Lietajúce robotické systémy

Zadanie 1.

Arutori : Adrián Komorný , Martin Kochan

Vytváranie trasy letu drona

Pre tvorbu trasy sme použili python program v tutorialoch ktorý vyberal jednu prednastavenú mapu s vopred zadefinovanými súradnicami počiatku a konca, ktorý vytvorí detailnú trasu medzi nimi. Tento program sme následne modifikovali. Bola pridaná funkcia ktorá pretriedila vytvorenú trasu tak že nechala len hranové body v ktorých sa menil smer trasy robota nazvaná ako „shorten_path“ ďalej bola pridaná funkcia na rozšírenie stien mapy „enlarge_walls“, taktiež v hlavnom cykle init() bolo pridané načítavanie dát so súradnicami trasy drona zo súbora kde podľa z súradnice sa vyberie príslušná mapa na ktorej sa bude hľadať trasa ako aj sa zapíšu začiatočné a koncové súradnice na tej mape ako počiatočné a koncové súradnice pre hľadanie trasy potom ako sa vytvorí príslušná skrátená trasa ta sa zapíše na koniec celkovej trasy drona. Tento proces sa bude opakovať až dokedy nie je kompletná trasa vytvorená, tá sa potom vypíše do txt súbora a je následne používaná pre riadenie drona.

Funkcia shorten_path funguje nasledovne.Viď. obr.1

Na začiatku si vytvorí všetky potrebné premenne ktoré bude potrebovať, potom prejde k spracovávaniu hustej trasy ktorá je jej jediný vstupný parameter. Vo for cikle postupne prechádza všetkými bodmi a kontroluje sú body v smere a ak zaregistruje že sa smer zmenil zapíše predošlí bod ako hranový bod do výstupnej trasy. Spôsob kontroly smeru je nasledovný. Funkcia si pamätá rozdiely súradníc medzi kurentným a 1. predošlým bodom a porovnáva tieto rozdiely s rozdielmi medzi 2. predošlým a 3. predošlým bodmi a ak sa nezhodujú tak zapíše 1. predošlý bod ako bod zlomu (zmeny smeru). Vo for cykle sú taktiež podmienky zabezpečujúce všetky nevhodné stavy kedy bi to mohlo zlyhať.

Funkcia enlarge_walls funguje nasledovne. Viď. obr.2

Funkcia má dva vstupné parametre matica znakov reprezentujúcu mapu a počet prechodov spustenia čo znamená o koľko sa steny budú rozširovať. Na začiatku si vytvorí všetky potrebné premenne ktoré bude potrebovať ,nasledovne je while cyklus ktorý sa opakuje toľkokrát koľko sa má mapa rozšíriť. Vo while je séria for cyklov ktoré zabezpečujú rozšírenie stien. To je riešené tak že sa prechádza cez celú maticu mapy, kontroluje sa každý znak a ak je to stena tak funkcia prejde cez všetkých 8 bodov okolo a ak nie sú stena ale prázdna pozícia prepíše ich na značku reprezentujúcu novú stenu. To sa opakuje na celej mape. Potom prejde cez celú mapu znova a prepíše každý dočasný bod novej steny na skutočnú stenu. Následne sa celí cyklus opakuje vo while podľa toho koľko sa má stena rozšíriť.

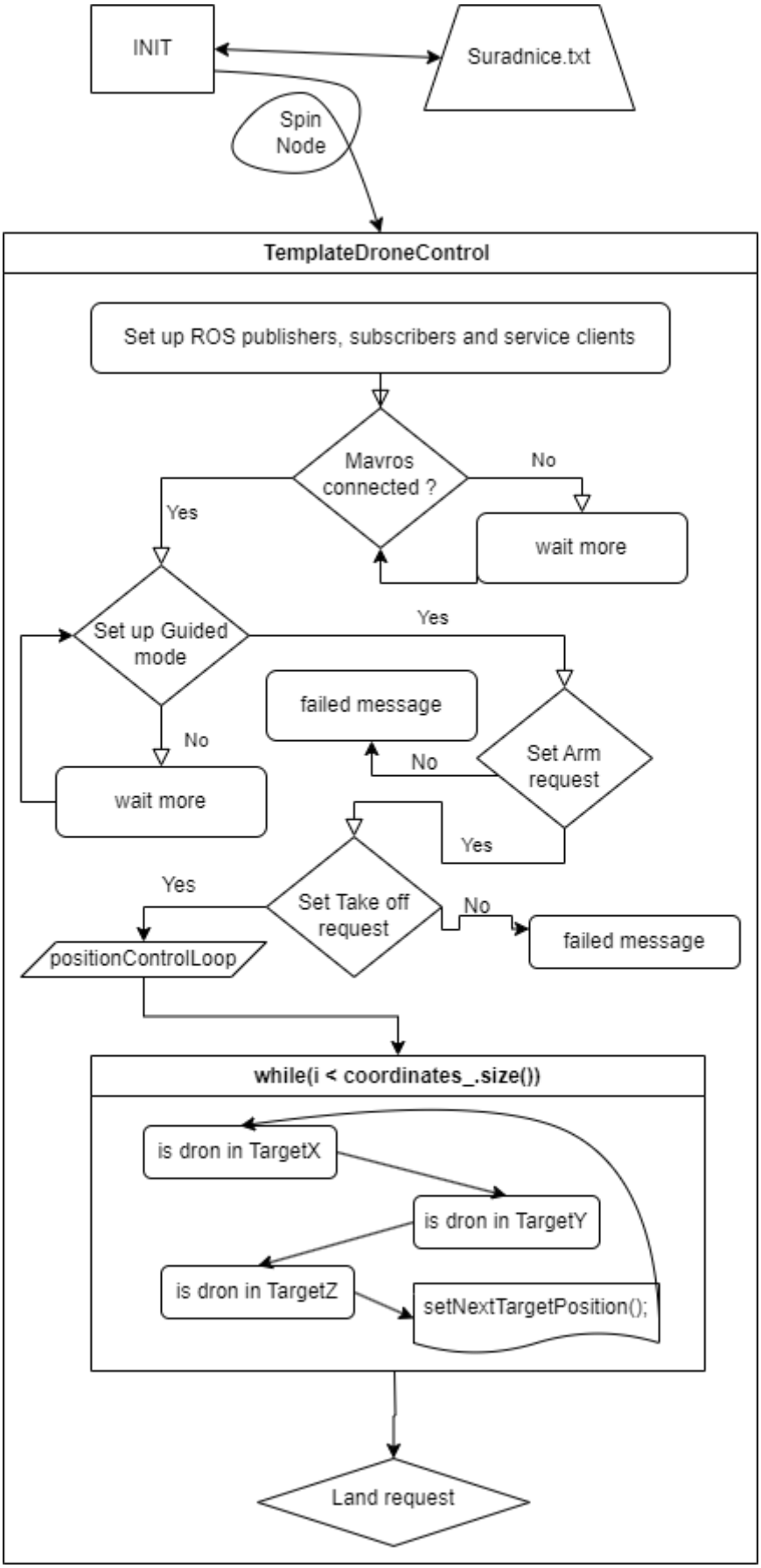
```python
def shorten_path(path):
    point_previous = path[0]
    point_previous2 = path[0]
    point_previous3 = path[0]
    point_previous4 = path[0]
    path_short = []
    sur1 = 0
    sur2 = 0
    index = 0
    index2 = 0
    check = 0

    for point_curent in path:
        if index == 0:
            path_short.append(point_curent)
            index2 += 1

        elif index <= 3:
            pass

        elif ((point_curent[0] - point_previous[0]) == sur3 and (point_curent[1] - point_previous[1]) == sur4) or check == 1:
            check = 0
            pass

        elif (point_curent[0] - point_previous[0]) != sur3 or (point_curent[1] - point_previous[1]) != sur4:
            path_short.append(point_previous)
            index2 += 1
            check = 1
```

Obrázok 1 Ukážka kódu funkcie shorten_path

```python
def enlarge_walls(filtered_data, runs):
    index = 0
    index2 = 0
    index3 = 0
    row_index = 0
    col_index = 0

    while index < runs:

        for row in filtered_data:
            for data in row:
                if filtered_data[row_index][col_index] == OBSTACLE_COL:
                    while index2 < 3:
                        while index3 < 3:
                            try:
                                if filtered_data[row_index - 1 + index2][col_index - 1 + index3] == "." :
                                    filtered_data[row_index - 1 + index2][col_index - 1 + index3] = FUTURE_OBSTACLE_COL
                            except:
                                pass
                            index3 += 1

                        index2 +=1
                        index3 = 0

                    index2 = 0
                    index3 = 0

                col_index += 1
```

Obrázok 2 Ukážka kódu funkcie enlarge_walls

ROS2


INIT
Suradnice.txt
Spin Node
TemplateDroneControl
Set up ROS publishers, subscribers and service clients
Mavros connected ?
No
Yes
wait more
Set up Guided mode
Yes
failed message
No
Set Arm request
No
wait more
Yes
Yes
Set Take off request
No
failed message
positionControlLoop
while(i < coordinates_.size())
is dron in TargetX
is dron in TargetY
is dron in TargetZ
setNextTargetPosition();
Land request