

| | |
|-------------------------------------|------------------|
| Upravljanje letjelicom kroz poligon | Verzija: <1.0> |
| Tehnička dokumentacija | Datum: 21.1.2022 |

Upravljanje letjelicom kroz poligon

Tehnička dokumentacija

Verzija <1.0>

Studentski tim: Dino Smirčić
Enio Krizman
Josip Hanak
Petar Sorić
Lovre Grašo
Lovro Široki
Viktor Horvat

Nastavnik: Matko Orsag

| | |
|-------------------------------------|------------------|
| Upravljanje letjelicom kroz poligon | Verzija: <1.0> |
| Tehnička dokumentacija | Datum: 21.1.2022 |

Sadržaj

- 1.Opis razvijenog proizvoda
- 2.Tehničke značajke
- 3.Upute za korištenje
- 4.Literatura

| | |
|-------------------------------------|------------------|
| Upravljanje letjelicom kroz poligon | Verzija: <1.0> |
| Tehnička dokumentacija | Datum: 21.1.2022 |

Tehnička dokumentacija

1. Opis razvijenog proizvoda

Za simuliranje ponašanja letjelice na računalu bilo je potrebno očitati i prenijeti arenu koju prelazi letjelica u programsko okruženje Gazebo koje služi za testiranje i simuliranje ponašanja robota. Arena je dimenzija 10 metara po x osi i 7 metara po y osi te se sastoji od prepreka čija se pozicija očitava preko OptiTrack kamere, zidova i podloge koji su napravljeni u programskom okruženju za 3d modeliranje „Blender”. Optitrack kamera snima informacije o translacijskom i rotacijskom položaju prepreke i prosljeđuju ih VRPN client serveru. Server trenutno očitane informacije prosljeđuje robotskom operacijskom sustavu ROS preko '/euroc3/vrpn_client/estimated_odometry' čvora (engl. node) na kojem su potom dostupne informacije za očitavanje. Izrađeni program za potrebe projekta „vrpnclient.py” koji je razvijen u programskom jeziku Python koristi informacije o položaju prepreka kako bi pohranio novu listu prepreka ili izmijenio postojeću te se lista sprema u formatu x;y;z u tekstualnu datoteku „data.txt”. Program „vrpnclient.py” ima 2 funkcionalnosti zato se prilikom pokretanja na zaslon terminala operacijskog sustava Linux postavlja pitanje hoće li se izraditi nova arena sa svim novih preprekama ako se unese broj 2 ili izmijeniti postojeća arena s preprekama, čije su pozicije dostupne u data.txt, ako se unese broj 1. Za unos svih novih prepeka potrebno je prilikom svakog unosa prepreke pritisnuti tipku enter na tipkovnici kada se prepreke postavljene na željenu poziciju u stvarnoj areni kako bi se učitale njihove pozicije u zadanu tekstualnu datoteku data.txt. Nadalje, za izmjenu postojeće arene s preprekama, prvotno se iz lista postojećih prepeka te se postavlja pitanje u terminalu koja se pozicija prepreke želi izmijeniti. Kada se odabere prepreka potrebno je također pritisnuti tipku enter kada se prepreka nalazi u areni na novoj željenoj poziciji. Prepreke se neprestano očitavaju u oba slučaja dok se program ne prekine pritiskom zadanih kombinacija tipki. Kada je program „vrpnclient.py” gotov s obavljanjem svoje zadaće, na računalu postaje dostupna datoteka data.txt iz koje se lako može pročitati popis pozicija svih prepreka koje se u tom trenutku nalaze u areni.

AddObstaclesToMap.py skripta je koja iskorištava mogućnosti programskog paketa Blender da se njime upravlja kroz programski jezik Python korištenjem BlenderPythonAPI-a. Pritom skripta učitava 3D model arene, čita prije spomenutu data.txt datoteku te parsiranjem podataka iz nje koji sarže x i y koordinatu u 3D modelu arene postavlja prepreke oblika kvadra duž visine arene na odgovarajuća mjesta. Skripta sprema novonastali model arene u svoj korijenski direktorij. path_to_arena.obj predstavlja putanju do modela 3D modela arene. Izlazne datoteke skripte stvaraju se u korijenskom direktoriju skripte. Izostavljanjem argumenta -b moguće je vidjeti stvoreni 3D model unutar samog programskog paketa Blender.

| | |
|-------------------------------------|------------------|
| Upravljanje letjelicom kroz poligon | Verzija: <1.0> |
| Tehnička dokumentacija | Datum: 21.1.2022 |

Kako bi našu mapu arene mogli koristiti za navigaciju, potrebno je bilo pretvoriti .world file u octomapu te zatim nju pretvoriti u 2D mapu za navigaciju. Tu pretvorbu smo ostvarili uz pomoć paketa https://github.com/larics/larics_gazebo_worlds. Zatim smo u stagerosu napravili robota koji se nalazio u našoj mapi i u rvizu učitali mapu i poziciju robota. Nakon toga pokrenuli smo move_base paket koji smo koristili za navigaciju robota. Potrebno je bilo preusmjeriti poruke koje šalje move_base na našeg drona u gazebu, to smo napravili uz pomoć python skripte bebopController.py koju smo sami napisali. Subscribeali smo se na topic /odom te zatim poruku proslijedili na topic /bebop/pos_ref. Sada smo mogli u rvizu zadati robotu nav goal i letjelica u gazebu bi doletila do zadanog cilja. Pokretanje svega zahtijevalo je nekoliko upaljenih terminala odjednom, pa smo napravili launch file koji sve to pokreće.

2. Tehničke značajke

Programska podrška za arenu

[addObstaclesToMap.py]

Ulazna datoteka skripti je data.txt, podaci u njoj predstavljaju koordinate prepreke u obliku [x,y,z]. Z koordinata se ne koristi jer je pretpostavljeno da će visina prepreke odgovarati visini arene, no zbog mogućih budućih implementacija ostavljena je mogućnost definiranja z komponente prepreke.

Ulazna datoteka skripti je arena.obj koja sadrži model arene.

Izlazne datoteke su export.stl, export.dae.

Skripta se oslanja na programski paket Blender.

Potrebna programska podrška za pokretanje skripte je:

Python >=3.10, Blender >=3.0

Letjelica

U projektu smo koristili letjelicu parrot bebop 2 prikazanu na slici.



Letjelica ima prednju kameru i može se njome upravljati preko ros-a koristeći paket bebop_autonomy.

Ispuštanje objekta

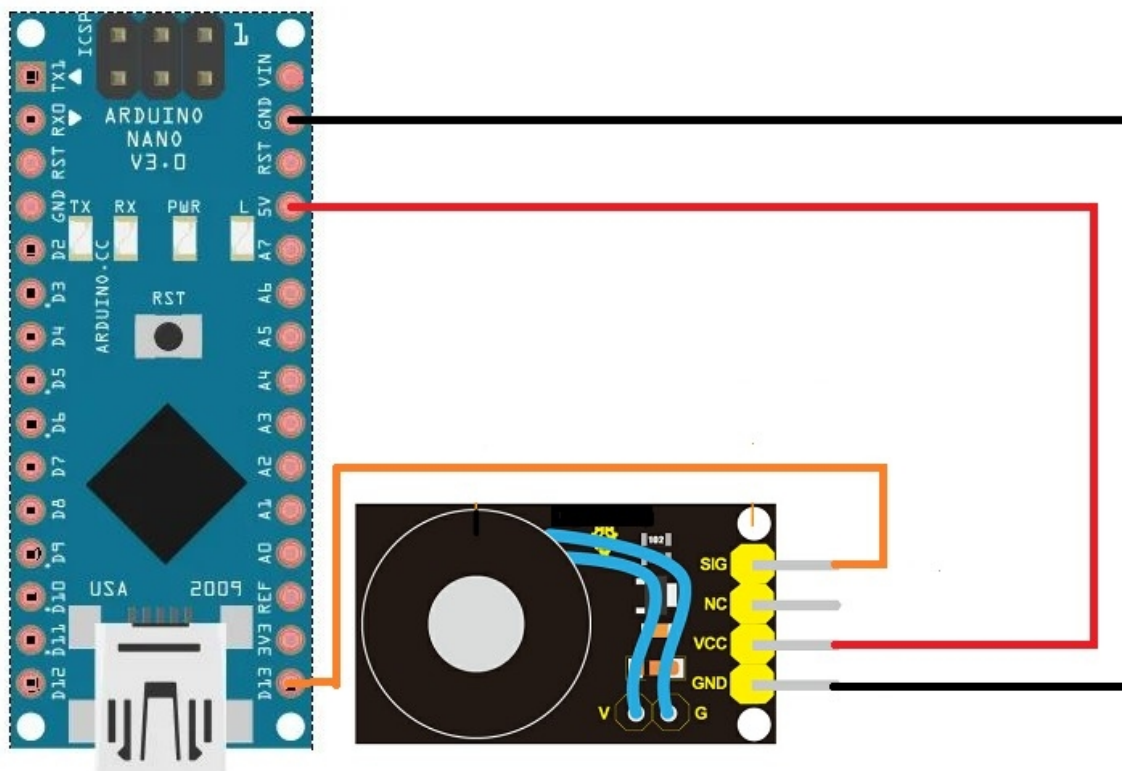
| | |
|-------------------------------------|------------------|
| Upravljanje letjelicom kroz poligon | Verzija: <1.0> |
| Tehnička dokumentacija | Datum: 21.1.2022 |

Ispuštanje objekta realizirano je s mikrokontrolerom arduino nano i elektromagnetom koji se nalaze na letjelici. Prilikom uključanja mikrokontrolera, elektromagnet je inicijalno aktivan te se objekt može spojiti na letjelicu.

Kako bi pokrenuli mikrokontroler u ROS-u potrebno je prvo u terminalu pokrenuti naredbu *roscore*.

Zatim se pokreće naredba: *roslun roserial_python serial_node.py /dev/ttyUSB0*.

Prilikom slanja naredbe: *rostopic pub magnet std_msgs/Empty –once*, elektromagnet se isključuje. Ponovnim slanjem iste naredbe možemo ponovo uključiti elektromagnet.



3. Upute za korištenje

Vrpnclient.py

Za testiranje funkcionalnosti programa potrebno je pokrenuti dogovarajući .bag file snimljen u postojećoj areni, koristiti stvarni postav s OptiTrack kamerom i preprekama ili pokrenuti program „obstacle_publisher.py” koji generira nasumične pozicije prepreka sljedećom naredbom ukoliko nalazite u mapi s programom:

```
$python3 „obstacle_publisher.py”
```

| | |
|-------------------------------------|------------------|
| Upravljanje letjelicom kroz poligon | Verzija: <1.0> |
| Tehnička dokumentacija | Datum: 21.1.2022 |

Kada je osigurano objavljivanje informacija o poziciji prepreka na `/*namespace*/vrpn_client/estimated_odometry` čvor, program za očitavanje ili izmjenu prepreka pokreće se:

```
$rosrun ros_vrpn_client vrpnclient.py
```

gdje je `ros_vrpn_client` simbolično ime ROS paketa u kojoj se program nalazi.

Ukoliko želite unijeti novu arenu pritisnite broj 1 te kada je prepreka na željenoj poziciji pritisnite tipku enter. Nakon što ste unijeli sve prepreke pritisnite kombinaciju tipki `ctrl+c` za prekid programa.

Ukoliko želite izmijeniti već postojeću arenu pritisnite 1 te će Vam se izlistati lista postojećih prepreka. Nakon što ste odabrali prepreku iz liste koju želite promijeniti, unesite njen indeks te kada je prepreka na željenoj poziciji u areni pritisnite enter. Za prekid pritisnite kombinaciju tipki `ctrl+d`.

Prilikom svakog unosa prepreke osvježava se datoteka `data.txt` u kojoj se nalazi popis svih prepreka te iz nje možete očitati pozicije prepreka koje ste prethodno unijeli te ih po potrebi i izmijeniti.

[`addObstaclesToMap.py`]

Skripta se pokreće iz komandne linije operativnog sustava korištenjem:

```
blender -b --python addObstaclesToMap.py -- [path_to_data.txt] [path_to_arena.obj]
```

Pritom:

`path_to_data.txt` predstavlja putanju do datoteke `data.txt`,

Korištenje navigacijskog paketa i pokretanje simulacije

1. Potrebno je klonirati github repozitorij uz pomoć naredbe

```
git clone https://github.com/vhorvat/Projekt_E
```

2. Koristeći naredbu

```
roslaunch drone_navigation launchSequence.launch
```

Pokreće se navigacija i simulacija letjelice u gazebo. Potrebno je u rvizu zadati nav goal kako bi se letjelica pomakla na željenu poziciju.

4. Literatura

<http://wiki.ros.org/>