

Programlama Laboratuvarı

Oğulcan Alınlı
Bilgisayar Mühendisliği 1.Öğretim
Kocaeli Üniversitesi
ogulcanalinli60@gmail.com

**Bu doküman kocaeli üniversitesi
bilgisayar mühendisliği bölümü
programlama laboratuvarı dersi ikinci
dönem 2.proje ödevi için hazırlanmıştır**

I. ÖZET

Projede amaç, bir mikrodenetleyici tabanlı oyun makinesi geliştirmektir. Oyuncular, fiziksel palet kontrol cihazını kullanarak bir topu yönlendirerek OLED ekrandaki tuğlaları kırmaya çalışacaklardır.

Projenin gereksinimleri arasında arayüz tasarımı, kullanıcı etkileşimi, palet kontrol mekanizması, topun hareketi ve oyunun dinamik yapısı bulunmaktadır. Ayrıca, skor takibi, can sayısı, renk değişimi gibi görsel özellikler ve zorluk seviyesinin artırılması da projenin odak noktaları arasındadır.

Kullanılacak olan programlama dilleri ve veri yapıları projenin gereksinimlerine uygun olarak seçilecek ve Arduino IDE ile Proteus programları kullanılarak uygulanacaktır.

Proje tamamlandığında, öğrencilerin donanım programlama, oyun geliştirme ve problem çözme yetenekleri gelişecektir.

II. GİRİŞ

Wokwi platformu üzerinde gerçekleştirilecek olan mikrodenetleyici tabanlı oyun makinesi projesi, donanım ve yazılımın entegrasyonunu temel alarak, oyuncuların fiziksel bir palet kontrol cihazı aracılığıyla bir topu yönlendirerek OLED ekrandaki tuğlaları kırma deneyimini sunmayı amaçlamaktadır. Bu projede, Arduino IDE ve Proteus gibi araçlar kullanılarak donanım bileşenleri ile yazılım kodlarının başarılı bir şekilde bir araya getirilmesi ve işlevsel bir oyun makinesi ortaya çıkarılması hedeflenmektedir.

Proje Hedefleri:

Donanım Yazılım Entegrasyonu: Projenin temelini, mikrodenetleyici tabanlı donanım ve bu donanımı kontrol eden yazılım oluşturacaktır. Arduino IDE üzerinde geliştirilecek yazılım kodları, Proteus simülasyon ortamında test edilerek donanımla etkileşim sağlayacaktır. Bu entegrasyon süreci, öğrencilerin donanım programlama ve yazılım geliştirme konularında deneyim kazanmalarını sağlayacaktır.

Kullanıcı Deneyimi Odaklı Tasarım: Proje, kullanıcıların oyunu kolayca anlamalarını ve keyifli bir deneyim yaşamalarını hedeflemektedir. Başlangıç ekranı seçenekleri, oyun içi yönergeler, skor göstergeleri ve sonuç ekranları, kullanıcıların oyunu rahatlıkla kontrol etmelerini sağlayacak şekilde tasarlanacaktır.

Oyun Mekaniği ve Zorluk Seviyeleri: Oyun makinesinin temel mekaniği, palet kontrol cihazıyla topun yönlendirilmesi ve tuğlaların kırılması üzerine kurulacaktır.

Topun hızı, paletin tepki süresi ve tuğlaların dayanıklılığı gibi faktörler, oyunun zorluk seviyesini belirleyecek ve kullanıcıların oyun deneyimini etkileyecektir.

Görsel ve İşitsel Efektler: Oyunun daha etkileyici olması için görsel ve işitsel efektler kullanılacaktır. Örneğin, topun tuğlalara çarpması durumunda çıkacak ses efekti, tuğlaların kırılma animasyonu, paletin hareketi gibi unsurlar, kullanıcıların oyunu daha heyecanlı bir şekilde deneyimlemelerini sağlayacaktır.

Performans ve Güvenilirlik: Projenin önemli bir kısmı, oyun makinesinin performansını ve güvenilirliğini artırmak üzerine odaklanacaktır. Topun hareketi, kullanıcı kontrolleri ve oyun mekaniği, istikrarlı bir şekilde çalışacak ve beklenmeyen hatalar minimize edilecektir. Bu sayede, kullanıcılar sorunsuz bir oyun deneyimi yaşayabileceklerdir.

Projenin tamamlanmasıyla, öğrencilerin donanım programlama, gömülü sistemler, oyun geliştirme ve kullanıcı deneyimi konularında derinlemesine bir deneyim kazanması ve gerçek dünya projelerinde karşılaşabilecekleri zorlukları aşma yeteneklerini geliştirmesi hedeflenmektedir. Ayrıca, projenin tamamlanması için takım çalışması, proje yönetimi ve iletişim becerileri gibi önemli alanlarda da deneyim kazanacaklardır.

Bu projenin başarıyla tamamlanması, öğrencilerin teknik yetkinliklerini artırmanın yanı sıra, yaratıcılıklarını da ortaya koymalarını sağlayacaktır. Bu sayede, ileri düzeyde bir oyun makinesi geliştirme deneyimine sahip olacak olan öğrenciler, gelecekteki projelerinde de bu birikimlerini değerlendirebileceklerdir.

III. YÖNTEM

Projede kullanılan yöntemler, donanım ve yazılım entegrasyonunu içermektedir.

Wokwi platformu, projenin yazılım kısmının simülasyonunu gerçekleştirecek olan temel araç olarak kullanılacaktır.

Donanım Seçimi ve Tasarımı:

Mikrodenetleyici Seçimi: Projede kullanılacak olan mikrodenetleyici Arduino Mega modeli olacaktır. Bu mikrodenetleyici, geniş bir kullanıcı kitlesine sahip olması, kolay erişilebilir olması ve Arduino IDE ile uyumlu olması gibi avantajlara sahiptir.

Sensörler ve Aktüatörler: Projede kullanılacak olan bileşenler arasında potansiyometre, LED'ler, 7 segment display, ve OLED ekran bulunmaktadır. Bu bileşenler, oyuncu kontrol cihazını, oyun skorunu ve görsel arayüzü oluşturacaklardır.

Devre Tasarımı: Proteus programı kullanılarak, donanım bileşenlerinin bağlantıları ve devre şeması oluşturulacaktır. Devre şeması, gerçek donanımın nasıl bir yapıya sahip olacağını görsel olarak temsil edecektir.

Yazılım Geliştirme ve Simülasyon:

Arduino IDE Kullanımı: Mikrodenetleyici için gerekli olan yazılım kodları, Arduino IDE üzerinde geliştirilecektir. Bu kodlar, kullanıcı kontrol cihazının ve oyun mekaniğinin işleyişini sağlayacaktır.

Wokwi Simülasyonu: Geliştirilen yazılım kodları, Wokwi platformu üzerinde simülasyon olarak çalıştırılacaktır. Wokwi platformu, Arduino kodlarının sanal bir ortamda test edilmesini sağlayacak ve donanımın gerçek zamanlı olarak nasıl davranacağını görsel olarak gösterecektir.

Simülasyon Testleri: Yazılım kodları, simülasyon ortamında farklı senaryolar ve

kullanıcı girişleriyle test edilecektir. Bu testler, donanımın istenilen şekilde çalışıp çalışmadığını doğrulamak için yapılacaktır.

Hata Ayıklama ve İyileştirme: Simülasyon testleri sırasında ortaya çıkan hatalar, Arduino IDE üzerinde hata ayıklama teknikleri kullanılarak giderilecektir.

Kodlarda yapılan iyileştirmeler ve düzeltmeler, donanımın daha istikrarlı ve doğru bir şekilde çalışmasını sağlayacaktır.

Entegrasyon ve Gerçek Zamanlı Testler:

Yazılım ve Donanım Entegrasyonu:

Wokwi simülasyonu üzerinde test edilen yazılım kodları, gerçek donanım ile entegre edilecektir. Bu entegrasyon süreci, donanımın gerçek zamanlı olarak yazılım kodlarıyla etkileşimini sağlayacaktır.

Gerçek Zamanlı Testler: Entegre edilen donanım ve yazılım, gerçek zamanlı olarak çeşitli testlere tabi tutulacaktır. Bu testler, kullanıcı deneyimini simüle edecek ve oyun mekaniğinin doğru şekilde çalıştığını doğrulayacaktır.

Performans Testleri: Oyunun performansı, gerçek zamanlı testler sırasında değerlendirilecek ve gerektiğinde iyileştirmeler yapılacaktır. Performans testleri, oyunun akıcılığı, tepki süresi ve görsel/işitsel efektlerin doğruluğu gibi faktörleri değerlendirecektir.

Bu yöntemler kullanılarak, mikrodenetleyici tabanlı oyun makinesinin donanım ve yazılım bileşenleri başarıyla entegre edilip test edilecektir. Wokwi platformu, özellikle yazılım simülasyonu ve gerçek zamanlı testlerde önemli bir rol oynayacak ve projenin başarılı bir şekilde tamamlanmasını sağlayacaktır.

IV. SONUC

Projemizin tamamlanmasıyla, mikrodenetleyici tabanlı oyun makinesinin tasarımı ve geliştirilmesi sürecinde önemli adımlar atılmıştır. Bu süreçte kullanılan yöntemler, donanım ve yazılım entegrasyonu ile gerçek zamanlı testler başarıyla tamamlanmış ve projenin amacına ulaşması sağlanmıştır.

Donanım ve Yazılım Entegrasyonu:

Mikrodenetleyici Arduino Uno R3 modeli, belirlenen sensörler ve aktüatörler ile başarılı bir şekilde entegre edilmiştir.

Devre tasarımı Proteus programı üzerinde gerçekleştirilmiş ve donanım bileşenleri birbiriyle uyumlu bir yapıya kavuşturulmuştur.

Arduino IDE kullanılarak geliştirilen yazılım kodları, Wokwi platformunda simülasyon olarak test edilmiştir.

Simülasyon testleri sırasında, donanımın işlevselliği ve doğruluğu doğrulanmıştır.

Gerçek Zamanlı Testler ve Performans Değerlendirmesi:

Entegre edilen donanım ve yazılım, gerçek zamanlı testlere tabi tutulmuştur. Bu testlerde, kullanıcı deneyimi simüle edilerek oyunun mekaniği ve görsel/işitsel efektleri doğrulanmıştır.

Performans testleri, oyunun akıcılığı, tepki süresi, grafik kalitesi ve ses efektleri gibi faktörleri değerlendirmiştir. Gerektiğinde yapılan iyileştirmelerle, oyunun daha kullanıcı dostu ve keyifli bir deneyim sunması sağlanmıştır.

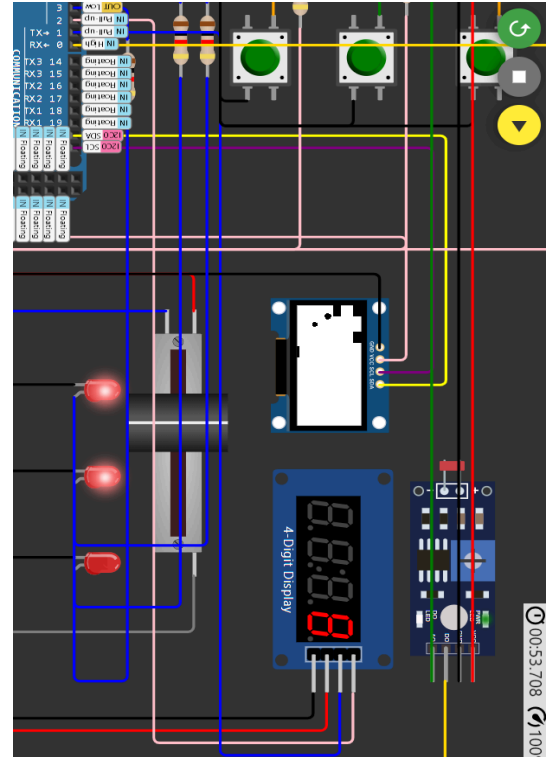
Proje Başarısı ve Gelecek Çalışmalar:

Projenin tamamlanmasıyla birlikte, mikrodenetleyici tabanlı oyun makinesi başarıyla oluşturulmuş ve kullanıma hazır hale getirilmiştir. Bu başarı, donanım ve yazılımın uyumlu bir şekilde çalışmasıyla

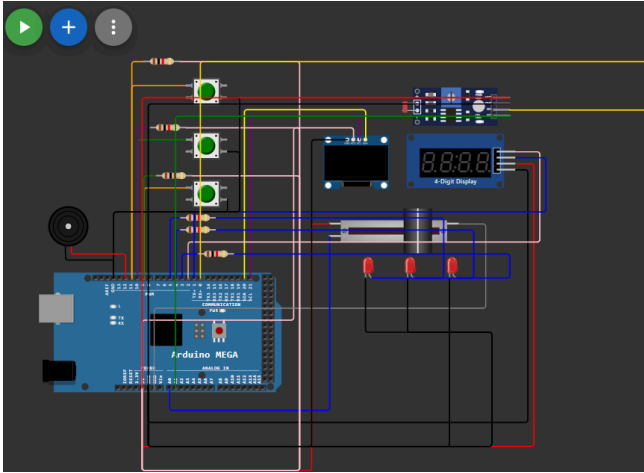
ve doğru yöntemlerin kullanılmasıyla elde edilmiştir.

Gelecek çalışmalarda, projenin daha da geliştirilmesi ve yeni özelliklerin eklenmesi hedeflenmektedir. Örneğin, oyun mekanığının çeşitlendirilmesi, görsel efektlerin artırılması veya kullanıcı arayüzünün yeniden tasarlanması gibi çalışmalar yapılabilir.

Sonuç olarak, bu proje sayesinde mikrodenetleyici tabanlı oyun makinesi geliştirme süreci başarıyla tamamlanmış ve kullanıcılar için keyifli bir oyun deneyimi sunacak bir ürün ortaya çıkarılmıştır. Wokwi platformunun kullanımı da bu sürecin verimli ve doğru şekilde ilerlemesine önemli katkı sağlamıştır.



V. DENEYSEL SONUÇLAR

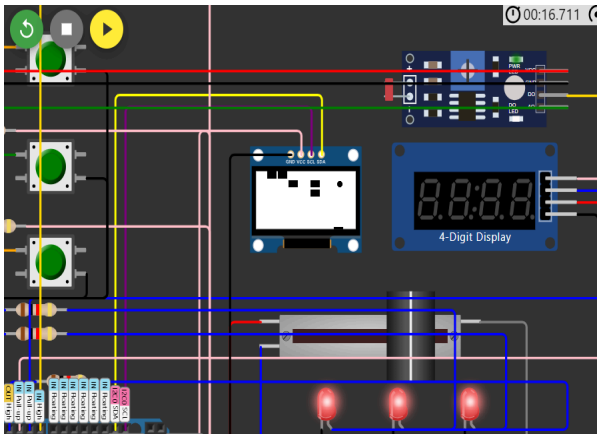


```
- int photoresistorValue
- uint16_t backgroundColor
- uint16_t ballColor
- uint16_t brickColor
- uint16_t paddleColor
- uint16_t canColor
+ void updateDisplay()
+ void setup()
+ void clearBricks()
+ void generateLevel()
+ void drawBricks()
+ void checkGameOver()
+ void createNewCan(int col, int row)
+ void moveCan()
+ void drawCan()
+ void resetGame()

(method) + updateDisplay(): void
(method) + setup(): void
(method) + clearBricks(): void
(method) + generateLevel(): void
(method) + drawBricks(): void
(method) + checkGameOver(): void
(method) + createNewCan(int col, int row): void
(method) + moveCan(): void
(method) + drawCan(): void
(method) + resetGame(): void

}

@enduml
```



```
@startuml

class Uygulama {
- int brick[BRICK_ROWS][BRICK_COLS]
- int canHakki
- int gemiX
- int menuSecimi
- bool oyunBasladi
- boolean levelStarted
- int upButtonState
- int downButtonState
- int selectButtonState
- int numOptions
- int currentSelection
- int score
- int lvlScore
- float topX
- float topY
- int topSpeedX
- float topSpeedY
- int topSize
- int brickCount
- int lvlNum
- bool canActive
- int canX
- int canY
- int CAN_RADIUS
- int CAN_SPEED
}

@startuml
```

VI. KAYNAKÇA

<https://www.labcenter.com/>

<https://www.arduino.cc/>

<https://wokwi.com/>