

ESKISEHIR OSMANGAZI UNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



OOP

Dönem Projesi

152120181126
İsmail Can Durak

Giriş

Proje konusu, 3B robot simülatöründeki robotun denetimi (hareket ettirilmesi ve sensörlerinden veri alınması) için yazılım geliştirmektedir. Bu projede, Webots adlı simulator programı ve bu programdaki robotu denetleyebilmek için gereken sınıf kütüphanesi verilmektedir.



Görev Planlama

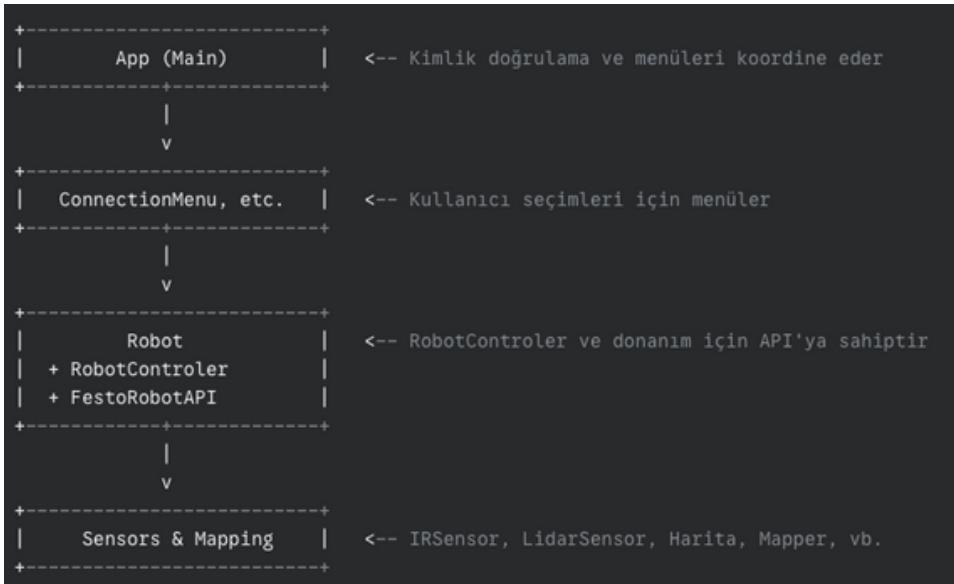


sadece benim tarafımdan
geliştirilmiştir

Genel Tasarım ve Mimari

Proje, her ana alanın özel bir sınıf veya alt sistemde kapsüllendiği modüler bir OOP yapısını takip eder. Kapsayıcı App sınıfı, aşağıdakileri koordine eder:

- Kimlik Doğrulama (kullanıcı kimlik bilgilerini doğrulama)
- Menü Navigasyonu (Bağlantı, Hareket, Sensör alt menüleri)
- Robot Etkileşimi (Robot ve RobotControler sınıfları aracılığıyla)



App, kullanıcıyı Authentication aracılığıyla kimlik doğrular. Başarılı olursa, kullanıcı birden fazla menü (örneğin, ConnectionMenu, MotionMenu, SensorMenu) ile etkileşime girer. Her menü, Robot'u kontrol edebilir veya sorgulayabilir; Robot, düşük seviyeli komutları RobotController'a ve ardından FestoRobotAPI'ya devreder. IRSensor ve LidarSensor, gerekirse ortam verilerini toplar. SafeNavigation, çarpışmaları önlemek için kullanılabilir. Harita ve Mapper, ortam haritalaması için sensör verilerini depolar veya yorumlar.

Menüler ve Etkileşim Akışı

Birincil kullanıcı deneyimi, bir dizi metin tabanlı menü üzerinden gerçekleşir. Örneğin:

- Kimlik Doğrulama
 - Kullanıcıdan ad, soyad ve erişim kodu istenir.
 - Kullanıcı girişini, RobotOperator'daki şifrelenmiş kodla karşılaştırır.
 - Eşleşme yoksa uygulama sona erer.
- Ana Menü
 - Bağlantı
 - Hareket
 - Sensör
 - Çıkış
- Alt Menüler
 - ConnectionMenu, robot'a bağlanabilir/bağlantıyı kesebilir veya geri donebilir/çıkabilir.
 - MotionMenu, ileri/geri git, dön, robot bilgisi yazdır veya çıkış gibi komutlar sunar.
 - SensorMenu, IR veya Lidar verilerini gösterebilir.

1. Connection

2. Motion

3. Sensor

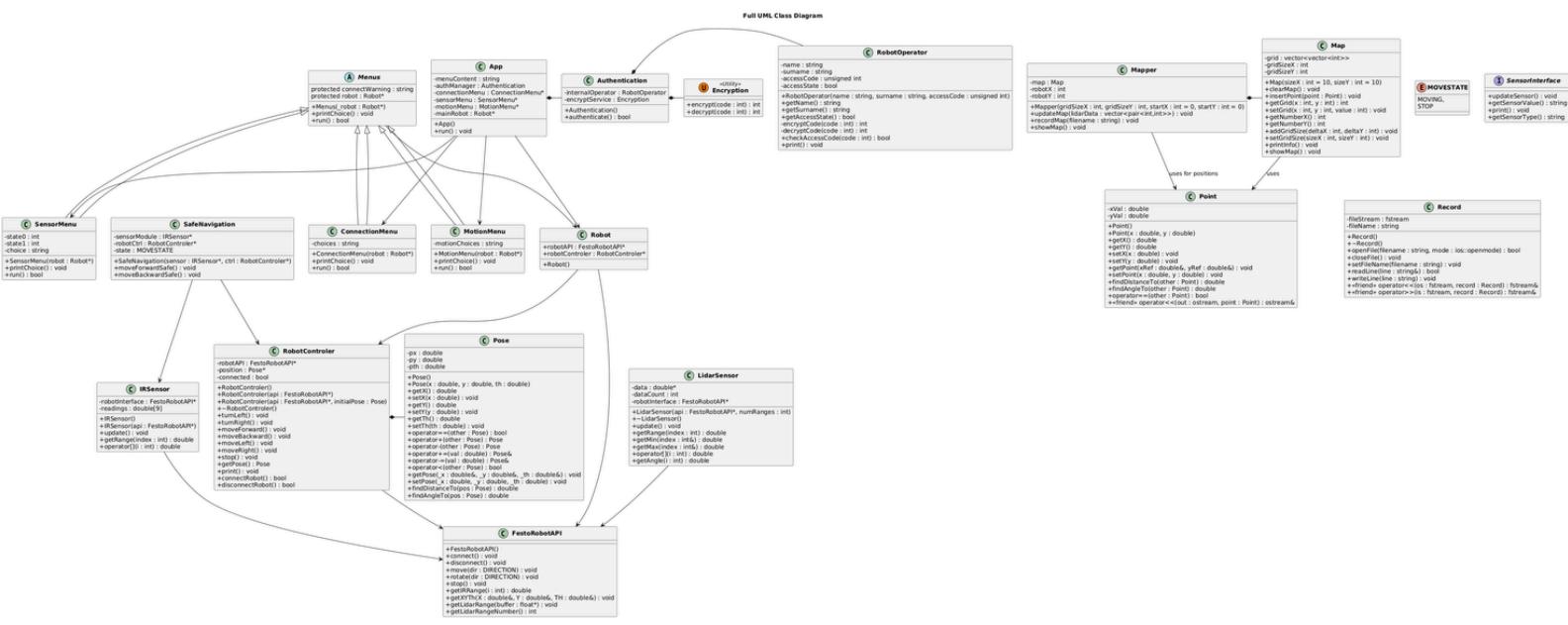
4. Quit

Önemli Sınıflar ve Sorumlulukları

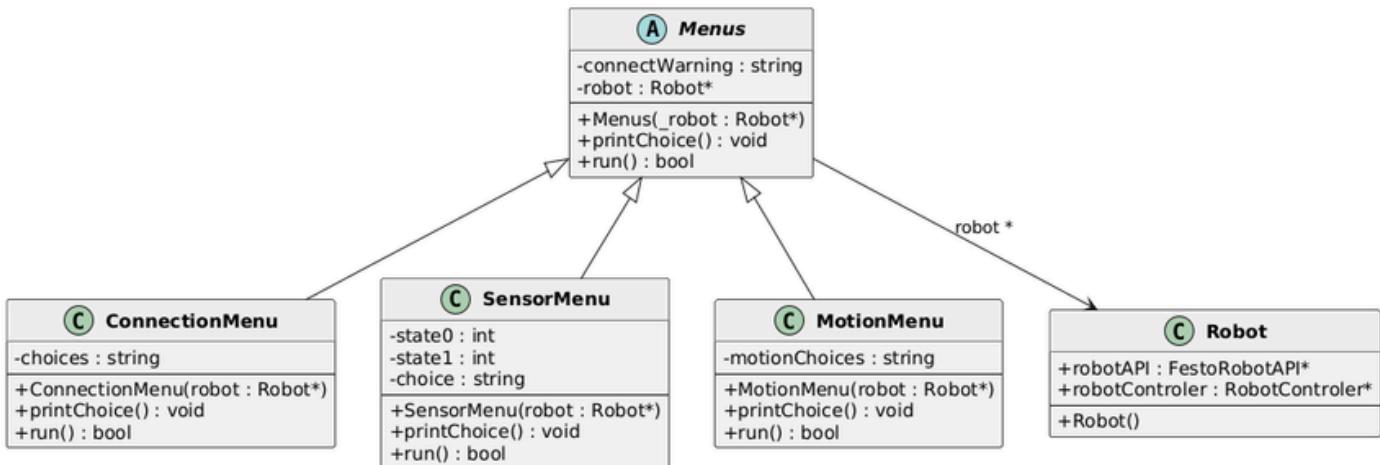
- App
 - Amaç: Oturum açma ve ana menü akışını yönetir.
 - Ana Metotlar: run().
- Authentication
 - Amaç: Kullanıcının adını, soyadını ve kodunu toplar; şifrelenmiş bir kodla karşılaştırır.
 - Ana Metotlar: authenticate().
- Menus (soyut temel)
 - Amaç: Menü seçeneklerini yazdırmak ve kullanıcı giriş döngülerini çalıştırmak için birleşik bir arayüz sağlar.
 - Türetilmiş: ConnectionMenu, MotionMenu, SensorMenu.
- Robot
 - Amaç: Donanım API'sine (FestoRobotAPI) ve kontrol edici mantığa (RobotController) atıfta bulunan merkezi nesne.
- RobotController
 - Amaç: Hareket komutlarını (dön, hareket et, dur) FestoRobotAPI'ye verir. Pose'u izler veya günceller.
- FestoRobotAPI
 - Amaç: Robotun gerçek donanım özelliklerini taklit eder/simüle eder. Bağlan, bağlantıyi kes, IR oku, hareket et, döndür, vb.
- IRSensor, LidarSensor
 - Amaç: IR veya Lidar'dan ortam verilerini alır. Muhtemelen SafeNavigation veya SensorMenu tarafından kullanılır.
- SafeNavigation
 - Amaç: IR verilerini RobotController'a güvenli hareket komutlarıyla birleştirerek çarpışma önleme sağlar.
- Map, Mapper

 - Amaç: 2B ızgara gösterimini depolamak ve güncellemek. Sensör veri noktalarını (örneğin engelleri) bir ızgara içine ekleyin.
- SensorInterface
 - Amaç: Sensör sınıflarını (updateSensor(), getSensorValue(), vb.) tanımlamak için tekdüze bir yol istiyorsanız soyut bir arayüz.

UML Diagramları



Menus UML Diagram



Örnek girdi ve çıktılar

Kimlik Doğrulama

```
Enter your first name: John  
Enter your last name: Smith  
Enter access code: 1234
```

```
Access Granted! Welcome, John Smith!
```

ConnectionMenu

```
Connection Menu  
  
1. Connect Robot  
2. Disconnect Robot  
3. Back  
4. Quit  
  
[Robot connected successfully...]
```

MotionMenu

```
Motion Menu  
  
1. Move Robot  
2. Safe Move Robot  
3. Turn Left  
4. Turn Right  
5. Backward  
6. Get Info  
7. Back  
8. Quit  
  
Moving forward...  
RobotController moved forward.
```

SensorMENU

```
Sensor Menu  
  
1. IR Sensor Info  
2. Lidar Sensor Info  
3. Back  
4. Quit  
  
IR Sensor [0]: 0.45  
IR Sensor [1]: 0.60  
IR Sensor [2]: 0.12
```

Mapper Recording

```
Map before sensor data:  
.....  
.....  
...  
  
Updating map with Lidar data...  
Recording map to 'testMapOutput.txt'...
```

Sonuç

Sağlam, Modüler Tasarım

Nesne yönelimli ilkelerin (kapsülleme, kalıtım, çok biçimlilik) kullanılmasıyla, proje, her bileşeninin – Menüler, RobotController, Sensörler ve Mapper – sürdürülebilir ve yeniden kullanılabilir bir modüle ayrıldığı temiz bir mimariye ulaşır.

Genişleme Kolaylığı

Soyut temel sınıflar (örneğin, Menus, SensorInterface), yeni işlevlerin, örneğin ek menüler veya özel sensörler gibi, mevcut kodu bozmadan sorunsuz bir şekilde eklenmesini sağlar. Bu, sistemi son derece genişletilebilir kılar.

Geliştirilmiş Kullanıcı Deneyimi

Sistemin konsol tabanlı menü akışı, kullanıcı etkileşimini basitleştirir. Teknik olmayan operatörler veya öğrenciler, robot'a kolayca bağlanabilir, hareket ettirebilir ve izleyebilir. Bu tasarım, çekirdek işlevleri yeniden kullanarak potansiyel yükseltmelere (örneğin, GUI veya web ucu) zemin hazırlar.

Güvenli Navigasyon ve Haritalama

SafeNavigation (çarpışmalardan kaçınmak için) ve Mapper (sensör verilerini bir ızgara üzerinde kaydetmek için) gibi özellikler, sistemin engel tanıma, ortam haritalama ve rota planlama gibi gerçek dünya robotik kullanım durumları için potansiyelini gösterir.

Test Edilebilirlik ve Güvenilirlik

Bireysel olarak test edilebilen bileşenler (her sınıf için test programları tarafından desteklenir), doğruluk konusunda güveni artırır. Bu, sürekli entegrasyon ve gelecekteki değişiklikleri daha kolay ve daha hatasız hale getirir.

Eğitsimsel Değer

Bu proje, robotikte OOP tasarımlının kapsamlı bir örneğini oluşturur.

Öğrenciler veya genç geliştiriciler sunları öğrenir:

- Uygulama mantığını (menüler) iş mantığından (robot kontrolü) ayırma
- Kimlik doğrulama modülü uygulaması
- Navigasyon kararlarına sensör verilerini entegre etme
- Ölçeklenebilirlik için sınıfları ve ilişkileri düzenleme

Sonuç olarak, proje tam işlevsel, modüler bir robotik uygulaması sunar.

Mimarisi, hem sınıf içi öğrenmeyi hem de gerçek robotik sistem entegrasyonunu destekler ve daha sofistike robotik algoritmalar, GUI'ler veya ek sensörlerin üzerine inşa edilebileceği bir temel sağlar.