

# PROGRAMLAMA LABORATUVARI-2

## PROJE-2

1<sup>st</sup> Bilge Çeşme  
Kocaeli Üniversitesi

Kocaeli/Türkiye  
220201045

2<sup>nd</sup> Selim Eren Kaya  
Kocaeli Üniversitesi

Kocaeli/Türkiye  
230201127

### I. ÖZET

Bu proje, mikrodenetleyici tabanlı bir oyun makinesi oluşturmak için çeşitli araçları kullanmayı amaçlamaktadır.

### II. PROJE TANIMI

Aşağıdaki isterlere göre mikrodenetleyici tabanlı oyun makinesi yapılması istenmiştir

### III. İSTERLER

- Ana Menü ve Seçenekler: Oyun başlangıcında, kullanıcıya "Başlat" ve "Çıkış" seçenekleri sunulur. Kullanıcı, yukarı ve aşağı tuşlarıyla seçenekler arasında gezinebilir ve "Başlat" seçeneğini onaylayarak oyuna başlayabilir.
- Oyun Başlangıcı: "Başlat" seçeneği seçildiğinde, oyun ekranı açılır ve oyun başlar. Oyuncu, paleti sağa sola hareket ettirerek topu oyun alanında tutmaya çalışır.
- Tuğlaların Kırılması ve Skor: Top, paletle çarpışarak tuğlaları kırar. Her kırılan tuğla, oyuncunun skorunu bir puan artırır. Bu skor, 7 segment displayde gösterilir.
- Can Sistemi ve LED Göstergesi: Oyuncunun başlangıçta 3 canı vardır. Her top paletin altına düştüğünde, oyuncunun bir canı azalır. Canlar, LED'lerle gösterilir ve her can kaybıyla bir LED söner.
- Özel Objeler ve Can Kazanımı: %10 olasılıkla düşen özel objeler, oyuncuya bir can kazandırır. Bu objeler, kırılan tuğlalardan aşağı doğru hareket eder ve alındığında oyuncunun can sayısı artar. Can kazanımıyla birlikte LED göstergesi güncellenir.

- Oyun Sonu ve Skor Gösterimi: Oyuncunun 3 canı bittiğinde, ekranda son skor görüntülenir ve oyuncu ana menüye geri döner.
- Renk Değişimi: Arka plan ve tuğlaların rengi, ışık sensörüne bağlı olarak değişir. Işık açıkken arka plan siyah, tuğlalar ise beyaz olur.
- Hız Artışı ve Bölüm Geçişi: Her bölüm geçildiğinde, topun hızı %20 artar. Bu şekilde oyun zorluk seviyesi artar ve oyuncunun becerisi sınanır.

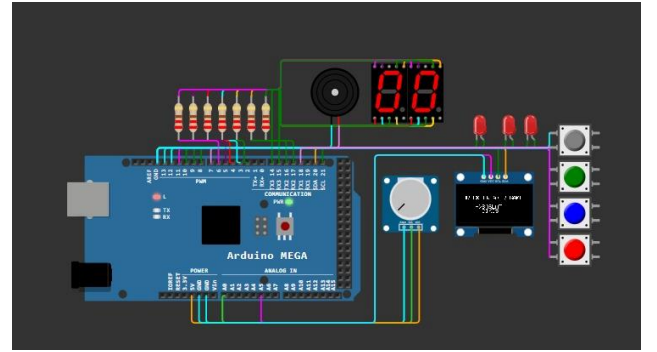


Fig.1: oyun

### IV.VOİD SETUP()

İşlevin amacı, projenin başlangıç ayarlarını yapmaktır. İşlevler arasında, bir OLED ekran başlatma, bir seri iletişim portunu başlatma, LED'leri, düğmeleri ve 7 segment ekranları başlatma, hoparlör pinini çıkış olarak ayarlama ve oyunu hazırlama gibi çeşitli ön ayarlar bulunmaktadır. Ayrıca, rastgele sayı üretimi için analog bir pin üzerinden gürültü kullanarak randomSeed() fonksiyonu çağrılır. Bu parça, projenin başlangıç aşamasında donanımın

hazırlanmasını sağlar ve oyunun başlamaya hazır  
V.VOID LOOP()

Bu loop() fonksiyonu Arduino programının ana döngüsünü temsil eder. Her döngüde, belirli işlevler çağrılır ve oyun durumuna bağlı olarak farklı ekranlar ve kontroller yönetilir.

segmentAyarla(): Bu işlev, 7 segment displayi ayarlar ve skor gibi bilgileri gösterir.

renkDegis(): Oyunun rengini kontrol eder ve ışık sensörüne bağlı olarak arka plan ve tuğlaların rengini değiştirir.

if (cikisEkran == 1) { ... }: Eğer çıkış ekranı açılmışsa, drawCikisEkran() işlevini çağırarak çıkış ekranını çizer.

else if (anaMenu == 1) { ... }: Eğer ana menü ekranı açılmışsa, drawMenu() işlevini çağırarak ana menüyü çizer ve kullanıcı butonları kontrol eder.

else { ... }: Ana menü veya çıkış ekranı açık değilse, oyundur. Oyun ekranı açıktır.

if (oyunBitme == 0) { ... }: Oyun devam ediyorsa, paletin ve topun hareketlerini kontrol eder, ayrıca objelerin hareketini kontrol eder ve bölüm tamamlanmasını kontrol eder.

drawGame(): Oyun ekranını çizer.

Bu döngü, kullanıcının oyunun farklı bölümlerinde gezinmesine ve oyun durumuna bağlı olarak gerekli işlevleri gerçekleştirmesine olanak tanır.

## VI.VOID drawMenu()

Bu fonksiyon, oyunun ana menü ekranını çizer. İşlevin içeriği şu adımları izler:

display.clearDisplay(): Ekranı temizler, böylece yeni içerik çizilmeye hazır hale gelir.

display.setTextSize(1): Yazı boyutunu ayarlar.

display.setTextColor(renkOyun): Yazı rengini, oyunun rengine uygun olarak ayarlar.

display.setCursor(SCREEN\_WIDTH / 15, SCREEN\_HEIGHT / 4): Yazının ekran üzerindeki konumunu belirler. Başlık metni ekranın üst kısmında yatay olarak ortalır.

display.println("BRICK BREAKER GAME\n"): Başlık metnini ekrana yazar. "\n" ile bir alt satıra geçirir.

if (seciliTus == 0) { ... }: Eğer "BASLAT" seçiliyse ">" işareti ile vurgular, aksi halde "CIKIS" seçili olarak gösterilir.

display.display(): Ekranı günceller, çizilen içeriği gösterir.

hale gelmesini sağlar.

Bu işlev, oyuncuya oyunu başlatma veya çıkış yapma seçeneklerini sunar ve kullanıcının seçimini belirtir.

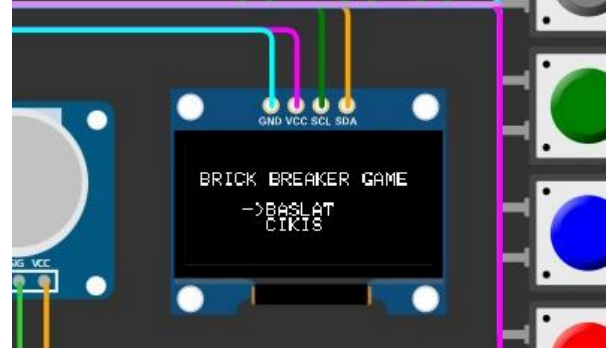


Fig.2: menü

## VII.VOID DRAWGAME

Bu drawGame() işlevi, oyun ekranını çizer. İşlevin içeriği şu adımları izler:

1. **display.clearDisplay()**: Ekranı temizler, böylece yeni içerik çizilmeye hazır hale gelir.
2. Tuğlaların çizimi: **if (bolumIndex >= 1 && bolumIndex <= bolumSayisi && oyunBitme == 0)**: Bu koşul, eğer oyun devam ediyorsa ve geçerli bir bölüm varsa tuğlaların çizimini yapar.

Bölümdeki tuğlaların konumlarını ve durumlarını kontrol eder ve ekrana çizer.

3. Can göstergesi:Kullanıcının can sayısına göre LED'leri açar veya kapatır.
4. Oyun bitiş durumu: **if (oyunBitme == 1)**: Oyunun bitmiş olup olmadığını kontrol eder. Eğer oyun bitmişse "GAME OVER" mesajı ve skorunu ekrana yazar. Ardından oyunu sıfırlar, bölümleri sıfırlar ve ana menüye döner.
5. Palet ve topun çizimi: Palet ve topun mevcut konumlarını ekrana çizer.
6. Objelerin çizimi: Ekranda hareket eden objelerin (örneğin, ek canları temsil eden objeler) çizimini yapar.
7. **display.display()**: Ekranı günceller, çizilen içeriği gösterir.

Bu işlev, oyun ekranındaki tüm unsurları güncelleyerek oyuncuya oyunu oynama imkanı sunar.

## VIII.VOİD MOVEPADDLE()

movePaddle() işlevi, kullanıcının paleti hareket ettirmesini sağlar. İşlev, potansiyometreden okunan değeri kullanarak paletin hareket hızını belirler. Ardından, paletin X konumunu günceller ve ekranın kenarlarına ulaşp ulaşmadığını kontrol eder, böylece palet ekran sınırlarını aşmaz. Bu işlev, oyuncunun oyun deneyimini kontrol etmesine olanak tanır ve paletin ekran içinde akıcı bir şekilde hareket etmesini sağlar.

## IX.VOİD MOVEBALL()

moveBall() fonksiyonu, oyun içindeki topun hareketini kontrol eder. Her döngüde, topun X ve Y konumlarına belirli bir hızda ilerletilir. Ardından, topun ekranın kenarlarına çarpıp çarpmadığını kontrol eder. Eğer top ekranın sol veya sağ kenarlarına çarparsa, yönünü tersine çevirir. Üst kenara çarparsa, yönünü yine tersine çevirir.

Topun paletle çarpışmasını kontrol eder. Eğer top paletin alt kenarına değerse ve paletin içinde kalıyorsa, yönünü yine tersine çevirir. Bu durumda bir ses efekti de çalınır.

Ardından, topun tuğlalarla çarpışmasını kontrol eder. Eğer top bir tuğlaya çarparsa, çarpışma noktasını belirler ve topun yönünü buna göre değiştirir. Çarpmanın ardından tuğla kırılır ve oyuncunun skoru artırılır.

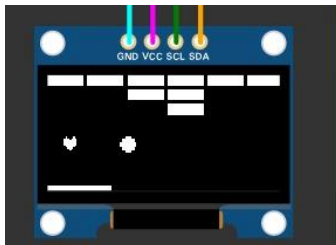
Son olarak, topun ekranın alt kenarına düşüp düşmediğini kontrol eder. Eğer top ekranın alt kenarına düşerse, oyunu sonlandırır.

## X.VOİD DRAWOBJELER()

drawObjeler() fonksiyonu, oyundaki özel objelerin çizimini yapar. Bu objeler, oyuncunun canını artıran kalp şeklinde sembollerdir.

Her döngüde, objeler dizisindeki her bir öge için kontrol yapılır. Eğer obje etkinse (objeler[index][2] == 1), objenin X ve Y konumları hesaplanır ve bir kalp şekli çizilir. Kalp şekli, 8x8 piksel boyutunda bir dizide saklanır.

Daha sonra, kalp şekli çizilirken, her pikselin durumu kontrol edilir. Eğer pikselin değeri 1 ise (beyaz piksel), o piksel çizilir. Bu şekilde, her bir objenin kalp sembolü ekran üzerinde çizilir.



## XI.VOİD OBJEDUSURME()

objeDusurme() fonksiyonu, belirli bir tuğla kırıldığında ekranda bir objenin düşme olasılığını kontrol eder.

Her tuğla kırıldığında, random() fonksiyonu kullanılarak 1 ile 10 arasında rastgele bir sayı oluşturulur. Bu sayı 5 ise (yaklaşık olarak 1/10 olasılıkla), bir objenin düşme olasılığı vardır. Bu durumda, ilgili tuğlanın konumunu belirten dizideki üçüncü değer (objeler[index][2]) 1 olarak ayarlanır, böylece o tuğlanın altında bir obje görüntülenir.

Bu sayede, oyuncular rastgele objelerin düşme olasılığını deneyimlerken, oyunun çeşitliliği artırılmış olur.

## XII.VOİD MOVEOBJELER ()

moveObjeler() fonksiyonu, ekrandaki objelerin düşme hareketini kontrol eder.

Her döngüde, 18 adet objeyi kontrol eder. Eğer bir obje düşüyorsa (objeler[index][2] == 1), objenin yatay konumu (objeX) ve dikey konumu (objeY) güncellenir. Düşen objelerin dikey konumları her adımda bir artırılarak aşağı doğru hareket ederler.

Eğer obje, ekranın alt kısmına ulaşırsa (objeY >= SCREEN\_HEIGHT), obje ekrandan çıkarılır ve yok edilir (objeler[index][2] = 0).

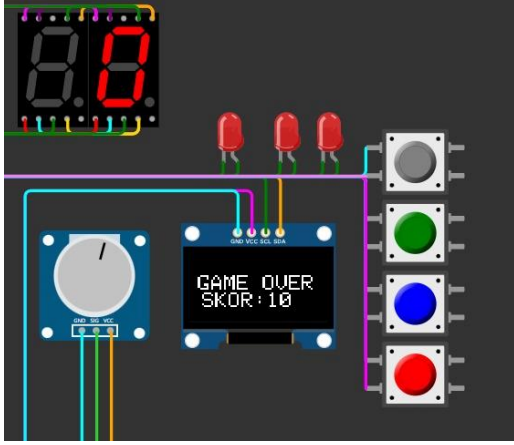
Ayrıca, objenin alt kısmının raketle çarpışıp çarpışmadığı kontrol edilir. Eğer obje, raketin altına değerse ve raketle çarpışma gerçekleşirse, oyuncunun can sayısı artırılır ve obje yok edilir.

## XIII.VOİD GAMEOVER()

gameOver() fonksiyonu, oyunun sonlandırılmasını ve gerekli işlemlerin yapılmasını sağlar.

Öncelikle, oyuncunun can sayısı azaltılır (canSayisi--). Daha sonra, eğer oyuncunun can sayısı 0'a düşerse, oyun bitirilir (canSayisi == 0). Bu durumda oyunBitme değişkeni 1 olarak ayarlanır ve topun hareketini durdurmak için ballSpeedY ve ballSpeedX değerleri sıfıra eşitlenir. Oyuncuya kaybetme efekti verilir (kaybetmeEfekti()). Ancak, eğer oyuncunun can sayısı hala 1 veya daha fazlaysa, oyuncuya can kaybetme efekti verilir (canKaybetmeEfekti()). Raketin ve topun başlangıç pozisyonlarına geri döner ve oyuncu devam edebilir.

Bu fonksiyon, oyuncunun canlarını kaybetmesi durumunda gerekli işlemleri gerçekleştirir ve oyunun devam edip etmeyeceğini kontrol eder.



### XIII.VOİD CHECKLEVELCOMPLETION()

Bu fonksiyon, bir bölümün tamamlanıp tamamlanmadığını kontrol eder ve gerekli işlemleri yapar. Eğer bölüm tamamlanmışsa bir sonraki bölüme geçer, eğer tüm bölümler tamamlanmışsa oyunu bitirir.

Bölümdeki tuğlaların tamamının kırılıp kırılmadığını kontrol eder. Eğer en az bir tuğla hala kırılmadıysa, fonksiyon geri döner ve bölüm tamamlanmamış demektir.

Eğer tüm tuğlalar kırıldıysa, o bölüm tamamlanmış demektir. Bu durumda, bir sonraki bölüme geçilir. Eğer tüm bölümler tamamlanmışsa, oyun bitirilir ve kazanma efekti verilir.

Eğer bir sonraki bölüme geçiliyorsa, ekran temizlenir ve bir sonraki bölüme geçiş mesajı ve güncel skor gösterilir. Ardından, bir sonraki bölümün başına dönülür ve topun hızı artırılır.

Bu fonksiyon, bir bölümün tamamlanıp tamamlanmadığını kontrol eder ve gerekli işlemleri yapar. Eğer bölüm tamamlanmışsa bir sonraki bölüme geçer, eğer tüm bölümler tamamlanmışsa oyunu bitirir.

### IXX.VOİD OYUNHAZIRLA ()

oyunuHazirla() fonksiyonu, oyunun başlangıç durumunu ayarlar. Bu fonksiyon, oyunun başlangıç ekranına döndüğünde veya bir oyun tamamlandığında çağrılır.

Paletin başlangıç konumunu yatayda ekranın ortasına yerleştirir. Topun başlangıç konumunu ve hızlarını ayarlar. Oyuncunun skorunu sıfırlar. Oyuncunun hangi bölümde olduğunu ve oyunun bitip bitmediğini sıfırlar. Oyuncunun can sayısını başlangıç değerine ayarlar.

### IXX.VOİD BOLUMLERISIFIRLA ()

bolumleriSifirla() fonksiyonu, oyun bölümlerini ve düşen objeleri sıfırlar. Bu fonksiyon, yeni bir oyun başlatıldığında veya oyun sıfırlanması gerektiğinde çağrılır.

Her bir bölüm için orijinal tuğla düzenini içeren matrisler oluşturur. Her bir bölüm matrisini, orijinal düzen matrisine kopyalar. Düşen objelerin matrisini sıfırlar. Bu işlemler, oyunun başlangıç durumunu ve bölüm düzenlerini sıfırlar, böylece yeni bir oyun başlatıldığında veya oyun sıfırlanırken doğru düzeni sağlar.

### XX.VOİD BUTONKONTROL ()

Fonksiyon, kullanıcının menüdeki seçenekleri değiştirmesini ve seçim yapmasını sağlar. Fonksiyon, kullanıcının butonlara basarak menüde gezinmesini ve seçim yapmasını kontrol eder.

### XXI.VOİD BUTONKONTROL ()

Bu fonksiyonu, çıkış ekranını ekrana çizdirir. Bu ekran, kullanıcının oyunu terk ettiğinde veya çıkış yaptığıda gösterilir. Ekranda "OYUNUMUZA GÖSTERDİĞİNİZ İLGI ICIN TESEKKURLER" gibi bir teşekkür mesajı bulunur.

### XXII .SES EFEKTLERİ

vurmaEfektiTugla(): Tuğlalara topun çarpması durumunda çalınacak ses efekti.

vurmaEfektiPalet(): Palet ile topun çarpması durumunda çalınacak ses efekti.

canAlmaEfekti(): Kullanıcının ek can kazandığı durumda çalınacak ses efekti.

kazanmaEfekti(): Oyunu kazandığınızda çalınacak özel bir melodi.

kaybetmeEfekti(): Oyunu kaybettiğinizde çalınacak bir kaybetme efekti.

canKaybetmeEfekti(): Kullanıcının can kaybettiği durumda çalınacak ses efekti.

Her işlevde, tone() fonksiyonu kullanılarak belirli bir frekansda bir nota çalınır. Ardından, belirli bir süre boyunca bekleme işlemi gerçekleştirilir ve ses, noTone() fonksiyonu ile durdurulur. Bu işlemler, Arduino'nun dahili zamanlama fonksiyonları kullanılarak gerçekleştirilir.

## XXIII .SEVEN SEGMENT DISPLAY

segmentAyarla() fonksiyonu, gösterilecek sayının onlar ve birler basamağına ayırır. Daha sonra, her bir basamağı sırasıyla ekrana yazdırmak için number() fonksiyonunu kullanır. Her bir basamağın gösterimi için ilgili basamağın pinini açar ve diğer basamağın pinini kapatır. Bu işlem, ekranda iki basamaklı bir sayının gösterimi için tekrarlanır.



Fig.

## XXIII .RENKDEĞİŞ

Bu fonksiyon, belirli bir düğmeye basıldığında ekrandaki renkleri tersine çevirir veya normal hale getirir.

digitalRead(BUTON\_PIN4) == LOW ifadesi, bir düğmeye basılıp basılmadığını kontrol eder.

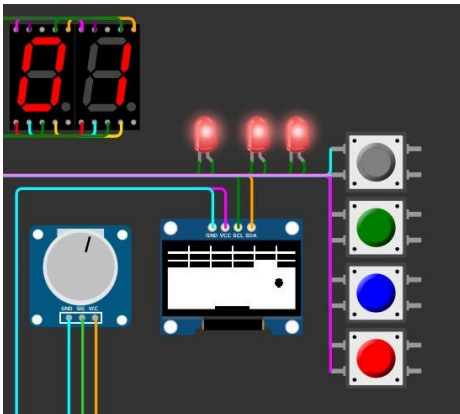
lightMode değişkeni, ekrandaki renk modunu takip eder ve başlangıçta 0 olarak ayarlanır.

Eğer belirli bir düğmeye basıldıysa, lightMode değeri kontrol edilir.

Eğer lightMode 0 ise (başlangıç durumu), display.invertDisplay(true) çağrılır ve lightMode değeri 1 olarak ayarlanır.

Eğer lightMode 1 ise, display.invertDisplay(false) çağrılır ve lightMode değeri 0 olarak ayarlanır.

Her iki durumda da, renk değişim efektinin daha belirgin olması için belirli bir süre beklenir.



## XXIV.UML DİYAGRAMI



## XXV.KATKILAR

Bu ekip çalışmasında Bilge Çeşme tuğla, top ve paletin kontrolünde; elemanların bağlanması görev almıştır. Selim Eren Kaya ise seven segment display ,objeler , efektler , hareket ve butonlar ile ilgilenmiştir.

## XXVI. KAYNAKÇA

<https://www.metehoca.com/akademi/arduino-modul/tm1637-4-digit-display-hw-069-arduino-ile-nasil-kullanilir-1025/>

[https://superomar.xyz/product\\_details/22030390.html](https://superomar.xyz/product_details/22030390.html)

<https://www.instructables.com/Arduino-Pong-3/>

