**Systemplanung und Projektentwicklung 4, Projektbeschreibung**

**Einleitung:**

Der MBot2 von mblock ist ein Roboter, welcher zum spielerischen Erlernen von Programmierkenntnissen entworfen wurde. Der Roboter besitzt viele verschiedene Sensoren, mit denen er mit seiner Außenwelt kommunizieren kann.

Programmiert kann er über die Programmiersprache „Scratch“ oder MicroPython werden. Bei MicroPython handelt es sich um eine „abgespeckte“ Variante von Python3, welche speziell für die Verwendung auf kleinen, leistungsschwächeren Microcontrollern optimiert wurde. Eine kleine Controllereinheit „cyberpi“ welche auf den mbot2 gesteckt wird, sorgt für die Kommunikation. In dieser Controllereinheit sind auch das WLAN-Module ein Lagesensor, Display und weitere Komponenten untergebracht.



Sensoren/Module des mbot2:

* Ultraschall-/Abstandssensor
* Lichtsensor
* LEDs
* Motoren
* Lagesensor
* Display
* Joystick

**Anforderungen:**

Es soll eine Anwendung entwickelt werden, welches es erlaubt den Roboter aus der Ferne zu steuern. Die Anwendung soll dabei auf einem Computer genauso laufen wie auf einem mobilen Endgerät. Da eine Sichtverbindung zwischen Roboter und Anwendung notwendig ist, kann davon ausgegangen werden, dass beide Geräte immer im selben Netzwerk sind. Roboter und Anwendung sollen sich selbständig im Netzwerk „finden“. Erst wenn eine Art Verbindung zwischen beiden hergestellt wurde, soll der Roboter Befehle entgegennehmen und Daten an die Anwendung senden. Darüber hinaus soll es auch möglich sein, über die Anwendung mehrere Geräte zu steuern (wenn auch nicht gleichzeitig). Der gerade verbundene Roboter soll mit einem eindeutigen Namen und dessen IP-Adresse angezeigt werden.

In der Anwendung sollen die Werte sämtlicher eingesetzter Sensoren/Aktoren angezeigt- und in regelmäßigen Abständen aktualisiert werden. Mit Hilfe der Anwendung soll es möglich sein dem Roboter Steuerbefehle zu senden. (Bewege dich vorwärts, rückwärts, links, rechts). Die Geschwindigkeit soll ebenfalls angegeben werden können.

Ein Sicherheitsmodus (der ein/ausgeschaltet werden kann) soll sicherstellen das der Roboter nicht mutwillig gegen eine Wand gefahren werden kann (suicide-prevention-feature).

Netzwerkanbindung:

Wird der Roboter eingeschaltet soll er sich automatisch mit dem Schul-WLAN verbinden. Die LEDs der Controller-Einheit sollen dabei den Status der Verbindung anzeigen.

* + Blau Verbindung wird aufgebaut;
  + Grün WLAN Verbindung erfolgreich;
  + Rot WLAN Verbindung fehlerhaft;

Konnte eine Verbindung aufgebaut werden, so soll für 5 Sekunden am Display der Controllereinheit die IP-Adresse angezeigt werden.

Es soll auch einen Modus geben, in dem der Roboter automatisch einer Linie folgt. Die Steuerbefehle sollen in diesem Modus weiterhin von der Anwendung kommen. Dazu wird diese regelmäßig die Werte des Lichtsensors auswerten müssen.

Die Farben der 5 LEDs der Controllereinheit sollen über die Anwendung eingestellt werden können.

**Projektmanagementanforderungen:**

Das Projekt soll mit Hilfe agiler Methoden (SCRUM) umgesetzt werden. Der Quellcode sowie sämtliche Dokumentation ist in einem GITHUB Repository abzulegen. Obwohl das Projekt mit agilen Methoden realisiert werden soll, ist es erforderlich, sich zu Beginn gewisse Fragen zu stellen und zu klären:

* Variantenbildung
* Machbarkeitsstudie
* Grobe Zeitplanung (welche, sobald die Teamvelocity bekannt ist, verfeinert werden soll).

Führen Sie zu Beginn einen „Sprint 0“ durch in welchem sie die Grobplanung vornehmen, UserStories schreiben und schätzen und ein grundsätzliches Design der Software vornehmen sowie sämtliche Software, die für die Entwicklung benötigt wird, installieren und entsprechend einrichten.

Eine Machbarkeitsstudie soll sicherstellen das folgende Eigenschaften realisierbar sind:

* Programmierung des mbot2 mit Hilfe von MicroPython
* WLAN-Konfiguration und Netzwerkkommunikation des mbot2 und einem Computer über TCP/UDP.
* Kommunikation/Ansteuerung folgender Sensoren/Module mit Hilfe von MicroPython: Display, Abstandssensor, Lichtsensor, Lagesensor, Motoren, LEDs

Dokumentation des Projektes:

* Softwarearchitektur
* Klassendiagramme
* Sequenzdiagramme, Aktivitätsdiagramm
* Komponentendiagramm, Verteilungsdiagramm
* Interaktionen
* Use-Case Beschreibung
* Sprintplanung
* Sprint-Demo
* Sprint-Retrospektive und Impediment-Management
* Projektcontrolling

Für die Projektdokumentation wird ein Template-File zur Verfügung gestellt.

Projektlaufzeit: 10. Jänner 2024 - 12. Juni 2024