหลืออะไรเลยนอกจากน้อง melodic จากการกระทำข้างต้นรับไปเลย 1 โควตาสำหรับแก้คะแนน FUN) หลังจากกินฝูงเตา copy เสร็จให้ melodic วิ่งทะลุไปฝั่งหน้าจอตเตา teleop และการไล่กิน pizza ทั้งหมดและเมื่อกินหมดแล้วให้ไปกินเตา teleop จากนั้นให้รีเซ็ตระบบทั้งหมดให้เป็นเหมือนตอนเริ่มต้นโปรแกรมโดยไม่ใช้วิธีเปิด-ปิดโปรแกรมใหม่ (รับไปเลยอีก 1 โควตาสำหรับแก้คะแนน FUN)

Grading

Part 1: Teleop_turtle	Score
ความถูกต้องความเสถียรของโปรแกรม	3
ความเหมาะสมในการประยุกต์ใช้ topics, service, parameter, namespace, remap	2
Part 2: Copy_turtle	
ความถูกต้องความเสถียรของโปรแกรม	3
ความเหมาะสมในการประยุกต์ใช้ topics, service, parameter, namespace, remap	2
Part 3: System integration	
ความเหมาะสมและความชัดเจนของหน้าที่แต่ละ node	2
การเขียน launch file ของระบบทั้งหมด	1.5
การออกแบบสถาปัตยกรรมของโปรแกรม + การเขียน doc	1.5

Night Market (12.30 – 13.30)

- System architecture (-1 คะแนน) – โครงสร้างสถาปัตยกรรมการทำงานของระบบทั้งหมด
- pizzapath. yaml (-0.1 คะแนน) – ไฟล์. yaml เพื่อทดลองนำไปเป็น path ให้กับเต้าตัว copy
- teleop_key.py (-1.5 คะแนน) – node รับและส่งคำสั่งที่ได้จากคีย์บอร์ด
- teleop_schedule.py (-3 คะแนน) – node สำหรับการจัดการ state ทั้งหมดของเต้า teleop
- teleop_turtle.launch.py (-1 คะแนน) – launch file สำหรับเรียกการทำงานของเต้า teleop
- package teleop_interfaces (-1 คะแนน) - รวม custom services ฟังก์ชัน teleop
- copy_schedule.py (-4 คะแนน) – node สำหรับการจัดการ state ทั้งหมดของฝูงเต้า copy
- copy_turtle.launch.py (-2 คะแนน) - launch file สำหรับเรียกการทำงานของเต้า copy
- controller.py (-2 คะแนน) – node สำหรับควบคุมเต้าให้ไปยังตำแหน่งที่กำหนด พร้อมสถานะการทำงานปัจจุบันของ controller โดยใช้ service

เงื่อนไขในการสอบ

- night market จะเปิดให้บริการเวลาตามช่วงเวลาดังกล่าวซึ่งแต่ละช่วงมีตัวคูณราคาตั้งต้นดังนี้
15.30 – 15.45 (x1.20), 18.30 – 18.45 (x1.30), 21.30 – 21.45 (x1.50), 00.00 – 00.10 (x1.75)
- ไม่อนุญาตให้ส่งต่อโค้ดใดๆ ทุกกรณี หากส่งต่อให้ไม่ว่าทางใดทางหนึ่งหรือรูปแบบใดรูปแบบหนึ่งจะนับเป็นการ**ทุจริตและปรับ F ทุกกรณี**
- ไม่อนุญาตให้พูดคุยเกี่ยวกับเนื้อหาในการสอบรวมถึงเนื้อหาใดๆที่เกี่ยวข้องกับรายวิชา ข้ามคู่ของตัวเอง ไม่ว่ากรณีใดๆ จะนับเป็นการ**ทุจริตและปรับ F ทุกกรณี**
- นักศึกษามีเวลาในการทำแบบทดสอบ 24 ชั่วโมง
- วิธีการส่งงาน zip ไฟล์ workspace ส่งทาง classroom ตั้งชื่อไฟล์ด้วยรหัสนักศึกษา ตัวอย่าง 6438_6460.zip แนะนำให้ลบไฟล์ build install log ออกก่อน หากโค้ดส่วนไหน error โค้ดส่วนนั้นจะไม่ได้รับการตรวจ