

Prolab-II proje-II

Meryem PAŞAOĞLU
Bilgisayar Mühendisliği
220201123@kocaeli.edu.tr

Abstract—Brick-breaker mantığında bir oyun istenmiştir. Oyuncu, bir fiziksel palet kontrol cihazını kullanarak bir topu yansıtarak OLED ekrandaki tuğlaları kırmaya çalışacaktır. Oyun, topun alt kısmından düşmesi durumunda sona erer.

I. GİRİŞ

Bu proje, bir tuğla kırma oyunundan oluşur. Oyun, bir oyun menüsü ile başlar. Menüde "Başlat" seçeneği seçilirse, oyun başlar. Oyun sırasında oyuncu bir paleti kontrol eder ve bir topu yukarı doğru yönlendirir. Oyuncu, topun tuğlaları kırmak için yönünü ayarlar. Top, tuğlalarla temas ettiğinde, tuğlalar kırılır ve oyuncunun puanı artar. Ayrıca, ekstra can paketi belirli bir olasılıkla görünür. Bu paketi toplamak, oyuncunun canını artırır. Tuğlaların tamamı kırıldığında veya oyuncunun canı tükendiğinde oyun biter.

A. isterler

Projede istenilen isterler şu şekildedir:

- açılacak ekranda "Başlat" ve "Çıkış" seçenekler olacak. Kullanıcı bunlara aşağı ve yukarı tuşlarıyla gelerek 3.br tuş le menüde bulunan seçeneklerden brn seçebilecektir.
- "Başlat" tuşuyla oyun ekranı açılacak ve oyun başlayacaktır."Çıkış" seçeneğinde ise "Oyunumuza gösterdiğiniz ilgi için teşekkürler" yazısı yazacaktır.
- Oyun başladıktan sonra kullanıcının kontrol ettiği palet potansiyometre ile sağ ve sol hareketler yaptırılacaktır
- Palet ile yönlendirme yapılan top çarptığı tuğlaları kırarak ve yön değiştirecektir. Ayrıca top platformun sağ, sol ve platformun bitişinin üst tarafındaki duvara çarpıncada yön değıştrmelidir
- Yön değıştirmeler fizik kuralları çerçevesinde olmalıdır.
- Her tuğla kırıldığında seven segment displayda gösterilecek skor puanı 1 artmalıdır.
- Her kullanıcı oyuna başladığında 3 canı vardır. Bu canlar led le gösterilmelidir.
- Top, palet ile kurtarılamayıp aşağı düştüğünde oyuncunun 1 canı gider. Daha sonra tekrar top paletin üstünde olarak oyun başlar.
- Her tuğla kırıldığında yüzde 10 şans ile bir obje düşürecektir.Bu obje kırıldığı tuğladan sabit olarak aşağı yöne doğru hareket edecektir.
- Oyuncu bu objeyi alırsa can sayısı 1 artacaktır. Bu eşzamanlı olarak led ile gösterilecektir.
- Oyuncunun 3 canı bittğinde ekranda son olarak 3 saniye skor puanı belirterek tekrar ana menüye geri döner.
- oyunun arka planı siyah, tuğlalar beyaz olacaktır. Bu 2 objenin rengi ışık sensörüne göre tam tersine değışebilecektir.

- Oyundaki tüm tuğlalar bitince 5 sanye ara ekran girecek ve daha sonra diğer bölüm başlayacaktır. Bu bölümdek tuğla dizaynının aynı olmaması beklenmektedir. Skorun kaldığı yerden devam etmesi beklenmektedir. Ayrıca her bölüm geçildiğinde topun hızı bir önceki tura göre yüzde 20 artacaktır.

II. YÖNTEM

Projede bulunan değışkenler ve metotlar şu şekilde özetlenebilir:

ilk olarak, iki global değışken tanımlanıyor: oyunnbayragii ve menuuseciimii. oyunnbayragii oyunun başlayıp başlamadığını belirleyen bir bayrak ve menuuseciimii ise menü seçeneğinin değerini saklayan bir değışken. anaamenuugosterr() fonksiyonu, LCD ekranda ana menüyü göstermek için kullanılıyor. Bu fonksiyon, ekrana oyun seçeneklerini ve mevcut seçeneği işaretleyen bir ok gösterir. menuisaretlee() fonksiyonu, kullanıcının menüde gezinmesini sağlar ve seçimlerine göre işlem yapar. Üç farklı butonun durumunu okur: yukarı, aşağı ve onay butonları (yukariibuttonuubass, asagiibuttonuubass, okeyybuttonuubass). Ardından, kullanıcının butonlara basıp basmadığını kontrol eder. Yukarı butonuna basıldığında ve seçenek 2 değilse (menuuseciimii != 2), seçeneği bir artırır ve menüyü günceller. Aşağı butonuna basıldığında ve seçenek 1 değilse (menuuseciimii != 1), seçeneği bir azaltır ve menüyü günceller. Onay butonuna basıldığında (okeyybuttonuubass == LOW), kullanıcının seçimine göre işlem yapılır. Eğer seçenek 2 ise (menuuseciimii == 2), oyunnbayragii bayrağı true yapılır, yani oyun başlatılır. Eğer seçenek 1 ise (menuuseciimii == 1), teşekkür mesajı gösterilir, bazı temizleme işlemleri yapılır ve program sonlandırılır.

lk olarak, bool tuglalaarr[tuglaninnsatirrsayisii][tuglaninnsutunnsayisii]; satırıyla tuglalaarrr adında bir iki boyutlu bool tipinde dizi tanımlanıyor. Bu dizi, tuğlaların durumunu tutacak; yani hangi tuğlaların kırık veya sağlam olduğunu belirleyecek. Ardından, setup() fonksiyonu geliyor. Bu fonksiyon Arduino'nun başlangıç ayarlarını yapar. Serial.begin(9600); satırı, seri haberleşme bağlantısını başlatır. Bu, Arduino'nun seri monitörü kullanarak bilgi göndermesini ve almasını sağlar. if(!display.begin(SSD1306SWITCHCAPVCC, 0x3C)) ... satırı, OLED ekranını başlatır. Eğer ekran başlatılamazsa, seri monitöre bir hata mesajı gönderir ve program durdurulur. Ardından, ekran temizlenir, bir süre beklenir ve ekran tekrar temizlenir. randomSeed(analogRead(0)); satırı, rastgele sayı üretmek için bir başlangıç değeri belirler. Bu, rastgele sayı

üreticinin daha tahmin edilemez olmasını sağlar. LED pinlerini çıkış olarak ayarlar. Fotoğraf direnci sensörünü giriş olarak ayarlar. Yukarı, aşağı ve enter butonlarını giriş olarak ayarlar. Sonra, tuğlaları başlangıçta oluşturur. İki döngü kullanarak tuğlalaarr dizisini doldurur. Başlangıçta, tüm tuğlaların sağlam olduğunu belirtmek için her bir elemanı "true" olarak ayarlar. leevveel ve tuglaadizaynii() değişkenlerini ve fonksiyonlarını belirler. Bu değişken ve fonksiyonların kodun geri kalanında kullanılacağını gösterir. Skoru ekrana yazdırır. 7 segment ekran için pinleri çıkış olarak ayarlar. İkinci 7 segment ekran için pinleri çıkış olarak ayarlar. Paletin başlangıç konumunu hesaplar ve potansiyometre değerini okur. Topun başlangıç konumunu paletin üstünde ayarlar

tuglaadizaynii fonksiyonu : Fonksiyon, belirli bir oyun seviyesine bağlı olarak tuğlaların düzenini oluşturur. leevveel değişkeni, oyunun hangi seviyede olduğunu belirtir. Bu değişkenin değerine göre tuğlaların dizilimi farklı olacaktır. Her seviye için farklı bir tuğla düzeni belirlenmiştir. Bu seviyelere göre belirlenen tuğla düzenleri: Seviye 1 (Level 1): Piramit şeklinde düzen. En üstte 1 tuğla, altında 3 tuğla, altında 5 tuğla, ve böyle devam eder. Her seviyede tuğla sayısı 2 artar. Seviye 2 (Level 2): Seviye 1'in tersi şeklinde bir düzen. En üstte en fazla sayıda tuğla, altında azalan sayıda tuğla bulunur. Seviye 3 (Level 3): Düz bir sıra şeklinde tuğlalar. Her seviyede sütun sayısı 1 artar. Seviye 4 (Level 4): Tüm tuğlaların olduğu bir düz sıra. Tüm satır ve sütunlarda tuğlalar mevcuttur. Seviye 5 (Level 5): Genişleyen bir üçgen şeklinde tuğlalar. Her satırda sütun sayısı azalır. Fonksiyon, seviyeye göre tuğlaların varlığını belirler. tuglalaarr adlı iki boyutlu diziye kullanarak, her bir tuğlanın var olup olmadığını belirtir. Tuğla dizilimi oluşturulduktan sonra, eğer seviye 1'den büyükse, ekranda seviye bilgisini gösterir. Bu bilgi, display nesnesi kullanılarak ekrana yazılır ve bir süre bekletilir. Tüm bu işlemler sonucunda, tuğlaların varlığına ve düzenine ilişkin bilgiler elde edilir. Bu bilgiler, oyunun belirlenen seviyesine uygun olarak ekran üzerinde gösterilir.

Fonksiyon, yediisegmenttskoruugosterr adıyla tanımlanır ve üç parametre alır: sayiii, segmenttpinleriii[] ve ortakkatott. İlk olarak, if (sayiii < 9) koşuluyla kontrol edilir. Eğer gösterilmek istenen sayı 9'dan büyükse, ekranın bir rakam göstermesi gerektiğinden emin olmak için sayı 1 olarak değiştirilir. Örneğin, 10 sayısı 1 olarak gösterilecektir, çünkü 7 segment ekranlar genellikle sadece bir rakam gösterebilir. Ardından, her bir segment pinine, sayının gösterilmesi için gerekli olan değerler atanır. rakamlaarr[sayiii][i] ifadesi, belirli bir rakamın 7 segment ekran üzerindeki her bir segmentini temsil eden bir dizi içindeki değerlere ulaşır. Bu değerler, her segmentin yanıp söneceği veya yanmayacağı bilgisini içerir. Son olarak, ortakkatott pinine LOW değeri verilerek ekran kapatılır. Çoğu 7 segment ekranın ortak katodu vardır. Bu nedenle, ekranın çalışması için ortak katot pinine HIGH, yani pozitif bir voltaj uygulanmalıdır. Ancak bu fonksiyon, ekranı kapatmak için ortak katot pinine LOW, yani düşük bir voltaj uygular. Bu, segmentlerin yanmayacağı ve ekranın kapanacağı anlamına gelir. Bu fonksiyon, belirli bir sayının 7 segment ekran üzerinde

gösterilmesini sağlar ve ekranı kapatır. Bu şekilde, belirli bir skor veya sayı kullanıcıya gösterilir.

Fonksiyonun adı skoruarttır() olarak tanımlanmıştır. Fonksiyon içerisinde skorpuannii değişkeni bir arttırılarak skorun güncellenmesi sağlanır. Bu işlem skorpuannii++ şeklinde ifade edilmiştir. Burada ++ operatörü, değişkenin değerini bir arttırır. Bu fonksiyonun çağrılmasıyla, oyun sırasında skorun arttırılması ve güncellenmesi sağlanır. ledddurumunuuguncellee fonksiyonu: Öncelikle, cannsayiisii değişkeninin değerine göre belirli koşullar kontrol edilir. Eğer cannsayiisii değeri 2 ise, yani oyuncunun 2 canı kalmışsa, ilk LED kapatılır (digitalWrite(ledd1pinii, LOW);). Eğer cannsayiisii değeri 1 ise, yani oyuncunun 1 canı kalmışsa, ikinci LED kapatılır (digitalWrite(ledd2pinii, LOW);). Eğer cannsayiisii değeri 0 ise, yani oyuncunun canı kalmamışsa, üçüncü LED kapatılır (digitalWrite(ledd3pinii, LOW);) ve oyunnbitttii() fonksiyonu çağrılarak oyun sonlandırılır. Ardından, eğer oyuncunun can sayısı sıfırdan büyük ise, yani oyuncu yeni bir can kaybetmemişse, paletin ve topun konumları yeniden ayarlanır. Paletin x konumu, ekranın genişliğinin ortasına yerleştirilir. Topun x konumu, paletin x konumuna eklenen paletin genişliğinin yarısı ve çıkarılan topun genişliğinin yarısı ile belirlenir. Böylece top, paletin üstünde merkezi bir konuma yerleştirilir. Topun y konumu, ekranın yüksekliğinden paletin yüksekliği ve topun yüksekliği çıkarılarak belirlenir. Bu şekilde top, ekranın üstünde ve paletin üstünde başlayacak şekilde konumlandırılır. Bu fonksiyon, oyuncunun can sayısına bağlı olarak LED'leri günceller ve oyun durumunu kontrol eder. Oyuncunun can sayısı değiştikçe, LED'ler kapatılır ve palet ile topun konumları yeniden ayarlanır.

oyunnbitttii() fonksiyonu: Bu fonksiyon, oyunun bitirilmesi durumunda çalışır. Öncelikle, ekran temizlenir ve "Game Over" mesajı ile oyuncunun skoru ekrana yazdırılır. Ardından, bir süre beklenir (delay(3000)) ve ekran temizlenir. Son olarak, oyuncuya tekrar oynamak isteyip istemediği sorulması için tekrarroynaa() fonksiyonu çağrılır.

tekrarroynaa() fonksiyonu: Bu fonksiyon, oyuncuya tekrar oynamak isteyip istemediğini sormak için kullanılır. İlk olarak, ekran temizlenir ve oyun seçenekleri ("BASLAT" ve "CIKIS") ekrana yazdırılır. Ardından, oyuncunun yukarı ve aşağı butonlarına basıp basmadığı sürekli olarak kontrol edilir. Eğer oyuncu "BASLAT" seçeneğini seçerse, oyun yeniden başlatılır. Bu durumda: Oyun bayrağı (oyunbayragii) true olarak ayarlanır. Seviye (leevveel) 1'e döndürülür. Can sayısı (cannsayiisii) maksimum can sayısı değerine döndürülür. Skor (skorpuannii) sıfırlanır. Topun hareket yönleri ve konumu başlangıç değerlerine döndürülür. Tuğla düzeni yeniden oluşturulur. LED'ler yeniden aktif hale getirilir. Eğer oyuncu "CIKIS" seçeneğini seçerse, oyun kapanır: Oyuncuya teşekkür mesajı gösterilir. Belirli bir süre beklenir. LED'ler kapatılır. Program sonlandırılır (exit(0)). Bu fonksiyonlar, oyunun sonlandırılması ve tekrar oynama seçeneğinin sunulması için gereken adımları içerir. Oyuncu, oyunu tekrar oynamak isterse, oyun yeniden başlatılır; aksi takdirde, program sonlandırılır.

loop() fonksiyonu:

Oyun Başladıysa Eğer oyunnbayragii bayrağı true ise, yani oyun başladıysa: Ekran temizlenir. Fotoğraf direnci sensöründen okunan değer alınarak arka plan rengi belirlenir. Eğer okunan değer 500'den büyükse arka plan beyaz, değilse siyah olur. Tuğlaların, paletin ve topun rengi, arka plan renginin tersine göre ayarlanır. Tuğlalar, palet ve top, belirlenen renklerle ekrana çizilir. Potansiyometre değeri okunarak paletin konumu güncellenir. Potansiyometre değeri değişimine bağlı olarak paletin x konumu ayarlanır. Ancak palet, ekranın sınırlarını aşmamalıdır, bu nedenle constrain() fonksiyonu kullanılarak paletin x konumu ekran genişliği ve palet genişliği arasında sınırlanır. Oyun Başlamadıysa Eğer oyunnbayragii bayrağı false ise, yani oyun başlamadıysa: Ana menü gösterilir (anaamenuugosterr()). Kullanıcının menü seçeneklerini belirlemesine olanak tanıyan fonksiyon çağırılır (menuuisaretlee()).

Topun Hareketini Güncelleme topunnxkonumuu ve topunnykonumuu, belirlenen yönde hareket miktarı kadar güncellenir. Bu, topun x ve y koordinatlarını günceller ve topu ekranda hareket ettirir. Topun Çarpışmalarını Kontrol Etme Sağ veya Sol Kenar Çarpışması: Eğer top ekranın sol veya sağ kenarına çarparsa, x yönündeki hareketi tersine çevrilir, böylece top ekranda geri döner. Tavan Çarpışması: Eğer top ekranın üst kenarına çarparsa, y yönündeki hareketi tersine çevrilir ve top aşağı yöne doğru devam eder. Taban Çarpışması: Eğer top ekranın alt kenarına çarparsa: Eğer top paletin üzerine düşerse, yönü tersine çevrilir ve top yukarı yöne doğru devam eder. Bu durum, topun paletten sekip yukarı hareket ettiği durumu simgeler. Eğer top paletin üzerine düşmezse, top ekranın ortasına geri konumlandırılır. Ayrıca can sayısı azaltılır ve LED durumu güncellenir. Bu durum, topun paleti kaçırıp düşmesi ve bir can kaybedilmesi durumunu simgeler. Tuğla Çarpışmalarını Kontrol Etme Tuğlaların her birini kontrol eder ve eğer top bir tuğlaya çarparsa, gerekli işlemleri gerçekleştirir: Topun yönünü tersine çevirir, böylece yukarı yöne doğru devam eder. Çarpılan tuğlayı kırılmış olarak işaretler. Skoru artırır. Eğer ekstra can paketi aktif değilse ve yüzde 10 olasılık sağlamıyorsa, ekstra can paketi aktif hale getirilir.

Ekstra Can Paketini Çizme Eğer ekstraacann durumuu aktifse, yani ekstra can paketi oluşturulduysa: Can paketini aşağı doğru hareket ettirir. Ekstra can paketini çizer. Eğer ekstra can paketi paletin üzerine düşerse, can sayısını artırır ve LED'leri günceller. Eğer ekstra can paketi ekranın altından geçerse, paketi kaldırır. Seviye Geçiş Kontrolü Eğer tüm tuğlalar kırıldıysa ve oyuncunun canı hala varsa, oyuncuyu bir sonraki seviyeye geçirir. Seviye numarasını artırır. Yeni bir seviye için tuğla düzenini oluşturur. Topu ve paleti ortaya yerleştirir. Her seviyede topun hızını Skoru Ekrana Yazdırma Skoru rengini belirler. Skoru ekrana yazdırır. Her saniyede bir, yedi segment ekranlara skoru yazdırır. Oyunun bitip bitmediğini kontrol eder. Eğer can sayısı 0 veya daha azsa, oyunu sonlandırır.

ELEKTRONİK DEVRENİN TASARIMI:

1. **LED'ler ('led1', 'led2', 'led3')**: - 'led1': Arduino Mega'nın 11. pinine (digital pin 11) bağlı kırmızı bir LED. - 'led2': Arduino Mega'nın 12. pinine (digital pin 12) bağlı kırmızı bir LED. - 'led3': Arduino Mega'nın 13. pinine (digital

pin 13) bağlı kırmızı bir LED.

2. **7 segment ekranlar ('sevseg1', 'sevseg2')**: - Her biri Arduino Mega'nın farklı pinlerine bağlı olan yedi segmentli ekranlar. - sevseg1: A segmenti: Uno'nun 27. pinine bağlı. B segmenti: Uno'nun 29. pinine bağlı. C segmenti: Uno'nun 31. pinine bağlı. D segmenti: Uno'nun 33. pinine bağlı. E segmenti: Uno'nun 35. pinine bağlı. F segmenti: Uno'nun 37. pinine bağlı. G segmenti: Uno'nun 39. pinine bağlı. Ortak pin: Uno'nun toprak pinine (GND) bağlı. -sevseg2: A segmenti: Uno'nun 41. pinine bağlı. B segmenti: Uno'nun 43. pinine bağlı. C segmenti: Uno'nun 45. pinine bağlı. D segmenti: Uno'nun 47. pinine bağlı. E segmenti: Uno'nun 49. pinine bağlı. F segmenti: Uno'nun 51. pinine bağlı. G segmenti: Uno'nun 53. pinine bağlı. Ortak pin: Uno'nun 5V pinine bağlı. - Bağlantılar, 'sevseg1' ve 'sevseg2' bileşenlerinin segmentlerini Arduino'nun dijital pinlerine ve toprak pinlerine (GND) bağlar.

3. **Basınç düğmeleri ('btn1', 'btn2', 'btn3')**: - 'btn1': Arduino Mega'nın 18. pinine (digital pin 18) bağlı bir basınç düğmesi (yukarı ok tuşu). - 'btn2': Arduino Mega'nın 17. pinine (digital pin 17) bağlı bir basınç düğmesi (aşağı ok tuşu). - 'btn3': Arduino Mega'nın 16. pinine (digital pin 16) bağlı bir basınç düğmesi (enter tuşu). - Tüm düğmeler, ayrıca toprak pinlerine (GND) bağlanır.

4. **Potansiyometre ('pot1')**: - Potansiyometre, Arduino Mega'nın A0 (analog pin 0) pinine bağlıdır. - Ayrıca, potansiyometre güç pimleri Arduino Mega'nın 5V ve GND pinlerine bağlanır.

5. **Fotoresistör ('ldr1')**: - Fotoresistör sensörü, Arduino Mega'nın A1 (analog pin 1) pinine bağlıdır. - Sensörün güç ve toprak pimleri Arduino Mega'nın 5V ve GND pinlerine bağlanır.

III. SONUÇLAR

Oyunun tasarımı, kullanıcı dostu bir arayüz ve eğlenceli bir oyun deneyimi sunmak amacıyla gerçekleştirilmiştir. Oyunda kullanıcının kontrol edebileceği bir palet bulunmakta olup, bu palet potansiyometre yardımıyla sağa ve sola hareket ettirilebilmektedir. Kullanıcı, paletle yönlendirdiği bir top aracılığıyla tuğlaları kırarak skor kazanmaktadır. Her tuğla kırıldığında skor artmakta ve bu skor 7 segment ekranlarında gösterilmektedir. Ayrıca oyuncunun can sayısı, topun paletin altına düşmesi durumunda azalmakta ve LED'ler aracılığıyla görsel olarak kullanıcıya bildirilmektedir.

Oyunun arka planı ve tuğlaların rengi, ışık sensörüne bağlı olarak dinamik olarak değişmektedir, bu da kullanıcıya farklı görsel deneyimler sunmaktadır. Her bölümün sonunda tuğla düzeni değişmekte ve topun hızı artmaktadır, bu da oyunun zorluk seviyesini arttırmaktadır.

Sonuç olarak, tasarlanan oyun kullanıcı dostu bir arayüze, eğlenceli bir oyun dinamiğine ve görsel olarak çekici bir deneyime sahiptir.

REFERENCES

- [1] <https://docs.wokwi.com/parts/wokwi-pushbutton>
- [2] <https://docs.wokwi.com/parts/wokwi-led>
- [3] <https://docs.wokwi.com/parts/wokwi-7segment>

- [4] <https://www.youtube.com/watch?v=7x1P80X1V3E>
[5] <https://github.com/DeanIsMe/SevSeg>

IV. UML DİYAGRAMI VE AKIŞ ŞEMALARI

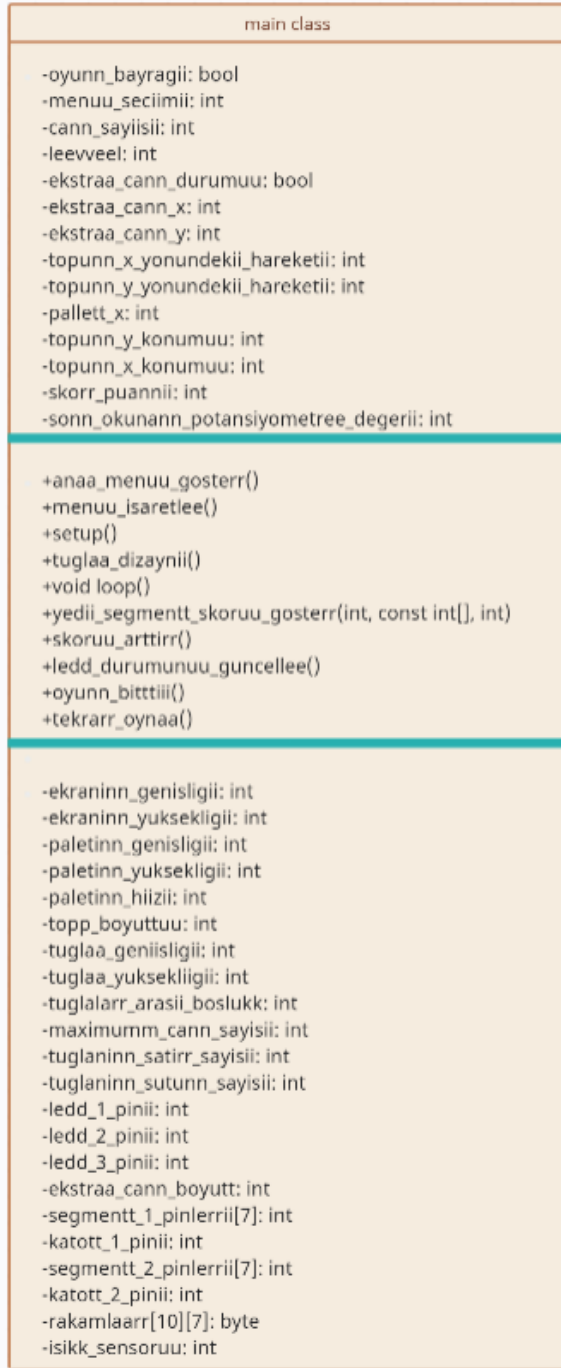


Fig. 1: UML diyagramı

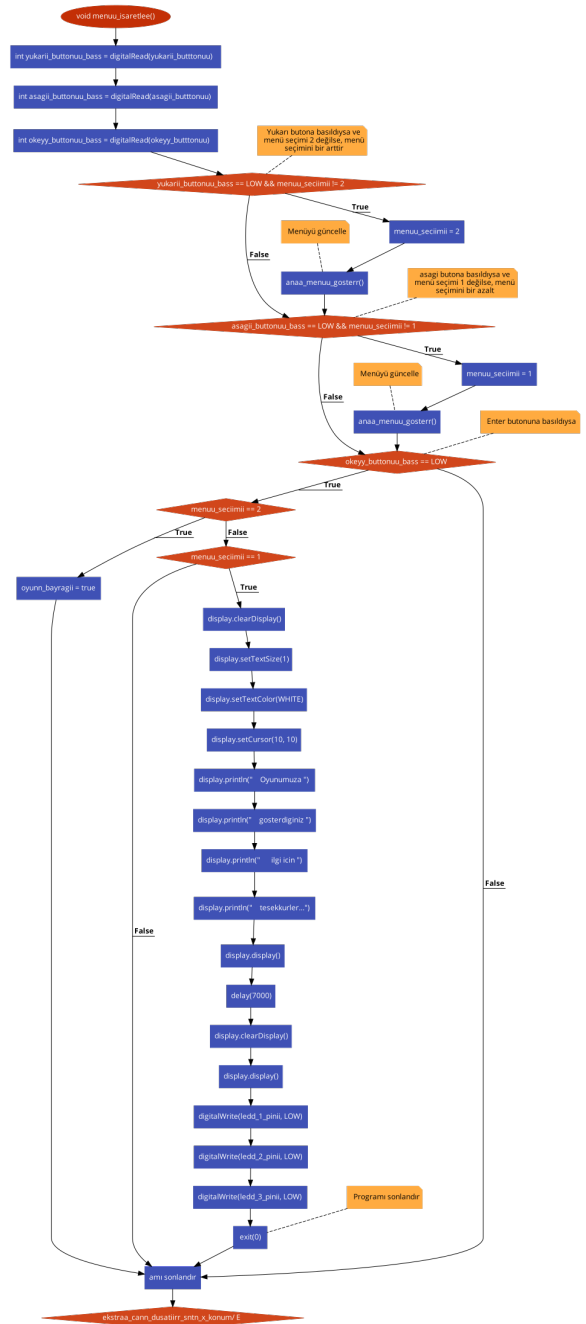
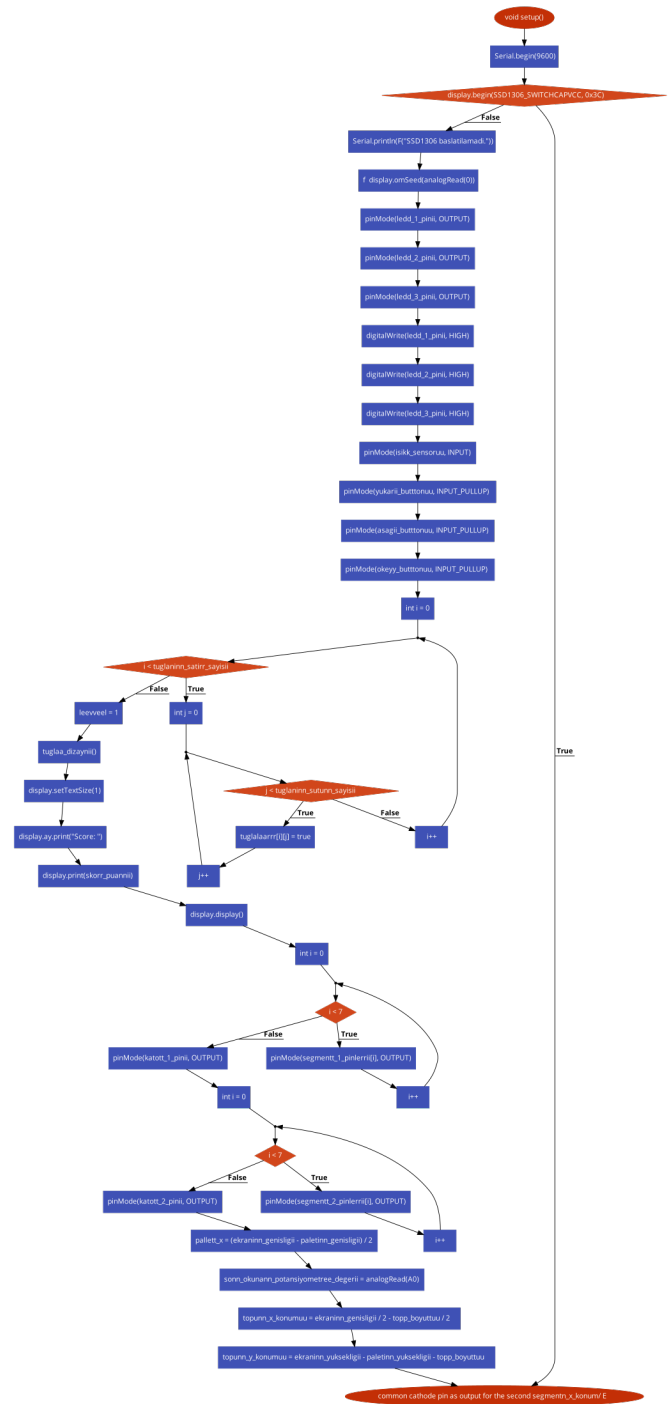
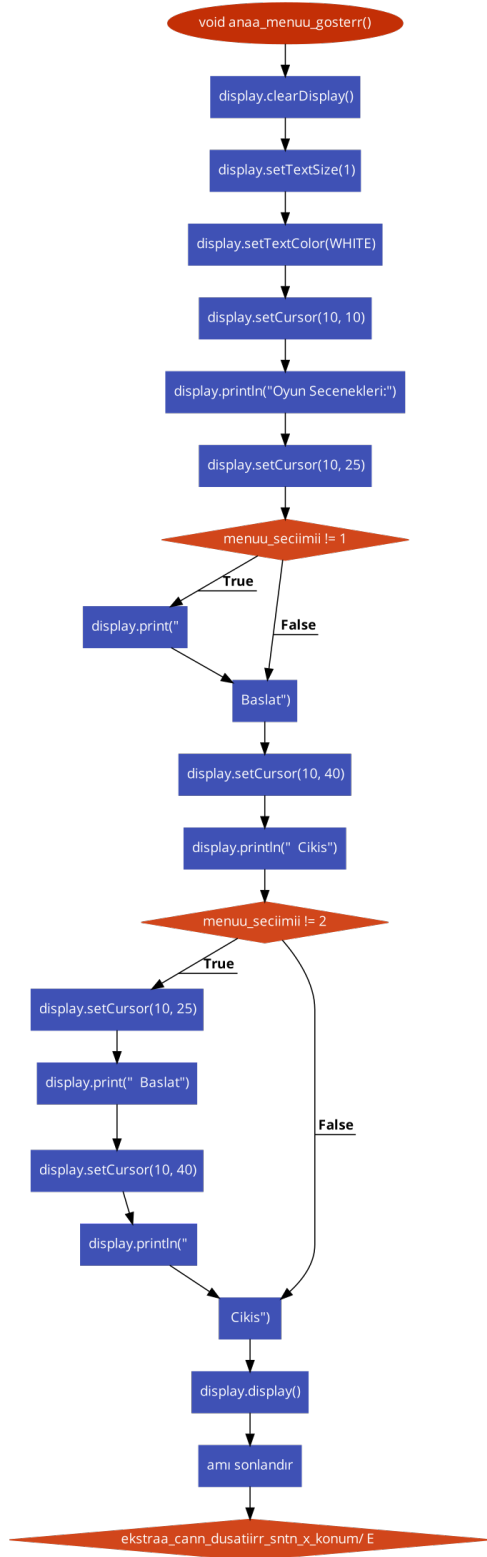


Fig. 2: menu isaretle fonksiyonu



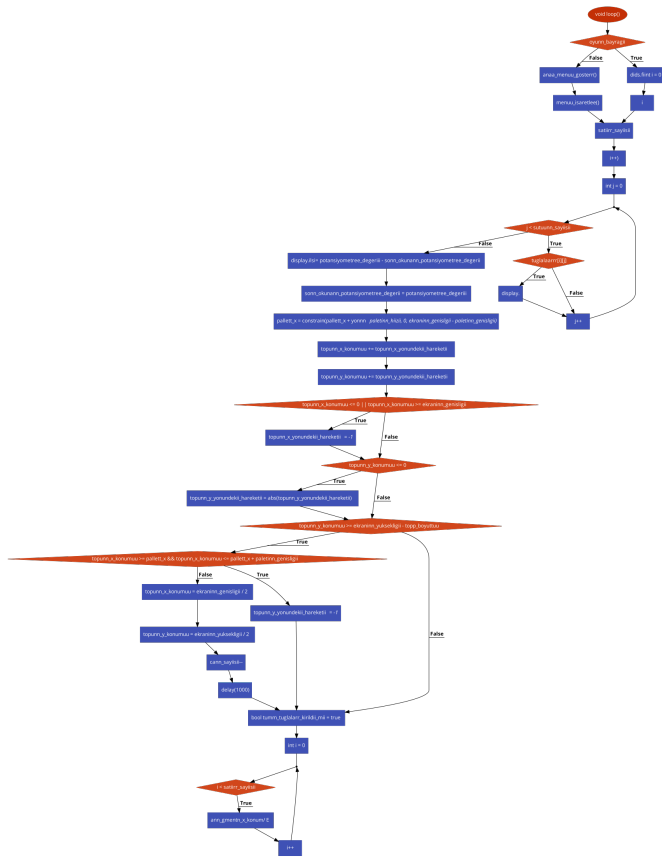


Fig. 5: loop fonksiyonu-1

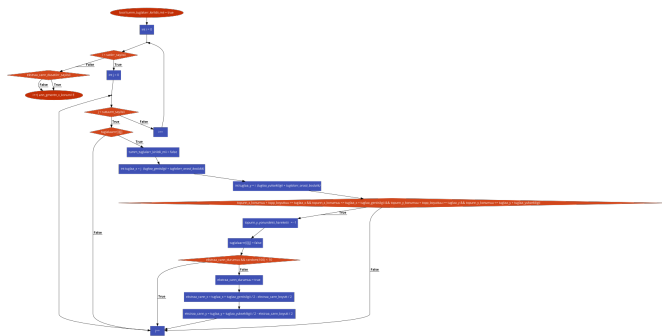


Fig. 6: loop fonksiyonu-2

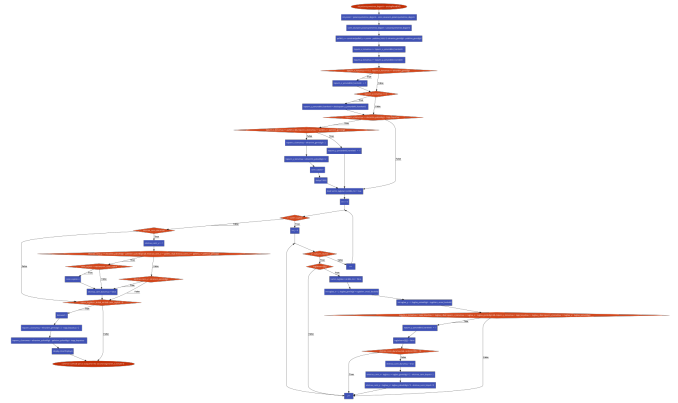


Fig. 7: loop fonksiyonu-3

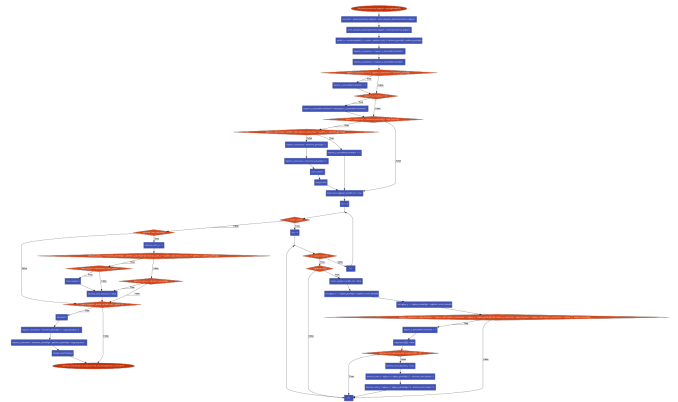


Fig. 8: loop fonksiyonu-4

