



T.C
KOCAELİ SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ
MÜHENDİSLİK VE DOęA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR/YAZILIM MÜHENDİSLİęİ

SAYISAL TASARIM

Hazırlayan
İbrahim Bener Karaca
220501019
<https://github.com/BenerKaraca>

Enes Küçük
220501017
<https://github.com/EnessKucukk>

DERS SORUMLUSU:
DR. ÖęR. ÜYESİ ERCAN ÖLÇER

01/06/2024

1 GİRİŞ

1.1 Projenin amacı

- Bu projede amaç, kullanıcıların grafik arayüz yardımıyla mantık kapılarını ekleyebilmelerini, bu kapıları bağlayabilmelerini ve sonuçlarını görebilmelerini sağlamaktır. Kullanıcı, arayüz üzerinden farklı mantık kapıları ekleyebilir (AND, OR, NOT vb.), bu kapıları istedikleri sayıda bağlayabilir ve ardından giriş değerlerini belirleyerek simülasyonu çalıştırabilir. Simülasyon sonucunda her bir kapının çıktısı görüntülenir.
- Projede beklenenler şunlardır:
- Grafik Arayüz (GUI) Oluşturma
- Mantık Kapıları Sınıflarının Tanımlanması
- Giriş Değerlerinin Alınması
- Mantık Kapılarının Eklenmesi
- Simülasyonun Çalıştırılması

2 GEREKSİNİM ANALİZİ

2.1 Arayüz gereksinimleri

Kapı Ekleme: Kullanıcı, araç çubuğundaki düğmelere tıklayarak çeşitli mantık kapılarını sahneye ekleyebilmelidir.

Giriş Değerlerini Belirleme: Kullanıcı, giriş değerlerini belirlemek için giriş kutularını kullanarak 0 veya 1 değerlerini girebilmelidir.

Kapıları Bağlama: Kullanıcı, mantık kapılarını birbirine bağlamak için fareyi kullanarak bağlantıları oluşturabilmelidir.

Uygulamayı Çalıştırma: Kullanıcı, uygulamayı çalıştırmak için "Çalıştır" düğmesine tıklayabilmelidir.

Çıkışları Gözleme: Kullanıcı, uygulama çalıştırıldığında, her bir mantık kapısının çıkışını konsol çıktısında görebilmelidir.

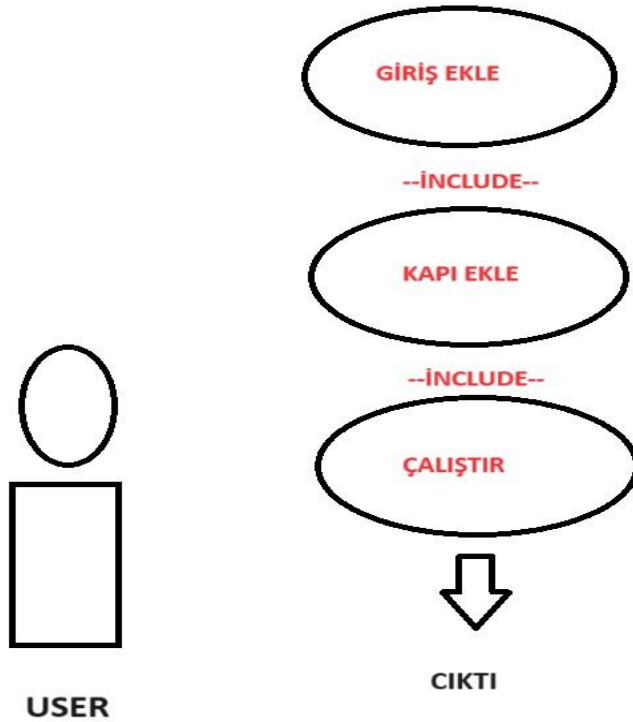
LED Durumunu Gözleme: Kullanıcı, uygulama çalıştırıldığında, LED'in durumunu görsel olarak görebilmelidir. LED kırmızıysa, mantık kapılarının çıkışı doğrudur; gri ise yanlıştır.

Girişleri ve Kapıları Taşıma: Kullanıcı, girişleri, kapıları ve bağlantıları fare ile sürükleyerek yeni konumlara taşıyabilmelidir.

2.2 Fonksiyonel gereksinimler

- Kullanıcı, arayüzdeki araç çubuğundan farklı mantık kapılarını seçebilmelidir.
- Seçilen mantık kapıları, sahneye eklenmelidir.
- Her bir kapı, adıyla birlikte görsel olarak temsil edilmelidir.
- Kullanıcı, girişlerin değerlerini belirleyebilmelidir.
- Belirlenen girişler, sahneye eklenmelidir.
- Kullanıcı, mantık kapıları arasında bağlantılar yapabilmelidir.
- Bağlantılar, sahneye eklenmelidir.
- Kullanıcı, tüm bağlantıları ve giriş değerlerini dikkate alarak uygulamayı çalıştırabilmelidir.
- Kapılar, belirlenen giriş değerlerine göre çalıştırılmalı ve çıkış değerleri hesaplanmalıdır.
- Her bir kapının çıkış değeri, ekrana yazdırılmalıdır.
- Bir LED bileşeni, sahneye eklenmelidir.
- LED'in durumu, kapıların çıkışlarına göre değişmelidir.

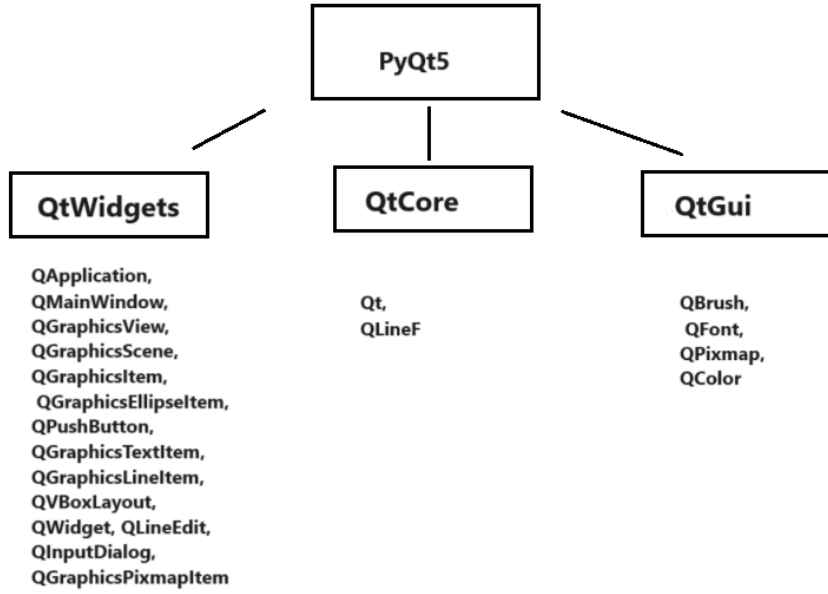
2.3 Use-Case diyagramı



3 TASARIM

3.1 Mimari tasarım

- **Sınıf Hiyerarşisi:** Uygulama, PyQt5 kütüphanesini kullanarak grafik arayüz oluşturur. Bu arayüzde mantık kapıları, bağlantıları ve girişleri temsil eden sınıflar bulunur. Bu sınıflar arasında bir hiyerarşi bulunur; örneğin, MantikKapisi sınıfı temel bir mantık kapısını temsil ederken, TekGirisliKapi ve CiftGirisliKapi bu sınıftan türetilmiştir ve daha özelleştirilmiş kapıları temsil eder.
- **Kullanıcı Arabirimi (UI):** PyQt5 kullanılarak oluşturulan kullanıcı arayüzü, QMainWindow üzerinde QGraphicsScene ve QGraphicsView ile tasarlanmıştır. Sahnedeki öğeler, mantık kapıları, girişler ve bağlantıları temsil eder. Ayrıca, araç çubuğu kullanıcıya kapıları eklemek için düğmeler sağlar.
- **Kapılar ve Bağlantılar:** Mantık kapıları ve bağlantılar, grafik öğeler olarak QGraphicsPixmapItem ve QGraphicsLineItem sınıfları tarafından temsil edilir. Her kapı ve bağlantı, sahnede konumlandırılabilir ve etkileşime açık hale getirilir.
- **Uygulama Kontrolü ve Yönlendirme:** Uygulamanın ana kontrolü MantikKapisiUygulaması sınıfı tarafından sağlanır. Bu sınıf, kullanıcının eklediği kapıları, girişleri ve bağlantıları yönetir. Ayrıca, kapıların çalıştırılmasını ve çıkışların hesaplanmasını yönetir.



3.2 Kullanılacak teknolojiler

- Yazılım Python dilinde yazıldı

QMainWindow: Ana uygulama penceresi oluşturmak için kullanılır.

QGraphicsView ve QGraphicsScene: Grafik sahne ve öğeleri oluşturmak, düzenlemek ve görüntülemek için kullanılır.

QGraphicsPixmapItem: Resim öğelerini sahneye eklemek için kullanılır.

QGraphicsLineItem: Çizgi öğelerini sahneye eklemek için kullanılır.

QPushButton: Düğme öğeleri oluşturmak için kullanılır.

QVBoxLayout ve QWidget: Düzen yönetimi için kullanılır.

QLineEdit ve QInputDialog: Kullanıcı girişi almak için kullanılır.

QBrush, QFont ve QColor: Grafik öğelerin biçimlendirilmesi için kullanılır. Varsa diğer teknolojiler ile ilgili açıklama

3.3 Veri tabanı tasarımı

- Mantık kapısı simülatörü, kullanıcının girişlerle mantık kapılarını birbirine bağlayıp çıktıları gözlemlemesini sağlayan bir araçtır. Bu uygulama, veri tabanına ihtiyaç duymaz çünkü kullanıcılar tarafından tanımlanan kapılar, girişler ve bağlantılar geçici olarak bellekte saklanabilir. Veri tabanı, genellikle uygulamanın kalıcı veri depolama gereksinimlerini karşılamak için kullanılır ve bu

uygulamada böyle bir ihtiyaç yoktur.

3.4 Kullanıcı arayüzü tasarımı

- Kullanıcı arayüzü tasarımı ile ilgili açıklama.
- Yazılımdan ekran çıktıları alınarak üzerinden açıklama yapılması
- Uygulamanın nasıl çalıştırılacağı ile ilgili açıklama

4 UYGULAMA

4.1 Kodlanan bileşenlerin açıklamaları

- MantikKapisi Sınıfı:
__init__(self, isim): Mantık kapısının ismini alarak bir MantikKapisi nesnesi oluşturur.
ciktiyi_al(self): Mantık kapısının çıktısını hesaplar ve döndürür.
- TekGirisliKapi Sınıfı:
__init__(self, isim): Tek girişli mantık kapısını oluşturur.
giris_ayarla(self, giris_degeri): Giriş değerini ayarlar.
- CiftGirisliKapi Sınıfı:
__init__(self, isim): Çift girişli mantık kapısını oluşturur.
girisleri_ayarla(self, giris1_degeri, giris2_degeri): Giriş değerlerini ayarlar.
- Diğer mantık kapısı sınıfları:
- NotKapisi: Tek girişli not kapısı.
- AndGate: Çift girişli AND kapısı.
- OrGate: Çift girişli OR kapısı.
- BufferGate: Tek girişli buffer (tekrarlayıcı) kapısı.
- XorGate: Çift girişli XOR kapısı.
- XnorGate: Çift girişli XNOR kapısı.
- NorGate: Çift girişli NOR kapısı.
- NandGate: Çift girişli NAND kapısı.
- KapiOgesi Sınıfı:
__init__(self, kapı, x, y): Grafik arayüzde bir mantık kapısını temsil eden nesneyi oluşturur.
mouseDoubleClickEvent(self, event): Kullanıcı çift tıkladığında kapı çıktısını hesaplar ve ekrana yazdırır.
- BaglantiOgesi Sınıfı:
__init__(self, baslangic_ogesi, son_ogesi): İki kapı arasında bir bağlantı çizgisi oluşturur.
guncel_pozisyonu_guncelle(self): Bağlantı çizgisinin pozisyonunu günceller.
- MantikKapisiUygulamasi Sınıfı:

`__init__(self)`: Uygulama penceresini oluşturur ve arayüz bileşenlerini ekler.
`ana_pencere_olustur(self)`: Ana pencereyi oluşturur.
`araç_cubugu_olustur(self)`: Araç çubuğunu oluşturur ve kapı düğmelerini ekler.
`giris_ekle(self, sayi)`: Belirtilen sayıda giriş ekler.
`kapi_ekle(self, kapı)`: Belirtilen mantık kapısını ekler.
`baglanti_ekle(self, baslangic_ogesi, son_ogesi)`: İki kapı arasında bir bağlantı ekler.
`calistir(self)`: Giriş değerlerini alır, kapıları bu değerlerle değerlendirir ve sonuçları ekrana yazdırır.

4.2 Görev dağılımı

- Projeyi geliştirirken tüm adımları ekip arkadaşım ile eş zamanlı olarak izledik.

4.3 Karşılaşılan zorluklar ve çözüm yöntemleri

- Kodu yazarken bağlantı çizme kısmında çok zorlandık birkaç araştırma ile çözmeye çalıştık

4.4 Proje isterlerine göre eksik yönler

- Kodda devre elemanları arasında istenilen gibi bağlantı çizilememekte.

5 TEST VE DOĞRULAMA

5.1 Yazılımın test süreci

- Test kodu yazılmadı

5.2 Yazılımın doğrulanması

- Herhangi bir test kodu yazılmadı.
- Bağlantılar sadece kapılar arası oluyor.