

Beroepsproduct Simulatie - Opdracht

Bram Knippenberg
9 oktober 2024

1. INLEIDING

In de course heb je geleerd om software te ontwerpen en te maken waarmee je eenvoudige actuatoren zoals robotarmen kunt ontsluiten en aansturen. Daarmee zou je bijvoorbeeld een robotsysteem kunnen bouwen die je een glas water kan aangeven. Voordat je een dergelijk systeem echt gaat maken is het vaak handig om eerst je algoritmen te testen in een gesimuleerde omgeving zodat je hardware niet kapot gaat en je makkelijker aanpassingen in de code kunt testen.

In de praktijk komt het vaak voor dat je bestaand werk van derden moet beoordelen op kwaliteit, bruikbaarheid en onderhoudbaarheid. Bijvoorbeeld het werk van een collega, een open source project, werk gedaan door externe partij voor een klant etcetera.

In deze opdracht komen beide aspecten aan bod, namelijk het maken van een simulatie en het beoordelen van het werk van een ander in de vorm van een peer-review. Als eerste maak je een simulatie van een robotarm en een “virtueel” bekertje water waarmee veilig getest en geëxperimenteerd kan worden. Daarna voer je een peer-review uit op het werk van een andere student.

2. OPDRACHT

2.1. SIMULATIE-PACKAGE

In deze opdracht moet je een Robot OS (ROS2) package ontwikkelen dat virtuele bekertjes kan simuleren. Andere ROS packages kunnen er dan gebruik van maken. Naast het bekertje ontwikkel je een demonstratie van een voorbeeldinteractie tussen een virtuele robotarm en een virtueel bekertje. Je zult hiervoor dus ook een virtuele implementatie van de AL5D controller moeten maken. Als laatste schrijf je een handleiding waarin je uitlegt hoe je deze demo kunt bouwen en uitvoeren, en welke eisen je ermee demonstreert.

In de bijlage worden de eisen beschreven waaraan je werk moet voldoen. Het werk dat je inlevert moet voor de peer-review minstens voldoen aan de MUST-eisen en twee andere (Should, Could, Would) eisen, anders mag je niet deelnemen aan de uiteindelijke toets (de peer-review).

2.2. PEER-REVIEW

Aan het eind van deze periode krijg je het werk van een andere student uitgereikt. Dat werk ga je in die les beoordelen op kwaliteit, bruikbaarheid en onderhoudbaarheid. Je bevindingen leg je vast in een rapport dat uiteindelijk door de docent wordt beoordeeld. Dit rapport, wat je op je laptop mag maken, is de uiteindelijke toets.

Deelname aan de peer-review kan alleen als je Simulatie-package (inclusief documentatie) door de docent als voldoende is aangemerkt in de les voorafgaand aan de peer-review. /newpage # Beoordeling

3. BEOORDELING

Er zijn twee toetsmomenten tijdens deze opdracht.

3.1. SIMULATIE-PACKAGE (UITWERKING VAN VOLDOENDE NIVEAU)

Ten eerste, voordat je mag deelnemen aan de peer-review moet je aantonen bij de docent dat je simulatie-package inclusief documentatie van voldoende niveau en omvang is. Het komt erop neer dat je simulatie-package minstens aan de MUST-eisen en minimaal twee ander eisen voldoet (zie bijlage).

3.2. PEER-REVIEW (SCHRIFTELIJKE TOETS)

Het tweede toetsmoment is de peer-review zelf. Tijdens een review-sessie krijg je al het werk van een andere student dat je moet beoordelen op kwaliteit, bruikbaarheid en onderhoudbaarheid. Je bevindingen leg je vast in een rapport dat je ter beoordeling inlevert bij de docent. Dat rapport maak je op je laptop en aan het eind van de toets lever je dat in via iSAS (inclusief het werk dat je beoordeelde).

De docent beoordeelt je rapport op inhoud en op onderbouwing. Met inhoud wordt bedoeld of wat je opschrijft klopt met het werk dat je hebt beoordeeld. Met onderbouwing wordt bedoeld of je steekhoudende argumenten gebruikt en die ondersteunt met verwijzingen naar en voorbeelden uit het werk dat je beoordeeld hebt.

4. PLANNING

De volgende tijdsplanning wordt gehanteerd:

- week 6 t/m week 8
Uitwerken van de simulatie-package, zowel realisatie als documentatie. Dit doe je deels in je eigen tijd, maar ook tijdens de lessen is er ruim tijd om daar aan te werken. De docent is tijdens die lessen aanwezig om je te helpen bij het realiseren.
- Week 9 (nog in te plannen)
In deze les laat je zien dat je werk van een dusdanige omvang is dat er een peer-review op gedaan kan worden. Is dit niet het geval, dan mag je niet deelnemen aan de peer-reviewsessie. Voor het begin van de les heb je ook een zip ingeleverd met al het materiaal waarmee een ander direct van start kan gaan. De docent test dat op zijn eigen systeem. Er wordt o.a. gekeken of het werk 'out of the box' compileert, of er documentatie is bijgevoegd die te openen valt, of er een handleiding is om er direct mee te beginnen, etc. Als dit niet het geval is, dan kun je niet deelnemen aan de toets.
- Week 9 of week 1 (nog in te plannen)
Tijdens een peer-review sessie voer je de review uit. Je krijgt het werk van een andere student uitgereikt, daarop voer je de peer-review uit. De docent zal je helpen als je het werk van de ander niet werkend krijgt.
Het rapport maak je op je laptop. Aan het eind van de toets lever je je rapport, inclusief het werk dat aan je was uitgereikt, in via iSAS. Je krijgt twee uur (120 minuten) voor deze review.

5. BIJLAGEN

A. EISEN ‘SIMULATIE-PACKAGE’

De uitwerking van je beroepsproduct zal op de volgende punten beoordeeld worden:

A.1. PACKAGE

ID	Omschrijving	Prio
PA01	Alle code is gepackaged volgens de ROS-directorystructuur.	Should
PA02	Package is te bouwen met colcon op ROS2 Jazzy Jalisco	Must
PA03	De applicatie wordt gebouwd met C++ volgens de Object Oriented principes die je geleerd hebt bij eerdere courses.	Must
PA04	Alle code voldoet aan de ROS C++ Style Guide.	Should

A.2. VIRTUELE SERVO CONTROLLER

ID	Omschrijving	Prio
VS01	De virtuele controller luistert naar een topic waarop string messages in het formaat van de SSC-32U ¹ worden geplaatst. Van de interface moeten ten minste commando's zijn opgenomen voor het verplaatsen van de servo's met een ingestelde duur en het stoppen van de servo's.	Must
VS02	De virtuele controller reageert op het topic (zie eis VS01) door bijbehorende joint_state messages te publiceren.	Must
VS03	De virtuele robotarm wordt gevisualiseerd in Rviz (een URDF-model van de arm is beschikbaar op OnderwijsOnline).	Must
VS04	De virtuele robotarm gedraagt zich realistisch m.b.t. tijdgedrag (servo's roteren kost tijd en gaat geleidelijk).	Must
VS05	De virtuele robotarm kan op een willekeurige plaats in de virtuele wereld geplaatst worden.	Should

A.3. VIRTUEEL BEKERTJE

ID	Omschrijving	Prio
VC01	Er kan op een willekeurige plek in de virtuele wereld een bekertje geplaatst worden.	Should
VC02	Publiceert een 3D-visualisatie van het bekertje voor Rviz.	Must
VC03	Detecteert de relevante punten van de gripper.	Should
VC04	Visualiseert de gedetecteerde punten van de gripper.	Could
VC05	Visualiseert wanneer de gripper het bekertje vastheeft.	Must
VC06	Het bekertje beweegt mee met de gripper (als hij vastgehouden wordt).	Must
VC07	Bekertje is onderhevig aan zwaartekracht wanneer losgelaten.	Must
VC08	Bekertje bepaalt en publiceert zijn positie.	Must
VC09	Bekertje bepaalt en publiceert zijn snelheid.	Should

¹Lynxmotion SSC-32U USB Servo Controller Board http://www.lynxmotion.com/images/data/lynxmotion_ssc-32u_usb_user_guide.pdf

ID	Omschrijving	Prio
VC10	Snelheid wordt getoond met <code>rqt_plot</code> .	Could

A.4. DEMONSTRATIE-INFRASTRUCTUUR

ID	Omschrijving	Prio
DI01	Een demoscript stuurt over de tijd een sequentie van commando's naar de armcontroller. ²	Must
DI02	Locatie van het bekertje wordt in de roslaunch-configuratie bepaald.	Could
DI03	Locatie van de arm in de wereld wordt in de roslaunch-configuratie bepaald.	Could

A.5. GEBRUIKERSHANDLEIDING

ID	Omschrijving	Prio
DM01	Beschrijft hoe de code gebouwd kan worden.	Must
DM02	Beschrijft stap voor stap hoe de arm bewogen kan worden middels enkele voorbeelden.	Must
DM03	Beschrijft welke eisen gerealiseerd zijn. En geeft hierbij een (korte) toelichting.	Must

A.6. ONTWERPDOCUMENTATIE

ID	Omschrijving	Prio
DD01	Beschrijft de structuur van de package (Nodes, topics, messages, et cetera).	Must
DD02	Beschrijft de structuur en samenhang van de broncode (class-diagrams, beschrijving, et cetera).	Must
DD03	Beschrijft hoe het gedrag van alle belangrijke componenten gerealiseerd is.	Could
DD04	Beschrijft de API van alle publieke interfaces.	Should

Mocht je om één of andere reden niet aan een must eis kunnen voldoen dan dien je dit, na het besproken te hebben met de docent, toe te lichten in de documentatie.

²hier wordt bedoeld dat je een lijstje van commando's, in het formaat van de SSC-32U, publiceert zodat de gesimuleerde arm gaat bewegen.