

# CengBall programlama oyunu uygulama geliştirme süreci

Alper Demir, Emre Can Küçüköglü, Doğa Uzuncukoğlu, Necati Çevik

## Programlama yarışması platformu

Programlama oyunu, oyuncuların oyuna doğrudan bir etki gösteremediği, bunun yerine kendi yazdıkları bilgisayar programı veya programcığı aracılığıyla oyun içi karakterlerin hareketlerini kontrol edebildiği bir bilgisayar oyunu çeşididir. CengBall bir programlama oyunu projesidir. CengBall projesi, yazılımla veya yapay zekâ tasarımıyla ilgilenen insanlara, kendi yapay zekâ kodlarını yazma ve başkalarıyla yarıştırmaya imkânı vermektedir. Proje, kuralları basitleştirilmiş bir futbol oyunudur. CengBall programlama oyunu, gerçek hayat spor stratejisini, yazılım mühendisliğini ve yapay zekâ algoritma tasarımı bir araya getirmektedir. Kullanıcıların yapay zekâ algoritmaları tasarlayıp bu algoritmalar için yazılım geliştirebileceği ve diğer kullanıcılarla yarışabileceği bir platform olmasının yanında; futbol oyununun bileşenlerini nesnel tabanlı bir şekilde dinamik bir altyapıda kontrol edilebilir ve değiştirilebilir tutarak yapay zekâ geliştirmeye olanak sağlamaktadır.

Proje, geniş şablondan bakmak gerekirse, eşit sayıda oyuncusu bulunan iki futbol takımının yazılmış kodlar aracılığıyla yarıştırılması temeline dayanmaktadır. Takımları kontrol eden kullanıcılar, oyuncular bazında pozisyon değiştirme, şut çekme, pas atma, araya kaçma gibi tercihleri yaparak, her an dinamik değişen top durumuna göre taktik geliştireceklerdir. Simülâtör tarafından kullanıcılara düzenli veri akışı sağlanacaktır. Bu veri akışı, kullanıcının takım ajanını öğrenebilir kılmasını sağlayacak dinamik etmenleri de barındırmaktadır. Örnek vermek gerekirse, ajan karşı takımdan şut çeken oyuncuların yüzdesini çıkartabilecek ve dinamik bir şekilde adam adama baskı gibi futbol taktikleriyle kontrol sağlayabilecektir.

Oyun kapsamında; futbol oyununun 2 boyuta indirgenmiş ve ofsayt, köşe vuruşu gibi kuralları çıkartılmış bir halinin benzetimi yapılmaktadır. Fiziksel değişkenler gerçeklikle örtüşür şekilde, bilimsel normlarla hesaplanmaktadır.

Kullanıcı kodunun sisteme dâhil edilmesinin tamamen güvenli hale getirilmesi için sistemin güvenlik protokollerinin tasarımı proje süresince geliştirilmektedir. Ayrıca projede; gerçekleştirme, yapay zekâ, görselleştirme parçaları ve sınıf tanımları arasındaki veri alışverişi güvenliği, gerçekleştirme katmanında veri yedekleme mantığıyla sağlanmaktadır. Bu sayede onay alamayacağı çağrılarda bulunan yapay zekâ ajanları uyarılabilecek ve asla gerçekleşen oyunu bozmayacaktır.

Kullanıcı, oyun ara yüzünden sisteme dâhil ettiği kodunu, oyun başı ayarlarını yaptıktan ve sistemde yer alan yerel rakiplerden birini seçtikten sonra gerçekleyebilecektir. Simülasyon sonrası oluşturulan disk dosyası, gerçekleşmiş maçın 2 boyutlu grafik ar yüzleriyle, gerçek zamanlı olarak sergilenmesini sağlayacaktır.

## Sistem Tasarımı

Sistem 3 alt sistemden oluşmaktadır. Bunlar oyuncuların sistemle etkileştiği yapay zekâ kodları, bu kodların çalıştırıldığı ve yarıştırıldığı gerçekleştirme kısmı ve sonuçların görselleştirildiği görselleştirme kısmıdır.



Takım sınıfı, kullanıcıların kapsamını belirleyeceği ana parça olma özelliği göstermektedir. İçerisinde her bir zaman diliminde gerçekleştirme parçası tarafından güncellenen Oyuncu sınıfından iki ayrı nesne listesi yer almaktadır. Bu nesnelere, yapay zekâ ortamının tamamen gözlemlenebilir olması amacıyla oyunculara ait pozisyon ve hız bilgilerini barındırmaktadır. Fakat kullanıcı; rakip Takım sınıfına ait Oyuncu nesnelerinin özelliklerini ancak bu özellik sergilendiğinde görebileceklerdir. Örnek vermek ve somutlaştırmak gerekirse, şut çekmemiş bir oyuncunun şut değeri bilinemeyecektir.

Oyuncu sınıfından oluşturulacak nesnelere, nesnelere tabana dökülmüş bir futbol oyununda takımların birer alt bileşenleridir. Bu sistematik yapı korunmaktadır. Oyuncu sınıfı nesnelere, bazı değişmez özelliklerin yanı sıra yorulma gibi zaman dilimleriyle doğru orantılı bir şekilde, oyunculara özgü katsayı kadar hızla düşüşe geçen değişken özellikler de barındırmaktadır. Bir Oyuncu nesnesi, şut, pas, top kapma, hareket etme gibi hareketlere olan niyetini, gerçekleştirme parçasına bildirmek için metotlar içermektedir.

Tekilliği saklanarak gerçekleştirme parçası içinde korunan Top sınıfı nesnesi, Takım sınıfının da birer bileşeni olma özelliği taşır. Bu sayede yapay zekâ ajanı, kontrol etmenin oldukça önemli olduğu Top nesnesinin pozisyon ve hız bilgilerine istediği zaman ulaşabilir.

Nesnelerinin özellikleri ve matematiksel fonksiyon hesaplamalarında kullanılacak olan katsayıların tanımlarını içeren *Saha* sınıfı, bazı özelliklerini oyun başı ayarlarından alacaktır.

### **Yapay Zekâ Modülü**

Bu alt sistem, kullanıcının sistemle etkileşime geçtiği parçadır. Bu kodlar sistemden düzenli olarak veri alacak ve bunlara dayalı kararlar vererek, girdileri sisteme iletecektir.

Kullanıcılar kendilerine sağlanan API aracılığıyla kodlarını geliştirebileceklerdir. Bu API kullanıcıya; takımı yönetme ve çevre hakkında bilgi alma imkânı sağlayacaktır. Kullanıcının daha rahat bir şekilde geliştirme sağlayabilmesi için API içerisinde yardımcı yapay zekâ metotları yer alacaktır.

Kullanıcı, Takım ara yüzünü geliştirerek, kendi stratejisini oluşturabilecektir. Bu süreçte dinamik olarak değişen veriler, oyuncu pozisyonları, top pozisyonu ve oyuncu yetenekleri ve değişken durumlarıdır.

Bu veriler, karar verme sırası gelen oyuncuya gerçekleştirme parçası tarafından sağlanacaktır.

Ayrıca oyun başında belirlenecek bazı değişmez özellikler bu parçada kullanıcılar tarafından değiştirilemeyecek şekilde ayarlanmaktadır.

### **Görselleştirme Modülü**

Gerçekleştirme parçasının oluşturup, sabit diske kaydettiği kayıt dosyalarını okuyarak, gerçekleştirilen müsabakanın iki boyutlu görselleştirilmesini sağlayan parçadır. Bu parça aracılığıyla, oyun gerçek zamandan çok daha hızlı bir gerçekleştirme aşamasından sonra, gerçek zamanlı olarak veya istenilen hızlarda kullanıcıya izlettirilebilecektir. Kaydedilmiş oyun dondurulup, tekrar başlatılabilecek, zaman dilimleri şeklinde ilerleyerek analiz edilebilecektir. Görselleştirme parçasının ihtiyacı olan veri, bir KayıtDosyası sınıfı nesnesinden oluşan fiziksel bir veri dosyasıdır. Her bir zaman

dilimi sırasında gerekleme parası tarafından Algı sınıfı nesnesi oluşturulacaktır. Oyunun gereklenmesi ařaması tamamlandığında, gerekleme parası ierisinde, tım zaman dilimlerine ait Algı sınıfı nesnelere bulunacaktır. Bunlar KayıtDosyası sınıfının bir nesnesi altında tutulmaktadır. Bu nesne ise google-gson kütüphanesinin yardımıyla JSON dosyası şeklinde oluşturulup, sıkıştırılarak saklanacaktır. Diske yazılacak bu dosya 2 dakikalık bir görselleřtirme için yaklaşık 2MB büyüklüğünde olacaktır. Simülasyon tarafından kullanılmak üzere, diske yazılan verinin boyutunu azaltmak için kullanılan sınıf tanımları mevcuttur. Bunlardan MetaVeri sınıfı bir sonuç dosyasına en bařta ve bir kere yazılmaktadır. Görselleřtirme parası Swing (Java) kütüphanesinden nesnelere kullanan alt sınıflar barındırmaktadır. Görselleřtirme bu sınıf nesnelere aracılığıyla yapılmaktadır.

## Veri Tasarımı

Sistem, paralar arası iletiřim için 2 adet veri tipine sahiptir. Bunlar gerekleme ve görselleřtirme paralarının bilgi alıřveriřinde kullandığı KayıtDosyası ve gerekleme ve yapay zekâ kodlarının iletiřimi için kullandığı Algı nesnelere dir.

### Algı verisi

Bu veri tipinden oluşturulan nesnelere, gerekleme parası tarafından yapay zekâ algoritmalarına bilgi gönderirken kullanılmaktadır. Algı, genel olarak oyun ierisindeki bütün nesnelere anlık pozisyon ve hız bilgisidir. Oyun, sırası gelen kullanıcıdan aldığı girdiye göre pozisyonları ve hızları deęiřtirir, nesnelere o anda sahip oldukları hız ve pozisyon verilerini bir Algı nesnesi haline getirir ve kayıt eder. Bu sebeple her gerekleme sonucunda ierisinde bütün nesnelere pozisyonlarının bulunduğu bir Algı listesi oluşur. Algı ierisindeki bölümlere řöyledir: oyuncuların pozisyon ve hız bilgileri, topun pozisyon ve hız bilgileri, maın anlık skoru.

### Simülasyon sonuç dosyası

Simülatör parası gerekleme sürecinde oyun ierisindeki nesnelere durumlarını belirli bir formata göre kaydetmektedir. Bu şekilde gerekleme ierisinde gerekleşen bütün olaylar görselleřtirme parasına bir dosya olarak aktarılır. Modüller arasındaki bağımsızlığı saęlayan temel veri tipidir. Görselleřtirme parasının bir simülasyonu görselleřtirmesi için bilmesi gereken tek veri, dosyanın disk üzerinde depolandığı yer bilgisidir. Simülasyon sonuç dosyası, KayıtDosyası sınıfından google-gson kütüphanesi kullanılarak oluşturulan bir JSON formatında fiziksel dosyadır.

KayıtDosyası ierisindeki bölümlere řu şekildedir: kullanıcıların kendi takımları için belirledikleri takım ismi ve takım renkleri, takım ierisindeki oyuncuların isimleri ve forma numaraları, gereklemenin oluşturduğu Algı listesi.

## Ölçeklenebilirlik

Uygulama bünyesinde kavram olarak futbol oyununun basitleřtirilmiş bir hali seçilmiştir. Fakat nesnel tabanlı tasarım korunduğı için uygulama kavramı, karşılıklı mücadelenin bulunduğu ve kontrolün, rakiplere nesnelere bazında yapıldığı her türlü

alana kaydırılabilir. Rakip durumundaki kontrol edicilerin ve bu kontrol edicilerin özgül nesnelerinin sayısının önemli ölçüde artması durumunda, gerçekleştirme bu durumdan 2 şekilde etkilenmektedir. Birincil olarak, gerçekleştirimin çok izlekli bir yapı barındırması ve her bir izleğin bir takıma değiştirilebilir bir üst zaman sınırı koyması sebebiyle, gerçekleştirme süresi uzamaktadır. Fakat bu süre, seçilen kavramın durum uzayının büyüklüğüne göre ayarlanabilir olduğu için istenilen sonucun elde edilmesini etkilememektedir. İkincil olarak ise, gerçekleştirme tarafından serileştirilerek, fiziksel olarak saklanan veri dosyasının boyutunda artış meydana gelmektedir. Fakat bu dosya boyutunu Java programlama dilinin, değişkenleri, nesnelerin kalıcı durumları hakkında bilgi içermediğini ifade etmek için kullandığı *transient* önekini kullanarak düşürmek mümkündür. Ayrıca oluşturulan JSON tipindeki dosya, sıkıştırma algoritmaları kullanılarak beşte birine kadar küçültülebilmektedir.

## Güvenlik

Uygulama kapsamında Algı verisinin bütünlüğü gerçekleştirme parçası tarafından korunmaktadır. Gerçekleştirme, ömür süresi boyunca, doğruluğunu korumakla yükümlü olduğu verilerin kopyasını kendi bünyesinde, dışarıya kapalı bir şekilde saklamaktadır. Dış etmen olarak nitelendirilebilecek yapay zekâ kodları ile gerçekleştirme arasındaki veri trafiği; yapay zekâ kodlarının kendilerine sunulan ve gerçekleştirme tarafından da erişilebilen değişkenleri (niyet) ayarlaması şeklinde olmaktadır. Simülatör bu değişkenleri, sakladığı birincil veriler bazında değerlendirmektedir. Simülatör tarafından yapay zekâ kodlarına periyodik olarak aktarılan verinin bütünlüğü ise, yapay zekâ kodlarına istediği zaman çağırabildiği metotlar sağlanarak korunmaya çalışılmıştır. Kullanıcı girdisi, doğrudan kod aracılığıyla olduğundan dolayı, bu kodun yapabilecekleri konusundaki güvenlik konuları önemlidir. Bu noktada, kullanıcı kodunun sistemi bozamaması için, proje kapsamında oyun çalışır haldeyken kullanıcı kodunu derleyip sisteme dâhil etmek mümkündür. Derleme işlemi, kullanıcıya, takım kodunu yazması için gerekli olan sınıf kodlarını içeren bir kütüphane aracılığıyla yapılmaktadır. Bu kütüphane, kullanıcının çalıştığı kod yazma platformunda, gerekli altyapıya sahip olmasını sağlayacaktır. Ayrıca, kullanıcı kodunu sınırlamak adına, Java güvenlik kütüphaneleri kullanılacak, kullanıcının oyunu durdurması veya kapatması engellenecektir.

## Sonuç

CengBall projesi, kullanıcılara yapay zekâ alanında kendilerini sınamaları için fırsat veren ve programlama yeteneklerini göstermelerini sağlayan bir projedir. Basit futbol tabanı, futbolun popülerliği ve izlemesi zevkli bir oyun olması dolayısıyla seçilmiştir. Bu noktada, programlamaya ilgisi olan ve futbol seven insanların ilgisi hedeflenmiştir. Proje, temelde iki takımı karşılıklı oynatacak bir platform oluşturmuş ve oyun türünü değiştirilebilir kılmıştır. Bu hususta, gelecekte aynı platform kullanılarak farklı oyun türlerinde de yapay zekâ geliştirmek ve oynamak mümkündür.

## **Kaynaklar**

Buckland M. (2005). Programming Game AI By Example. Wordware Publishing, Inc.