

Project 11

CengBall: Yapay Zekâ Futbol Algoritmaları Oyunu

Yazılım Tasarım Raporu

Project 11
14.03.2014

İçindekiler

1. Giriş	3
a. Problem tanımı	3
b. Amaç	3
c. Kapsam.....	3
d. Literatür taraması ve ilişkili çalışmalar.....	4
e. Projenin özgünlüğü	4
f. Tanımlar, kısaltmalar.....	5
2. Sistem Tasarımı	6
a. Simülatör modülü	8
b. Yapay zekâ modülü	11
c. Görselleştirme modülü	12
3. Veri Tasarımı	16
a. Algı	18
b. Simülasyon sonuç dosyası.....	18
4. Arayüz Tasarımı.....	19
a. Menü ekranı.....	19
b. Takım ekleme ekranı	19
c. Maça başlama ekranı	19
d. Maç ekranı	20
5. Araçlar ve Kütüphaneler	21
a. Swing (Java) kütüphanesi.....	21
b. Google-gson kütüphanesi	21
6. Proje Takvimi.....	21
a. Proje tarih ve görev tablosu.....	22
b. Proje Gantt çizelgesi.....	23
7. Referanslar	26

Şekiller Listesi

Şekil 1: Sistem tasarımı genel görünüş	6
Şekil 2: Genel sistem tasarımı	7
Şekil 3: Simülatör modülü	8
Şekil 4: Takım sınıfı	9
Şekil 5: Oyuncu sınıfı	10
Şekil 6: Top sınıfı	10
Şekil 7: Saha sınıfı	11
Şekil 8: Görselleştirme modülü	13
Şekil 9: Algı sınıfı	14
Şekil 10: KayıtDosyası sınıfı	14
Şekil 11: MetaVeri sınıfı	15
Şekil 12: Görselleştirme modülü sınıfları	15
Şekil 13: Tasarım sınıfları	15
Şekil 14: Genel veri tasarımı	17
Şekil 15: Menü ekranı	19
Şekil 16: Maça başlama ekranı	20
Şekil 17: Maç ekranı	21
Şekil 18: Proje planı, Şubat ayı	23
Şekil 19: Proje planı, Mart ayı	24
Şekil 20: Proje planı, Nisan ayı	24
Şekil 21: Proje planı, Mayıs ayı	25
Şekil 22: Proje planı, tam tarihleriyle plan sonu	25

1. Giriş

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 [1]. 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 tasarımını bir araya getirmektedir.

a. Problem tanımı

- Kullanıcıların yapay zekâ algoritmaları tasarlayıp bu algoritmalar için yazılım geliştirebileceği ve diğer kullanıcılarla yarışabileceği, kolay kullanılabilir ve eğlenceli bir platform oluşturmak,
- Yapay zekâ tasarlama ve bunu bir platforma uygulama aşamalarını da projenin bir parçası yapmak,
- Futbol oyununun bileşenlerini nesnel tabanlı bir şekilde dinamik bir altyapıda kontrol edilebilir ve değiştirilebilir tutarak yapay zekâ geliştirmeye olanak sağlama.

b. Amaç

- Yapay zekâ yazılımlarıyla ilgili insanların, eğlenceli bir platform aracılığıyla, birbirleriyle bu konuda mücadele edebileceği imkânı sağlamak amacıyla bu platformu geliştirmek,
- Platformu modüler ve geliştirilebilir bir şekilde oluşturup, ilgili araştırma, geliştirme ve eğitim kurumlarında geleneksel bir turnuvaya dönüştürülebilir olmasını sağlamak,
- Uluslararası arenadaki örnekleri gibi, yapay zekâ geliştirme platformunu, üniversitelerde derslere materyal olarak sunabilmek veya yarışma şeklinde bu derslere katkı sağlayabilmek,
- Uluslararası alanda saygınlık sahibi yarışmalara milli bir rakip olması ve bu yarışmalarda yarışan yapay zekâ algoritma geliştiricilerinin, kendilerini katılımcılar arasında test edebileceği bir platform oluşturmak.

c. Kapsam

Yapay zekâ (YZ veya İng. Artificial Intelligence'den AI), insanın düşünme yöntemlerini analiz ederek bunların benzeri yapay yönergeleri geliştirmeye çalışmaktır [2]. CengBall projesi, bu alana ilgi duyan ve kendini bu alanda geliştirmeye çalışan insanlar için eğlenceli bir öğrenme ve uygulama platformudur. Proje kapsamında kullanıcı, nesnel programlama tabanında dinamik değişen duruma göre, öğrenerek tepki verebilen futbol takım ajanı kodlayacaktır.

Proje, geniş şablondan bakmak gerekirse, altışar 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. Simülasyon 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 sağlayabilecek.

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; simülatör, yapay zekâ, görselleştirme modülleri ve sınıf tanımları arasındaki veri alışverişi güvenliği simülatör katmanında veri yedekleme mantığıyla sağlanmaktadır. Bu sayede onay alamayacağı çağrılarda bulunan yapay zekâ ajanları uyarılabilecek ve asla simüle edilen oyunu bozamayacaktır.

Kullanıcı, oyun arayüzünden sisteme dâhil ettiği kodunu, oyun başı ayarlarını yaptıktan ve sistemde yer alan yerel rakiplerden birini seçtikten sonra simüle edebilecektir. Simülasyon sonrası oluşturulan disk dosyası, simüle edilmiş maçın 2 boyutlu grafik arayüzleriyle, gerçek zamanlı olarak sergilenmesini sağlayacaktır.

d. Literatür taraması ve ilişkili çalışmalar

CengBall projesiyle en çok benzerlik gösteren çalışma, her sene uluslararası düzenlenen RoboCup turnuvasıdır [3]. Bu turnuva dâhilinde, yarışmacılar elektronik olarak geliştirdikleri robotlar aracılığıyla fiziksel bir ortamda kuralları basitleştirilmiş bir futbol oyununda mücadele etmektedirler. Boğaziçi Üniversitesi bünyesinde Prof. Levent Akın ve öğrencileri bu fiziksel turnuvaya katılıyorlar.

Programlama oyunu kategorisinde çeşitli oyunlar yer almaktadır. Bu oyunlar genel olarak robotik savaş simülasyonları ve yapboz oyunlarıdır. Spor alanında, yapay zekâ geliştirilerek oynanabilen tek oyun çeşidi ise TORCS isimli araba yarışıdır [4]. Bu araba yarışında kullanıcı arayüzü üzerinden yarış ve araba seçimi yapılmasına karşın, 3 boyutlu yarış anı yalnızca bir görselleştirmeden ibarettir. Oyun, kullanıcının C++ programlama dilinde yazılmış, *drive* fonksiyonu dâhilinde ilerlemektedir.

Ayrıca Massachusetts Institute of Technology bünyesinde, her yıl düzenli olarak BattleCode adında bir yapay zekâ programlama yarışması yapılmaktadır [5]. Ara dönemde, geliştirme süreçleri ders olarak sunulan ve internet üzerinden takibi mümkün olan BattleCode oyunu, gerçek zamanlı bir savaş stratejisi oyunudur. Yarışma ise, bu oyun için en iyi oyuncu kodunu geliştirmek üzerinedir.

Proje geliştirme sürecinde, Mat Buckland'a ait *Programming Game AI by Example* isimli kitap, yapay zekâ algoritma tasarımları konusunda kaynak olarak kullanılmaktadır [6].

e. Projenin özgünlüğü

Programlama oyunları kategorisinde [7],[8] CengBall, futbol-benzeri milli bir simülasyon olarak ilktir. Futbol dünyasının dinamiklerini, programlama becerileriyle birleştirmesi açısından CengBall, sayıları oldukça sınırlı programlama oyunları arasındaki yerini alacaktır.

Hedef aldığı kitle ve oynanış biçimi açısından benzer futbol oyunlarından da ayrılan CengBall, yapay zekâ programlamasıyla ilgilenenleri, üniversite öğrencilerini ve yazılım kabiliyetlerini eğlenceli bir şekilde geliştirmek isteyenleri bir araya getirerek, ulusal ve uluslararası düzeyde müsabaka ve yarışmalar düzenlenebilecek ortamı sağlamaya çalışması açısından da diğer oyunlardan ayrılmaktadır.

f. Tanımlar, kısaltmalar

Sınıfların proje dâhilinde kod olarak karşılıkları	Sınıfların raporda kullanılan Türkçe karşılıkları
Simulator	Simülâtör
Team	Takım
Player	Oyuncu
Visualizer	Görselleştirme
Pitch	Saha
SaveFile	KayıtDosyası
Percept	Algı
Intent	Niyet
Metadata	MetaVeri

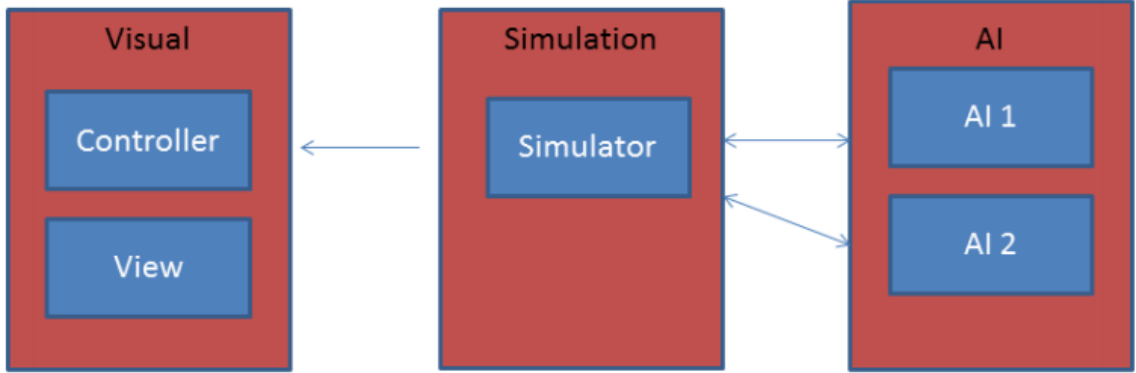
Diğer tanım ve kısaltmalar

JSON [9]	JavaScript Object Notation: Veri alışverişinin boyutunu küçültmek, kolaylaştırmak ve sistematikleştirme amacıyla kullanılan bir içerik kategorileştirme dilidir.
Java [10]	Bir programlama dili ve bilgi işlem platformudur.
API	Application Programming Interface: Uygulama Programlama Arayüzü

2.Sistem Tasarımı

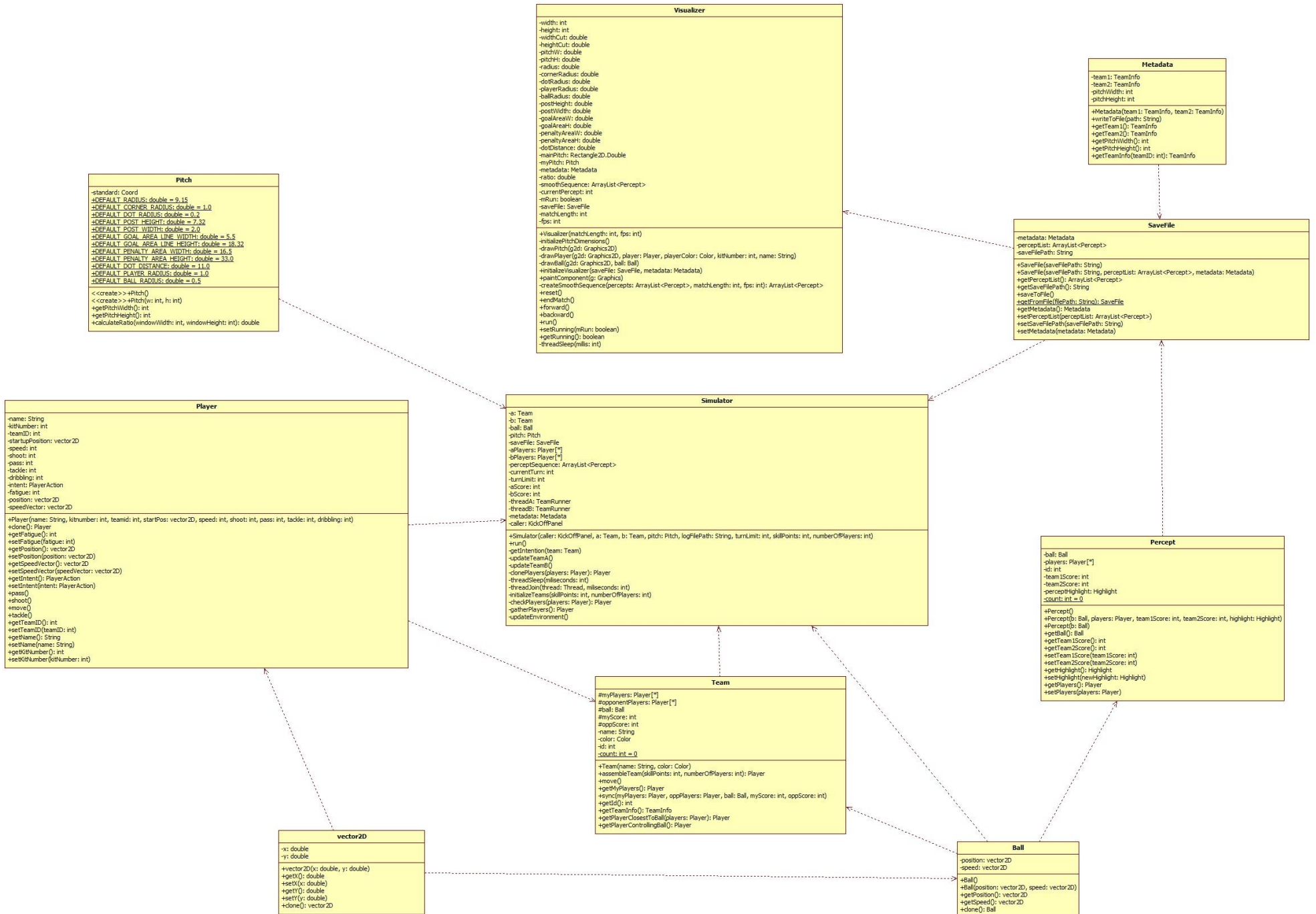
Sistem 3 alt sistemden oluşmaktadır. Bunlar oyuncuların sistemle etkileştiği *yapay zekâ kodları*, bu kodların çalıştırıldığı ve yarıştırıldığı *simülâtör* kısmı ve sonuçların görselleştirildiği *görselleştirme* kısmıdır.

Bileşen şeması *şekil 1'*de görülmektedir.



Şekil 1: Sistem tasarımı genel görünüş

Sınıf diyagramı ise ana hatlarıyla *şekil 2'*de sunulmuştur.



Şekil 2: Genel sistem tasarımı

a. Simülâtör modülü

Bu modül, periyodik olarak aldığı kullanıcı girdisini, yapay futbol kurallarına uygunluğunu kontrol ederek, simüle edecektir. Yapay zekâ kodları arasındaki iletişimi sağlayacaktır. Simülasyon sonlandığı zaman sonuçları kaydedecektir. Sonuçlar daha sonra *görselleştirme* modülü tarafından görselleştirilecektir.

Simulator
<pre>-a: Team -b: Team -ball: Ball -pitch: Pitch -saveFile: SaveFile -aPlayers: Player[*] -bPlayers: Player[*] -perceptSequence: ArrayList<Percept> -currentTurn: int -turnLimit: int -aScore: int -bScore: int -threadA: TeamRunner -threadB: TeamRunner -metadata: Metadata -caller: KickOffPanel</pre>
<pre>+Simulator(caller: KickOffPanel, a: Team, b: Team, pitch: Pitch, logFilePath: String, turnLimit: int, skillPoints: int, numberOfPlayers: int) +run() -getIntention(team: Team) -updateTeamA() -updateTeamB() -clonePlayers(players: Player): Player -threadSleep(milliseconds: int) -threadJoin(thread: Thread, milliseconds: int) -initializeTeams(skillPoints: int, numberOfPlayers: int) -checkPlayers(players: Player): Player -gatherPlayers(): Player -updateEnvironment()</pre>

Şekil 3: Simülâtör modülü

Simülâtör modülü (şekil 3) içinde barınan *Takım* sınıfı objelerinden, kendi metotları aracılığıyla aldığı verileri simüle edecektir. *Takım* sınıfından türettiği sınıfı kodlayacak olan kullanıcı, takım oyuncularının her biri için *niyet* ayarlaması yapacaktır. Örnek vermek gerekirse kullanıcı şut çekmesini istediği oyuncu üzerinden ilgili metodu, kuvvet ve doğrultu verileriyle çağırarak, bu işlemde ilgili oyuncuda bir *niyet* ayarlaması yapacaktır. *Simülâtör* modülü, takım verilerindeki tutarlılık ve bu verilerin oyunun fiziksel özellikleriyle uyumluluğunu kontrol edecek, bir sonraki sıra için, saha içi nesnelere dinamik özelliklerini güncelleyecektir.

Team
<pre>#myPlayers: Player[*] #opponentPlayers: Player[*] #ball: Ball #myScore: int #oppScore: int -name: String -color: Color -id: int -count: int = 0</pre>
<pre><<create>>+Team(name: String, color: Color) +assembleTeam(skillPoints: int, numberOfPlayers: int): Player +move() +getMyPlayers(): Player +sync(myPlayers: Player, oppPlayers: Player, ball: Ball, myScore: int, oppScore: int) +getId(): int +getTeamInfo(): TeamInfo +getPlayerClosestToBall(players: Player): Player +getPlayerControllingBall(): Player</pre>

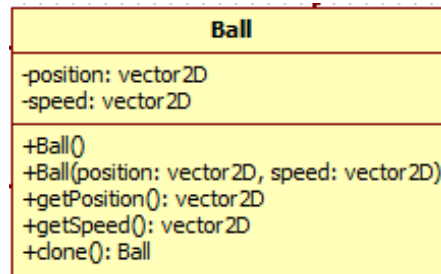
Şekil 4: Takım sınıfı

Takım sınıfı (şekil 4), kullanıcıların kapsamını belirleyeceği ana modül olma özelliği göstermektedir. İçerisinde her bir zaman diliminde simülatör 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.



Şekil 5: Oyuncu sınıfı

Oyuncu sınıfından (şekil 5) oluşturulacak nesnelere, nesnel tabana dökülmüş bir futbol oyununda takımların birer alt bileşenleridirler. Bu sistematik yapı korunmaktadır. *Oyuncu* sınıfı nesnelere, bazı değişmez özelliklerin yanı sıra *fatigue* 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 *niyet*'ini, simülatora bildirmek için metotlar içermektedir.



Şekil 6: Top sınıfı

Tekilliği saklanarak *Simülasyon* modülü içinde korunan *Top* sınıfı (şekil 6) 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.

Pitch
-standard: Coord +DEFAULT_RADIUS: double = 9.15 +DEFAULT_CORNER_RADIUS: double = 1.0 +DEFAULT_DOT_RADIUS: double = 0.2 +DEFAULT_POST_HEIGHT: double = 7.32 +DEFAULT_POST_WIDTH: double = 2.0 +DEFAULT_GOAL_AREA_LINE_WIDTH: double = 5.5 +DEFAULT_GOAL_AREA_LINE_HEIGHT: double = 18.32 +DEFAULT_PENALTY_AREA_WIDTH: double = 16.5 +DEFAULT_PENALTY_AREA_HEIGHT: double = 33.0 +DEFAULT_DOT_DISTANCE: double = 11.0 +DEFAULT_PLAYER_RADIUS: double = 1.0 +DEFAULT_BALL_RADIUS: double = 0.5
<<create>> +Pitch() <<create>> +Pitch(w: int, h: int) +getPitchWidth(): int +getPitchHeight(): int +calculateRatio(windowWidth: int, windowHeight: int): double

Şekil 7: Saha sınıfı

Nesnelerinin özellikleri ve matematiksel fonksiyon hesaplamalarında kullanılacak olan katsayıların tanımlarını içeren *Saha* sınıfı (şekil 7), bazı özelliklerini oyun başı ayarlarından alacaktır.

b. Yapay zekâ modülü

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

Kullanıcılar kendilerine sağlanan kolay kullanılabilir 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* arayüzünü geliştirerek, kendi stratejisini oluşturabilecektir. Bu süreçte dinamik olarak değişen veriler şunlardır:

- Oyuncu pozisyonları
- Top pozisyonu
- Oyuncu yetenekleri ve değişken durumları

Bu veriler, karar verme sırası gelen oyuncuya simülatör tarafından sağlanacaktır.

```

package imports;

import java.awt.Color;

import common.Ball;
import common.vector2D;

import simulator.Player;
import simulator.Team;

public class Agent2 extends Team {
    private static String teamName = "Besiktas";
    private static Color teamColor = new Color(73, 78, 12);
    private static String[] names = {"Sergen", "Stefan", "Oktay", "Ertugrul"};
    private static String[] surnames = {"Yalcin", "Kuntz", "Derelioglu", "Saglam"};

    public Agent2() {
        super(teamName, teamColor);
    }

    @Override
    public void move() {
        System.out.println(teamName + " moves. ID: " + this.getId());
    }

    @Override
    public Player[] assembleTeam(int skillPoints, int numberOfPlayers) {
        Player[] myPlayers = new Player[numberOfPlayers];
        for(int i = 0; i < numberOfPlayers; i++) {
            // TODO
            myPlayers[i] = new Player(names[0], i+1, this.getId(), new vector2D(50,50), 10,10,10,10,10);
        }
        return myPlayers;
    }
}

```

Yapay zekâ modülünden giriş koduna bir örnek.

Ayrıca oyun başında belirlenecek bazı değişmez özellikler bu modülde kullanıcılar tarafından değiştirilemeyecek şekilde ayarlanmaktadır. Bu modüle ait giriş kodu örneği yukarıda gösterilmiştir.

Kullanıcılara sağlanacak API modülü; sistem metotları, yardımcı fonksiyonlar ve simülatörün varsayılan ve yapay zekâ kodu tarafından değiştirilmesi mümkün olan değerlere dair bilgi içeren viki-sayfası ile birlikte sunulacaktır.

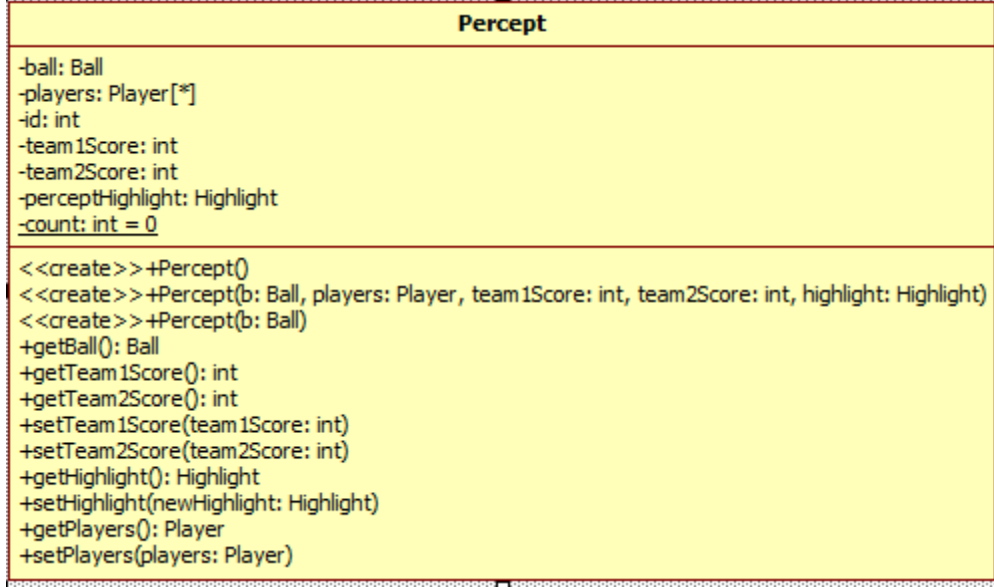
c. Görselleştirme modülü

Simülatör modülünün oluşturup, sabit diske kaydettiği kayıt dosyalarını okuyarak, gerçekleştirilen müsabakanın iki boyutlu görselleştirilmesini sağlayan modüldür.

Visualizer
<pre> -width: int -height: int -widthCut: double -heightCut: double -pitchW: double -pitchH: double -radius: double -cornerRadius: double -dotRadius: double -playerRadius: double -ballRadius: double -postHeight: double -postWidth: double -goalAreaW: double -goalAreaH: double -penaltyAreaW: double -penaltyAreaH: double -dotDistance: double -mainPitch: Rectangle2D.Double -myPitch: Pitch -metadata: Metadata -ratio: double -smoothSequence: ArrayList<Percept> -currentPercept: int -mRun: boolean -saveFile: SaveFile -matchLength: int -fps: int </pre>
<pre> + Visualizer(matchLength: int, fps: int) -initializePitchDimensions() -drawPitch(g2d: Graphics2D) -drawPlayer(g2d: Graphics2D, player: Player, playerColor: Color, kitNumber: int, name: String) -drawBall(g2d: Graphics2D, ball: Ball) + initializeVisualizer(saveFile: SaveFile, metadata: Metadata) + paintComponent(g: Graphics) + createSmoothSequence(percepts: ArrayList<Percept>, matchLength: int, fps: int): ArrayList<Percept> + reset() + endMatch() + forward() + backward() + run() + setRunning(mRun: boolean) + getRunning(): boolean + threadSleep(milis: int) </pre>

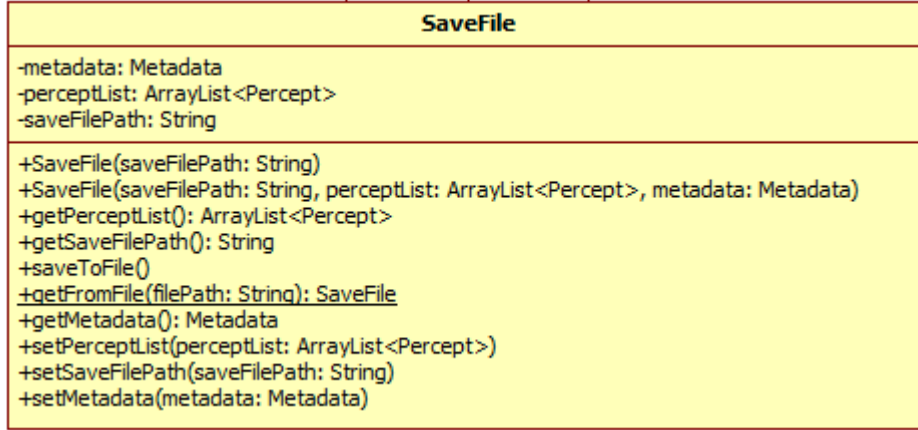
Şekil 8: Görselleştirme modülü

Bu modül aracılığıyla, oyun gerçek zamandan çok daha hızlı bir simülasyon aşamasından sonra, gerçek zamanlı olarak veya istenilen hızlarda kullanıcıya izlettirebilecektir. Kaydedilmiş oyun dondurulup, tekrar başlatılabilecek, zaman dilimleri şeklinde ilerleyerek analiz edilebilecektir.



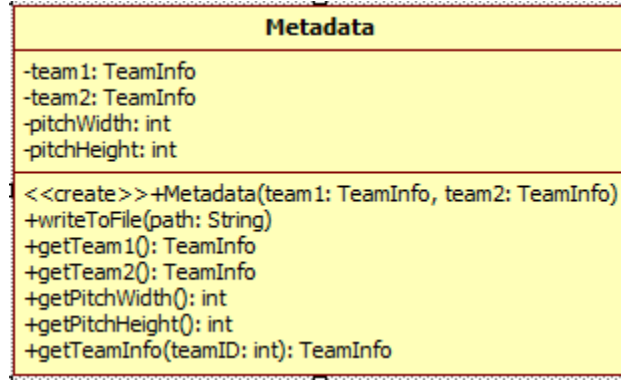
Şekil 9: Algı sınıfı

Görselleştirme modülünün ihtiyacı olan veri, bir *KayıtDosyası* sınıfı nesnesinden oluşan fiziksel bir veri dosyasıdır. Her bir zaman dilimi sırasında *simülatör* tarafından *Algı* sınıfı (şekil 9) nesnesi oluşturulacaktır.



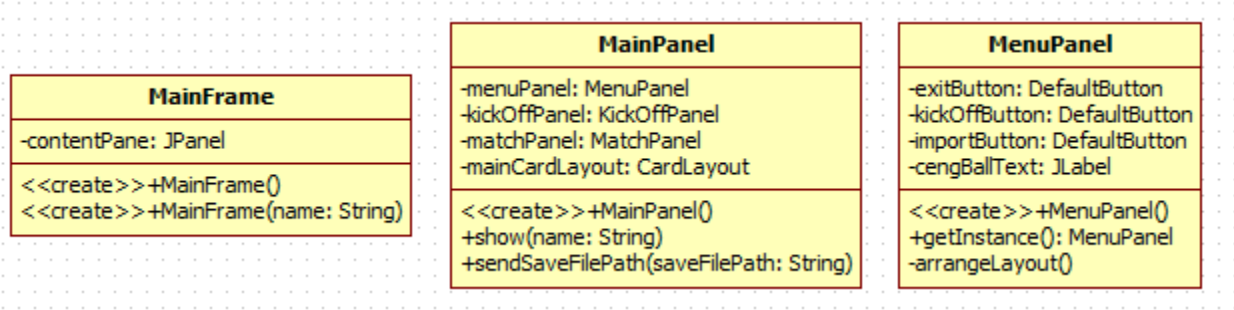
Şekil 10: KayıtDosyası sınıfı

Oyunun simülasyon aşaması tamamlandığında, *simülatör* modülü içerisinde, tüm zaman dilimlerine ait *Algı* sınıfı nesnelere bulunacaktır. Bunlar *KayıtDosyası* sınıfının (şekil 10) bir nesnesi altında tutulmaktadır. Bu nesne ise google-gson kütüphanesinin yardımıyla JSON dosyası şeklinde diske yazılacaktır. Diske yazılacak bu dosya 5 dakikalık bir görselleştirme için yaklaşık 100-200 KB büyüklüğünde olacaktır.



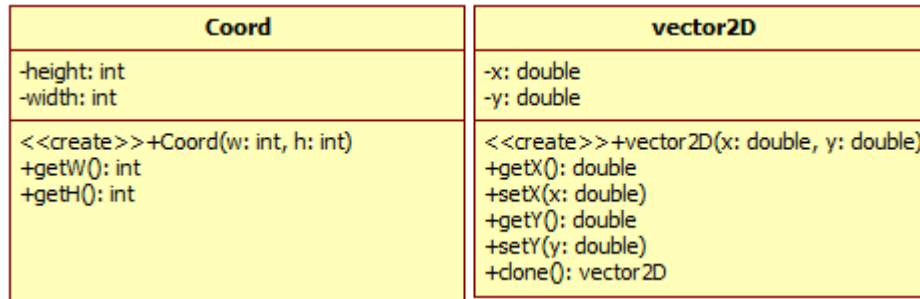
Şekil 11: MetaVeri sınıfı

Simülasyon tarafından kullanılmak üzere, diske yazılan verinin boyutunu azaltmak için kullanılan sınıf tanımları mevcuttur. Bunlardan MetaVeri (şekil 11) sınıfı bir sonuç dosyasına en başta ve bir kere yazılmaktadır.



Şekil 12: Görselleştirme modülü sınıfları

Görselleştirme modülü 5. Araçlar ve Kütüphaneler bölümünde de anlatılacağı üzere Swing (Java) kütüphanesinden nesnelere kullanan alt sınıflar barındırmaktadır. Görselleştirme bu sınıf nesnelere aracılığıyla yapılmaktadır. Modül içinde kullanılan bu sınıflar şekil 12’de gösterilmiştir.



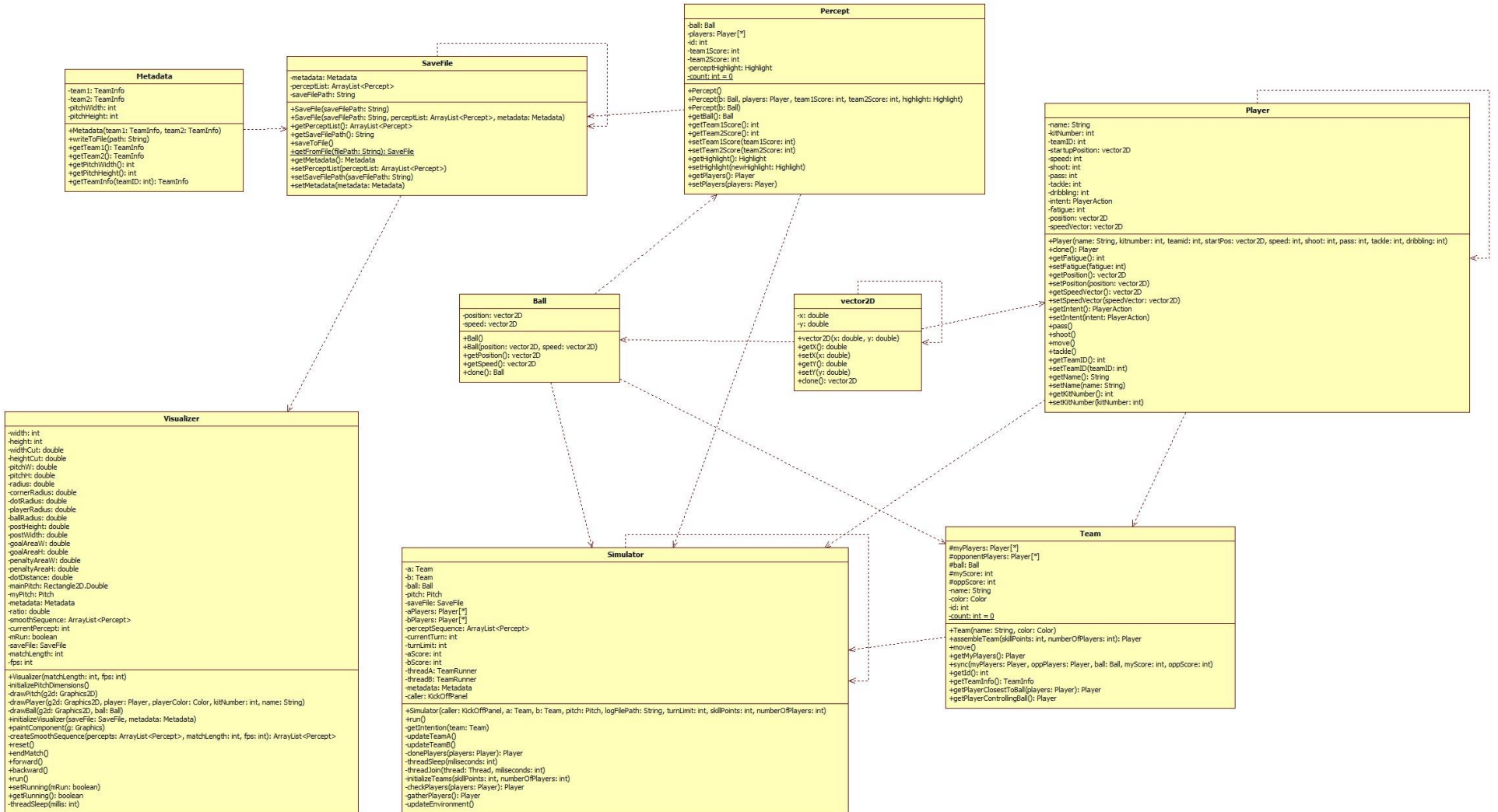
Şekil 13: Tasarım sınıfları

Nesnel tabanlı programlama teknikleri ile oluşturulan tasarım sınıfları (şekil 13), tüm modüller tarafınca kullanılabilir.

3. Veri Tasarımı

Sistem, modüller arası iletişim için 2 adet veri tipine sahiptir. Bunlar *simülatör* ve *görselleştirme* modüllerinin bilgi alışverişinde kullandığı *KayıtDosyası* ve simülatör ve yapay zekâ kodlarının iletişimi için kullandığı *Algı* nesnelere aittir.

Genel veri tasarımı ana hatlarıyla *şekil 14'de gösterilmiştir.*



Şekil 14: Genel veri tasarımı

a. Algı

Bu veri tipinden oluşturulan nesnelere, *simülatör* modülü tarafından yapay zekâ algoritmalarına bilgi gönderirken kullanılmaktadır. Algı, genel olarak oyun içerisindeki bütün nesnelere anlık pozisyon ve hız bilgisidir. Oyun, sırası gelen kullanıcıdan aldığı girdiye göre pozisyonları ve hızları değiştirir, nesnelere o anda sahip oldukları hız ve pozisyon verilerini bir *Algı* nesnesi haline getirir ve kayıtlar. Bu sebeple her *simülasyon* sonucunda içerisinde bütün nesnelere pozisyonlarının bulunduğu bir *Algı* listesi oluşur.

Algı içerisindeki bölümler aşağıda sıralanmıştır:

- Oyuncuların pozisyon ve hız bilgileri
- Topun pozisyon ve hız bilgileri
- Maçın anlık skoru

b. Simülasyon sonuç dosyası

Simülatör modülü simülasyon sürecinde oyun içerisindeki nesnelere durumlarını belirli bir formata göre kaydetmektedir. Bu şekilde simülasyon içerisinde gerçekleşen bütün olaylar *görselleştirme* modülüne bir dosya olarak aktarılır. Modüller arasındaki bağımsızlığı sağlayan temel veri tipidir. *Görselleştirme* modülünün bir simülasyonu görselleştirmesi 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.

SaveFile içerisindeki bölümler aşağıda sıralanmıştır:

- Kullanıcıların kendi takımları için belirledikleri takım ismi ve takım renkleri
- Takım içerisindeki oyuncuların isimleri ve forma numaraları
- Simülasyonun oluşturduğu *Algı* listesi

4. Arayüz Tasarımı

Oyunun arayüzü, basit, kullanıcı dostu ve temel fonksiyonları yerine getirebilecek şekilde tasarlanmıştır. Kullanıcı bu arayüz sayesinde, yazdığı takım kodunu sisteme dâhil edebilir, arzu ettiği takımlar arası maç yaptırabilir ve bu maçı izleyebilir.

a. Menü ekranı

Menü ekranı, oyun açıldığında kullanıcının ilk göreceği ekrandır. Kullanıcı bu ekran sayesinde oyunun temel fonksiyonlarını yerine getirebileceği diğer ekranlara geçiş sağlar. Bu ekran aracılığıyla, Takım ekleme ekranına ve maça başlama ekranına geçebilir veya oyundan çıkabilir. Menü ekranı *şekil 15'*de gösterilmektedir.



Şekil 15: Menü ekranı

b. Takım ekleme ekranı

Takım ekleme ekranı, kullanıcının yazdığı takım kodunu sisteme dâhil etmesini sağlar. Kullanıcı daha sonra maça başlama ekranına giderek dâhil ettiği takımını seçip maça başlayabilir. Takım ekleme işlemi, kullanıcının ekleme tuşuna basması ve kodunun dizinini sisteme bildirmesiyle başlar. Daha sonra bu kullanıcı kodu, sistem tarafından derlenir ve kontrol edilir. Kontrol aşamasında, kullanıcı kodunun maç yapabilmesi için gerekli olan fonksiyonlara sahip olup olmadığı esas alınır. Eğer derleme ve kontrol aşamasında bir hata ile karşılaşılırsa, sistem Takım ekleme ekranında kullanıcıya geri bildirim yapar ve oluşan hatalar hakkında kullanıcıyı bilgilendirir. Ekleme işlemi başarılı olursa, kullanıcı menü ekranına geri dönebilir ya da başka bir takım kodu ekleyebilir.

c. Maça başlama ekranı

Maça başlama ekranı, kullanıcının seçeceği, iki takım arasında maç yaptırmasını sağlar. Bu ekranda, takım seçmek ve maçın ayarlarını yapmak mümkündür. Seçilebilecek takımlar, daha önceden sisteme dâhil edilmiş takımlardır. Kullanıcı bu takımlar arasında gezip, arzu ettiği takımlarda karar kılar. Ayrıca,

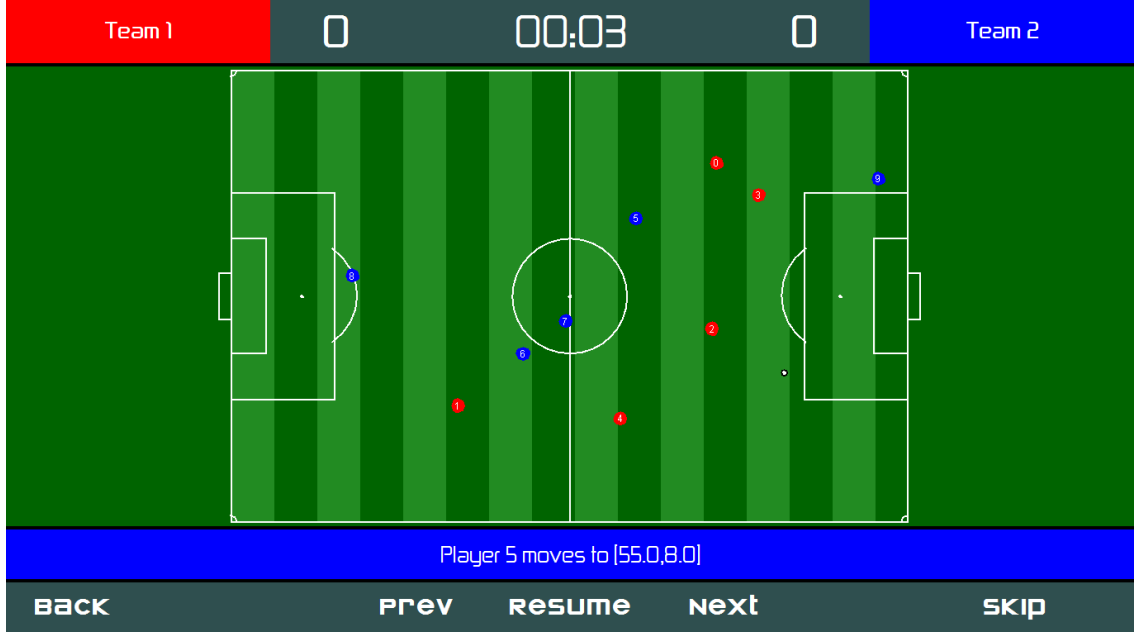
maç süresi ve maç ekranında yorum panelinin olup olmayacağı ayarları bu ekran aracılığıyla yapılabilir. Kullanıcı istediği ayarları yaptıktan sonra maçı başlatır ve sistem maç ekranına geçiş yapar. Bu ekran *şekil 16*'da gösterilmektedir. Bu şekilde *Home1* ve *Away1* sisteme eklenmiş takımları ifade etmektedir.



Şekil 16: Maça başlama ekranı

d. Maç ekranı

Maç ekranı, simülasyonu gerçekleştirilmiş bir maçın oynatılmasından sorumludur. Bu ekran simülasyon modülünün oluşturduğu maç dosyasını alır ve ekranda oynatır. Bu ekranda, takım isimler, skor, maç zamanı, yorum ekranı ve iki boyutlu olarak gösterilen futbol sahası vardır. Kullanıcı arzu ederse, Maça başlama ekranına geri dönebilir, oyunu durdurup devam ettirebilir ve maçı atlayıp doğrudan maç sonucunu görebilir. Maç sonuçlanınca, kullanıcı tekrar menü ekranına döner. Maç ekranı *şekil 17*'de gösterilmiştir.



Şekil 17: Maç ekranı

5. Araçlar ve Kütüphaneler

- Sistem, arayüz ve simülasyon görselleştirmesi için Swing (Java) kütüphanesini [11],
- Simulator ve visualizer arasındaki dosya alışverişi için google-gson kütüphanesini [12] kullanmaktadır.

a. Swing (Java) kütüphanesi

Javax.Swing paketi Java Foundation Classes (JFC) kütüphanesi içindedir. 134 sınıfı olan büyük bir pakettir. Java.AWT (Java Abstract Window Toolkit) paketinin üzerine kuruludur ve uygulamalarda onun işlevini üstlenmiştir. Java ile Grafiksel Arayüzler (GUI) hazırlamak için gerekli bütün araçlara sahiptir.

b. Google-gson kütüphanesi

Google-gson, açık kaynak kodlu bir Java kütüphanesidir. Java nesnelerinin JSON biçimine dönüştürülmesini ve bu türden tekrar Java nesnesine çevrilmesine yarar.

6. Proje Takvimi

Proje ekibi, 22 Şubat – 16 Mayıs tarihleri arasında projeyi sonuçlandırılabilir hale getirmeyi, 16 Mayıs – 31 Mayıs tarihleri sonunda da projeyi kapatmayı hedeflemektedir.

Proje ekibi, Alper Demir, Doğa Uzuncukoğlu, Emre Can Küçükkoğlu ve Necati Çevik'ten oluşmaktadır. Proje iş bölümleri, planlanan zaman aralıklarıyla tablo şeklinde ay bazında verilmiştir.

a. Proje tarih ve görev tablosu

CengBall şeklinde ifade edilen iş kalemleri tüm proje çalışanlarının katılımıyla tamamlanacaktır.

Projenin iş kalemleri	Süre	Başlangıç	Bitiş	Görevi üstlenen
Yapay zekâ API Tasarımı	11 hafta	02.22.14	05.10.14	CengBall
Sınıflar arası hiyerarşi tasarımı	1 hafta	02.22.14	03.01.14	Alper Demir
Gerekli sınıf niteliklerinin araştırılması	1 hafta	02.26.14	03.08.14	Doğa Uzuncukoğlu
Yapay zekâ yardımcı fonksiyonlarının kodlanması	1 hafta	03.01.14	05.01.14	Emre Can Küçükkoğlu
API sınıflarının kodlanması	7 hafta	03.22.14	05.10.14	CengBall
Test aşaması	2 hafta	04.26.14	05.10.14	Alper Demir
Simülasyon modülü tasarımı	7 hafta	03.08.14	04.26.14	CengBall
Simülasyon modülünün eşzamanlılık tasarımı	1 hafta	03.08.14	03.15.14	Doğa Uzuncukoğlu
Sıra-tabanlı simülasyon davranışının kodlanması	1 hafta	03.11.14	03.18.14	Emre Can Küçükkoğlu
İş mantığının kodlanması	5 hafta	03.22.14	04.26.14	CengBall
Test aşaması	2 hafta	04.12.14	04.26.14	Alper Demir
Yapay zekâ ajanı tasarlama	12 hafta	02.23.14	05.17.14	CengBall
Futbol mantığı araştırması	1.5 hafta	02.23.14	03.04.14	Doğa Uzuncukoğlu
Sezgisel futbol taktiklerine karar verilmesi	1 hafta	03.02.14	03.09.14	Emre Can Küçükkoğlu
Sonlu durum makinasının planlanması	1 hafta	03.05.14	03.12.14	Necati Çevik
Yapay zekâ ajanlarının kodlanması	10 hafta	03.09.14	05.17.14	CengBall
Test aşaması	2 hafta	05.04.14	05.17.14	Doğa Uzuncukoğlu
Kullanıcı arayüzü ve visualizer modülü geliştirmeleri	1.5 hafta	05.11.14	05.21.14	CengBall
Logo tasarımı ve kullanıcı arayüzü geliştirmesi	1 hafta	05.11.14	05.18.14	Emre Can Küçükkoğlu
Menu ekranlarının tasarımı	1 hafta	05.14.14	05.21.14	Necati Çevik
Görselleştirmede iyileştirme çalışmaları	1 hafta	05.14.14	05.21.14	Alper Demir
Test aşaması	0.5 hafta	05.18.14	05.21.14	Doğa Uzuncukoğlu
Proje web sitesi ve kullanım kılavuzu oluşturma	1.5 hafta	05.21.14	05.31.14	CengBall
Web sitesi tasarım geliştirmeleri	1 hafta	05.21.14	05.28.14	Emre Can Küçükkoğlu
Viki sayfalarının oluşturulması	1 hafta	05.22.14	05.29.14	Necati Çevik
Kullanım kılavuzu sunulması	1 hafta	05.25.14	05.31.14	Alper Demir

b. Proje Gantt çizelgesi

Task Name	Assigned To	Feb 17							Feb 24						
		M	T	W	T	F	S	S	M	T	W	T	F	S	S
1 Yapay zeka API Tasarımı	CengBall														
2 Sınıflar arası hiyerarşi tasarımı	Alper Demir														Al
3 Gerekli sınıf niteliklerinin araştırılması	Doga Uzuncukoglu														
4 Yapay zeka yardımcı fonksiyonlarının kodlanması	Emre Can Kucukoglu														
5 API sınıflarının kodlanması	CengBall														
6 Test aşaması	Alper Demir														
7															
8 Simülasyon modülü tasarımı	CengBall														
9 Simülasyon modülünün eşzamanlılık tasarımı	Doga Uzuncukoglu														
10 Sıra-tabanlı simülasyon davranışının kodlanması	Emre Can Kucukoglu														
11 İş mantığının kodlanması	CengBall														
12 Test aşaması	Alper Demir														
13															
14 Yapay zeka ajanı tasarlama	CengBall														
15 Futbol mantığı araştırması	Doga Uzuncukoglu														
16 Sezgisel futbol taktiklerine karar verilmesi	Emre Can Kucukoglu														
17 Sonlu durum makinasının planlanması	Necati Cevik														
18 Yapay zeka ajanlarının kodlanması	CengBall														
19 Test aşaması	Doga Uzuncukoglu														
20															
21 Kullanıcı arayüzü ve visualizer modülü geliştirmeleri	CengBall														
22 Logo tasarımı ve kullanıcı arayüzü geliştirmesi	Emre Can Kucukoglu														
23 Menu ekranlarının tasarımı	Necati Cevik														
24 Görselleştirmede iyileştirme çalışmaları	Alper Demir														
25 Test aşaması	Doga Uzuncukoglu														
26															
27 Website ve kullanım kılavuzu oluşturma	CengBall														
28 Website tasarım geliştirmeleri	Emre Can Kucukoglu														
29 Viki sayfalarının oluşturulması	Necati Cevik														
30 Kullanım kılavuzu sunulması	Alper Demir														

Şekil 18: Proje planı, Şubat ayı

Task Name	Assigned To	Mar 3					Mar 10					Mar 17					Mar 24					
		M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
1 Yapay zeka API Tasarımı	CengBall																					
2 Sınıflar arası hiyerarşi tasarımı	Alper Demir																					
3 Gerekli sınıf niteliklerinin araştırılması	Doga Uzuncukoglu																					
4 Yapay zeka yardımcı fonksiyonlarının kodlanması	Emre Can Kucukoglu																					
5 API sınıflarının kodlanması	CengBall																					
6 Test aşaması	Alper Demir																					
7																						
8 Simülasyon modülü tasarımı	CengBall																					
9 Simülasyon modülünün ezamanlılık tasarımı	Doga Uzuncukoglu																					
10 Sıra-tabanlı simülasyon davranışının kodlanması	Emre Can Kucukoglu																					
11 İş mantığının kodlanması	CengBall																					
12 Test aşaması	Alper Demir																					
13																						
14 Yapay zeka ajanı tasarlama	CengBall																					
15 Futbol mantığı araştırması	Doga Uzuncukoglu																					
16 Sezgisel futbol taktiklerine karar verilmesi	Emre Can Kucukoglu																					
17 Sonlu durum makinasının planlanması	Necati Cevik																					
18 Yapay zeka ajanlarının kodlanması	CengBall																					
19 Test aşaması	Doga Uzuncukoglu																					
20																						
21 Kullanıcı arayüzü ve visualizer modülü geliştirmeleri	CengBall																					
22 Logo tasarımı ve kullanıcı arayüzü geliştirmesi	Emre Can Kucukoglu																					
23 Menu ekranlarının tasarımı	Necati Cevik																					
24 Görselleştirmede iyileştirme çalışmaları	Alper Demir																					
25 Test aşaması	Doga Uzuncukoglu																					
26																						
27 Website ve kullanım kılavuzu oluşturma	CengBall																					
28 Website tasarım geliştirmeleri	Emre Can Kucukoglu																					
29 Wiki sayfalarının oluşturulması	Necati Cevik																					
30 Kullanım kılavuzu sunulması	Alper Demir																					

Şekil 19: Proje planı, Mart ayı

Task Name	Assigned To	Mar 31					Apr 7					Apr 14					Apr 21					
		M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
1 Yapay zeka API Tasarımı	CengBall																					
2 Sınıflar arası hiyerarşi tasarımı	Alper Demir																					
3 Gerekli sınıf niteliklerinin araştırılması	Doga Uzuncukoglu																					
4 Yapay zeka yardımcı fonksiyonlarının kodlanması	Emre Can Kucukoglu																					
5 API sınıflarının kodlanması	CengBall																					
6 Test aşaması	Alper Demir																					
7																						
8 Simülasyon modülü tasarımı	CengBall																					
9 Simülasyon modülünün ezamanlılık tasarımı	Doga Uzuncukoglu																					
10 Sıra-tabanlı simülasyon davranışının kodlanması	Emre Can Kucukoglu																					
11 İş mantığının kodlanması	CengBall																					
12 Test aşaması	Alper Demir																					
13																						
14 Yapay zeka ajanı tasarlama	CengBall																					
15 Futbol mantığı araştırması	Doga Uzuncukoglu																					
16 Sezgisel futbol taktiklerine karar verilmesi	Emre Can Kucukoglu																					
17 Sonlu durum makinasının planlanması	Necati Cevik																					
18 Yapay zeka ajanlarının kodlanması	CengBall																					
19 Test aşaması	Doga Uzuncukoglu																					
20																						
21 Kullanıcı arayüzü ve visualizer modülü geliştirmeleri	CengBall																					
22 Logo tasarımı ve kullanıcı arayüzü geliştirmesi	Emre Can Kucukoglu																					
23 Menu ekranlarının tasarımı	Necati Cevik																					
24 Görselleştirmede iyileştirme çalışmaları	Alper Demir																					
25 Test aşaması	Doga Uzuncukoglu																					
26																						
27 Website ve kullanım kılavuzu oluşturma	CengBall																					
28 Website tasarım geliştirmeleri	Emre Can Kucukoglu																					
29 Wiki sayfalarının oluşturulması	Necati Cevik																					
30 Kullanım kılavuzu sunulması	Alper Demir																					

Şekil 20: Proje planı, Nisan ayı

Task Name	Assigned To	Apr 28					May 5					May 12					May 19								
		M	T	W	T	F	S	M	T	W	T	F	S	M	T	W	T	F	S	M	T	W	T	F	S
1 Yapay zeka API Tasarımı	CengBall																								
2 Sınıflar arası hiyerarşi tasarımı	Alper Demir																								
3 Gerekli sınıf niteliklerinin araştırılması	Doga Uzuncukoglu																								
4 Yapay zeka yardımcı fonksiyonlarının kodlanması	Emre Can Kucukoglu																								
5 API sınıflarının kodlanması	CengBall																								
6 Test aşaması	Alper Demir																								
7																									
8 Simülasyon modülü tasarımı	CengBall																								
9 Simülasyon modülünün eşzamanlılık tasarımı	Doga Uzuncukoglu																								
10 Sıra-tabanlı simülasyon davranışının kodlanması	Emre Can Kucukoglu																								
11 İş mantığının kodlanması	CengBall																								
12 Test aşaması	Alper Demir																								
13																									
14 Yapay zeka ajanı tasarlama	CengBall																								
15 Futbol mantığı araştırması	Doga Uzuncukoglu																								
16 Sezgisel futbol taktiklerine karar verilmesi	Emre Can Kucukoglu																								
17 Sonlu durum makinasının planlanması	Necati Cevik																								
18 Yapay zeka ajanlarının kodlanması	CengBall																								
19 Test aşaması	Doga Uzuncukoglu																								
20																									
21 Kullanıcı arayüzü ve visualizer modülü geliştirmeleri	CengBall																								
22 Logo tasarımı ve kullanıcı arayüzü geliştirmesi	Emre Can Kucukoglu																								
23 Menu ekranlarının tasarımı	Necati Cevik																								
24 Görselleştirmede iyileştirme çalışmaları	Alper Demir																								
25 Test aşaması	Doga Uzuncukoglu																								
26																									
27 Website ve kullanım kılavuzu oluşturma	CengBall																								
28 Website tasarım geliştirmeleri	Emre Can Kucukoglu																								
29 Wiki sayfalarının oluşturulması	Necati Cevik																								
30 Kullanım kılavuzu sunulması	Alper Demir																								

Şekil 21: Proje planı, Mayıs ayı

Task Name	Duration	Start	Finish	Assigned To	May 26								
					M	T	W	T	F	S	S		
1 Yapay zeka API Tasarımı	11 hafta	02/22/14	05/10/14	CengBall									
2 Sınıflar arası hiyerarşi tasarımı	1 hafta	02/22/14	03/01/14	Alper Demir									
3 Gerekli sınıf niteliklerinin araştırılması	1 hafta	02/26/14	03/08/14	Doga Uzuncukoglu									
4 Yapay zeka yardımcı fonksiyonlarının kodlanması	1 hafta	03/01/14	05/01/14	Emre Can Kucukoglu									
5 API sınıflarının kodlanması	7 hafta	03/22/14	05/10/14	CengBall									
6 Test aşaması	2 hafta	04/26/14	05/10/14	Alper Demir									
7													
8 Simülasyon modülü tasarımı	7 hafta	03/08/14	04/26/14	CengBall									
9 Simülasyon modülünün eşzamanlılık tasarımı	1 hafta	03/08/14	03/15/14	Doga Uzuncukoglu									
10 Sıra-tabanlı simülasyon davranışının kodlanması	1 hafta	03/11/14	03/18/14	Emre Can Kucukoglu									
11 İş mantığının kodlanması	5 hafta	03/22/14	04/26/14	CengBall									
12 Test aşaması	2 hafta	04/12/14	04/26/14	Alper Demir									
13													
14 Yapay zeka ajanı tasarlama	12 hafta	02/23/14	05/17/14	CengBall									
15 Futbol mantığı araştırması	1.5 hafta	02/23/14	03/04/14	Doga Uzuncukoglu									
16 Sezgisel futbol taktiklerine karar verilmesi	1 hafta	03/02/14	03/09/14	Emre Can Kucukoglu									
17 Sonlu durum makinasının planlanması	1 hafta	03/05/14	03/12/14	Necati Cevik									
18 Yapay zeka ajanlarının kodlanması	10 hafta	03/09/14	05/17/14	CengBall									
19 Test aşaması	2 hafta	05/04/14	05/17/14	Doga Uzuncukoglu									
20													
21 Kullanıcı arayüzü ve visualizer modülü geliştirmeleri	1.5 hafta	05/11/14	05/21/14	CengBall									
22 Logo tasarımı ve kullanıcı arayüzü geliştirmesi	1 hafta	05/11/14	05/18/14	Emre Can Kucukoglu									
23 Menu ekranlarının tasarımı	1 hafta	05/14/14	05/21/14	Necati Cevik									
24 Görselleştirmede iyileştirme çalışmaları	1 hafta	05/14/14	05/21/14	Alper Demir									
25 Test aşaması	0.5 hafta	05/18/14	05/21/14	Doga Uzuncukoglu									
26													
27 Website ve kullanım kılavuzu oluşturma	1.5 hafta	05/21/14	05/31/14	CengBall									
28 Website tasarım geliştirmeleri	1 hafta	05/21/14	05/28/14	Emre Can Kucukoglu									
29 Wiki sayfalarının oluşturulması	1 hafta	05/22/14	05/29/14	Necati Cevik									
30 Kullanım kılavuzu sunulması	1 hafta	05/25/14	05/31/14	Alper Demir									

Şekil 22: Proje planı, tam tarihleriyle plan sonu

7. Referanslar

- | | | |
|----------------------|---|---|
| [1] | Programlama oyunu viki sayfası | http://en.wikipedia.org/wiki/Programming_game |
| [2] | Yapay zekâ tanımı viki sayfası | http://tr.wikipedia.org/wiki/Yapay_zek%C3%A2 |
| [3] | RoboCup turnuvası | http://en.wikipedia.org/wiki/RoboCup |
| [4] | TORCS programlama oyunu | http://torcs.sourceforge.net/ |
| [5] | MIT BattleCode yarışması ve dersleri | https://www.battlecode.org/ |
| [6] | Programming Game AI by Examples, Mat Buckland | http://www.ai-junkie.com/books/toc_pgaibe.html |
| [7] | Programlama oyunları listesi | http://www.aiforge.net/game-links.htm |
| [8] | Programlama oyunları viki arşiv sayfası | http://programminggames.org/ |
| [9] | JSON hakkında viki sayfası | http://en.wikipedia.org/wiki/JSON |
| [10] | Java programlama dili | https://java.com/tr/download/faq/whatis_java.xml |
| [11] | Swing (Java) kütüphanesi | http://docs.oracle.com/javase/7/docs/api/javafx/swing/package-summary.html |
| [12] | Google-gson kütüphanesi | http://en.wikipedia.org/wiki/Gson |