Selbstfahrender Roboter mit LIDAR

Niklas Otten Yesenia Möhring

Abstract

This is the abstract.
It consists of two paragraphs.

Reflexion

Algorithmus

Nachdem wir uns in der Projektmanagement Woche das Ziel unseres Projektes ausformuliert hatten, begonnen wir uns zu überlegen wie wir aus zwei Punktewolken die Verschiebung zu errechnen. Wir hatten bereits die Idee uns auf gerade Wende zu konzentrieren und wussten auch schon, wie wir dieses Erkennen konnten. Um den Algorithmus weiter zu vereinfachen hatte ich die Idee, anstatt mit geraden mit den Schnittpunkten dieser Geraden zu rechnen. Der Vorteil dieser Methode liegt darin das ein Punkt nur aus einem Vektor besteht, im Gegensatz zu einer Geraden, welche aus Zwei Vektoren besteht, außerdem gibt es unendlich viele Möglichkeiten eine gerade zu speichern wohingegen punkte eindeutig sind.

Netzwerk

TCP

Um unsere App mit dem Roboter zu verbinden, brauchten wir ein einfaches Protokoll, um daten zu übertragen. Da wir Wlan als Übertragungsweg nutzen war das erste Protokoll, welches mir eingefallen ist HTTP, dieses wird auch für Internetseiten verwendet und ist weit verbreitet. Nach dem ich einen einfachen HTTP Server in Python aufgesetzt hatte wollte ich einen Client für diesen in Unity schreiben. Allerdings waren alle Informationen zu “Networking” in Unity mit TCP, daher wäre es in Unity wesentlich aufwändiger gewesen einen HTTP Client zu schreiben, und so habe ich mich für ein einfacheres Protokoll entschieden, TCP, welches das HTTP zugrunde liegende Protokoll ist.

Protokoll

TCP gibt uns die Möglichkeit daten zu übertragen, um das Protokoll leicht verständlich zu machen, was auch bei der Fehlersuche hilft, haben wir uns entschieden Text zu senden welcher von Menschen gelesen werden kann.

Eine Beispielnachricht wäre

```
roboter summlidardata 20
```

Das erste Wort der Nachricht ist der Ort, an welchen es geleitet wird, die Idee hierbei ist das dieses Wort dem TCP Server/Client sagt zu welchem weiterem Programm es die empfangenen daten senden soll. Zum Beispiel war ursprünglich die Idee das es ein Python Programm gibt, welches daten empfängt, um den Roboter zu Bewegen und ein C++ Programm, welches die LIDAR daten des Sensor auswertet. Das Zweite Wort ist der Befehl, dieser sagt dem Programm was nun zu tun ist. In diesem Fall Daten messen und zurücksenden. Das letzte Wort ist das Argument, dieses gibt extra Informationen zu dem Befehl, in diesem Fall das der Sensor 20 Umdrehungen machen soll.

Um die daten im Argument besser zu organisieren haben wir uns ein System ausgedacht, um dies zu tun. Zuerst können einzelne Informationen durch Kommata getrennt werden. Dies ist nötig da zum Beispiel die Lidar daten aus vielen Messpunkten bestehen. Allerdings besteht jeder einzelne Messpunkt aus zwei Zahlen. Daher werden diese mit dem nächsten trenn Zeichen getrennt. Wir hätten für solche Daten auch Klammern nutzen können, allerdings sind diese wesentlich Schwieriger auszuwerten als diese Zeichen. Denn die meisten Programmiersprachen bringen bereits die Möglichkeit mit Zeichenketten an bestimmten Zeichen aufzuteilen. Dies macht es sehr einfach ein Argument Rekursiv zu verarbeiten, man teilt die erste Ebene nach den Leerzeichen auf, ersetzt Kommata durch Leerzeichen und die anderen trenn Zeichen mit dem trenn Zeichen der Ebene darüber und gibt diese Liste an Argumenten einzeln an die entsprechende Funktion.

Ein Beispiel ist die Multi Funktion, welche mehrere Bewegung Befehle an den Roboter sendet.

```
multi move,1;1;1,rotate,90;1
```

Da Multi von demselben Code interpretiert wird wie move und rotate, könnte man sogar mehrere multi Funktionen ineinander schachteln, auch wenn dies keinen Sinn erfüllt. ### Python zu C++ Ursprünglich hatte ich den gesamten code, welcher auf dem Roboter lief in Python geschrieben. Allerdings war es nicht möglich den code für den LIDAR Sensor in Python zu schreiben, da Python zu langsam für diesen Sensor war daher musste der Code, um den Sensor auszulesen in C++ geschrieben werden. Da wir allerdings schon den TCP Server und die Motorsteuerung in Python implementiert hatten wollten wir einfach ein Programm schreiben, welches den Lidar ausliest und diese Daten zurück an Python überträgt. Da Tim, welcher den C++ Code geschrieben hatte, dies aber nicht zum Laufen bringen konnte, haben wir uns entschieden den code in C++ umzuschreiben. Zum Glück war das Python Programm nur 122 Zeilen lang und so recht einfach umzuschreiben.

AI

Da unser Projekt eine Karte und eine Position erzeugt, ist es eine Gute Demonstration wenn der Roboter in der Lage ist selbstständig zu angegebenen Zielen zu navigieren. ### A *Ich habe mich für A* entschieden da es ein weit verbreiteter Pathfinding Algorithmus ist und es gute Erklärungen gibt wie dieser Funktioniert. Um A* nutzen zu können und weil es allgemein das Nutzen unserer Daten einfacher macht, haben wir unsere Messdaten Rastarisiert, das heißt wir speichern mit einer Auflösung von einem Zentimeter, ob ein Punkt eine Wand ist oder nicht. Man gibt diesem Algorithmus nun einen start- und Endpunkt zusammen mit einer Liste der Wände und er findet einen Weg. Ein Problem, welches ich zu Beginn hatte, war das A* seine Daten in einer Liste gespeichert hatte, was sehr ineffizient ist. Daher habe ich einen Sogenannten Heap implementiert. Dies reduzierte die Laufzeit drastisch.

Fazit

Meine Aufgabe in diesem Projekt waren das Networking, die AI und einwenig UI. Zu allen diesen Themen gab es relativ viele Beispiele im Internet was den Prozess relativ einfach für mich gemacht hat. So habe ich leicht noch ein par Dinge über Unity lernen können, und habe auch eine neue Datenstruktur kennengelernt, den Heap. Der größte unterschied zu vorherigen projekten war, das dieses tatsächlich von anderen Menschen benutzt werden sollte. Was bedeutet das wir nicht mit einem „Proof of Concept“ aufgehört haben.

Was noch zu tun ist

Ein Ziel welches für mich noch offen geblieben ist, wäre einen Algorithmus zu schreiben welcher autonom einen Raum erkundet und volständig kartiert.

Quellen

26.01.2012

installed software, created Unity project. ## 28.01.2021 HTTPs://gist.github.com/danielbierwirth/0636650b005834204cb19ef5a
HTTPs://forum.unity.com/threads/how-do-i-detect-when-a-button-is-being-pressed-held-on-eventtype.352368/

09.02.2021 HTTPs://www.youtube.com/watch?v=0G4vcH9N0gc HTTPs://pressstart.vip/tutorials/2018/07/12/44/pan–zo

A*

HTTPs://www.youtube.com/watch?v=-L-WgKMFuhE HTTPs://github.com/SebLague/Pathfinding/tree/master/Episode%20
%20heap ## 12.02.2021 HTTPs://pressstart.vip/tutorials/2018/06/22/39/mobile-joystick-in-unity.html ##
27.03.2021 Yesenia geholfen die Roboter Simulation zu schreiben