



**T.C**  
**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ**  
**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ**

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

**ÖĞRENCİ ADI:**  
**HATİCE REYHAN ÇALIŞKAN**  
**BETÜL CANOL**

**ÖĞRENCİ NUMARASI:**  
**220501001**  
**220501009**

**DERS SORUMLUSU:**  
**DR. ÖĞR. ÜYESİ ERCAN ÖLÇER**

**TARİH:**  
**14.05.2024**

# 1 GİRİŞ

## 1.1 Projenin amacı

Bu projede basit mantık devrelerinin tasarımı ve simülasyonunun gerçekleştirilmesi için bir program geliştirilmiştir.

Bu projenin amacı; mantık kapılarının işleyişini kavrayıp yaptıkları işlemleri kodlarımızda gerçekleştirip gerçekte verecekleri çıktıları web sitesinde oluşturduğumuz arayüzde de vermesidir.

Oluşturduğumuz sitenin temel hedefi; kullanıcıların tasarım sürecini görselleştirmek ve etkileşimli bir şekilde deneyimlemelerini sağlamaktır. Kullanıcılar, tasarım alanına elemanları sürükleyip bırakarak devrelerini oluşturabilecek, bağlantıları kolayca kurabilecek ve simülasyon modunda devrenin davranışını gözlemleyebileceklerdir. Böylece, karmaşık mantık devrelerini anlama ve analiz etme becerileri geliştirilecektir.

## 2 GEREKSİNİM ANALİZİ

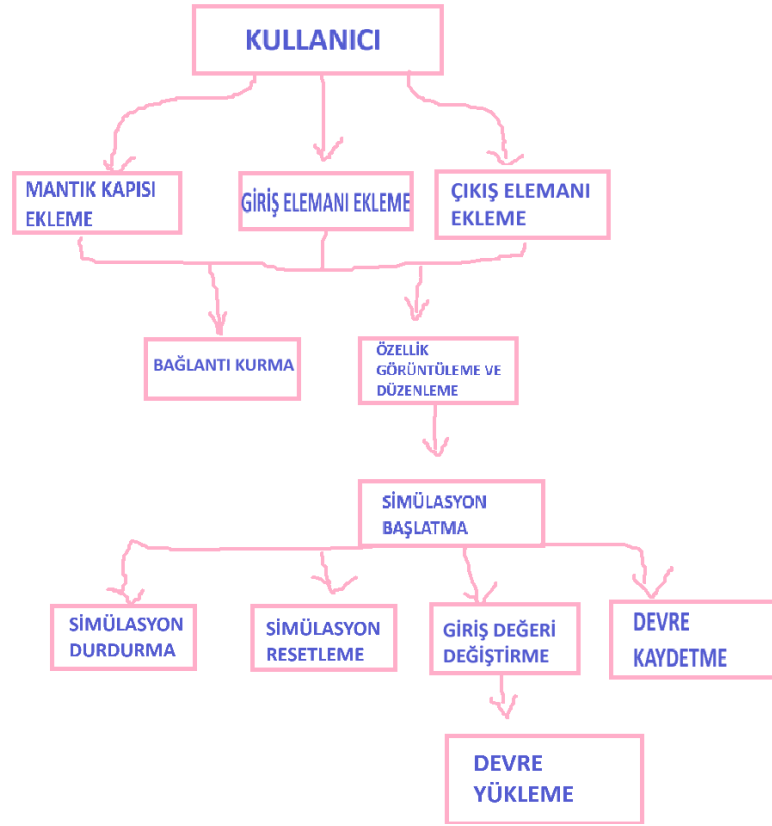
### 2.1 Arayüz gereksinimleri

- \*Kullanıcıların mantık kapılarını; (not, buffer, and, or, nand, nor, xor, xnor) ve giriş-çıkış elemanlarının ikonlarını ekranda görüp erişip işleyebilmeleri.
- \*Bağlantı elemanları ve bağlantı düğümlerinin istenildiğinde çalışabilmesi.
- \*Kullanıcının devre elemanlarını yerleştirebildikleri ve düzenleyebildikleri alanın görselleştirilmesi ve mantık kapılarının istenildiği zaman ekranda görülüp imleçle tutulup istenilen yere sürüklenmesi.
- \*Çalıştır(run), bitir(reset), durdur(stop) butonlarının bulunması.
- \*Çalıştır butonu ile devrenin çalıştırılması ve giriş değerlerinin dinamik olarak değiştirilmesi.
- \*Led veya output göstergesi ile devrenin çıkış değerinin gözlemlenmesi.

### 2.2 Fonksiyonel gereksinimler

- \*Mantık kapılarının kurallarına göre kullanımları.
- \*Çıkış sinyallerinin gösterildiği elemanların oluşturulması.
- \*Giriş sinyallerinin gönderildiği elemanların oluşturulması ve hangi sinyali gönderdiklerini kullanıcının düzenleyebilmesi.
- \*Oluşturulan devrenin kullanıcının isteğine göre durdurması veya silmesinin sağlanması.

## 2.3 Use-Case diyagramı



## 3 TASARIM

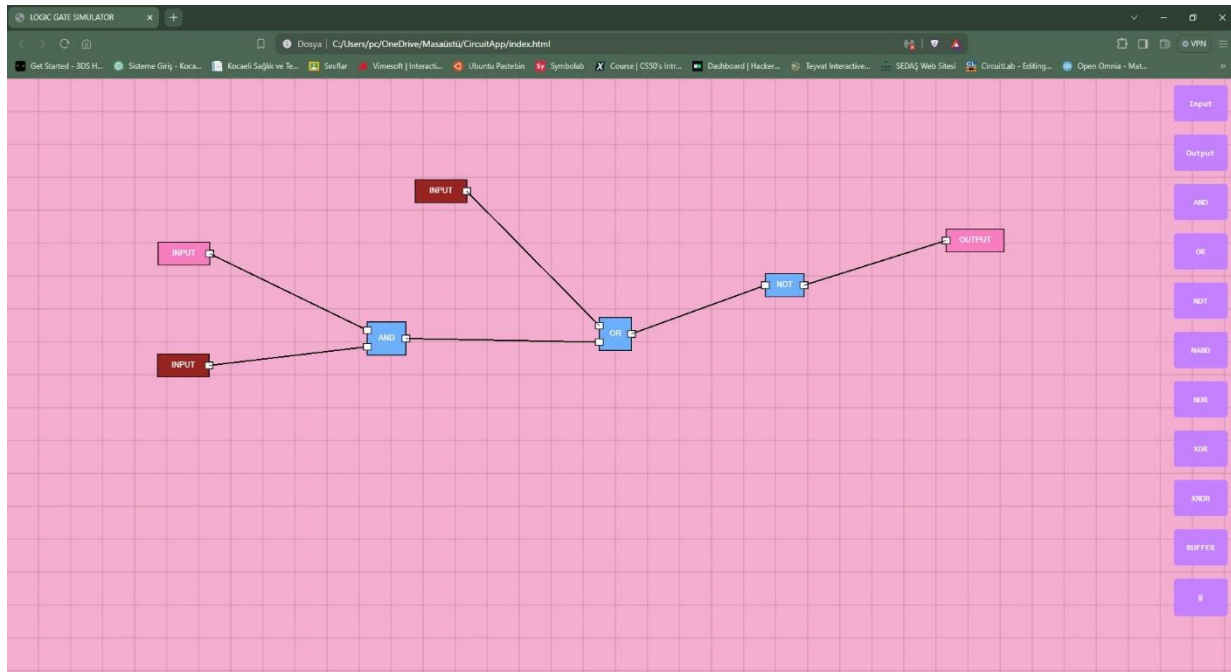
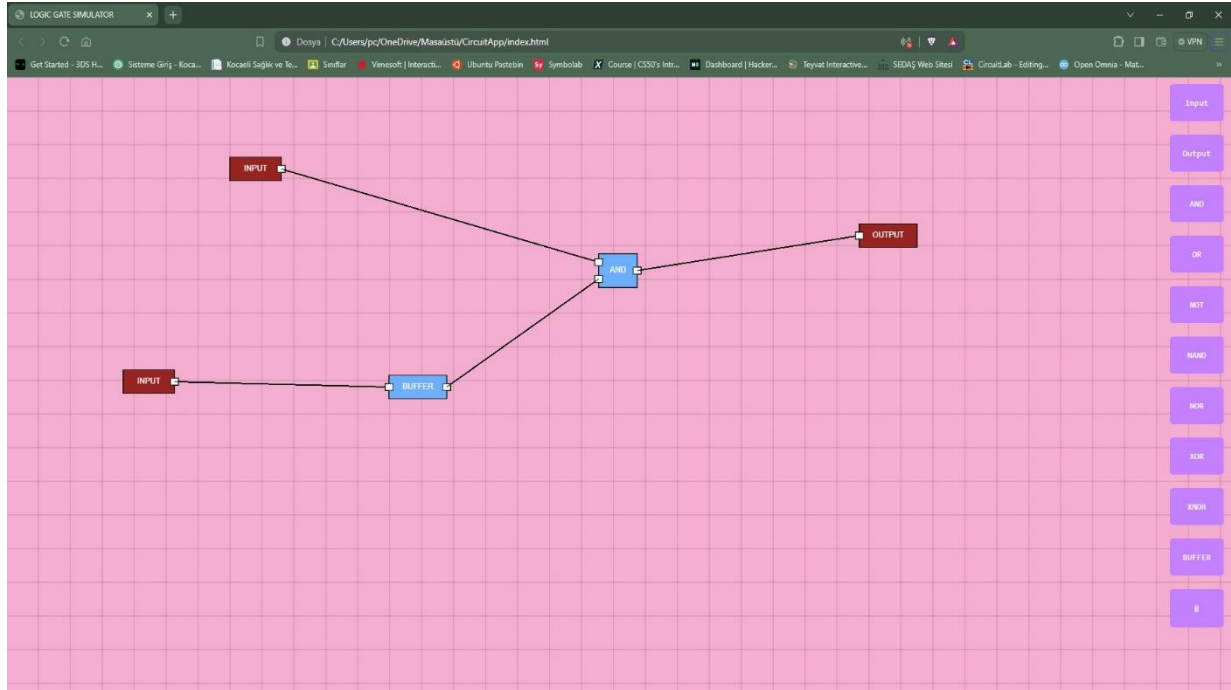
### 3.1 Mimari tasarım

- \*Araç çubuğunun eklenmesi
- \*Tasarım alanının oluşturulması
- \*Simülasyon kontrolleri

### 3.2 Kullanılacak teknolojiler

Bu yazılımı yazarken, JavaScript, html, css programlama dilleri kullanılmıştır.

## 3.2 Kullanıcı arayüzü tasarımı



## 4 UYGULAMA

### 4.1 Kodlanan bileşenlerin açıklamaları

Function draw: Bu fonksiyon, sürekli olarak çalışır (bir döngüde) ve **setNearest()** fonksiyonunu çağırır.  
**setNearest()**, fare veya dokunma konumuna en yakın düğümü belirler.

Function keyPressed: Bu fonksiyon, bir tuşa basıldığında çalışır.

Delete tuşu basılırsa, fare konumu en yakın düğümün içindeyse, **deleteNode()** fonksiyonunu çağırarak düğümü siler.

CTRL + A tuş kombinasyonu basılırsa, **renameNode()** fonksiyonunu çağırarak düğümü yeniden adlandırır.

CTRL + B tuş kombinasyonu basılırsa, **breakOutputConnection()** fonksiyonunu çağırarak düğümün bağlantılarını keser.

Function mousePressed: Bu fonksiyon, fareye tıklandığında çalışır.

Eğer tıklanan yer en yakın düğümün içindeyse, düğümü taşımak için **moving** ve **offset** ayarlanır.

Düğümün giriş veya çıkış noktalarına tıklanırsa, **lining** ayarlanarak bağlantı yapma işlemi başlatılır.

Function mouseDragged: Bu fonksiyon, fare sürüklendiğinde çalışır.

Eğer **moving** aktifse, düğüm taşınır.

Eğer **lining** aktifse, bağlantı çizgisi fare ile birlikte çizilir.

Function Mouse released: Bu fonksiyon, fare bırakıldığında çağrılır.

Eğer bir çizgi çiziliyorsa, çizginin bitiş noktasını kontrol eder ve bağlantıları ayarlar.

Taşıma veya çizgi çizme işlemlerini sonlandırır ve arka planı yeniden çizer.

Function touchStarted: Bu fonksiyon, dokunma başladığında çağrılır.

**setNearest()** fonksiyonunu çağırarak en yakın düğümü belirler.

**mousePressed()** fonksiyonunu çağırarak dokunma işlemini başlatır.

**touchHold()** fonksiyonunu 500 ms sonra çağırmak için bir zamanlayıcı ayarlar.

Function touchMoved: Bu fonksiyon, dokunma hareketi sırasında çağrılır.

**mouseDragged()** fonksiyonunu çağırarak dokunma hareketi sırasında fare sürüklemeye işlevini yerine getirir.  
**setInterval** zamanlayıcısını temizler, böylece **touchHold** fonksiyonu tetiklenmez.

Function touchHold: Bu fonksiyon, kullanıcı bir düğmeye uzun süre dokunduğunda çalışır.

**nearest** düğümüne uzun süre dokunulduğunda, seçenekleri gösterir (**showOptions**).

Function touchEnded: Bu fonksiyon, dokunma sona erdiğinde çağrılır.

**mouseReleased()** fonksiyonunu çağırarak dokunma bırakıldığında fare bırakma işlevini yerine getirir.

**setInterval** zamanlayıcısını temizler.

**mousePressedTimeout** bayrağını yanlış yapar ve 100 ms sonra tekrar doğru yapar.

Function renameNode: Bu fonksiyon, bir düğümün adını değiştirir.

Düğümün tipi giriş veya çıkış ise, kullanıcıdan yeni bir isim ister.

İsim geçerliyse, düğümün adını günceller.

Arka planı yeniden çizer ve seçenek

Function deleteNode: Bu fonksiyon, belirli bir düğümü siler.

İlk olarak, düğümün çıkış bağlantılarının bağlı olduğu diğer düğümlerdeki **inputConnected** bayrağını kapatır.

Ardından, tüm düğümler arasında gezerek, silinecek düğüme yapılan tüm çıkış bağlantılarını temizler.

Düğümü **nodes** listesinden çıkarır.

Arka planı yeniden çizer ve seçenekleri gizler.

Function breakOutputConnection: Bu fonksiyon, belirli bir düğümün tüm çıkış bağlantılarını keser. Düğümün çıkış bağlantılarının her birine giderek, bağlı düğümlerdeki girişleri sıfırlar ve **inputConnected** bayrağını kapatır.

Düğümün **outputNodes** listesini boşaltır (**null** ile doldurur).

Arka planı yeniden çizer ve seçenekleri gizler.

Function revDFS: Bu fonksiyon, verilen bir düğümden başlayarak tersine derinlik öncelikli arama (revDFS) yapar ve Boolean fonksiyonunu oluşturur.

Yani mantık işlemlerinin gerçekleştirilmesi için en önemli adım diyebiliriz.

Giriş veya pulse tipindeki düğümlerde düğüm adını döner.

Düğümün **inputNodes** listesini kullanarak, değişkenlerin bağlantılarını belirler ve **operations** nesnesine göre Boolean fonksiyonunu oluşturur.

Function placeNode: Bu fonksiyon, yeni bir düğümü düzenli bir şekilde yerleştirir.

**rack** dizisindeki son düğümün konumuna göre yeni düğümün koordinatlarını belirler.

Eğer yatay alan yeterli değilse, düğümü yeni bir satıra yerleştirir.

Düğümü **rack** ve **nodes** dizilerine ekler.

Function createBG: Arka plan oluşturur.

Function setNearest: Bu fonksiyon, fare veya dokunma konumuna en yakın düğümü belirler.

Her düğüm için fare veya dokunma konumuna olan uzaklığı hesaplar ve en yakın düğümü **nearest** değişkenine atar.

Her düğümün **show** ve **operate** fonksiyonlarını çağırır.

p5 Kütüphanesi: p5.js, görsel, interaktif ve yaratıcı projeler geliştirmek için kullanılan JavaScript tabanlı bir kütüphanedir. p5.js, Processing dilinin JavaScript versiyonu olarak ortaya çıkmış ve web tarayıcıları üzerinde çalışması amaçlanmıştır. Bu kütüphane, sanatçılar, tasarımcılar ve her seviyeden programcılar için kolay ve anlaşılır bir şekilde kod yazma imkanı sunar.

## 4.2 Görev dağılımı

Sürekli fikir alışverişinde bulunularak hareket edilmiş ve raporda da tasarım ve içerik konulup cümleler beraber kurulmuştur.

## 4.3 Karşılaşılan zorluklar ve çözüm yöntemleri

Mantık kapılarının işleme sokulması ve bağlantı kablosu ekleme konusunda zorluklar yaşanmış olup sonrasında gerekli araştırmalar yapıp bazı fonksiyonlar yazılarak bu problem çözülmüştür.

## 4.4 Proje isterlerine göre eksik yönler

Sağ tık yapılıncı özellik tablosu yok.

Başlat ve durdur tuşu bulunmamakta, devre sürekli çalışır haldedir.

## KAYNAKÇA

<https://p5js.org/download/>

[https://www.w3schools.com/js/js\\_functions.asp](https://www.w3schools.com/js/js_functions.asp)

[https://www.w3schools.com/js/js\\_classes.asp](https://www.w3schools.com/js/js_classes.asp)

[https://www.w3schools.com/js/js\\_class\\_inheritance.asp](https://www.w3schools.com/js/js_class_inheritance.asp)

[https://www.w3schools.com/html/html\\_basic.asp](https://www.w3schools.com/html/html_basic.asp)

[https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)

[https://youtu.be/8dWL3wF\\_OMw?si=zHyRRnp8PYRwVR04](https://youtu.be/8dWL3wF_OMw?si=zHyRRnp8PYRwVR04)

## GITHUB LİNKLERİ

[https://github.com/ReyhanClskn/Logic\\_Circuit\\_Sim](https://github.com/ReyhanClskn/Logic_Circuit_Sim)

<https://github.com/BetulCnl/logic-circuit-sim>