

PROJE 2

Egemen Çakır

Kocaeli Bilgisayar Mühendisliği
Email: cakiregemen0@gmail.com

I. GİRİŞ

Bu projenin amacı, çeşitli sensörler kullanarak mikro-denetleyici 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. Oyun ortamı; genişliği 8, uzunluğu 16 olan bir matristen oluşmaktadır. Oled ekran kullanılarak arayüz gösterilecek ve matrislerden oluşan bu uzay ortamı kullanıcılara sunulacaktır. 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. Kullanılacak araç dışında, rastgele şekilde oluşturulacak karakterler; engel karakteri (meteor veya uzay çöpu), 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 çarpırsa bile oyuncunun hakkı gitmez) oluşturulur. 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 çöpu ise 1 kere atış yapması gerekmektedir.Ayrıca bizden istenen isterler:

- 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 yüzde 20 oranında hızlanacaktır. Menüde seçim yapabilmek için aşağı, yukarı hareketi sağlayacak ve seçim işlemi gerçekleştirecek 3 buton kullanılması beklenmektedir.
- 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.
- 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.

- Kullanıcın hakları led ışıklarla aynı şekilde gösterilecektir. Uzay aracı engele çarpıp can kaybettiğinde otomatik olarak 3 saniye engele çarpırsa 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.
- Oyuncunun tüm hakları bittiğinde sistem tekrar ana menüye dönecektir.
- 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.
- Arduino LDR ışık sensörü kullanılarak oyun ortamının renk değiştirmesi beklenmektedir. LDR ışık sensöründen alınan verilere göre oyundaki siyah beyaz renk dağılımı değiştirilecektir.

Bu proje arduino'yu öğrenme açısından ve bazı devre elemanlarını kullanma açısından oldukça önemlidir. Bu projeyi c++ dilinde yazdım ve arduino kartını kodlamak için arduino ide uygulamasını kullandım.

II. ADIMLAR

Adım 1

İlk adım olarak proteus da arduino kartları için bir kütüphane indirdim.Daha sonra arduino mega kartımı yerleştirdim.Daha sonra OLED ekranımı yerleştirip bağlantılarını yaptım.Ekranın çalışması ve üzerinde işlemler gerçekleştirebilmek için Adafruit'in kütüphanelerini kullandım ve son olarak sanal makineyi kurdum ki bazı test işlem sonuçlarını orada da görebileyim diye.

Adım 2

Bu adımda menü tasarımı yapmaya başladım.Öncelikle seri haberleşme ve OLED ekran için başlatma kodlarını setup() kısmına yazdıktan sonra menüde hareket etmek ve seçmek için kullanacağım 3 adet butonu karta bağladım.Daha sonra bunlardan digital veri alacağım için 11,12,13. digital pinlere pinMode() fonksiyonu ile INPUT modunu belirttim.Her loop() fonksiyonu başında bu üç butona basılıp basılmadığını kontrol etmek için digitalRead() ile verileri okuyorum.Daha sonra her seferinde bir butonu aktif edecek şekilde 3 adet if koşullarını yazdım.Daha sonra benim giriş kısmını oluşturan giris() yazdım.Bu fonksiyon:

- `giris` metodu:ekranda YUKLENİYOR yazısı basar ve `delay()` fonksiyonnu yardımı ile biraz bekledikten sonra ekrana bir resim basar ve `page1()` fonksiyona gecirecek olan global değişkenim olan `page` değişkeninin değerini 1 yapar.Bu `page` ler arası geçişi `loop` fonksiyonunda `if` koşulları ile gerçekleştirdim o anda `page` kaç değerini gösteriyor ise kullanıcı o sayfada oluyor.

`giris()` metodunu ve butonları bağladıktan sonra diğer adım gectim.

Adım 3

Bu adımda menumu tamamlamak için diğer metotları yazdım.`page1()`,`page2()`,`page3()` ve `page4()` metotlarını ve `yon` ve seçim kontrolleri için `select()` ve `option()` metotlarını yazdım.Bu metotlar:

- `page1` metodu:`Giris` ekranından sonra geçilen kısımdır.Bu fonksiyonda `select` ve `option` fonksiyonlarını kullanarak kullanıcıya ayar kısmına zorluk kısmına ve oyun oynama kısmına geçmesi için bu kısımları 16 px aralıklarla alt alta yazdım ve kullanıcının hangi satırda olduğunu göstermek için bir işaret bastırdım.Bu işaret ilk önce ilk kısım adını gösterecek şekilde konumu verildi fakat `option` metodunda yukarı ya da aşağı buton basımına göre bu imlecin y kordinatı 16 px oynuyor aynı şekilde `select` metodu ile imlecin bulunduğu konuma göre kısımlar arası geçiş için `page` değişkeninin değeri değiştiriliyor.
- `select` metodu:`Secme` tusunun aktif olma durumuna göre ve bulunduğu sayfadaki imlecin konumuna göre `page` değerini , zorluk değeri gibi değişkenlerin değerini değiştiriyor.
- `option` metodu:Yukarı ya da aşağı metodunun aktif olmasına göre imlecin y konumunu 16 px değişime sebep oluyor bu da mevcut sayfada imlecin tekrar bastırılmasında yukarı aşağı bir hareket gözlemlenmesine sebep oluyor.
- `page2` metodu:`page` değişkeninin değeri 2 olması durumunda çalışır.Bu metod 16 px aralıklarda alt alta zorluk seviyeleri var aynı zamanda imleci ekrana bastırarak fonksiyon ve `select` ve `option` metotlarının da içeriyor.
- `page3` metodu:Bu kısım daha sonra yazdım fakat oyun bu kısımda oynanıyor ve kullanıcı zorluk seçmeden bu kısma geçemiyor.
- `page4` metodu::`page` değişkeninin değeri 4 olması durumunda çalışır.Bu metod 16 px aralıklarda alt alta ayarlar var aynı zamanda imleci ekrana bastırarak fonksiyon ve `select` ve `option` metotlarının da içeriyor.Bu kısım uygulamanın şu anki haliyle herhangi bir katkısı yoktur yani ses ve ışıklandırma geliştirilirse bu kısımlar daha fazla katkı verebilir.

Bu fonksiyonlarımı yazdıktan sonra `loop` fonksiyonunda bu `page` metotlarını `if` blokları içinde yazdım ve artık menümü yaptım ve istediğim gibi dolasıp kısımlar arasında gezabiliyorum.Daha sonraki bölümde oyunu yavaştan yapmaya başladım.

Adım 4

Bu kısımda `gemi`,`silah` , `hedef` objeleri ve oyun için sınıflar yazdım fakat bu sınıfları `.h` ve `.cpp` uzantılı hallerini yazıp ana kodlarımın yazdığı yere `include` ettim.Bu sınıflar:

- `ship` class:Genel özellik olarak `byte` tipinde `cephane` ve `silah` değişkenlerini `integer` olarakda `x` , `y` özelliklerini ve `silah` tipinde bir dizi tutuyor.Fonksiyon olarak iki adet `constructor` , `update`,`updateSilah` ve `insertSilah` adında fonksiyonlar içeriyor.Bu fonksiyonları daha sonra açıklayacağım.
- `silah` class:Genel özellik olarak `bool` tipinde `active` değişkenini `integer` olarakda `x` , `y` özelliklerini ve `silah` tipinde bir pointer tutuyor.Fonksiyon olarak iki adet `constructor` , `update` adında fonksiyon içeriyor.Bu fonksiyonları daha sonra açıklayacağım.
- `target` class:Genel özellik olarak `byte` tipinde `type` ve `hasar` değişkenlerini `integer` olarakda `x` , `y` özelliklerini ve `target` tipinde bir pointer tutuyor.Fonksiyon olarak iki adet `constructor` , `update` fonksiyon içeriyor.Bu fonksiyonları daha sonra açıklayacağım.
- `oyun` class:Genel özellik olarak `ship` tipinde,`target` tipinde pointerlar ve `target` adında 2 boyutlu bir matris tutuyor.Fonksiyon olarak bir tane `constructor` ,`updateTarget` ,`insertTarget` ,`delTarget` , `kontrol1` ve `kontrol2` adında fonksiyonlar içeriyor.Bu fonksiyonları daha sonra açıklayacağım.

Sınıflarımı genel olarak oluşturdum ve oyunu yazmaya başladım.

Adım 5

Öncelikle oyun oynanırken ya da menüdeyken çevredeki ışık miktarına göre ekran renklerini tersine çeviren bir fonksiyon yazdım.Bu fonksiyon:

- `LDR` metodu:Burada bir adet `LDR` kullandım ve karta `A0` analog pinine bağladım.Bu fonksiyonda `A0` sıfırda `analogRead()` metodu ile verileri okudum ve gelen değeri 500'ün altındaysa ekran rengini tersine çevirecek bir fonksiyon yazdım.

Daha sonra geminin can ve silah değerini gösterecek 6 adet ledi karta bağladım ve bunların yanmalarını kontrol etmek için `ledler1` ve `ledler2` adında fonksiyonlar yazdım bunlar sadece geminin can ve silah değerine göre ledleri yakıp söndürecekler.Bunlardan sonra `page3` metodunun içini doldurmaya başladım.Öncelikle bir `while` actım ve içine kullanıcının zorluk seçimlerine göre farklı oyun koşulları olacağı için bir `if` bloğu ile iki kısma ayırdım.Zorluk seviyesi 1 için öncelikle `millis()` fonksiyonunu kullanarak geminin 60 milisaniyede hedeflerin 1 sn de hareket etmesi için iki adet `if` bloğu yazdım.Daha sonra hedeflerin hareketi için gerekli koşulun içinde sırasıyla `delTarget`,`insertTarget`,`updateTarget` metotlarını kullandım.Bu metotlar:

- `delTarget` metodu:Bu metod oyun classında bulunuyor, 1 adet parametre alıyor ve parametre de mevcut satırı belirtiyor.Bu fonksiyonun amacı yeni bir satır oluşturulmadan o satırı temizleme işlemi yapmaktır.

- insertTarget metodu: Bu metod oyun classında bulunuyor, 5 adet parametre alıyor. Öncelikle bir koşul yazdım ve bu koşulda her 17 obje içinde 3 ödül için 0-17 arası rastgele konumlar atadım. Daha sonra mevcut satıra kaç tane obje yerleştirileceğini belirledim ve her obje sayısına göre farklı koşullar yazdım. Bu koşullar içinde öncelikle mevcut obje sayısına göre ödülün mü yerleşeceğini yoksa meteor mu yerleşeceğini karar verdim ve burada oyun classında yer alan matrisin içine target pointrini kullanarak ekledim. Ayrıca bu objelerin 8 birime yerleşimi tanımladığım mod koşullarına göre rastgele şekilde yerleştiriliyor.
- updateTarget metodu: Bu metod oyun classında bulunuyor. Matristeki her nesneyi targetin içindeki update fonksiyonunu kullanarak y kordinatlarını güncelliyor.

Daha sonra bu nesleri ekranda göstermek için ana kod ekranımda drawTarget adında bir fonksiyon tanımladım ve bu targetin type değerine göre drawBitmap fonksiyonunu kullanarak farklı görseller bastırdım. Daha sonra 60 milisaniyede ucagın hareketi için kosulu doldurmaya başladım. Öncelikle gemi hareketi ve ekranda görünmesi için için ana kod kısmında moveShip ve drawShip adında bir fonksiyonlar yazdım. Bu fonksiyonlar:

- moveShip metodu: Öncelikle oyun classını ana kod kısmına include etmiştim ve bundan bir global Oyun *oyun adında değişken oluşturup new Oyun() yapıyla bir yeni oyun sınıfından nesne adresi atamıştım. Bu oyun classının constructur içinde classın özelliği olan Ship *s değerine bir gemi nesnesi adresi atanıyor. Bu fonksiyonda geminin hareketi için potansiyometre kullandım ve bu fonksiyonda A1 analog değerine bağladığım potansiyometrenin değerlerini bazı koşullardan sonra ship classındaki update fonksiyonunu kullanarak geminin x eksenindeki konumunu güncelledim.
- drawShip metodu: Sadece drawBitmap ile geminin görselini ekrana bastırdım.

Bu iki fonksiyonu yazdıktan sonra ayrıca drawShip metodunu target hedefleri için olan koşul blogunada ekledim.

Adım 6

Bu adımda her iki hareket blogunda kullandığım can kontrolü için kontrol1 ve canKontrol metotlarını yazdım. Bu metotlar:

- kontrol1 metodu: Oyun classında ve 4 adet parametre alıyor. Genel olarak matristeki NULL olmayan objelerin konum bilgisi ile geminin konum bilgisini karşılaştırıp objelerin type değerine göre farklı olaylar gerçekleştiriyor. Örneğin type 1 ya da 2 ise gemi canı 1 azalır , type 4 ise cephanesinin değeri 1 artar gibi değişken özellikleri değeri değişir.
- canKontrol metodu: Ana kod kısmında yazdım içinde kontrol1 metodunu ve çarpışma anında ses çıkışı için buzzer kullandım. Kontrol1 den dönen gemi canı değeri ile önceki can değeri arasında azalma varsa bir koşula girer buzzer çalışır. Ayrıca çarpışmadan sonra 3 sn hasar

görmeme olayı içinde bazı değişken değerlerini değiştirip 3 sn içinde can kontrol1 içindeki type 1 ve 2 için çarpışma kontrol işlemi devre dışı bırakılıp geminin hasar görmesi engelleniyor.

Bu metotları yazdıktan sonra geminin silah objesini fırlatma ve uzay çöpi gibi maddeleri yok etmesi ve silahın ekranda gösterilmesi için atis(), kontrol2() ve drawSilah() metotlarını yazdım ve her iki hareket blogunda kullandım. Bu metodlar:

- atis metodu: Atış yapmak için bir adet butonunu kartın 10. digital pinine bağladım ve setup metodunda pineMode ile giriş alacağımı söyledim. Bu metod butonun durumuna ve geminin cephanesinin 0'dan büyük olması durumunda geminin cephanesinin sayısını 1 azaltıyor ve Ship classında yer alan insertSilah fonksiyonu ile Ship classındaki silah dizisine 1 tane silah nesnesi ekliyor.
- kontrol2 metodu: Bu metod Oyun classında ve temel işlevi silah dizisindeki neslerin konumları ile yerget matrisindeki nesnelerin konumlarının değerlerine ve targetin type değerine göre targetin yok edilmesini sağlıyor.
- drawSilah: Ana kod kısmında tanımlı olan bu fonksiyon sadece mevcut silah dizisinde yer alan silahların görüntüsünü ekrana yansıtıyor.

Bu fonksiyonları yazdıktan sonra oyuncunun her satır atladığında skor olarak gösterilmesi için 3 adet seven-segment displayı karta bağladım ve skor adında bir fonksiyon yazıp bu fonksiyonda target hareket koşul blogunda kullandım. Bu fonksiyon:

- skor metodu: Bu fonksiyon global olarak tanımlı satır adımlarını tutan geişkenin basamak değerlerine göre segment-display da rakamları yazdırıyor.

Skor gösteriminide bitirdikten sonra en son olarak eğer geminin can değeri 0'a eşit olursa page değişkeninin değerini 1 yapıp hem while dan çıkmayı hemde ana menüye dönmeyi sağladım.

Adım 7

Bu adımda zorluk seviyesi 2 için düzenlemeler yapmaya başladım içerik olarak seviye 1 ile aynı fonksiyonları kullanıcak fakat her while döngüsü başında bazı kontroller ekledim. Bu kontrollerin amacı her 10 sn de targetların düşme hareketleri yüzde 20 oranında hızlanacak olması. Bunu sağlamak için birçok global değişken tanımladım. Örneğin sn değişkeni bu targetların hareket sürelerini tutuyor ve syc olarak tanımladığım diğer değişken her harekette artıyor ve kaç hareket 10 sn yapacak onu tutan bir tmp değişkenimde var. Eğer burada syc tmp'e eşit olursa bu kontrol yapısına girer ve sn değişkeni yüzde 20 değerinde azaltılır tmp değişkeni yeni sn değişkeninin kaç defada 10 sn ye ulaşacağı sayıyı günceller ve bu şekilde target hareket yapısı her 10 sn de yüzde 20 hızlanmış bir görüntü oluşur diğer yapılar bu kontrol yapısı dışında zorluk 1 ile aynıdır. Ayrıca ek olarak deflt adında bir fonksiyonum var. Bu fonksiyon:

- deflt metod: Bu fonksiyon ana kod kısmında tanımlıdır ve geminin canı bittiginde ana ekrana döndüğünde ve daha sonra tekrar yeni oyuna başlanacağı zaman

daha önce kullanılmış olan neredeyse tüm değişkenlerin başlangıctaki değerlerini atar böylelikle her şey baştan başlanmış gibi olur.

Bu fonksiyonuda projeye ekledikten sonra projeyi bitirmiş oldum.

III. ÖZET

Bu projenin amacı, çeşitli sensörler kullanarak mikro-denetleyici tabanlı bir oyun makinesi geliştirmektir. Böylelikle bilmediğim bileşenlerin nasıl kullanıldığını ne işe yaradıklarını uygulayarak ve test ederek daha iyi öğrendim ve benimsedim. Ayrıca arduino ide yi C++ kullanarak yazdım ve C++ bilgilerim ve arduino ile bilgilerim yenilendi ve daha bilmediğim birçok şey öğrendim.

IV. SONUÇ

Tüm kontroller yapıldıktan sonra artık menü de eklediğim 3 buton ile gezinip seçim yapabiliyor seçtiğim zorluk seviyesine göre farklı dinamikte uzay gemisi oynu oynayabiliyorum. Ayrıca gemiyi hareket ettirmek için potansiyometrem , ışığa balı ekran renk değişimi için LDR , çarpışmada uyarı sesi için buzzer , skor tabelası için seven-segment display larım ve ledlerim başarılı bir şekilde çalıştırıp oyunumu oynayabildim. Fakat projeyi yaparken bazı problemlerle karşılaştım bunlar:

- Öncelikle arduino uno kullanırken projeye daha fazla tanımlamalar ve eklemeler yapılırken hafıza sorunu ile karşılaştım. Bu sorunu arduino mega kullanarak giderdim.
- Bazı bileşenleri kullanırken çalışmama gibi durumlar ile karşılaştım fakat bunlarda bazı bağlam ya da INPUT ya da OUTPUT mod durumlarından kaynaklandığını buldum ve bunları düzelttim.

Genel olarak projenin sonuçları böyleydi.

V. KAYNAKÇA

- <https://www.mischianti.org/2021/07/14/ssd1306-oled-display-draw-images-splash-and-animations-2/>
- <https://www.circuitgeeks.com/oled-display-arduino/>
- <https://randomnerdtutorials.com/guide-for-oled-display-with-arduino/>
- <https://www.circuitbasics.com/programming-with-classes-and-objects-on-the-arduino/>

