PTP-01

SENSOR VERİSİNDEN FAYDALANARAK HEDEF KONUMU TESPİTİ

	Ad Soyad	**İmza**
Hazırlayan	Tuğba Tosun	

GENEL- PUBLIC

İÇİNDEKİLER

GENEL- PUBLIC

ŞEKİL LİSTESİ

GENEL- PUBLIC

TABLO LİSTESİ

Şekil tablosu öğesi bulunamadı.

GENEL- PUBLIC

1. KAPSAM

Bu doküman sensör konum ve kerteriz verisinden faydalanarak hedef tespiti yapan uygulama kapsamını açıklamaktadır.

Proje aşağıdaki github adresinde paylaşılmıştır.

http: https://github.com/tugba-tosun/PredictTargetPosition.git

SSH: git@github.com:tugba-tosun/PredictTargetPosition.git

GENEL- PUBLIC

2. GİRİŞ

2.1. Dünya

1000x1000 boyutunda bir alan

Bu alan üzerinde herhangi bir yerde konumlanabilecek 2 adet sensor

Bu alan üzerinde herhangi bir yerde konumlanacak 1 adet hedef

Sensörlerden alınan verileri işleyecek merkezi birim

2.2. Görevler

Her bir sensor kendi konumlandığı lokasyonu merkezi birime iletecektir.

Her bir sensor tespit ettiği hedefin sadece kerteriz bilgisini hesaplayarak merkezi birime iletecektir. Merkezi birim sensörlerden aldığı bilgileri işleyerek hedefin koordinat sisteminde konumladığı noktayı hesaplayarak sergileyecektir.

2.3. Notlar

Sensör ve işleyici birim uygulamaları Java ile geliştirilecektir.

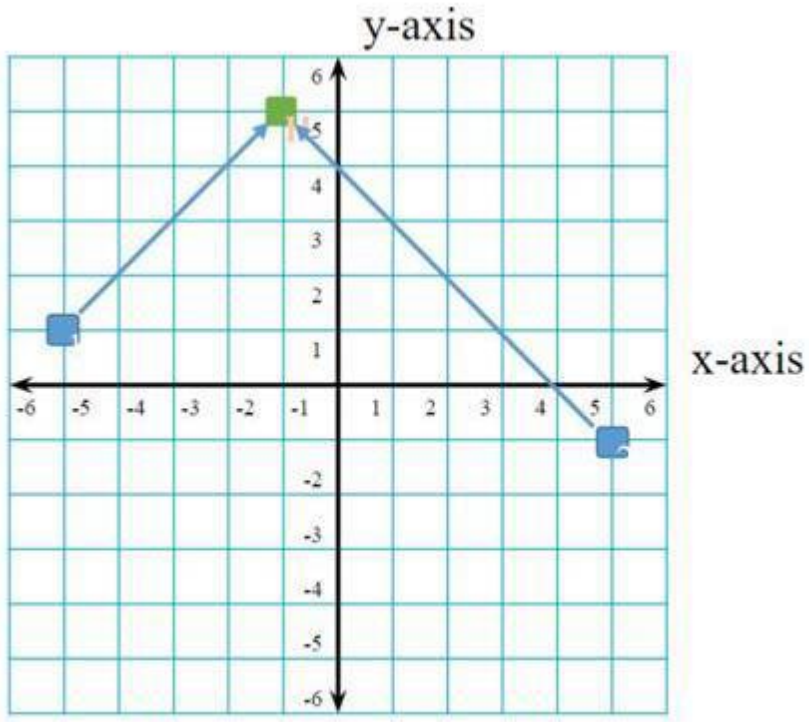
Uygulamalar arası haberleşme için Apache Kafka kullanılacaktır.

Proje kapsamında; projenin derleme ve çalıştırma aşamalarının açıklandığı, kullanılan altyapı bilgilerinin (kütüphane, teknoloji,versiyon vb.) belirtildiği bir rapor hazırlanması beklenmektir.

2.4. Örnek

- Sensör1: (-5,1) Noktasında
- Sensör2: (5,-1) Noktasında
- Hedef: (-1,5) Noktasında
- Sensör1 için hedefin kerterizi Y pozitif ekseninden saat yönündeki açısı 45 derece
- Sensör2 için hedefin Y Pozitif ekseninden saat yönündeki açısı 315 derece

GENEL- PUBLIC



Şekil 1 Örnek Sensör ve hedef konumlandırılması

GENEL- PUBLIC

3. KONUM TESPİTİ

Hedef konum tespiti için iki doğrunun kesişim noktası bulunmuştur. iki adet sensörden gelen konum ve kerteriz bilgisinden faydalanarak doğru denklemi oluşturulmuştur. Oluşturulan doğru denklemlerinin kesişim noktası hedefi vermektedir. İkiden fazla sensör kullanılması durumunda birden fazla kesişim noktası olabilecektir. Bu durum için en küçük kareler yöntemi ile çözüm oluşturulabileceği öngörülmektedir.

Konum tespiti için org.ptp.controlcenter.business paketi altında EntityLocater sınıfı oluşturulmuştur.

GENEL- PUBLIC

4. UYGULAMA BİLGİSİ

4.1. GELİŞTİRME ORTAMI VE KÜTÜPHANELER

Programı geliştirmek için kullanılan program ve kütüphaneler aşağıda verilmiştir.

- Eclipse versiyon: 2023-06 (4.28.0)
- Apache Kafka Windows : 2.13-3.5.1
- JUnit 5
- Java SE 1.8

4.2. PAKETLER

Yazılım üç ana modülden oluşmaktadır.

- ControlCenter
- Sensor
- Utility

4.2.1. CONTROLCENTER MODÜLÜ

Org.ptp.controlcenter paketi altında yer alan tüm alt paketleri içermektedir. Org.ptp.controlcenter.main paketi altında yer alan APP.java isimli dosya ile uygulama çalıştırılabilmektedir. Uygulama bir grafik arayüzü de sahiptir.

ControlCenter uygulaması ilk açılışta KAFKA consumer olarak görev almaktadır. Bu uygulamadan sadece bir tane açılmalıdır. Birden çok uygulama açılmak istenirse KAFKA'da farklı grup yaratılması gerekecektir.

Uygulama açıldığında KAFKA'dan sensor verisi beklemektedir. Gelen sensör verisi bir HashMap'te saklanmaktadır. Hashmap anahtar değeri sensör UUID'sidir. Böylelikle sensörden gelen en son veri listede tutulmaktadır.

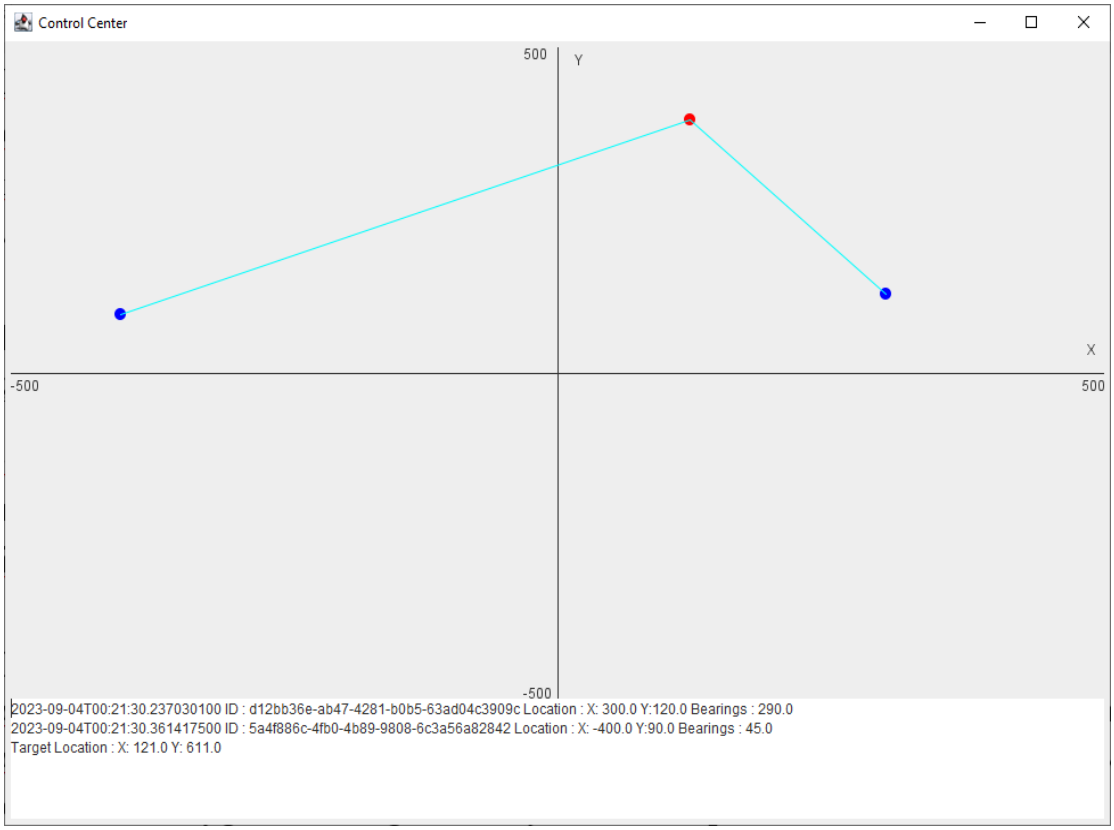
Veri işleme aşamasında en güncel veriye sahip iki sensör verisinden faydalanılarak hedef konumu hesaplanmaktadır. İkiden fazla sensörün verisi gelmesi durumunda uygulama tarafından en güncel tarihe sahip iki farklı sensör verisi seçilecektir. Bu durumunda haricinde sensörden gelen veri ile güncel zaman arasında 10 saniyenden fazla fark varsa bu veri kullanılmamaktadır.

Kafka'dan veri çekilirken LATEST indeks ayarı yapılmıştır. Bu kapsamda KAFKA'da veri birikmiş olsa bile en güncel veri ile ilerlenecektir. Birikmiş veri geçmişe ait olacağı için anlamsız olabileceği değerlendirilmiştir.

Veri KAFKA'dan alındıktan sonra asenkronluğu sağlamak amacıyla uygulama içinde farklı servisler araçlığıyla katmanlı bir yapıda arayüze ilerilemektedir. Arayüzde gösterim amacıyla bir iki boyutlu grafik hazırlanmıştır.

GENEL- PUBLIC

Şekil 1 ControlCenter yazılım temel akışı



Şekil 2 Kontrol Merkezi Arayüzü

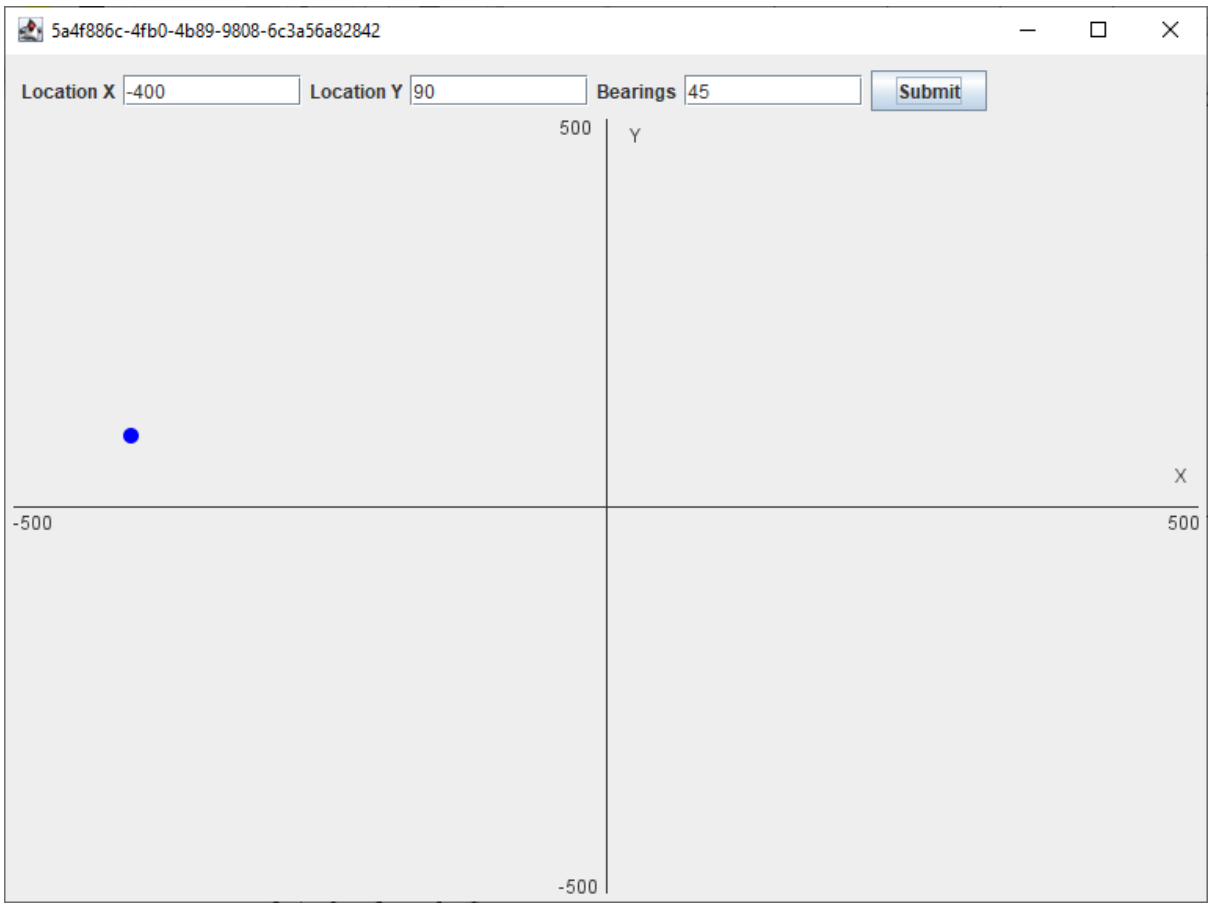
4.2.2. SENSÖR MODÜLÜ

Org.ptp.sensör paketi altında yer alan tüm alt paketleri içermektedir. Org.ptp.sensör.main paketi altında yer alan APP.java isimli dosya ile uygulama çalıştırılabilmektedir. Uygulama bir grafik arayüzü de sahiptir.

Uygulama KAFKA producer olarak çalışmaktantır. Birden fazla sensör uygulaması çalıştırılabilmektedir. Uygulama ekranından sensör konumu ve kerteriz bilgisi girilmesi ardından sensör için otomatik bir seçkin ID üretilmektedir. Tüm yaratılan sensörler saniyede bir konum ve kerteriz bilgisini KAFKA topiğine üretmektedir. Kullanılan kafka topic ismi “predicttarget” olarak AppConstants.java isimli sınıfa girilmiştir.

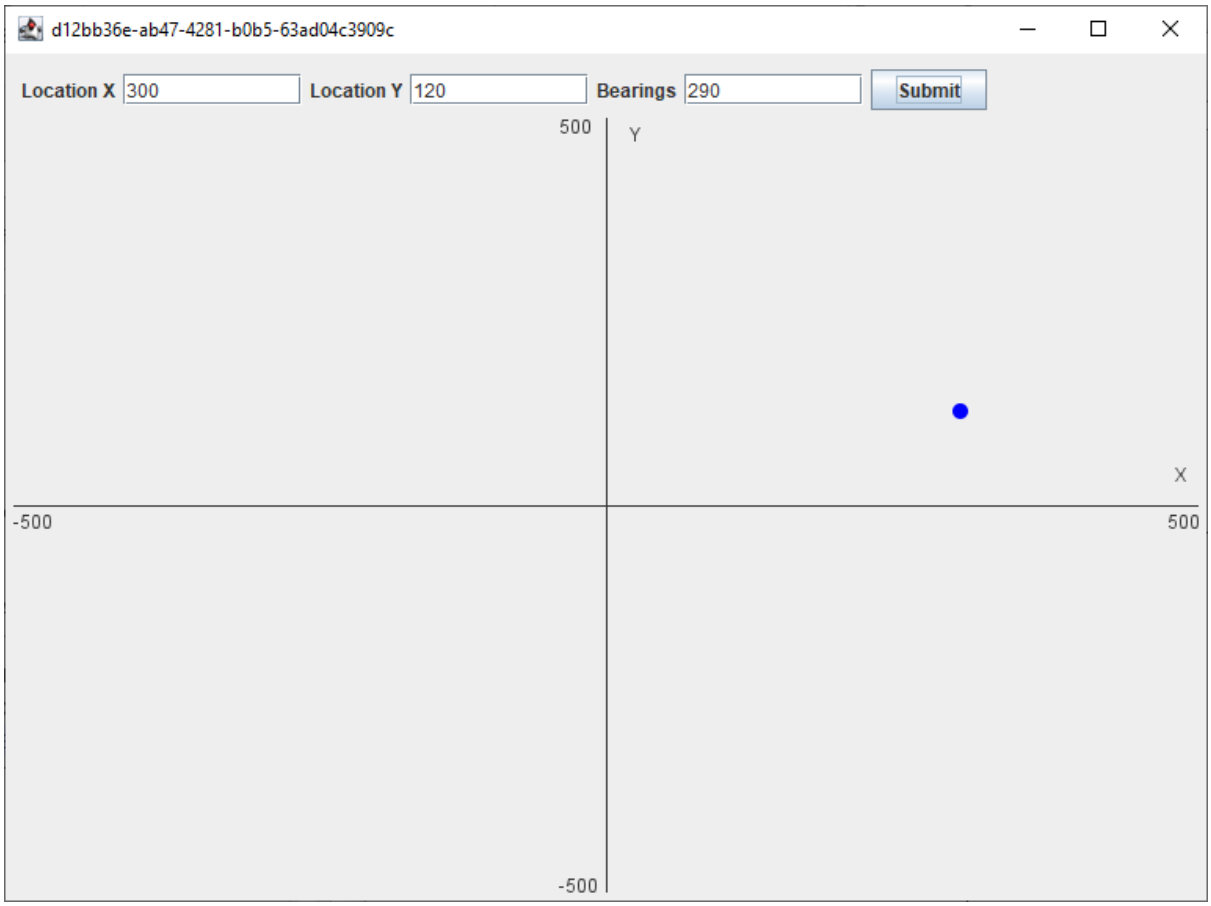
GENEL- PUBLIC

Şekil 3 Sensör uygulaması yazılım temel akışı



Şekil 4 Sensör Modülü Grafik Arayüzü Sensör 1

GENEL- PUBLIC



Şekil 5 Sensör Modülü Grafik Arayüzü Sensör 2

4.2.3. UTILS

ControlCenter ve Sensör uygulamaları için ortak java kütüphaneleri sağlayan pakettir. Çalıştırılabilir değildir. Org.ptp.utils paketi altında yer alan dosyaları içermektedir.

4.3. UYGULAMA ÇALIŞTIRMA

Uygulamanın çalıştırılacağı bilgisayarda APACHE KAFKA kurulu olmalıdır.

Apache KAFKA indirildikten sonra aşağıdaki komutlar ile çalışır duruma getirilir.

```
.\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties
```

```
.\bin\windows\kafka-server-start.bat .\config\server.properties
```

Control Center (org.ptp.controlcenter.main altında) ve Sensör (org.ptp.sensor.main altında) uygulamaları jar dosyası üretilerek ya da açılan projeler APP.java isimli dosya çalıştırılarak aktif hale getirilir. Bir adet Control Center uygulaması ve istenen sayıda Sensör uygulaması çalıştırılabilir. Control Center uygulaması en güncel iki adet sensör verisi ile hedef konumu tahmini yapacaktır.

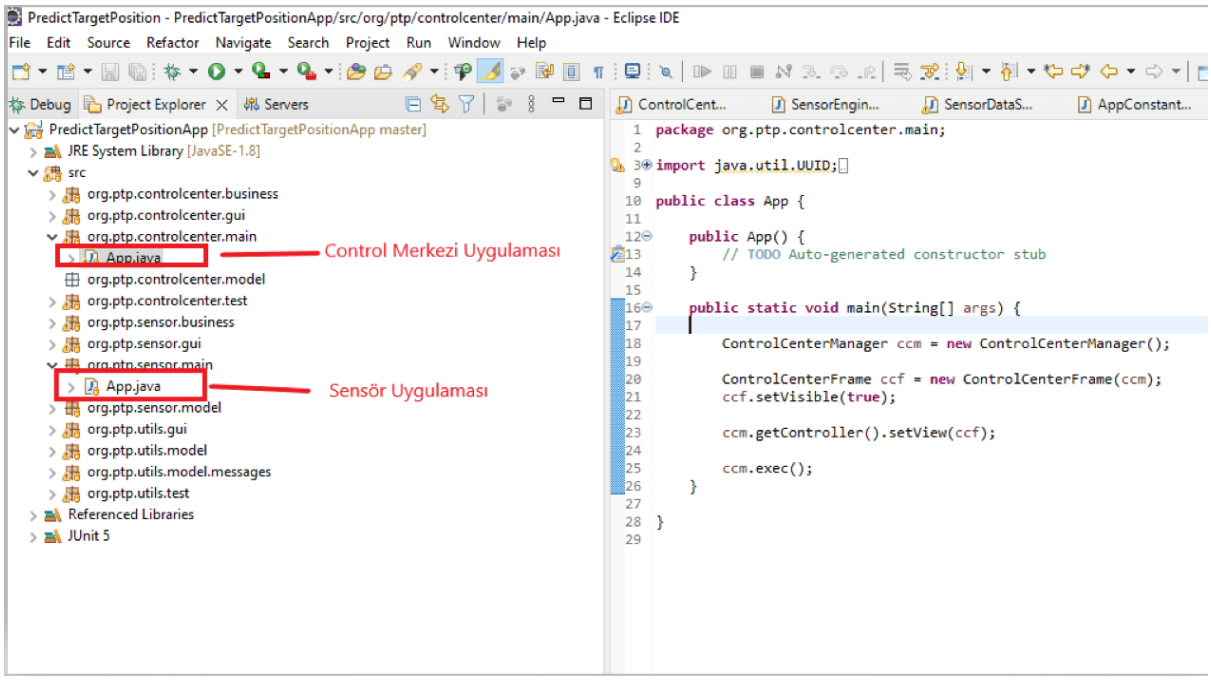
Akış:

GENEL- PUBLIC

- Kullanıcı Kafka'yı çalıştırır
- Kullanıcı Control Center uygulamasını çalıştırır. Bu uygulama KAFKA "predicttarget isimli "topiğini tüketmektedir. Mesajlar geldikçe dinamik gösterim yapmaktadır. İki farklı sensör bilgisi geldiğinde hedef konum tahminleme başlayıp görselleştirilecektir.
- Kullanucu Sensör uygulamasını çalıştırır. Sensör uygulamasına X [-500,+500] ve Y[-500,+500] değer aralığında bir sensör konumu girilir.Sensörün tespit ettipi kerteriz bilgisi girilir ve "submit" butonuna basılır. Bu işlem sensörün random bir UUID almasını sağlar ve ilgili uygulama çaık olduğu sürece üretilen UUID kullanılır. Sensör veriri saniyede bir otomatik olarak kafka topiğine üretilir. Kullanıcı birden fazla sensör açabilir. İki adet sensör aktif olduğunda Control Center uygulaması tahminlemeye başlar.

4.4. BİRİM TESTİ

Hedef konum belirleme için hazırlanan iki doğru kesişim noktası bulma algoritmasının birim testleri org.ptp.controlcenter.test paketialtında EntityLocaterTest sınıfında yazılmıştır. Doğruların parallel olma durumu ve bağımsız olma durumları test edilmiştir.



Şekil 6 IDE üzerinden uygulamaları çalıştırma

Uygulama çalıştırma sırasında KAFKA paketleri eksik olması durumunda indirilen kafka kütüphanesi içinde yer alan kafka_2.13-3.5.1\libs altındaki jar dosyaları Project properties-> Java Build Path -> Libraries -> Add External Jar adımları uygulanarak eklenmelidir.

GENEL- PUBLIC

GENEL- PUBLIC