metin, simge, sembol, logo, amblem içeren bir resim

Açıklama otomatik olarak oluşturuldu

**T.C.**

**ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ**

**MÜHENDİSLİK – MİMARLIK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ**

**NESNE TABANLI PROGRAMLAMA I PROJE RAPORU**

**Yürütücü: Prf. Dr. METİN ÖZKAN**

**“Her Yöne Hareketli (Omni-directional) Robot Denetim Sistemi”**

**Üyeler**

**152120211127 - FURKAN GEMİCİ**

**152120211082 - YAKUB ŞİYAR DEMİRBAŞ**

**152120211071** - **CEM LEVENT AVCI**

**152120211039 – EMİRHAN KALKAN**

**ARALIK 2024**

İçindekiler Tablosu

[1.GİRİŞ 3](#_Toc185795763)

[1.1 Projenin Amacı 3](#_Toc185795764)

[1.2 Kapsam 3](#_Toc185795765)

[1.3 Kullanılan Teknojiler 3](#_Toc185795766)

[2.TASARIM 4](#_Toc185795767)

[2.1 Genel Sistem Tasarımı 4](#_Toc185795768)

[2.2 Görev Planlama 4](#_Toc185795769)

[2.3 Örnek Girdiler ve Çıktılar 6](#_Toc185795770)

[3. SONUÇLAR VE DEĞERLENDİRME 12](#_Toc185795771)

[3.1 Proje Sonuçlarının Değerlendirilmesi 12](#_Toc185795772)

[3.2 Takım Çalışması Hakkında Yorumlar 12](#_Toc185795773)

[3.3 Artılar ve Eksiler 12](#_Toc185795774)

[3.4 Öneriler 13](#_Toc185795775)

# 1.GİRİŞ

## 1.1 Projenin Amacı

Bu proje, Omni-directional (her yöne hareket edebilen) bir robotun kontrolünü ve sensör verilerinin yönetimini sağlamak amacıyla tasarlanmıştır. Robotun, simülatör ortamında hareket kontrolünü ve çevresel verilerin işlenmesini kolaylaştıran bir yazılım geliştirilmiştir. Webots simülatörü kullanılarak yürütülen bu proje, robotik sistemlerin geliştirilmesi ve yönetilmesi konusunda bir temel oluşturmayı amaçlamaktadır.

## 1.2 Kapsam

Proje, robotun hareket kontrolü, güvenli navigasyonu, sensörlerden veri toplanması, haritalama işlemleri ve bu işlemlerin yönetilmesi gibi alanları kapsamaktadır. Robot, verilen görevleri yerine getirebilmek için çeşitli sınıflar ve metotlar aracılığıyla programlanmıştır. Bununla birlikte, yazılım yalnızca simülatör ortamında çalışacak şekilde sınırlanmıştır. Harici donanımlarla entegrasyon ve gerçek dünya ortamında çalışma gibi özellikler bu projenin kapsamı dışında bırakılmıştır.

## 1.3 Kullanılan Teknojiler

Bu projenin geliştirilmesinde, C++ programlama dili kullanılmış ve nesne yönelimli programlama prensiplerinden faydalanılmıştır. Webots simülatörü, robotun kontrol ve test sürecinde ana platform olarak kullanılmıştır. Ayrıca, FestoRobotAPI adlı sınıf kütüphanesi, robotun hareket ve sensör yönetimi gibi işlevlerini yerine getirmek için projeye entegre edilmiştir. Yazılım geliştirme sürecinde modüler bir yapı benimsenmiş, UML diyagramları ve test programları ile tasarım desteklenmiştir.

# 2.TASARIM

## 2.1 Genel Sistem Tasarımı

Bu projede, Omni-directional robotun hareket kontrolünü sağlamak ve sensör verilerini işlemek üzere modüler bir sistem tasarımı benimsenmiştir. Sistemin genel yapısı, nesne tabanlı programlama prensipleri doğrultusunda sınıflar ve alt sistemler olarak organize edilmiştir.Aşağıda da menü interface kısmının uml diyagramı verilmiştir.

metin, ekran görüntüsü, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

## 2.2 Görev Planlama

Proje ekibindeki her bir üyenin üstlendiği görevler şu şekilde planlanmıştır:

1. **Cem**:
   * Robotun hareket kontrolünü gerçekleştiren **RobotController** sınıfını geliştirmiştir.
   * Robotun pozisyon bilgilerini yöneten **Pose** modülünü oluşturmuştur.
   * Gerekli sınıflar oluşturulduktan sonra arayüz entegrasyonu için aşağıdaki dosyalar üzerinde çalışmıştır:
     + **App.h**
     + **Menus.h**
     + **MotionMenu.h**
     + **Robot.h**
     + **RobotInterface.h**
   * Projede kullanılacak olan sınıflar ve modüller için **Doxygen** belgelendirme çalışmaları yapmıştır.
   * Projenin son halini Bitbucket’a yükledi.

1. **Furkan**:
   * Robotun çevresini algılaması için **Lidar Sensor** modülünü kodlamıştır.
   * Robotun mesafesi için sensörlerden gelen verileri işlemek amacıyla **IRSensor** modülünü geliştirmiştir.
   * Güvenli bir şekilde hareket etmesini sağlayan **Safe Navigation** algoritmalarını geliştirmiştir.
   * Aşağıdaki modüller ve yapılar üzerinde çalışmıştır:
     + **FestoRobotInterface.h**
     + **FestoRobotSensorInterface.h**
     + **SensorInterface.h**
     + **SensorMenu.h**
   * **Giriş** kısmında projenin tanıtımını yazmıştır.
   * **İçindekiler** ve **Kapak Sayfası** tasarımlarını hazırlamıştır.
2. **Emirhan**:
   * Robotun çevresel haritasını oluşturmak için **Map** modülünü geliştirmiştir.
   * Çevresel verileri işleyen **Mapper** modülünü oluşturmuştur.
   * Toplanan verileri kaydeden **Record** modülünü yazmıştır.
   * **Tasarım** kısmında UML diyagramlarının oluşturulması ve açıklamalarının yazılmasını üstlenmiştir.
   * Görev dağılımlarının açıklamalarını düzenlemiştir.
   * Aşağıdaki modüller üzerinde çalışmıştır:
     + **ConnectionMenu.h**
     + **MapManager.h**
     + **MapMenu.h**
3. **Yakub**:
   * Robotun kullanıcı tarafından yönetilmesini sağlayan **Robot Operator** sınıfını geliştirmiştir.
   * Veri güvenliğini artırmak için **Encryption** algoritmalarını uygulamıştır.
   * Robotun hedef noktalarını yöneten **Point** modülünü kodlamıştır.
   * Kullanıcı kimlik doğrulama işlemleri için **Authentication.h** modülünü geliştirmiştir.
   * Doküman düzenlenmesinde yardımcı olundu.

## 2.3 Örnek Girdiler ve Çıktılar

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi, siyah beyaz içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, yazı tipi, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, ekran, görüntüleme, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, multimedya yazılımı, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, ekran, görüntüleme, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

# 3. SONUÇLAR VE DEĞERLENDİRME

## 3.1 Proje Sonuçlarının Değerlendirilmesi

Proje kapsamında belirlenen hedeflerin büyük bir kısmına ulaşılmış olup, genel olarak istenen işlevsellikler yerine getirilmiştir. Proje süresince Mapper, MAP ve diğer sınıfların geliştirilmesi sırasında bazı teknik zorluklarla karşılaşılsa da, ekip olarak bu sorunlara çözüm bulma konusunda etkili bir çaba gösterilmiştir. Özellikle MAP sınıfının işlevsellikleri, haritalama ve gösterim konularında istenen sonuçları vermemiştir. Ancak, sınıfların entegrasyonu sırasında ortaya çıkan hatalar, projeyi tamamlamada beklenenden daha fazla zaman harcamamıza neden olmuştur.

## 3.2 Takım Çalışması Hakkında Yorumlar

Proje süresince takım olarak etkili bir işbirliği gerçekleştirilmiştir. Birden fazla kez fiziksel veya çevrimiçi buluşmalar düzenlenerek grup çalışması yapılmış ve proje üzerinde ortak bir şekilde ilerlenmiştir. Ekip üyelerinin düzenli iletişim kurması ve görev paylaşımında esnek davranması, projenin ilerlemesini olumlu yönde etkilemiştir. Ayrıca, Github kullanılarak kodların versiyon kontrolü sağlanmış, yapılan değişikliklerin takibi kolaylaştırılmıştır. Bu yöntem, takım çalışmasındaki verimliliği önemli ölçüde arttırmıştır.

## 3.3 Artılar ve Eksiler

**Artılar:**

* Takım üyeleri arasındaki iletişim ve işbirliği oldukça başarılıydı.
* Github kullanımı sayesinde kodlama süreci düzenli bir şekilde ilerledi ve hata takibi kolaylaştı.

**Eksiler:**

* Bazı sınıfların entegrasyonu sırasında teknik zorluklarla karşılaşıldı ve bu sorunlar ekstra zaman kaybına neden oldu.
* Bazı sınıflarda fark edilmeyen hatalar nedeniyle, beklenenden daha fazla düzeltme yapılması gerekti.
* Proje sürecinde verilen sürenin kısa olması nedeniyle istenilen verimde çalışmalar yapılamadı.

## 3.4 Öneriler

* Gelecek projelerde, sınıfların entegrasyonunun daha kolay ve uyumlu olacağı bir yapı tercih edilmelidir.
* Takım çalışmasına daha uygun ve zorluk seviyesi daha düşük bir ödev verilmesi, sürecin daha verimli geçmesini sağlayabilir.
* Proje kapsamı ve zorluk derecesi, belirlenen zaman dilimine uygun şekilde optimize edilmelidir.
* Daha sık aralıklarla kodlama oturumları düzenlenerek, hataların erken tespit edilmesi sağlanabilir.