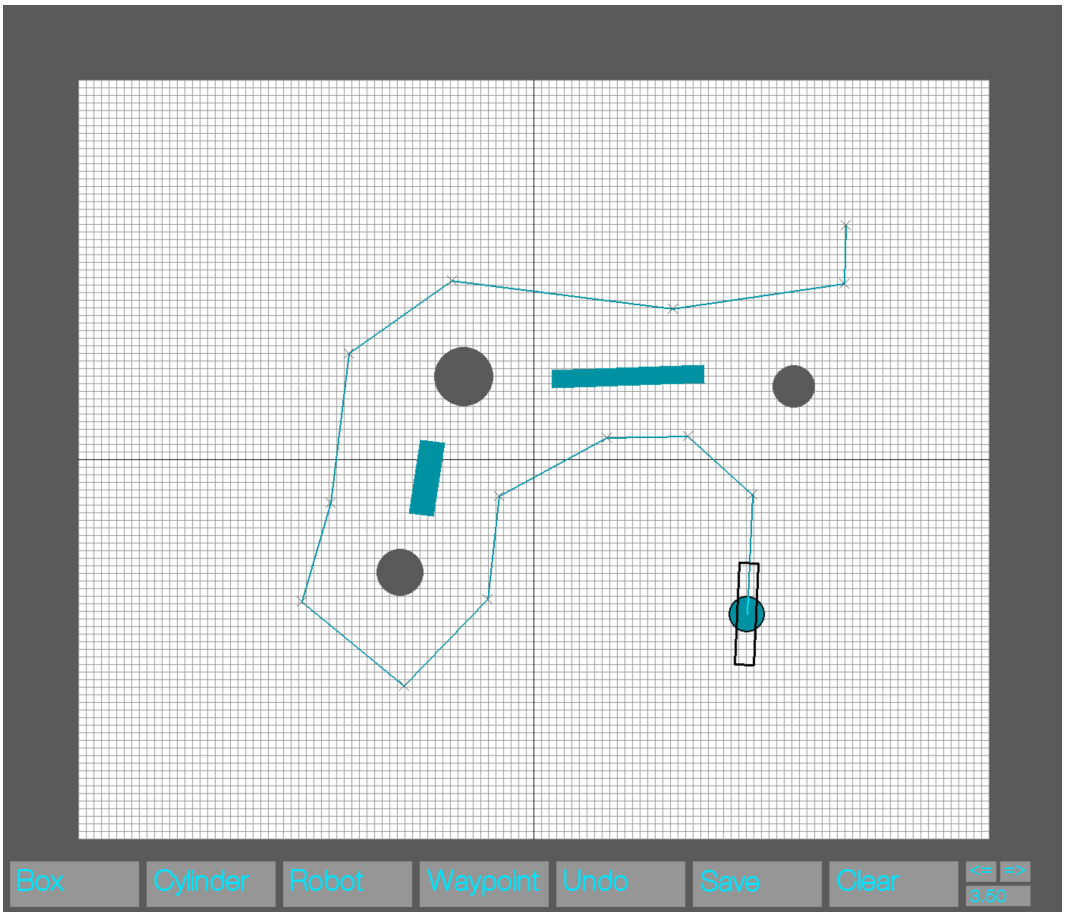
Bakalářká práce - Návod rozšíření pro PUIA5

hamarm

November 2024

1 Generátor prostředí pro Gazebo

Tento program v jazyce python byl vytvořen za účelem usnadnění vytváření prostředí pro simulaci v Gazebu. Uživatel si v grafickém rozhraní může navolit počáteční pozici robota, body, ke kterým se robot má dostat a překážky ve formě kvádrů a válců libovolné velikosti.



Obrázek 1: Ukázka okna generátoru

1.1 Funkce Tlačítek

Tlačítka Box, Cylinder, Robot a Waypoint jsou tlačítka přepínací, zatímco šipky změny výšky překážek, Undo, Save a Clear lze jen stisknout.

- Box - slouží k vytvoření překážky typu kvádr (na obrazovce tmavě modrozelený kvádr). Po zapnutí lze v mřížce vytvářet kvádry tak, že první kliknutí stanoví na pozici kurzoru střed kvádru, druhé kliknutí stanoví na pozici kurzoru jeden z jeho rohů a tedy jeho tvar a velikost, zatímco třetí kliknutí stanoví úhel natočení kvádru dle pozice kurzoru. V jednotlivých fázích konstrukce program zobrazuje náhledy, jak kvádr bude vypadat před jeho vytvořením.
- Cylinder - slouží k vytvoření překážky typu válec (šedý kruh). Po zapnutí lze v mřížce vytvářet válce tak, že první kliknutí stanoví na pozici kurzoru střed válce a druhé kliknutí poloměr. V jednotlivých fázích konstrukce program zobrazuje náhledy, jak válec bude vypadat před jeho vytvořením.
- Robot - slouží ke změně startovní pozice robota (tmavě modrozeleného kruhu). Po zapnutí se nová pozice robota definuje pozicí kurzoru při kliknutí na mřížku.
- Waypoint - slouží k přidávání bodů cesty robota. Po zapnutí lze nový bod přidat na pozici kurzoru kliknutím na mřížku. Robot se automaticky otáčí za prvním bodem (jeho orientace je znázorněna světle modrou čarou na jeho ikoně).
- Undo - po stisknutí zruší předchozí akci (přidání překážky/bodu cesty/změnu pozice robota).
- Save - po stisknutí vygeneruje 3 soubory se vším potřebným k simulaci.
- Clear - pos stisknutí vrátí program do stavu hned po spuštění (vše v mřížce se vymaže, robot se vrátí na střed).
- Šipky vlevo/vpravo - stisknutím se zmenší/zvětší výška všech následujících překážek ve 3D prostoru o 0.1.

2 Použití vygenerovaných souborů

Pomocí tlačítka Save se vygenerují 3 soubory (jejich jméno lze nastavit v kódu programu):

1. .world soubor - spustitelný Gazebem, obsahuje všechny údaje pro simulaci překážek v Gazebu.
2. .txt soubor - obsahuje počáteční pozici robota pro jeho vytvoření pomocí ros2 node spawn_entity.py. Počáteční pozice je v něm uvedene jako 4 hodnoty (pozice v X, Y, Z a otočení v Z), které se použijí jako parametry node spawn_entity.py.
3. .json soubor - obsahuje json řetězec složený ze seznamu bodů (podseznamy se souřadnicemi X a Y) cesty robota (první bod je jeho počáteční poloha) a počátečního otočení robota. Je potřeba pro node control_node.py, která řídí robota tak, aby projela popořadě skrze všechny body.

3 Řízení kolejového vozidla v simulaci

Řízení má v simulaci na starosti node control_node.py napsaný v pythonu. Jeho princip je ten, že řídí směr robota tak, aby se vždy snažil mít totožnou orientaci s úhlem mezi ním a dalším bodem jeho cesty vygenerované ve scénáři pomocí aplikace popsané v kapitole **??**. Zatáčení node řídí tak, že vypočte rozdíl mezi správnou a současnou orientací, který pak přenásobí koeficientem angle_sensitivity a pošle jako příkaz úhlové rychlosti pro Gazebo plugin libgazebo_ros_diff_drive.so. Pro její fungování potřebuje 5 parametrů:

1. target_frame - název objektu, jehož transformace (údaje o relativní poloze vzhledem ke středu) má node snímat
2. path_to_json - cesta k výše zmíněnému .json souboru.
3. angle_sensitivity - koeficient rozdílu úhlů - vyšší způsobí prudší zatačení.

4. min_dist - minimální vzdálenost (v m), na kterou se robot musí přiblížit k bodu, aby jej považoval za projetý a cílil k dalšímu.

5. max_speed - rychlost jízdy robota v m/s.