

BLM 103- Algoritmalar ve Programlama I  
2019-2020 Güz Dönemi  
  
PONG OYUNU  
  
Proje Teslim Raporu  
6 Ocak 2020

Ogün Berat Gürses, Mehmet Çolak

1 Giriş 1

1.1 Projenin Amacı 1

1.2 Proje Ekibi 1

2 Geliştirilen Uygulama 2

2.1 Kullanılan Araçlar 2

2.2 Tasarım 2

3 Sonuçlar 3

# Giriş

## Projenin Amacı

***Bir paddle’ı yapay zeka kullanarak dinamik hale getirip, belirli zorluk düzeylerine göre vermesi gereken tepkileri ayarlayıp, bu yapay zeka tepkisinin zorluk derecelerine göre aktif edilmesini sağlayıp. Çalışır bir PONG oyunu tasarlamak.***

## Proje Ekibi

***Mehmet Çolak 10.07.2001 yılında Kadiköy'de dünyaya geldi. 2019 yılında Kadiköy Nazmi Arıkan Fen Bilimleri Temel Lisesi'nden mezun oldu. Şu anda Fenerbahçe Üniversitesi Bilgisayar Mühendisliği bölümünde lisans eğitimi almakta. C ve benzeri dillerle ilgilenmektedir. 190301022***

***Ogün Berat Gürses 10.11.2000 yılında Sakarya ili Adapazarı ilçesinde doğdu. 2018 yılında Sakarya Anadolu Lisesi'nden mezun oldu. Şu anda Fenerbahçe Üniversitesi Bilgisayar Mühendisliği bölümünde lisans eğitimi almakta. C ve benzeri dillerle ilgilenmektedir. 190301005***

# Geliştirilen Uygulama

## Kullanılan Araçlar

***Visual Studio***

***Adobe Photoshop CC***

***Paint***

## Tasarım

##### ***Algoritmamızın nasıl çalıştığı konusunda biraz bilgi vermek gerekirse, algoritmamızı başlattığınızda ilk olarak karşınıza bir ana menü çıkıyor. Bu ana menüde oyunumuzun ismi, zorluklar ve bu zorlukları seçebilmek için basmanız gereken tuşlar gösteriliyor. Bu ekrandan herhangi zorluğu seçmek için bastığınız tuş, arka planda gereken kod satırları aktif hale geliyor. Ardından oyunumuz başlıyor ve top ilk olarak sizin tarafınıza geliyor. Buradan sonra topu karşılarsanız yapay zeka paddle’miz harekete geçiyor ve hangi zorluğu seçtiğinize bağlı olarak topu karşılamaya çalışıyor. Eğer kolay modu seçtiyseniz büyük ihtimalle kazanmanız pek uzun sürmeyecektir. Eğer zor modu seçtiyseniz paddle sizi biraz zorlayacaktır. Eğer imkansız modu seçtiyseniz gol atabilmeyi düşünmeyin.***

##### ***Yazdığımız kod parçalarını anlatacak olursak;***

##### ***Tasarladığımız algoritma dört ana başlıktan oluşmaktadır.***

##### ***1) Paddle hareketi***

##### *Yapay zeka paddle’nin dinamik hareket etmesi sağlandı.*

##### ***2)Zorluk belirleme***

##### *Yazdığımız algoritmanın hız değişkenlerini değiştirerek paddle’nin hızını arttırıp azaltarak istenilen zorluk derecelerini elde ettik.*

##### ***3)Tuş atamaları***

##### *Algoritmamızdan beklenen zorluk dereceleri istenilen tuşlara atayarak zorluk dereceleri arasında seçim yapılabilmesini sağladık.*

##### ***4)Ana ekran tasarımı***

##### *Bize verilen görseller üzerinde çeşitli oynamalar yaparak, yaptığımız tasarıma uygun hale getirdik.*

##### ***1) Paddle hareketi***

int center = paddle[0].y + 25;

int screen\_center = screen->h / 2 - 25;

int ball\_speed = ball.dy;

if (ball\_speed < 0) {

ball\_speed = -ball\_speed;

}

//top sağa hareket

if (ball.dx > 0) {

//merkeze geri dön

if (center < screen\_center) {

paddle[0].y += ball\_speed;

}

else {

paddle[0].y -= ball\_speed;

}

}

Bu kısım bize verilen merkeze dön kodu olup burada paddle’nin merkeze dönerken ki hızı ve merkeze nasıl döneceği gösterilmiştir.

else {

//top aşağı hareket ederken

if (ball.dy > 0) {

if (ball.y > center) {

paddle[0].y += 1.96;

}

else {

paddle[0].y -= 1.96;

}

}

//top yukarı haraket ederken

if (ball.dy < 0) {

if (ball.y < center) {

paddle[0].y -= 1.96;

}

else {

paddle[0].y += 1.96;

}

}

//top düz hareket ederken

if (ball.dy == 0) {

if (ball.y < center) {

paddle[0].y -= 1.96;

}

else {

paddle[0].y += 1.96;

}

}

}

##### }

##### Bu kısımlar kolay mod için yazdığımız kısım olup burada aşağıdan, yukarıdan ve merkezden gelen toplarda paddle’nin nasıl hareket edeceği ve ne kadar hızlı olacağı tasarlanmıştır.

##### ***2)Zorluk belirleme:***

##### Bu kısımda az önce gösterdiğimiz kod parçasının farklı hızlarda oluşturulmuş hali bulunmaktadır. (Kolay mod=1.96, Zor mod=3.5, İmkansız mod=10)

##### ***3)Tuş atamaları***

//draw background

SDL\_RenderClear(renderer);

SDL\_FillRect(screen, NULL, 0xFF120a8f);

//ana menüyü getir

if (state == 0) {

if (keystate[SDL\_SCANCODE\_F1]) {

state = 1;

}

if (keystate[SDL\_SCANCODE\_F2]) {

state = 3;

}

if (keystate[SDL\_SCANCODE\_F3]) {

state = 4;

}

//arayüzü getir

draw\_menu();

//oyun bitti ekranını getir

##### }

##### Bu kısımda basılan tuşa göre hangi durumun başlayacağı ayarlanmıştır.

else if (state == 1) {

//check score

r = check\_score();

//if either player wins, change to game over state

if (r == 1) {

state = 2;

}

else if (r == 2) {

state = 2;

}

//paddle ai movement

move\_paddle\_ai\_kolay();

//\* Move the balls for the next frame.

move\_ball();

//draw net

draw\_net();

//draw paddles

draw\_paddle();

//\* Put the ball on the screen.

draw\_ball();

//draw the score

draw\_player\_0\_score();

//draw the score

draw\_player\_1\_score();

##### }

##### Bu kısımda sadece kolay modun aşamaları verilmiştir. Diğer modların kodları da bu şekildedir, sadece state durumları ve hangi fonksiyonun başlayacağı farklıdır.

##### Oyun içerisindeki kademeler ve bu kademelerin hangi sıra ile çalıştırılacağı bu kısımda belirlenmektedir.

##### ***4) Ana ekran tasarımı***

##### Gerekli tüm tasarım dosyaları github adresimizde mevcuttur.

##### ***Tasarım sürecinde karşılaştığımız sıkıntılar:***

##### *1)* *Paddle’nin hareket etmesini ayarlamaya çalışırken paddle’nin oyun alanının dışına çıkması.*

##### *2) Topun düz bir hat üzerinde hareketinde paddle’nin topa karşı tepkisiz oluşu.*

##### *3)Zorluk derecelerini ayarlarken verdiğimiz değerler nedeniyle paddle’nin titreme sorunu.*

# Sonuçlar

***Bu proje ile ilk oyunumuzu tasarlamış olduk. SDL kütüphanesinin ne olduğunu, nasıl ve neden kullanıldığını öğrendik. Bir oyun algoritması tasarlarken, bu algoritmanın nasıl daha optimize bir şekilde yazılabileceğini ve menü vb. birimlerle nasıl etkileşim kurduğunu tecrübe etmiş olduk.***

Hazırlanan sunum video’su adresi: <https://youtu.be/dwF2nQ1UKfw>

Dosyaların github adresi: <https://github.com/Siencer/BLM-103-PONG-OYUNU.git>