

**T.C**

**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ**

**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR/YAZILIM MÜHENDİSLİĞİ**

**PROJE KONUSU:** **SAYISAL TASARIM PROJESİ**

**ÖĞRENCİ ADI: Gürel Bilgin**

**ÖĞRENCİ NUMARASI: 220502041**

**https://github.com/gurelbilgin**

**ÖĞRENCİ ADI: Berkay Aras**

**ÖĞRENCİ NUMARASI:** **220501033**  
**https://github.com/brkyaras**

**DERS SORUMLUSU:**

**Dr. Öğr. Üyesi Ercan ÖLÇER**

**TARİH: 02.06.2024**

1. **GİRİŞ**

#### 1.1 Projenin Amacı

Projenin amacı, kullanıcıların basit mantık devrelerini tasarlamak için bir platform sağlamaktır. Bu platform, kullanıcılara grafik arayüzü üzerinden mantık kapıları, giriş/çıkış elemanları ve bağlantı elemanları gibi bileşenleri sürükleyip bırakarak devreleri oluşturma imkanı sunar. Projenin ana amacı şunlardır:

* Kullanıcıların kendi mantık devrelerini oluşturmasını sağlamak.
* Mantık kapılarını, giriş/çıkış elemanlarını ve bağlantı elemanlarını eklemek için kullanıcı dostu bir arayüz sunmak.
* Oluşturulan devreleri simüle ederek devrelerin çalışmasını görsel olarak izlemek.
* Kullanıcıların simülasyon sırasında devre elemanlarının özelliklerini düzenlemesine olanak tanımak.

#### 1.2 Beklenenler

Projede gerçekleştirilmesi beklenenler şunlardır:

* Mantık kapıları için etiket ve giriş bağlantı sayısı gibi özelliklerin bulunduğu bir özellik tablosu oluşturulması.
* Giriş/çıkış elemanları için etiket, renk ve başlangıç değeri gibi özelliklerin bulunduğu bir özellik tablosu oluşturulması.
* Bağlantı elemanları için etiket ve renk gibi özelliklerin bulunduğu bir özellik tablosu oluşturulması.
* Tasarım alanında aynı devre elemanının birden fazla eklenebilmesi.
* Bağlantı hatlarının herhangi bir noktada birleştirilmesi gerektiğinde bağlantı düğümlerinin kullanılması.
* Simülasyon kontrolleri için çalıştır, sıfırla ve durdur düğmelerinin bulunması.

1. **GEREKSİNİM ANALİZİ**

#### 2.1 Arayüz Gereksinimleri

* **Kullanıcı Arayüzü Gereksinimleri:**
  + Kullanıcıların araç çubuğu üzerinden mantık kapıları, giriş/çıkış elemanları ve bağlantı elemanları gibi bileşenleri seçip tasarım alanına sürükleyip bırakabilmesi.
  + Her bir elemanın özelliklerinin sağ fare tuşu ile tıklanarak görüntülenebilmesi ve düzenlenebilmesi.
  + Simülasyon sırasında kullanıcıların devre elemanlarının giriş değerlerini değiştirebilmesi.
  + Tasarım alanında çift tıklama veya sağ tıklama gibi interaktif yöntemlerle bileşenlerin özelliklerinin düzenlenebilmesi.
* **Donanım Arayüzü Gereksinimleri:**
  + Donanım arayüzü gereksinimleri bulunmamaktadır.

#### 2.2 Fonksiyonel Gereksinimler

* Kullanıcıların mantık kapılarını, giriş/çıkış elemanlarını ve bağlantı elemanlarını araç çubuğundan seçerek tasarım alanına ekleyebilmesi.
* Her bir elemanın kendine özgü bir özellik tablosuna sahip olması ve bu özelliklerin kullanıcı tarafından düzenlenebilmesi.
* Bağlantı elemanları aracılığıyla mantık kapıları, giriş/çıkış elemanları ve diğer bağlantı elemanları arasında bağlantıların kurulabilmesi.
* Simülasyon sırasında kullanıcıların devre elemanlarının giriş değerlerini değiştirebilmesi ve simülasyonun güncellenmesi.
* Simülasyonun başlatılması, sıfırlanması ve durdurulmasını sağlayan kontrol düğmelerinin bulunması.

#### 2.3 Use-Case Diyagramı

Use-Case Diyagramı, aşağıdaki temel kullanıcı etkileşimlerini gösterir:

* **Kullanıcı**, araç çubuğundan mantık kapıları, giriş/çıkış elemanları ve bağlantı elemanları seçerek tasarım alanına ekler.
* **Kullanıcı**, tasarım alanında bulunan elemanların özelliklerini sağ fare tuşu ile tıklayarak düzenler.
* **Kullanıcı**, bağlantı elemanları aracılığıyla devre elemanları arasında bağlantılar kurar.
* **Kullanıcı**, simülasyonu başlatır, sıfırlar veya durdurur.
* **Kullanıcı**, simülasyon sırasında giriş değerlerini değiştirir ve simülasyon sonuçlarını izler.

1. **TASARIM**

#### 3.1 Mimari Tasarım

* **Mimari Tasarımın Açıklanması:**
  + Projede Model-View-Controller (MVC) mimarisine uygun bir yapı kullanılacak.
  + Model kısmında mantık kapıları, giriş/çıkış elemanları ve bağlantı elemanları gibi bileşenlerin veri modelleri oluşturulacak.
  + View kısmında kullanıcı arayüzü, araç çubuğu, tasarım alanı ve kontrol düğmeleri gibi görsel bileşenler yer alacak.
  + Controller kısmında ise kullanıcı etkileşimlerini kontrol edecek ve model ile görüntü arasındaki etkileşimi sağlayacak iş mantığı kodları olacak.
* **Modül Diyagramı:**
  + Modül diyagramı eklenmiştir.

#### 3.2 Kullanılacak Teknolojiler

* **Yazılım Dili:**
  + Python programlama dili kullanılacak.
* **Harici Kütüphaneler:**
  + Tkinter: Kullanıcı arayüzü oluşturmak için kullanılacak.
  + PIL (Python Imaging Library): Resim dosyalarını açmak ve görüntüleri işlemek için kullanılacak.
* **Diğer Teknolojiler:**
  + Proje için özel bir veritabanı kullanılmayacak. Veriler, Python programlama dilinin veri yapıları kullanılarak saklanacak.

#### 3.3 Veri Tabanı Tasarımı

* **Veri Tabanı Tasarımı Açıklaması:**
  + Proje için veritabanı kullanılmayacak. Veriler, Python programlama dilinin listeleri, sözlükleri ve diğer veri yapıları kullanılarak bellekte saklanacak.
  + ER Diyagramı eklenmemiştir.

#### 3.4 Kullanıcı Arayüzü Tasarımı

* **Kullanıcı Arayüzü Tasarımı:**
  + Kullanıcı arayüzü, Tkinter kütüphanesi kullanılarak oluşturulacak.
  + Araç çubuğu, tasarım alanı, kontrol düğmeleri ve diğer görsel bileşenler arayüze entegre edilecek.
* **Ekran Çıktıları:**
  + Kullanıcı arayüzü örnekleri ve çalışma zamanında elde edilen ekran görüntüleri rapora eklenecektir.
* **Uygulamanın Çalıştırılması:**
  + Python betiği çalıştırılarak uygulama başlatılacaktır. Çalışma zamanında, kullanıcı arayüzü üzerinden tasarım yapılabilir ve simülasyon başlatılabilir.

1. **UYGULAMA**

#### 4.1 Kodlanan Bileşenlerin Açıklamaları

* **LogicGateSimulator Sınıfı:**
  + Proje ana kontrol sınıfıdır. Kullanıcı arayüzünü oluşturur, bileşenlerin eklenmesini, simülasyonun çalıştırılmasını ve sıfırlanmasını sağlar.
* **LogicGate Sınıfı:**
  + Mantık kapılarını temsil eder. Çeşitli mantıksal işlevlere sahip kapılar oluşturulabilir.
* **InputElement, OutputElement ve LEDElement Sınıfları:**
  + Giriş, çıkış ve LED elemanlarını temsil eder. Her birinin farklı özellikleri ve davranışları vardır.
* **Connection ve ConnectionNode Sınıfları:**
  + Bağlantıları ve bağlantı noktalarını temsil eder.
* **Diğer Yardımcı Sınıflar ve Fonksiyonlar:**
  + Kullanıcı arayüzünü oluşturmak ve etkileşimleri yönetmek için yardımcı sınıflar ve fonksiyonlar bulunmaktadır.

#### 4.2 Görev Dağılımı

* **Tasarım ve Geliştirme Aşaması:**
  + Kullanıcı arayüzü ve araç çubuğunun tasarımı ve geliştirilmesi: [Öğrenci Adı]
  + Mantık kapılarının ve diğer bileşenlerin sınıflarının oluşturulması: [Öğrenci Adı]
  + Simülasyon işlemlerinin yazılması: [Öğrenci Adı]
* **Raporun Hazırlanması Süreci:**
  + Giriş ve Gereksinim Analizi Bölümlerinin Hazırlanması: [Öğrenci Adı]
  + Tasarım ve Uygulama Bölümlerinin Hazırlanması: [Öğrenci Adı]
  + Son Kontrol ve Düzenleme: [Öğrenci Adı]

#### 4.3 Karşılaşılan Zorluklar ve Çözüm Yöntemleri

* **Problemler:**
  + Kullanıcı arayüzünün dinamik olarak güncellenmesi gerekiyordu.
  + Mantık kapılarının bağlantılarının yönetimi karmaşıktı.
* **Çözümler:**
  + Tkinter'in güncelleme yeteneklerinden ve olay yönetiminden yararlanıldı.
  + Bağlantıların ve kapıların yönetimi için daha modüler bir yapı tasarlandı ve sınıflar arası etkileşim dikkatle planlandı.

#### Proje İsterlerine Göre Eksik Yönler

* **Görevlerden Kodlanamayanlar:**
  + Simülasyon esnasında ledler “run” tuşuna basıldığı zaman yanmıyor. Aynı zamanda bağlantılar yapıldıktan sonra “inputların” değerleri kapılara göre dinamik olarak değişmiyor.

1. **TEST VE DOĞRULAMA**

#### 5.1 Yazılımın Test Süreci

* **Test Uygulaması:**
  + Yazılımın doğruluğunu kontrol etmek için TestLogicGateSimulator adında kapsamlı bir test sınıfı geliştirildi. Bu sınıf, LogicGateSimulator sınıfının tüm yöntemlerini ve bileşenlerini sistematik olarak test etmek için tasarlandı.
* **Test Edilen Bileşenler:**
  + Test uygulaması, LogicGateSimulator sınıfının her bir yöntemini ve bileşenini detaylı bir şekilde test etti. Bunlar arasında mantık kapıları (AND, OR, NOT vb.), giriş/çıkış elemanları (Input Box, Output Box, LED), bağlantılar ve diğer araçlar yer almaktadır.
* **Tekrar Edilebilirlik:**
  + Test uygulaması, her çalıştırıldığında aynı sonuçları üretmektedir. Bu, yazılımın tekrar tekrar test edilmesine olanak tanımaktadır. Ayrıca, testlerin tekrarlanabilir olması hataların tekrarlanabilir bir şekilde tespit edilmesine ve düzeltilmesine yardımcı olur.
* **Örnek Senaryolar:**
  + Farklı mantık kapıları ve bağlantılarla çeşitli devrelerin tasarımı ve simülasyonu üzerinde test senaryoları oluşturuldu. Bu senaryolar, yazılımın farklı kullanım durumlarında nasıl davrandığını test etmek için kullanıldı.

#### 5.2 Yazılımın Doğrulanması

* **Doğruluk Testi Sonuçları:**
  + Test uygulaması ile yazılımın doğruluğu kapsamlı bir şekilde değerlendirildi. Bu testler sonucunda, yazılımın beklendiği gibi çalıştığı ve tasarım gereksinimlerini karşıladığı doğrulandı.
* **Eksik veya Hatalı Çalışan Bileşenler:**
  + Dinamik değer değişimi ve led yanması konusunda eksiklikler var. Bunun dışında bir hata bulunmamaktadır.