

# Programlama Laboratuvarı II

## Proje 2

Furkan Demirsoy 210201052, Önder Alp Cıvcır 200201083, Kocaeli Üniversitesi  
Bilgisayar Mühendisliği

### I. ÖNSÖZ

Bu proje Kocaeli Üniversitesi Bilgisayar Mühendisliği Programlama Laboratuvarı II dersi 2. projesidir. Bu raporda bu projeyi nasıl yaptığımız, yaparken nelerden faydalandığımız ve karşılaştığımız hataları nasıl çözdüğümüz gibi birçok konuda bilgilendirme yapılıacaktır.

### II. GİRİŞ

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. Uzay gemisi yalnızca sağa, sola hareket ederek objelerden kaçabilmekte ve yeterli mühimmata sahip olabilirse objeleri yok edebilmektedir.

### III. ÖZET

#### A. Proje Açıklaması

- 1. Oyun ortamı; genişliği 8, uzunluğu 16 olan bir matristen oluşmaktadır. Oled ekran kullanılarak arayüz gösterilecek ve matrislerden oluşan bu uzay ortamı kullanıcılara sunulacaktır.
- 2. 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 inecek şekilde tasarlanmalıdır. Böylece platform aşağıya doğru kayacak şekilde tasarlanmış olacaktır.

- 3. Kullanılacak araç dışında, rastgele şekilde oluşturulacak karakterler; engel karakteri (meteor veya uzay çöpi), 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 çarparsa bile oyuncunun hakkı gitmez) oluşturulur.
- 4. Her oluşturulacak satıra (8 kareye) rastgele objeler, 1 adet boşluk bırakılacak şekilde yerleştirilmelidir. Uzay aracı bu boşluktan hareketine devam edebilecektir. Rastgele oluşturulan nesnelerden her 14 engelde bir kere, 1 can ve 1 silah hakkı da eklenmelidir. Bu nesnelerin dizilimleri rastgele oluşturulmalıdır. Kullanıcının engelleri yok edebilmesi için; meteor ise 2 kere atış yapması, uzay çöpi ise 1 kere atış yapması gerekmektedir.

#### B. Proje İsterleri

- 1. Oyun ekranı açıldığında menüde başlat ve zorluk seviyesi olmak üzere 2 seçenek olacaktır. 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 sağlayacak ve seçim işlemi gerçekleştirecek 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 tasarlanması beklenmektedir.
- 3. Oyun başladığında kullanıcıda 3 silah hakkı olacaktır. Bunlar led ışıkla gösterilecektir. Kullanıldığında sayı 1 azalacak ve buna göre ledlerden biri sönecektir. Hak tekrar kazanıldığında buna göre led yanacaktır. 3 adet led kullanılması beklenmektedir. Atış yapabilmek için ekstra buton kullanılacaktır.
- 4. Kullanıcın 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 atlادığında 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 değıştirmesi beklenmektedir. LDR ışık sensöründen alınan verilere göre oyundaki siyah beyaz renk dağılımı değıştirilecektir.
- 7. Rastgele oluşturulan nesneler için farklı şekiller belirlenecektir. Buna imkan sağlayacak bir OLED ekran kullanılması beklenmektedir.

### C. Proje Kısıtları

- Proje Arduino IDE ve Proteus programları kullanılarak geliştirilecektir.
- Belirtilen bütün sensörlerin kullanılması zorunludur.
- Projeye başka sensörler eklenmemelidir.

## IV. KULLANILAN KÜTÜPHANELER

- AdafruitGFX.h
- AdafruitSSD1306.h
- Wire.h
- U8glib.h
- stdio.h
- stdlib.h

## V. YÖNTEM

### A. AMAÇ

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. Uzay gemisi yalnızca sağa, sola hareket ederek objelerden kaçabilmekte ve yeterli mühimmata sahip olabilirse objeleri yok edebilmektedir.

### B. DEVRE

Proteus programı ve Proteus için Arduino kullanılablmesini sağlayan kütüphaneler ile proje için gerekli olan devre kurulumu dijital ortamda projenin isterlerini karşılayacak şekilde oluşturulmuştur.

- İnternette yaptığımız araştırmalar sonunda arduino mega kartının bellek alanı daha fazla olduğu için Arduino kart kullanıldı. Simulino kartının kullanılmasının sebebi oled ekrana veriyi daha kolay bir şekilde aktarmak için tercih edildi.

- Proje isterlerinden ötürü OLED ekranın 28x64 tercih edildi.
- Ses çıkartmak için buzzer kullanıldı. Can haklarını göstermek için ledler, ortam ışığına göre programın isterler doğrultusunda tepki verebilmesi için LDR kullanıldı.
- Oyunun sonucunu göstermek 7 segment display kullandı.

### C. GÖRSELLİK

Movement ve object isimli iki adet fonksiyon kuruldu. Bu fonksiyonda da uzay çöplerinin düzenli bir şekilde hareketi sağlandı. Hareket sağlanırken de en başta birinci dizinin tamamı açıldıktan sonra her seferinde her generateObject() fonksiyonu tekrar çağrılarak dizi tekrar değiştirildi. Ancak önceki diziden kalma şeylerin ekrandan silinmemesi için bu yapılmadan bir adım önce, önceki dizi başka bir diziye kopyalandı ve o dizi içerisindeki elemanlar bir counter sayacıyla teker teker basıldı.

- Meteor, uzay çöpü gibi objelerin oluşturulması için bir fonksiyon kuruldu. O fonksiyon içerisinde iki tane çok boyutlu dizi oluşturularak generateObject() isimli bir fonksiyon oluşturdu. Her seferinde bu fonksiyon içerisindeki objeler random fonksiyonuyla ve aralarında bir boşluk olacak şekilde dizinin içerisine atıldı. Daha sonra bu dizi loop içerisinde açıldı. Açıldıktan sonra da bu objelerin her biri meteor ise meteor, uzay çöpü ise uzay çöpü olacak şekilde OLED ekrana yerleştirdi.
- Movement ve Object isimli iki adet fonksiyon kuruldu. Bu fonksiyonlarda uzay çöplerinin düzenli bir şekilde hareketi sağlandı.
- Hareket sağlanırken de en başta birinci dizinin tamamı açıldıktan sonra her seferinde her GenerateObject() fonksiyonu tekrar çağrılarak dizi tekrar değiştirildi. Ancak önceki diziden kalma veriye göre OLED ekrana aktardığımız görselliğin ekrandan silinmemesi için bu yapılmadan bir adım önce, önceki dizi başka bir diziye kopyalandı ve o dizi içerisindeki elemanlar bir counter sayacıyla teker teker basıldı.

### D. ARAYÜZ

Loop döngüsünün içerisinde bir kontrol sağlandı. Bunun için Ug8 kütüphanesindeki display.drawReact() fonksiyonu kullanıldı. Ayrıca yazı yazdırmak için de display.setCursor() fonksiyonu ile yazıların konumu belirlendi. Display.setColor() fonksiyonu ile renkler belirlendi. Menüün oluşturulması için şu şekilde bir algoritma oluşturuldu; Arkadaki ekranın rengine göre program ilk oluşturulduğunda a.Control değeri eğer 0 ise a.Control olan bölgeye girilmesi ve içerisindeki fonksiyonlara erişilerek ekrana basılması sağlandı. digital.read() fonksiyonuyla her buton bir pine bağlı, butonlar pin değerlerine göre okunarak bir olması durumunda butona basıldığı anlaşıyor ve butonState değişkeni bir olduğundan içi boş sadece kenarları görünen bir dikdörtgen oluşturularak yine display.drawReact() fonksiyonuyla aşağı durumuna göre x koordinatı 30 azaltıldı yukarı durumuna göre x koordinatı 30 artırıldı. Seçim butonuna basıldığında da o anki x koordinatının yeri kaydedildi bir struct veri yapısının içerisinde ve içerisindeki değerin durumuna göre hangi zorluk seviyesinin istendiği anlaşıldı.

## VI. DENEYSEL SONUÇLAR

- İnternette yaptığımız araştırmaların sonucunda OLED ekranda çalışmaya en uygun olarak Gfx Kütüphanesi ve Ug8 kütüphanesinin olduğuna karar verildi. İlk olarak hangi arduino Uno kartının kullanılması gerektiği araştırıldı Bunun için Arduino Uno Arduino mega kartlarının olduğu öğrenildi. Ancak arduino simulino içerisinde varsayılan olarak gelen arduino Uno kartı kullanıldığında analog okuma yaptı, ancak dijital okumalarda OLED ekranda hiçbir görüntü oluşmadığı fark edildi. Yaptığımız araştırmaların sonucunda Arduino simulino Uno kart kullanıldı. Bunun için github'dan özel bir simulino kütüphanesi eklenildi.
- Projeye başlandığında Arduino simulino Uno kullanıldı ancak proje ilerledikçe Arduino kartının bellek kapasitesinin az olduğu fark edildi. Bellek kapasitesi az olduğu için programda bellek taşması meydana geldi. Bu sebepten Arduino Uno'dan vazgeçildi ve Arduino Mega kullanıldı.

- Arduino Mega'ya geçilmesinin bir diğer sebebi de yedi segment display'lerin çok fazla pin ihtiyacı olması ve Arduino Mega'nın çok fazla dijital pininin olmasıdır.
- Programın gelen ortam ışığına göre çeşitli ayarlamalar yapabilmesi için LDR kullanıldı.
- LDR'nin üzerinden elektrik geçtiğinde digital.read() yapılarak bir olması durumunda ışığın var olduğu, sıfır olması durumunda ışığın yok olduğu fark edildi ve bu duruma göre ortamın rengi değiştirildi.

## VII. SONUÇ

Bu projede Proteus programını kullanmayı, Proteus programında Arduino ile bir geliştirme ortamı kurabilmeyi ve Arduino kullanmayı öğrendik. Proteus ile bize verilen isterler doğrultusunda elektronik bir ürün ortaya çıkartabilmeyi öğrendik. Arduino'nun fiziksel yapısını, Arduino'nun çalışma mantığını kavrayarak projemizde kullandık.

Uygulama C dilinde Arduino IDE platformunda Proteus üzerinde simüle edilmiştir ve ekstra kütüphaneler kullanılmıştır.

Proje hazırlanırken en zorlayan kısım sınırlı bir bellek alanında çalışmak oldu. Veri yapılarını ve bunları daha bellek dostu bir şekilde tutmayı öğrenmiş olduk.

Projenin tamamlanması 3 hafta sürmüştür. Bu 3 haftalık sürecin ilk kısmı projeyi anlamak ve algoritma oluşturma ile geçmiş sonraki kısım ise deneme yanılma ile kodu yazma olmuştur.

## VIII. KAYNAKÇA

- 1) <https://www.youtube.com/watch?v=MB4x0KjgtEUt=261s>
- 2) <https://www.youtube.com/watch?v=AI2OitTDVLUt=521s>
- 3) [https://www.youtube.com/watch?v=EqNeh8wu\\_E](https://www.youtube.com/watch?v=EqNeh8wu_E)
- 4) <https://www.instructables.com/How-to-Simulate-Arduino-in-Proteus/>