

ARBEITSPAKET 2

- Beigelegte Dokumente von „MobileSim“, „World Inc“ und die ReadMe des ARIA Programms durchgelesen
- MobileSim durch Kommandozeile gestartet, Befehl:
 - `"C:\Program Files\MobileRobots\MobileSim\MobileSim.exe" -m`
 - `"C:\Users\Sysadmin\Desktop\WS2018_projNav\HM_Karlstrasse_F8100_OG3_mod.map" -r "p3dx:robot1" --start "12710,11880,0" -r "p3dx:robot2" -r box`
- *Nullpunkt des Roboters festgelegt*

ARBEITSPAKET 3

- Vorgegebenes Matlab-File „HM_TA_Aria_getting_Started.m“ angepasst und Robotersteuerung hinzugefügt mit Hilfe der README-Datei des ARIA Programms
- Vorgefertigte Methoden verwendet(z.B. `arrobot_getvel(300)` zur Bewegung in einer Richtung)
- Tasteneingabe erfolgt über Kommandozeile(Matlab), WASD zur Fortbewegung und Drehung, B zum Beenden der Verbindung zum MobileSim Roboter

ARBEITSPAKET 4

- 16 Sonardaten relativ zum Roboter aus der WorldInc-Datei ausgelesen(Position,Lagewinkelxyalpha)
- Anhand der gemessenen Sensordistanzen(`arrobot_getsonarrange()`) werden durch Polares anhängen(relativer Winkel+Winkelroboter) und Aufaddieren der Lage des Roboters, die neuen XY-Koordinaten der Umgebung berechnet
- Plotten der Umgebung

ARBEITSPAKET 5

- Erstellen eines Konstruktors (`robotics.BinaryOccupancyGrid(25,25,7)`) mit vorgegebener Breite, Höhe und Auflösung
- Durch Aufrufen von `setOccupancy()` werden die verrechneten Koordinaten dargestellt und mit `show()` im Grid dargestellt

ARBEITSPAKET 6

- Zurückgelegter Roboterweg wird in einer txt-Datei abgespeichert(XY-Koordinaten) und im Occupancy Grid mit dem Befehl ‚p‘ abgebildet

ARBEITSPAKET 7

- Es werden Briefkästen in die Map miteingefügt, diese bekommen feste XY-Koordinaten
- Eine neue Methode „boxLocation“ berechnet die Distanz vom Roboter zu den Briefkästen
- Sobald ein Schwellwert von $<1.5\text{m}$ erreicht wird, wird die Distanz als ein String ausgegeben
- Es fehlt der UDP Server welcher diese Meldung ausgeben soll

ARBEITSPAKET 8
<ul style="list-style-type: none"> • Erstellen einer Methode „packageDropFinalize“, diese soll den Roboter ab dem Schwellwert autonom ansteuern • Löschen der erreichten Box, berechnen der nächsten Distanzen • Manuelles ansteuern der nächsten Box bis Schwellwert wieder erreicht, danach autonom • Winkel wird automatisch in die Richtung der Briefkästen angepasst
ARBEITSPAKET 9
<ul style="list-style-type: none"> • Funktion Homing definiert, diese bekommt vier feste Koordinate an denen sich der Roboter orientieren kann • Er fährt durch die Berechnung der Distanz zur nächst gelegenen Koordinate • Nach ansteuern der Koordinate bei welcher er bremst (durch einen Schwellwert), richtet sich der Roboter neu aus und „schaut“ in Richtung der nächsten Koordinate • Dieser Vorgang wird solange durchgeführt bis er den festgelegten Startpunkt erreicht
ARBEITSPAKET 10
<ul style="list-style-type: none"> • Erstellen der Funktion „collisionAvoidance“, diese ruft die Funktion (arrobot_getsonarrange()) auf und gibt uns die Reichweite der Sensoren wieder. • Die Methode wird vor der nächsten Bewegung ausgeführt und die Sensordaten geprüft • Durch einstellen eines Schwellwertes (<300) wird eine Meldung „Achtung Gefahr“ angezeigt, der Roboter bleibt stehen, dreht sich um 180° in die entgegengesetzte Richtung und fährt 0.3m weg • Es sollten die Sensoren noch einzeln geprüft werden, da die Türen ziemlich eng sind und die seitlichen Sensoren eine niedrigere Toleranz benötigen. Ansonsten führt der Roboter die Methode wieder aus und dreht ihn eventuell in die Wand.
ARBEITSPAKET 11
<ul style="list-style-type: none"> • Nicht durchgeführt
ARBEITSPAKET 12
<ul style="list-style-type: none"> • Nicht bearbeitet
ARBEITSPAKET 13
<ul style="list-style-type: none"> • Nicht bearbeitet
ARBEITSPAKET 14
<ul style="list-style-type: none"> • Nicht bearbeitet