

Kocaeli Üniversitesi
Bilgisayar Mühendisliği Bölümü
Programlamaaboratuvarı II
Proje 2
Arduino Uzay Gemisi Projesi

Ahmad Alhomsi
ID:210201140
tarikuka.aa@gmail.com

ÖZET

Bu rapor Programlama Laboratuvarı II dersi 2. Projesinin çözümünü açıklamak üzere hazırlanmıştır. Bu proje C diliyle yapılmıştır. Öncelikle Arduino' u kullanarak oyun ortamı yapılmıştır. Belirli kurallara göre hareket eden bir uzay gemisi önündeki engelleri aşarak ulaşmasını sağlayan bir oyun tasarlanmıştır. Uzay gemisi yalnızca sağa, sola hareket ederek objelerden kaçabilmekte ve yeterli mühimmata sahip olabilirse objeleri yok edebilmektedir. Oyun ortamı; genişliği 8, uzunluğu 16 olan bir matristen oluşmaktadır. Bu projenin amacı, çeşitli sensörler kullanarak mikrodenetleyici tabanlı bir oyun makinesi geliştirmektir. Oyun, uzay ortamında çeşitli meteor ve uzay çöplerinden kaçmaya çalışan bir uzay gemisini ele almaktadır. Proje Arduino IDE ve Proteus programları kullanılarak geliştirilmektedir. SSD 1306 kütüphanesi kullanarak geliştirilmiştir

GİRİŞ

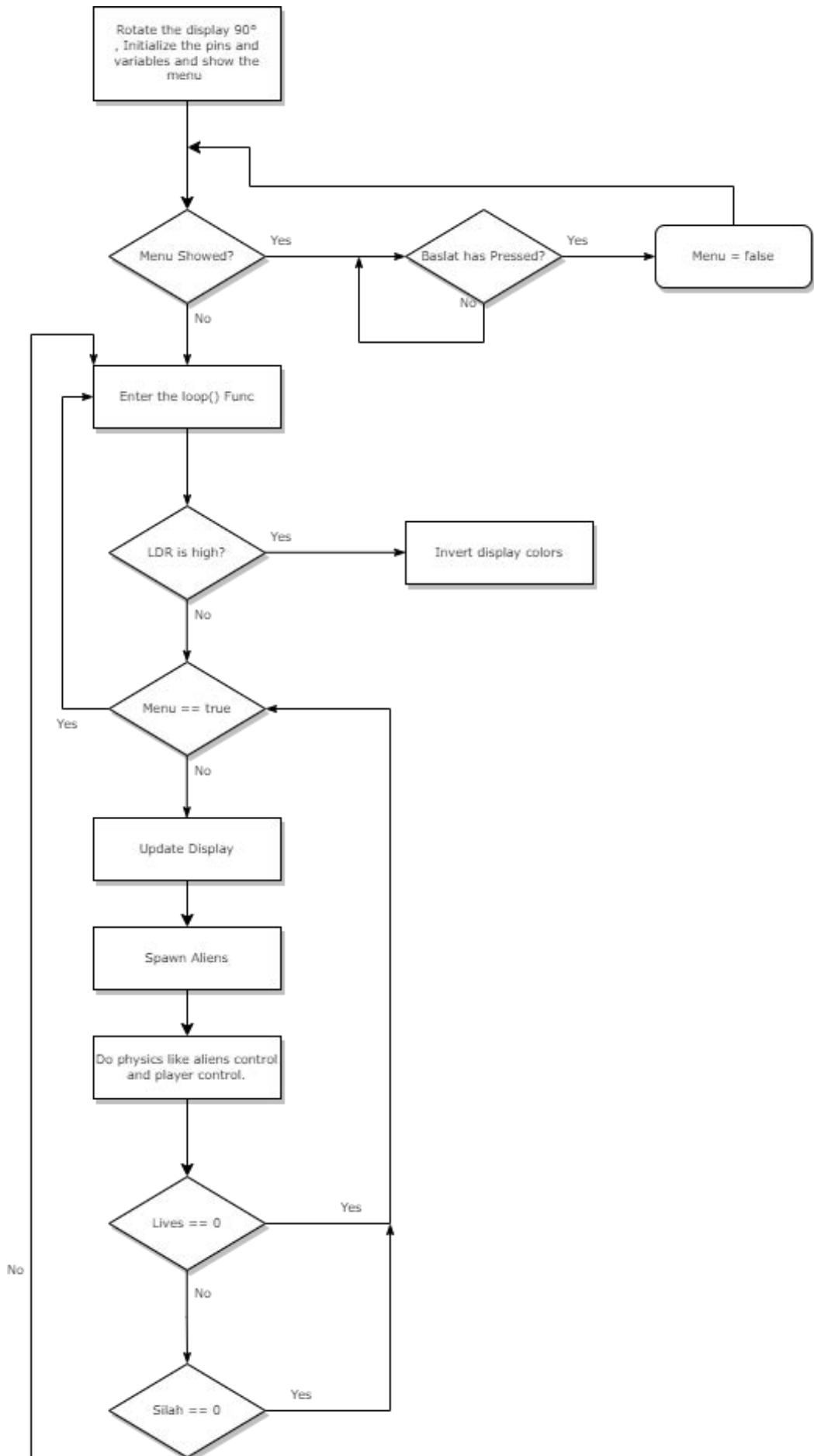
. Oled ekran kullanılarak arayüz gösterilecek ve matrislerden oluşan bu uzay ortamı kullanıcılara sunulmaktadır. Oyun; matriste kullanıcının kontrol ettiği karakter her zaman 1.satırda kalacak şekilde, matrisin en son bölgesinde rastgele oluşturulan engel objeleri ise her saniye 1 kare aşağıya inerek şekilde tasarlanmıştır. Böylece platform aşağıya doğru kayacak şekilde tasarlanmıştır. Kullanılacak araç dışında, rastgele şekilde oluşturulacak karakterler; engel karakteri (meteor veya uzay çöpü), ödül puanı (bu alındığında kullanıcı ekstra 1 hak kazanır, yedek can), silah (bununla kullanıcının yönettiği karakter önüne doğru bir obje fırlatır, ilk çarptığı engel karakterinin canını 1 azaltır), dokunulmazlık (alındığı zaman 3 saniye boyunca engele çarpsa bile oyuncunun hakkı gitmez) oluşturulmaktadır. Her oluşturulacak satır (8 kareye) rastgele objeler, 1 adet boşluk bırakılacak şekilde yerleştirilmektedir.

Uzay aracı bu boşluktan hareketine devam edebilmektedir. Rastgele oluşturulan nesnelerden her 14 engelde bir kere, 1 can ve 1 silah hakkı da eklenmektedir. Bu nesnelerin dizilimleri rastgele oluşturulmaktadır. Kullanıcının engelleri yok edebilmesi için; meteor ise 2 kere atış yapması, uzay çöpü ise 1 kere atış yapmaktadır. En üst bölgesinde oluşturmaya başlamaktadır ve her saniye aşağıya doğru bir kare ilerlemektedir. Her saniyede yeni rastgele nesneler oluşturulmaktadır. Uzay aracı ise yalnızca en alt bölgesinde bulunacak ve sağa, sola hareket edebilecektir.

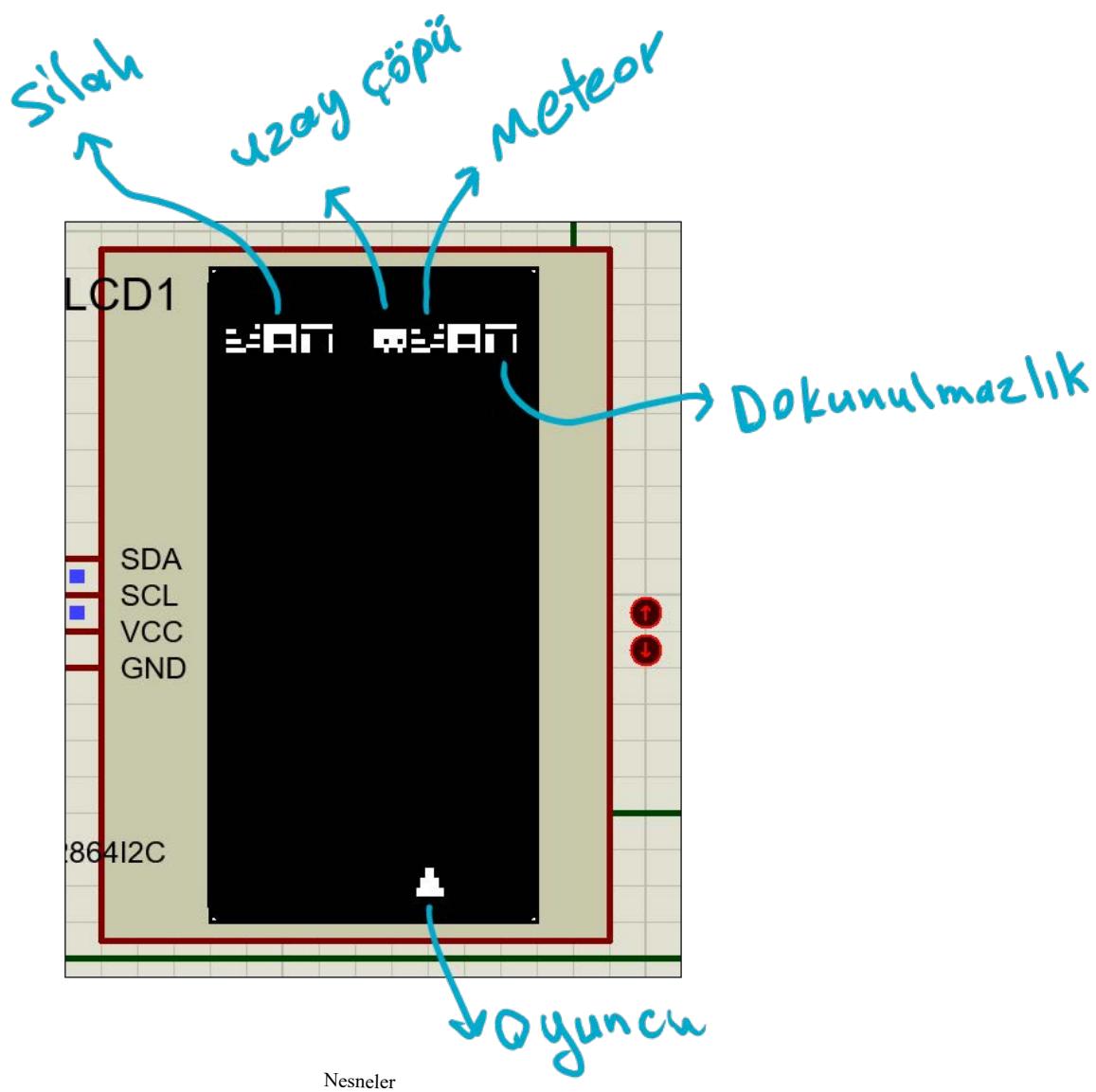
YÖNTEM

1. Oyun ekranı açıldığında menüde başlat ve zorluk seviyesi olmak üzere 2 seçenek olmaktadır. Kullanıcı oyuna 3 can hakkı ile başlayacaktır. 1. zorluk seviyesinde platform her zaman saniyede 1 kare aşağıya doğru hareket edecektir. 2. zorluk seviyesinde platform her 10 saniyede bir %20 oranında hızlanacaktır. Menüde seçim yapabilmek için aşağı, yukarı hareketi sağlayacak ve seçim işlemi gerçekleştirerek 3 buton kullanılması beklenmektedir. 2. Başlat düğmesine bastıktan sonra oyun başlayacaktır. Uzay aracı matrisin 1.satırında olacak şekilde sağ ve sola hareket edebilecektir. Sağa ve sola hareket için potansiyometre kullanılarak yeni bir oyun kolu tasarılmıştır. 3. Oyun başladığında kullanıcıda 3 silah hakkı olacaktır. Bunlar led ışıkla gösterilmektedir. Kullanıldığından sayı 1 azalacak ve buna göre ledlerden biri sönecektir. Hak tekrar

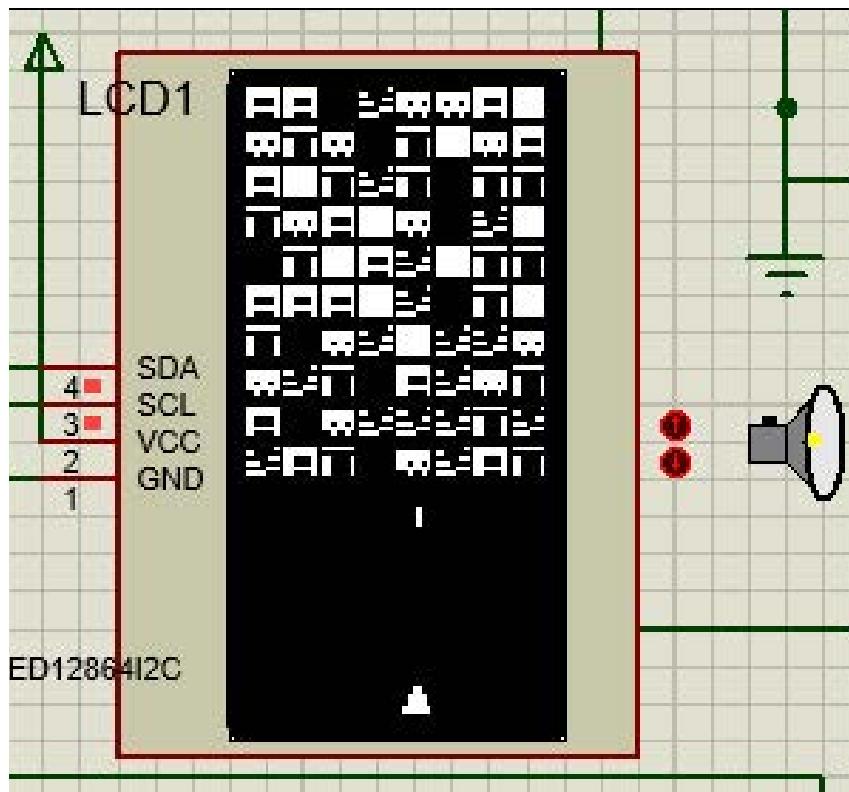
kazanıldığından buna göre led yanacaktır. 3 adet led kullanılması beklenmektedir. Atış yapabilmek için ekstra buton kullanılacaktır. 4. Kullanıcının hakları led ışıklarla aynı şekilde gösterilecektir. Uzay aracı engele çarpıp can kaybettiğinde otomatik olarak 3 saniye engele çarpsa bile 2.kere can kaybetmeyecektir. Can hakkı kazandığında led ışıkta artma olacaktır. Bu ledler silah hakkındaki ledlerden hariç tasarlanmalıdır. Ayrıca uzay aracı engele çarptığında buzzer kullanılarak uyarı verilecektir. 5. Oyuncunun tüm hakları bittiğinde sistem tekrar ana menüye dönecektir. 6. Oyuncu her satır atladığında skor puanı 1 artacaktır ve skor puanı 7 segment display ile gösterilmelidir. Skor tablosu için 3 adet 7 segment display bulunmalıdır. 7. Arduino LDR ışık sensörü kullanılarak oyun ortamının renk değiştirmesi beklenmektedir. LDR ışık sensöründen alınan verilere göre oyundaki siyah beyaz renk dağılımı değiştirilecektir. 8. Rastgele oluşturulan nesneler için farklı şekiller belirlenecektir. Buna imkan sağlayacak bir OLED ekran kullanılması beklenmektedir. SSD1306 ekranı tercih edilmiştir. Adafruit_GFX, Wire, Adafruit_SSD1306 ve Splash kütüphaneleri kullanılmıştır. ayrıca ekrana grafiği çizdirmek için drawBitmap fonksiyonunu kullanılmıştır. oyuncu ve engellerin şekilleri "const unsigned char PROGMEM" şekilde bir dizi ile tasarılmıştır. engeller, oyuncu ve gameObject (engeller ve oyuncu içermektedir) struct tipinde tanımlanmıştır. global diziler de tanımlanmıştır InitDown, previousMillis, lives ve silah gibi. struct ise: Missile, Alien (iki boyutlu struct tipi) ve en son Player struct gibi global değişkenler tanımlanmıştır.



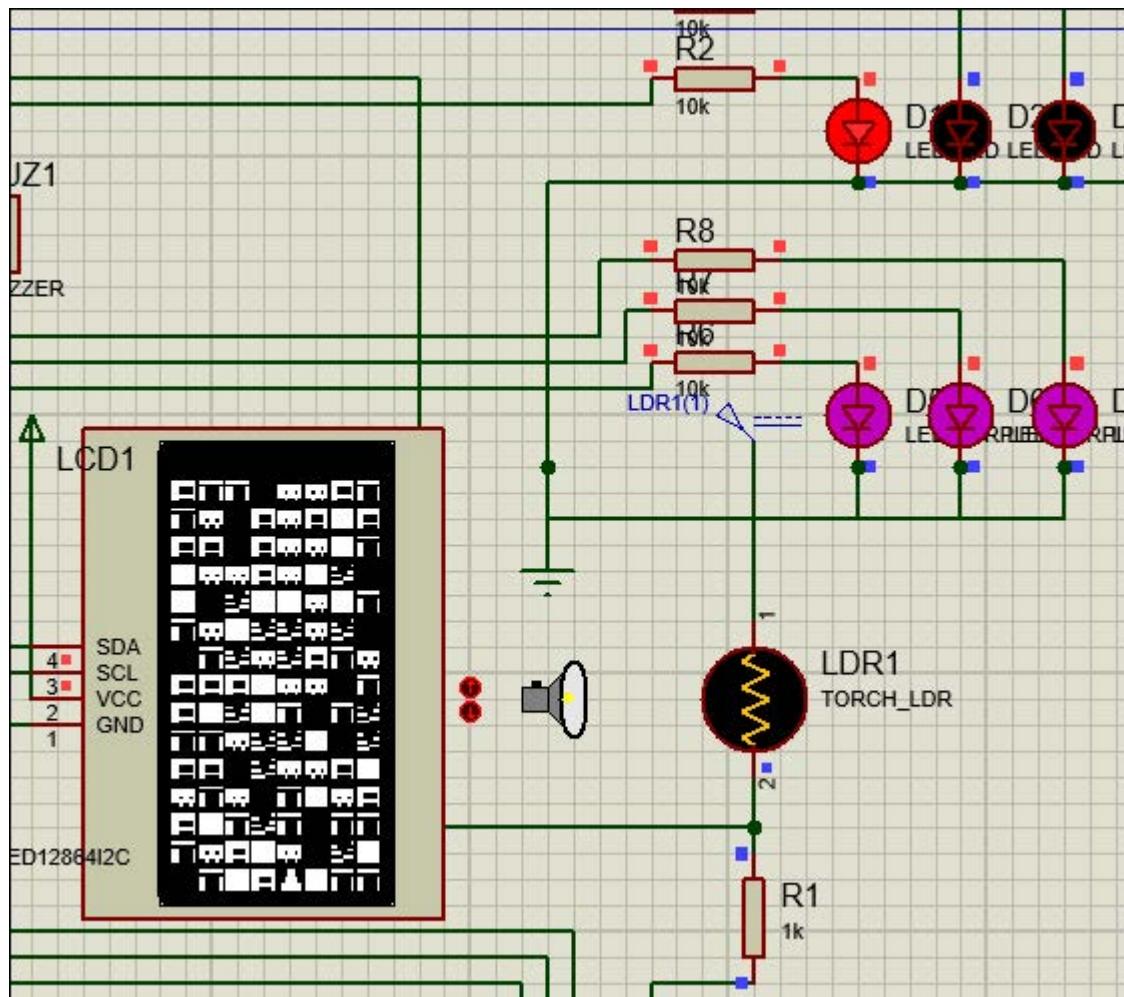
Flow Chart



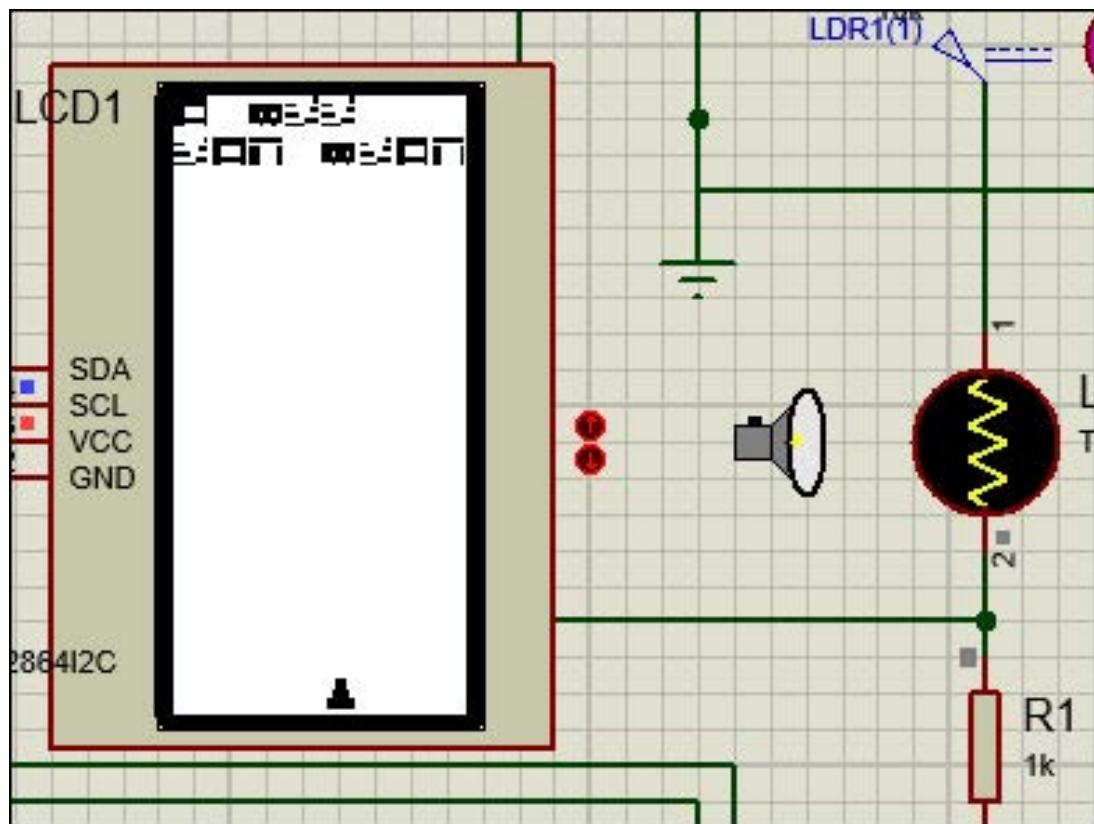
Nesneler



Oyun 1



Oyun 2



Oyun 3

KAYNAKÇA

REFERENCES

- [1] <https://www.xtronical.com/projects/space-invaders-parts-1-7/part-1-oled-plotting-graphics-and-basic-movement/>
- [2] <https://forum.arduino.cc/t/128x32-i2c-oled-display-can-it-be-dimmed/668244>
- [3] <https://www.programiz.com/dsa/priority-queue>
- [4] <https://docs.oracle.com/javase/7/docs/api/java/util/PriorityQueue.html>
 - [5] <https://github.com/mgeeee35/easyUML>
 - [6] <https://www.geeksforgeeks.org/binary-heap/>
- [5] <https://www.sanfoundry.com/c-program-implement-heap/>
- [6] <https://www.youtube.com/watch?v=HqPJF2L5h9Ut=1335s>
 - [9] <https://www.youtube.com/watch?v=wptevk0bshY>
 - [7] <https://www.youtube.com/watch?v=U1AJZQxYTU>
- [8] <https://www.codesansar.com/c-programming/recursive-function.htm>
- [12] https://www.tutorialspoint.com/cprogramming/c_recursion.htm
- [13] <https://www.programiz.com/c-programming/c-recursion>
- [14] <https://www.youtube.com/watch?v=kepBmgvWNDw>
- [15] Book Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology by Dan Gusfield
- [16] <https://www.youtube.com/watch?v=ngCos392W4w>
- [17] <https://www.youtube.com/watch?v=ggk7HbcnLG8>
- [18] <https://www.youtube.com/watch?v=QWDCKPEWSWot=425s>
- [19] <https://www.youtube.com/watch?v=cv7CY8UmFL0>
- [20] <https://www.geeksforgeeks.org/readwrite-structure-file-c/>
 - [21] <https://cboard.cprogramming.com/c-programming/77351-writing-array-struct-file.html>
- [22] <https://courses.cs.washington.edu/courses/cse373/17wi/lectures/priority-queues.pdf>
- [23] <http://www.cs.cornell.edu/courses/cs2110/2015fa/L17-PriorityQueuesAndHeaps/cs2110PqueuesHeaps.pdf>
- [24] <https://www.geeksforgeeks.org/priority-queue-using-binary-heap/>
- [25] <https://runestone.academy/runestone/books/published/pythonds/Trees/BinaryHeapImplementation.html>