

Kötelező Program

Nehézségi fokozatok és érdemjegyek

A kötelező programok három *nehézségi fokozatban* teljesíthetők. A *nehézségi fok* meghatározza a **legjobb** érdemjegyet, amely a teljesítéséért kapható!

Nehézségi fok	Legjobb megszerezhető érdemjegy
Basic	3
Advanced	4
Epic	5

Tip

A feladatok úgy vannak megadva, hogy érdemes a **Basic** szinttel kezdeni, és onnan fokozatosan építkezni az **Epic** szintig.

A kötelező programok a következő szempontok szerint kerülnek értékelésre:

- Bizonyítottan saját munka
- Értékelhető eredményeket produkáljon
- Verziókövetés használata, feltöltés GitHub/GitLab/egyéb repoba
- Értékelési szempontok:
 - a megoldás teljessége
 - megfelelő ROS kommunikáció alkalmazása
 - program célszerű szerkezete
 - az implementáció minősége
 - a kód dokumentálása

Ütemezés

Okt. hét	Dátum	Számonkérés
2.	szept. 13	Kötelező programok ismertetése.
8.	okt. 25	Kötelező program mérföldkő.
14.	nov. 6	Kötelező programok bemutatása.

Évközi jegy

A félév elfogadásának feltétele, hogy mind a két ZH, mind a kötelező program értékelése legalább elégséges. A két ZH közül az egyik az utolsó óra alkalmával pótolható.



Félév végi jegy

$$\backslash \text{Jegy} = (\text{ZH1} + \text{ZH2} + 2 \backslash \text{times KötProg}) / 4 \backslash$$

Kötelező program témák

1. PlatypOUs



1.1. PlatypOUs pályakövetés

- **Basic:** Szimulátor élesztése, SLAM tesztelése. ROS node/node-ok implementálása szenzorok adatainak beolvasására és a robot mozgatására.
- **Advanced:** ROS rendszer implementálása pályakövetésre szimulált környezetben bármely szenzor felhasználásával(pl. fal mellett haladás adott távolságra LIDAR segítségével).
- **Epic:** Implementáció és tesztelés a valós hardware-en és/vagy nyűgözz le!

1.2. PlatypOUs akadály elkerülés

- **Basic:** Szimulátor élesztése, SLAM tesztelése. ROS node/node-ok implementálása szenzorok adatainak beolvasására és a robot mozgatására.
- **Advanced:** ROS rendszer implementálása akadály felismerésére és az akadályt kikerülő trajektória tervezésére és megvalósítására szimulált környezetben bármely szenzor felhasználásával.
- **Epic:** Implementáció és tesztelés a valós hardware-en és/vagy nyűgözz le!

1.3. PlatypOUs objektum követés

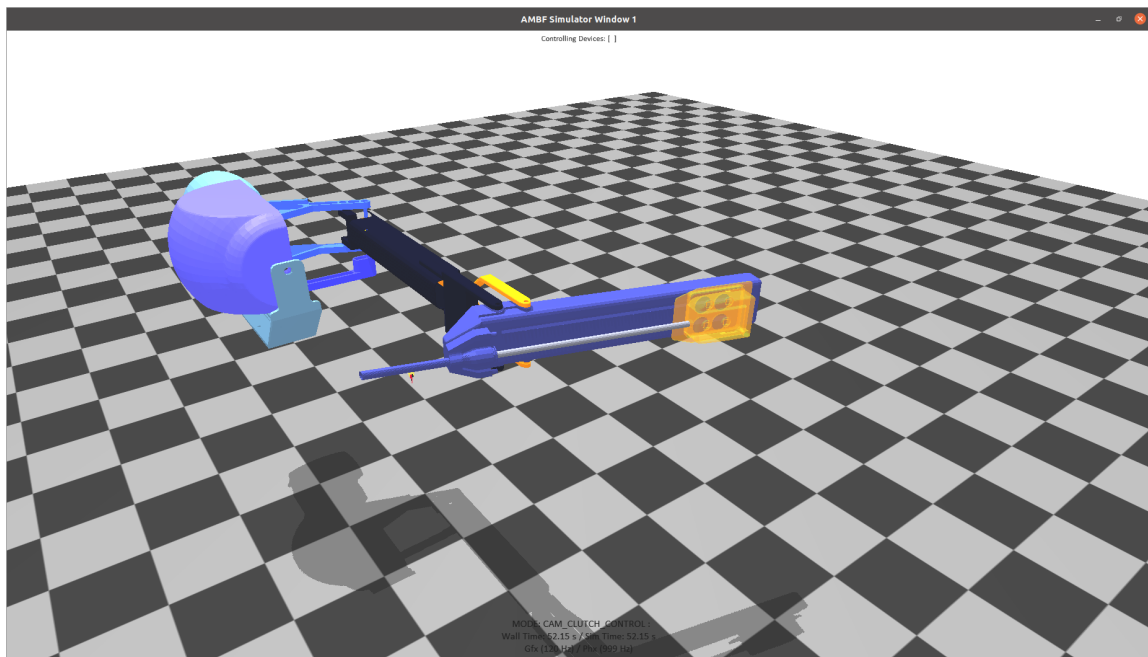
- **Basic:** Szimulátor élesztése, SLAM tesztelése. ROS node/node-ok implementálása szenzorok adatainak beolvasására és a robot mozgatására.
- **Advanced:** ROS rendszer implementálása objektum megkeresésére/felismerésére és követésére/megközelítésére szimulált környezetben bármely szenzor felhasználásával (pl. visual servoing).
- **Epic:** Implementáció és tesztelés a valós hardware-en és/vagy nyűgözz le!

1.4. PlatypOUs action library

- **Basic:** Szimulátor élesztése, SLAM tesztelése. ROS node/node-ok implementálása szenzorok adatainak beolvasására és a robot mozgására.
- **Advanced:** Egyszerű műveleteket tartalmazó, ROS action alapú könyvtár és ezeket végrehajtó rendszer implementálása (pl. push object, move to object, turn around).
- **Epic:** Implementáció és tesztelés a valós hardware-en és/vagy nyűgözz le!

2. AMBF

2.1. da Vinci sebészrobot ROS integrációja AMBF szimulátorban



- **Basic:** Szimulátor élesztése, robot vezérlése joint space-ben és task space-ben (IK már implementálva AMBF-ben) ROS-ból CRTK szerinti topic-okon keresztül
- **Advanced:** Objektumok detektálása *Peg transfer puzzle*-ben
- **Epic:** Autonóm manipuláció *Peg transfer*-en és/vagy nyűgözz le!

2.2. KUKA robotkar ROS integrációja AMBF szimulátorban



- **Basic:** Szimulátor élesztése, robot vezérlése joint space-ben ROS-ból
- **Advanced:** Robot vezérlése task space-ben, IK?
- **Epic:** Trajektóriatervezés

2.3. PR2 humanoid robot ROS integrációja AMBF szimulátorban



- **Basic:** Szimulátor élesztése, robot vezérlése joint space-ben ROS-ból
- **Advanced:** Robot vezérlése task space-ben, IK?

- **Epic:** Trajektóriatervezés/Navigáció/Manipuláció

X. Saját téma

Megegyezés alapján.

Hasznos linkek

- [Gazebo ROS packages](#)
- [PlatypOUs](#)
- [AMBF](#)
- [My fork of AMBF](#)
- [CRTK topics](#)
- [Navigation stack](#)
- [Paper on LiDAR SLAM](#)
- [Paper on vSLAM](#)
- [Paper on Visual Servoing Mobile Robot](#)