**YAZILIM GEREKSİNİM ANALİZ RAPORU**

PROJE ADI: Labirent İçindeki Kişi Konumlandırma Programı

**Hazırlayan**: Bahattin ADİYAMAN

1. GİRİŞ

Bu proje, bir labirent içinde bulunan kişilerin koordinatlarını alarak Öklid veya Manhattan mesafe hesaplama algoritmaları ile en yakın ve en uzak kişiyi belirlemeyi amaçlamaktadır. Kullanıcı, belirlediği tespit mesafesi dahilindeki kişileri görüntüleyebilir

2. KAPSAM VE AMAÇ

Programın hedef kitlesi, harita ve konum verileri ile çalışan geliştiriciler ve mesafe hesaplama algoritmalarını öğrenmek isteyen kullanıcılar olabilir. Bu uygulama, iki nokta arasındaki farklı mesafe hesaplama yöntemlerini deneyimleme imkânı sunar.

**Kullanım Senaryosu:**

\*Kullanıcı bir labirent genişliği ve yüksekliği belirler.

\*Kişi sayısını girerek komutlarını belirler.

\*kendi konumunu girer.

\*Mesafe hesaplama yöntemi seçerek en yakın ve en uzak kişiyi bulur.

\*elirlenen tespit mesafesi içindeki kişileri görüntüleyebilir.

3. FONKSİYONEL GEREKSİNİMLER

Kullanıcı aşağıdaki işlemleri gerçekleştirebilmelidir:

Labirentin boyutlarını belirleme.

Kişi sayısını girme ve koordinatlarını belirleme.

Kendi konumunu girme.

Mesafe hesaplama yöntemi seçme (Öklid veya Manhattan).

En yakın ve en uzak kişiyi bulma.

Tespit mesafesi içindeki kişileri listeleme.

Yanlış girişleri önleme: Labirent sınırları dışına çıkamaz.

4. PERFORMANS VE SİSTEM GEREKSİNİMLERİ

**Minimum Gereksinimler:**

C derleyicisi (GCC, Clang veya MSVC desteklenir).

Terminal tabanlı kullanım için Windows, Linux veya MacOS uyumlu sistemler.

Hafif bellek kullanımı ile büyük veri kümeleri için optimize edilebilir.

**Performans:**

Büyük kişi sayıları için dinamik bellek yönetimi (malloc/free) önerilir.

Hesaplamalar mümkün olduğunca verimli yapılmalıdır.

Kullanıcı girişleri geçerli olup olmadığını kontrol etmek için doğrulanmalıdır.

5. GÜVENLİK VE SINIRLANDIRMALAR

Kullanıcının hatalı giriş yapmasını önlemek için doğrulama mekanizması eklenmiştir.

Negatif veya hatalı koordinat girilemez.

Labirent sınırları dışına çıkmak mümkün değildir.

6. SONUÇ VE ÖNERİLER

Mevcut sistem doğru bir şekilde çalışmaktadır ancak ileride şu özellikler eklenebilir:

Grafiksel Arayüz: Kullanıcı dostu bir GUI ile labirent haritası eklenebilir.

Dinamik Bellek Kullanımı: Daha büyük veri kümeleri için malloc/free yönetimi önerilmektedir.

Farklı Mesafe Algoritmaları: Ekstra algoritmalarla seçenek artırılabilir.