

Altın Toplama Oyunu

Gold Collecting Game

Cumali TOPRAK

Mühendislik Fakültesi, Bilgisayar Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
cumalitoprakk@gmail.com

Berkay Efe ÖZCAN

Mühendislik Fakültesi, Bilgisayar Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
berkayefeozean@gmail.com

Özetçe— Projede C# dili kullanılarak windows form uygulaması kullanılarak altın toplama oyunu yapılmıştır. Amaç dört farklı oyuncu ile oyun tahtasında bulunan altınları toplamaya ve en çok altını kazanmaya yöneliktir. Her oyuncu altınları toplarken farklı strateji uygulamaktadır. Bu farklı stratejilere göre en uygun arama algoritmaları kullanılmaya çalışılmıştır. Projede ayrıca iki farklı altın kategorisi bulunmakta olup bunlardan ilki gizli ikincisi açık altın kategorisidir. Oyunculardan herhangi biri bu gizli altınların üzerinden geçerse bu altın tüm oyunculara görünür hale gelecek şekilde düzenlenir. Oyunun sonunda tüm oyuncuların kazandığı, harcadığı altın sayısı ve toplam atılan adım sayısı sonuç olarak ekrana yazdırılır.

Anahtar Kelimeler—C#, Windows Form, Oyun, Arama Algoritması, Dinamik Programlama.

Abstract — In the project, a gold collection game was made using the windows form application using the C# language. The aim is to collect the gold in the game board with four different players and earn the most gold. Each player applies a different strategy when collecting gold. The most appropriate algorithms have been tried to be used according to these different strategies. The project also has two different gold categories, the first of which is hidden and the second is open gold. If any of the players pass over these hidden golds, this gold is edited to be visible to all players. At the end of the game, the number of gold won and spent by all players and the total number of steps taken are ultimately written on the screen.

Keywords— C#, Windows Form, Game, Searching Algorithm, Dynamic Programming.

I. GİRİŞ

Projede C# dili ve windows form görsel arayüzü kullanılarak altın toplama oyunu yapılmıştır. Oyun A, B, C, D isimli 4 oyuncu ile oynanmaktadır. Oyuncu kuralları gereği her oyuncu başlangıçta verilen altın miktarı ile oyuna başlar. Her oyuncu oyuna başlarken sırasıyla kendi sırası geldiği zaman hedefi daha önce yoksa hedef belirler. Daha önce eğer hedefi varsa hedefine doğru hamle yapar. Her oyuncu hamle sırası kendine geldiği zaman adım miktarı olarak 3 adım atar. Bu hamle hedefine gidecek en mantıklı hamle olmalıdır. Her oyuncu için hedefine giderken farklı algoritmalar kullanılır. Oyuncuların hedef belirlemeleri için

ikinci şart eğer ilgili oyuncu hamle sırası kendisine geldiği zaman kendi hamle kriterlerine göre yeni hedef belirler ve bir hamle miktarı ilerler. Oyuncuların hedef belirlemelerinin üçüncü şartı ise oyuncu hedefine ulaşmıştır, ve yeni hedef belirler hedefine ilerler.

Oyunda altın oranı toplam kare sayısının varsayılan olarak %20'si kadardır. Ama istenildiği takdirde bu değer bir parametre ile değiştirilebilir. Bu altın oranında varsayılan olarak %10'u da gizli altındır. Gizli ve açık altınlar oyun tahtası üzerinde farklı renklerle görselleştirilmiştir. Her bir altın değeri sırasıyla 5, 10, 15, 20 olan değerlerden birini rastgele olarak alır.

A oyuncu kural gereği kendisine en yakın altını kendine hedef olarak seçer. Bu oyuncu için her hamle varsayılan 5, hedef belirleme varsayılan 10 altın maliyetindedir. B oyuncusu en karlı olan altın kareyi hedefler. Bu oyuncu için her hamle varsayılan 5, hedef belirleme varsayılan 10 altın maliyetindedir. C oyuncusu her hedef belirleme sırasında kendisine en yakın belli sayıdaki gizli altınların olduğu kutuları açar. Biz bunu varsayılan olarak 2 altındır. Sonradan bu altınların da içinde olduğu altın karelerinden B oyuncusu gibi en karlı altını seçer. D oyuncusu diğer oyuncuların hedeflerini önceden sezme özelliği bulunur. Bu altınlara onlardan önce erişebiliyorsa bu altınlar arasında en karlısına gider. Eğer bu altınlara diğer oyuncuların önce erişemiyorsa bu altınları hariç tutar ve hedef olarak diğer altın kareleri içerisinde en karlı olanı seçer. Her hamle varsayılan 5 her hedef belirleme varsayılan olarak 20 altın değerindedir.

Oyuncular hedeflerine ulaştıkları zaman bu altın değeri ilgili kullanıcının toplanan altın değerine eklenir. Her hamle ve hedef belirleme maliyetleri de ilgili kullanıcının altın değerinden çıkarılır. Herhangi bir kullanıcının altını bittiği zaman oyuncu oyundan elenir. Diğer oyuncular oyuna devam eder. Eğer tüm oyuncuların altını biter veya tüm altınlar toplanırsa oyun biter. Her oyuncu hareketi için ayrı bir arama algoritması kullanılmıştır.

Oyun başlangıcında parametre olarak değişebilir değerler form ekranından değiştirilebilir. Başlangıç ekranı şekil 1 de gösterilmiştir.

Şekil 1 : Başlangıçta açılan parametre alma ekranı

Oyun ayrı bir form üzerinde görsel olarak dinamik olarak gösterildi. Oyun bitiş ekranı Şekil 2 de gösterilmiştir. Oyun bittiği zaman her oyuncu için toplanan toplam altın miktarı, atılan toplam adım sayısı ve harcanan toplam adım sayısı ayrı bir formda gösterilir. Uygulamada herhangi bir third party kütüphane, paket kullanılmamış olup tüm kodları bizim tarafımızdan yazılmıştır.

	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>
<u>Toplam Adım Sayısı</u>	86	72	47	23
<u>Harcanan Altın Miktarı</u>	380	380	380	275
<u>Toplanan Altın Miktarı</u>	180	180	50	90
<u>Kasadaki Altın Miktarı</u>	0	0	0	15

Şekil 2 : Oyun bitiş ekranı

II. YÖNTEM

A. Hedefe Doğru İlerlerken Algoritmayı Kullandık ?

Hedef belirledikten sonra hamle yaparken en yakın altına gideceksek tüm altınların bizden x ekseninde ve y ekseninde ne kadar uzaklıkta olduğunu hesaplar bu değerlerden bize en yakın olanına bir hamle miktarı ilerleriz. En karlı altına doğru gitmek istersek öncelikle tüm altınların bize olan uzaklıklarının yanında bu altınların değerini de hesaba kattık. Altın değerinin hedefe ilerlerken ki hamle maliyetinden farkı bize o altına giderken ki kar değerini

verir. Oyuncular hedeflerine ilerlerken gizli bir altının üzerinden geçerse bu gizli altın tüm oyunculara görünür hale gelir. Ve açık altın listesine eklenir. Gizli altın listesinden çıkarılır.

B. Kodumuzda Hangi Kod Tasarımını(Pattern) Kullandık ?

Kodumuzu yazarken mümkün olduğunca soyutlamadan (abstraction) faydalandık. Öncelikle base bir abstract gamer classı oluşturup burada tüm oyuncular için ortak olan birçok property ve metodları yazdık. Her kullanıcı için özel olan işlemleri ise abstract metodu override edecek şekilde tanımladık. Böylelikle birçok kod tekrardan kurtulmuş olduk. Görsel kısım ile oyunculara ait kod kısımlarını birbirinden tamamen ayrı yazdık sadece nesne çağırılıyla dependency injection kurallarını uygulamaya çalıştık. Oyuncular classlarının UML diyagramı Şekil 3 te gösterilmiştir.

C. Görsel Kısımda Ne Kullandık?

Kodumuzu dotnet framework ortamında yazdığımız için windows form arayüzünü kullandık. Altın karelerini oluştururken x ve y değerlerince label kullandık. Altınları temsilen sarı renkli labeller gizli altınları temsilen kırmızı renkli labelleri ve diğer yolları göstermek için gri renkli labellar kullandık.

D. Oyun Sonuçlarını Kaydetmek İçin Ne Kullandık ?

Oyuncu sonuç verilerini oyunun sonunda hem form ekranında oyun sonunda gösterilir hem de kod içerisine gömdüğümüz dosya içerisini kaydedilir. Bu dosya yolunu biz manuel olarak değil uygulamanın konumuna göre tanımladığımız sistem sayesinde relative olarak verilir.

III. DENEYSEL SONUÇLAR

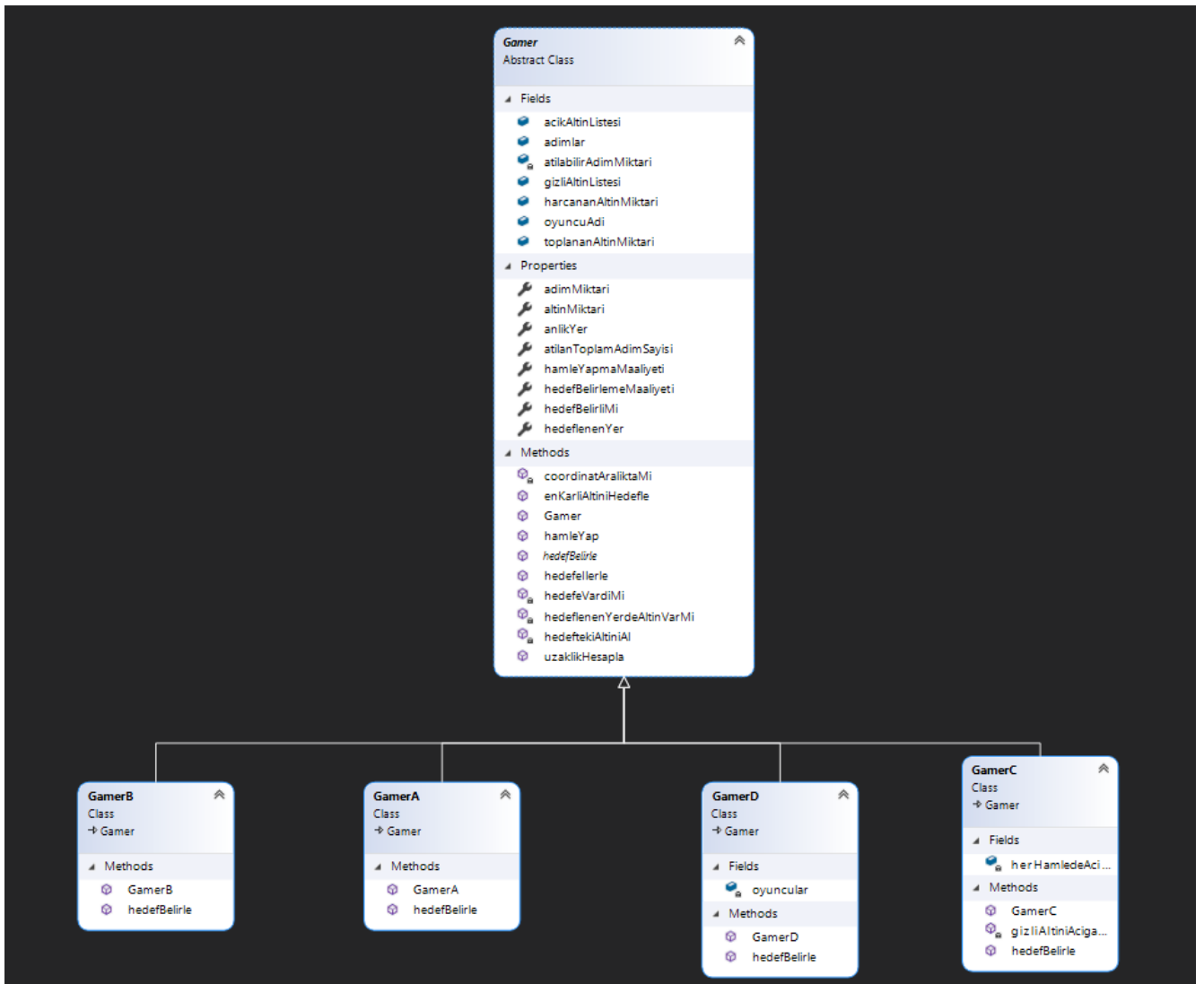
Projemiz dotnet framework platformunda yazıldığı için sadece windows ortamında çalışmaktadır. Bu programımızı birçok bilgisayarda denedik ve sorun çıkmadı. Oyun sonucu her bir kullanıcının attığı adımlar dosyaya ayrı ayrı yazılır. Dosya kod içerisinde gömülmüştür. Eğer dosya daha önce yoksa oluşturulur. Varsa bu dosya kullanılır. Program kullanılan sistem minimum sistem gereksinimlerine sahipse akıcı bir şekilde çalışmaktadır.

IV. ÖZET VE SONUÇLAR

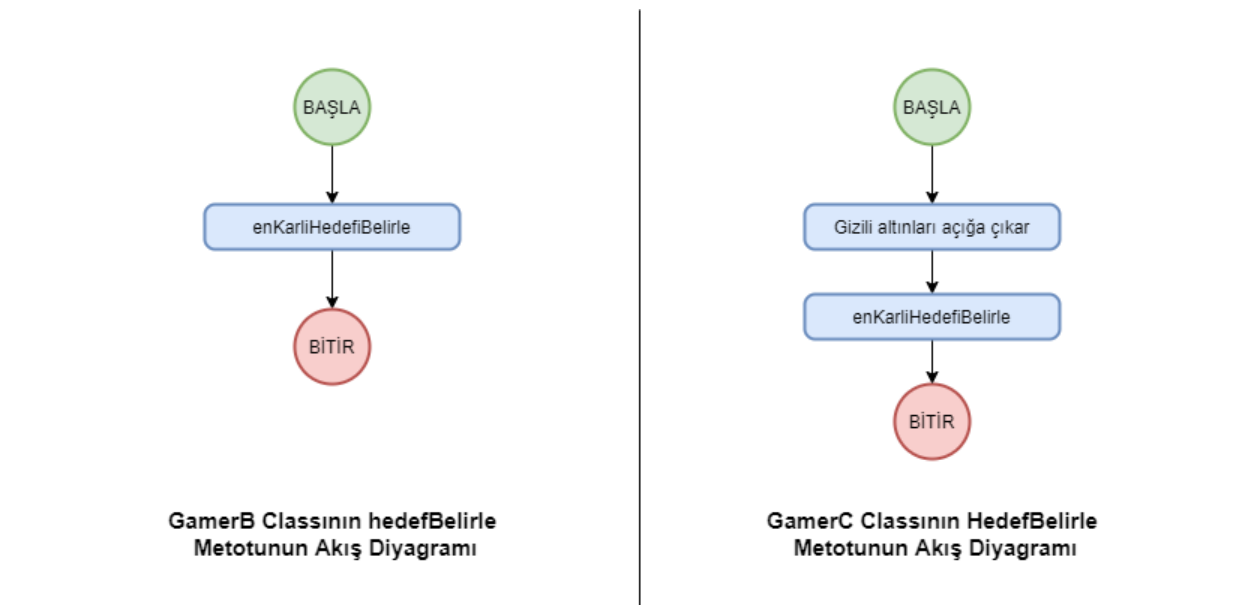
Projemizin konusu kapsamında yaptığımız altın toplama oyunu C# dili kullanılarak windows form görsel arayüzü kullanılarak gerçekleştirilmiştir. Uygulama ödev kriterlerini uyularak tüm isterler uygulanarak tamamlanmıştır. Uygulama tamamladıktan sonra birçok cihazda denenmiş hata alınmamıştır. Bu proje kapsamında arama algoritmaları kullanılmış arama algoritmalarının kullanımına aşına olunmuştur.

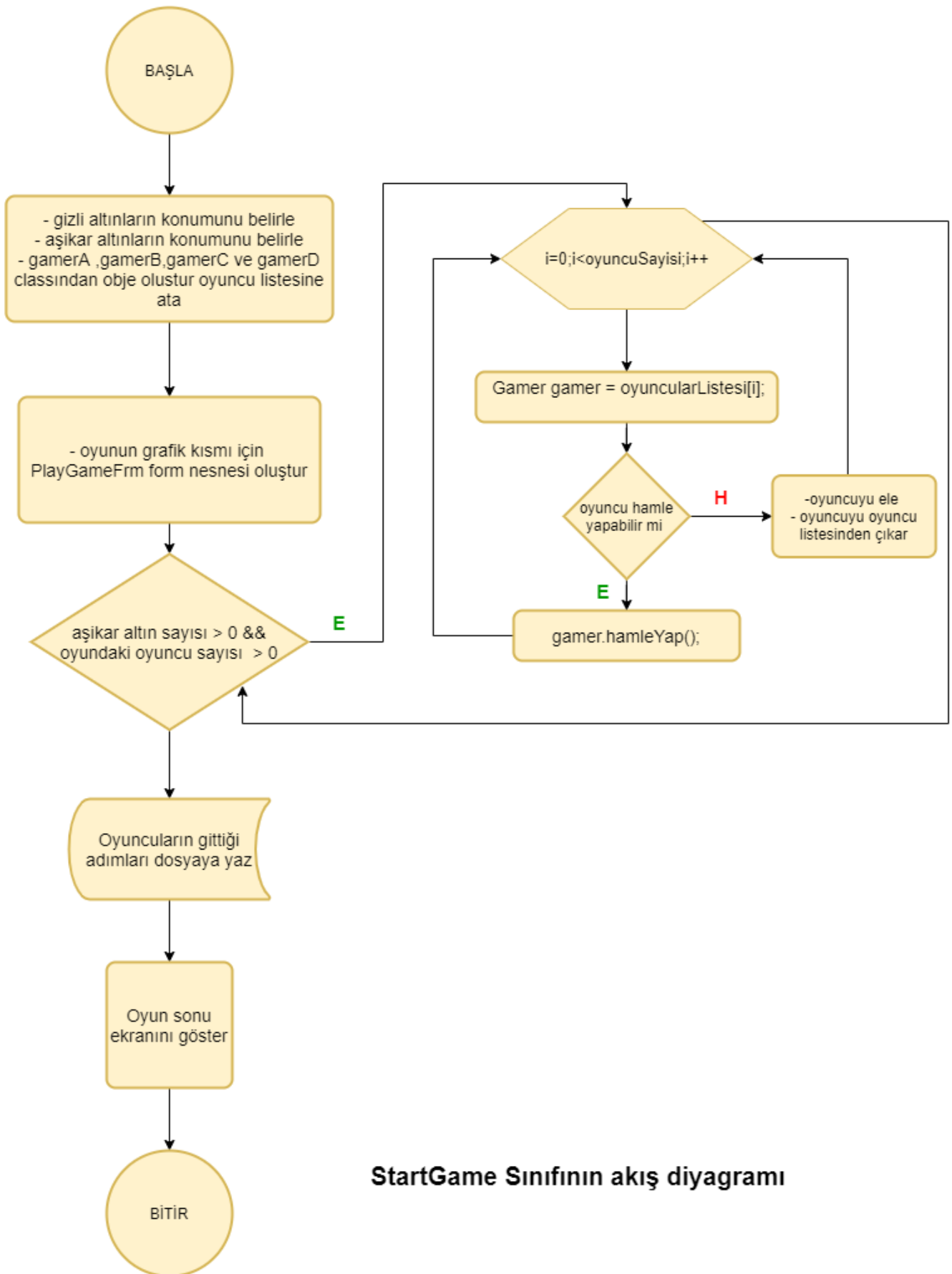
KAYNAKÇA

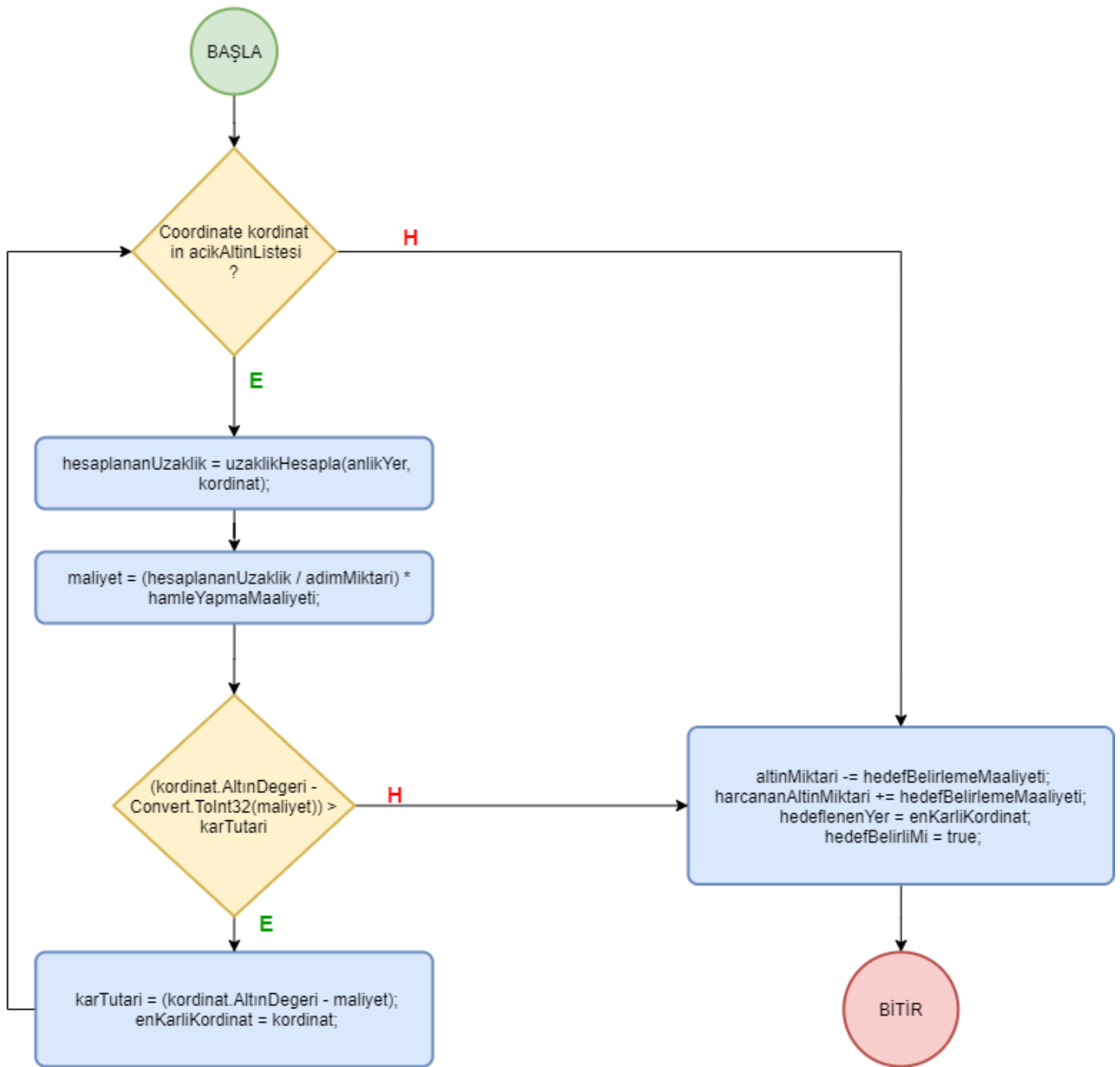
- [1] <https://www.youtube.com/>
- [2] <https://stackoverflow.com/>
- [3] <https://www.c-sharpcorner.com/>
- [4] <https://www.w3schools.com/cs/>



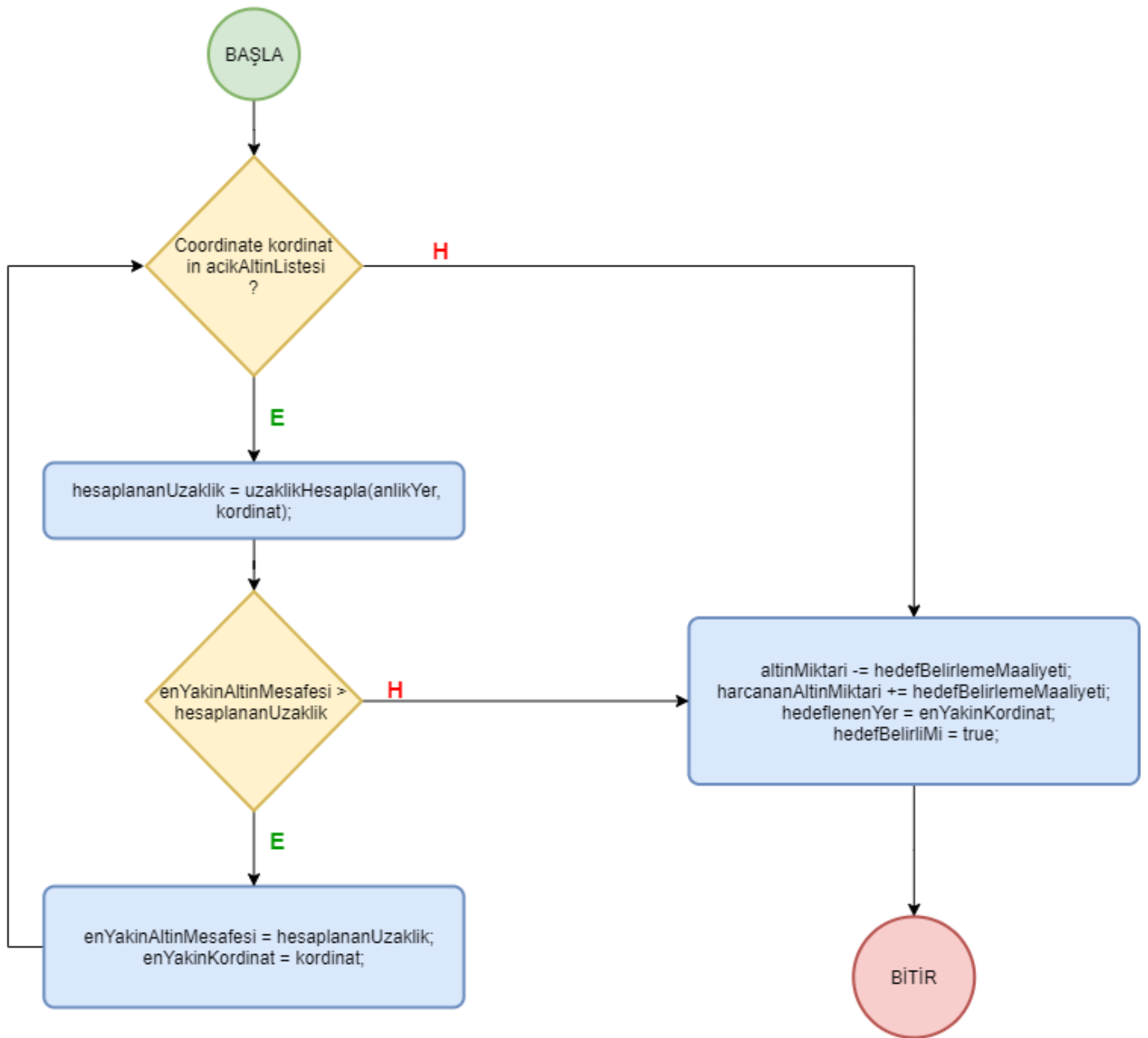
Şekil 3 : Oyuncu classlarının UML diyagramı



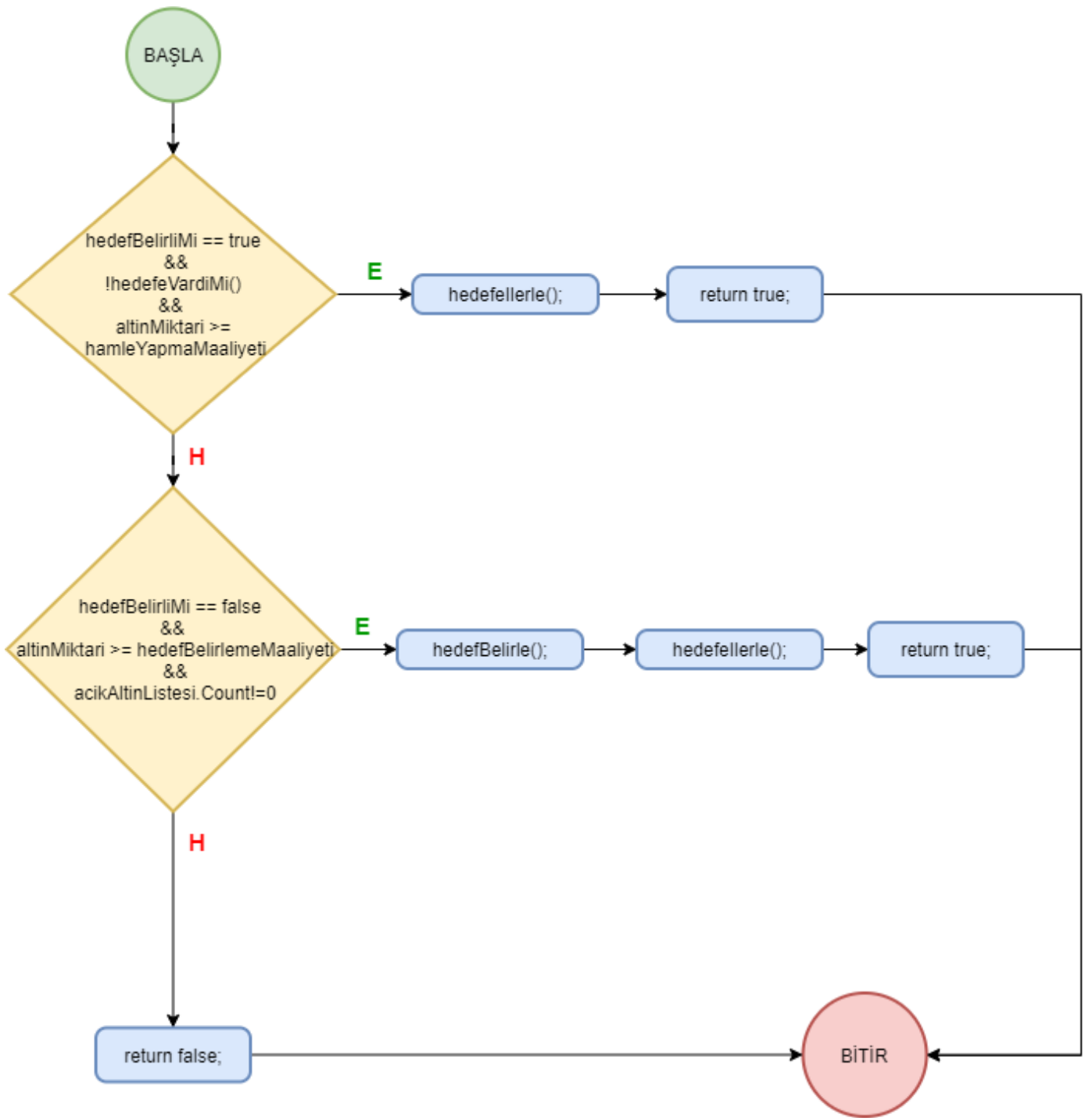




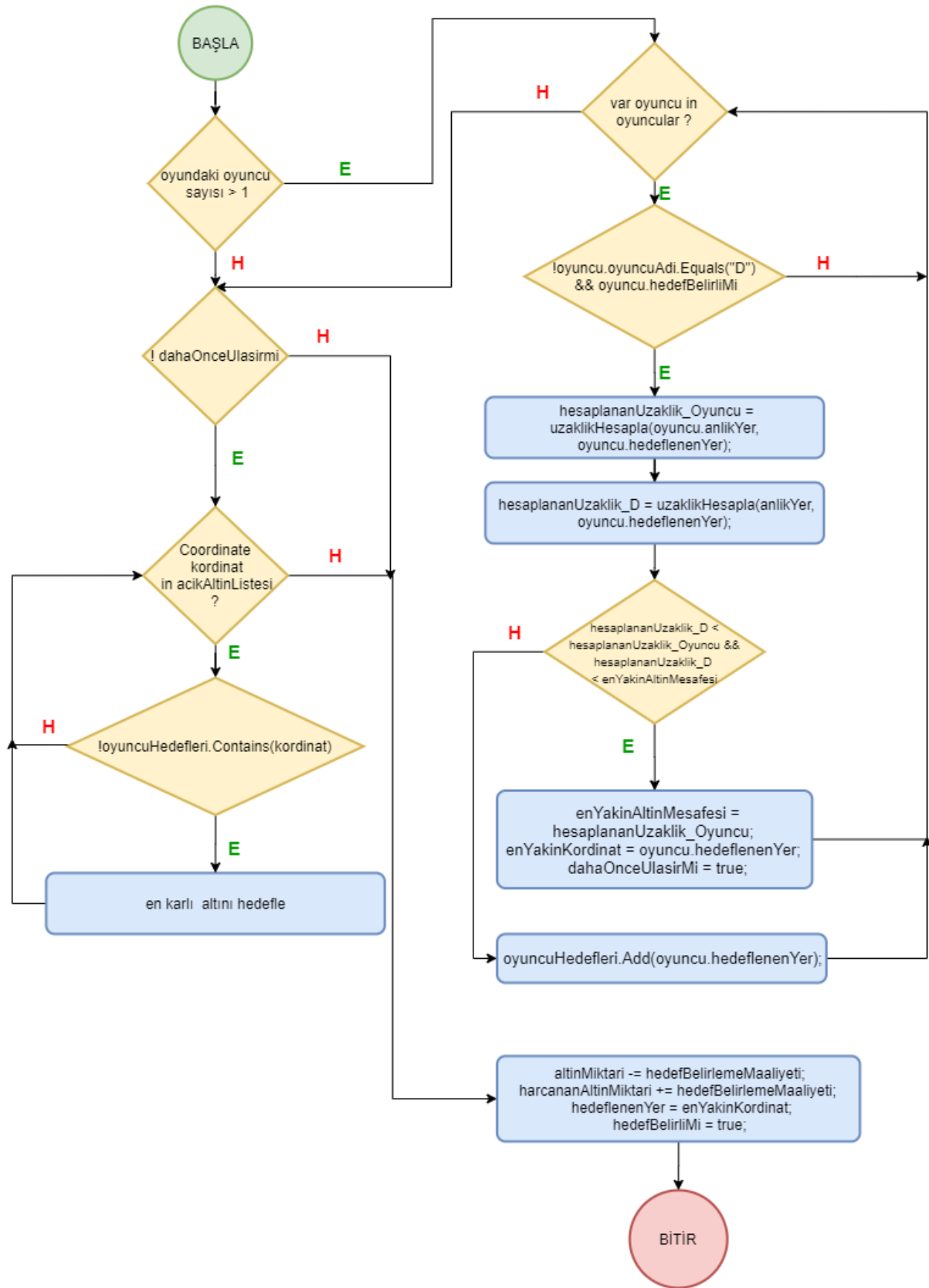
Gamer Classının enKarliAltiniBelirle Metotunun Akış Diyagramı



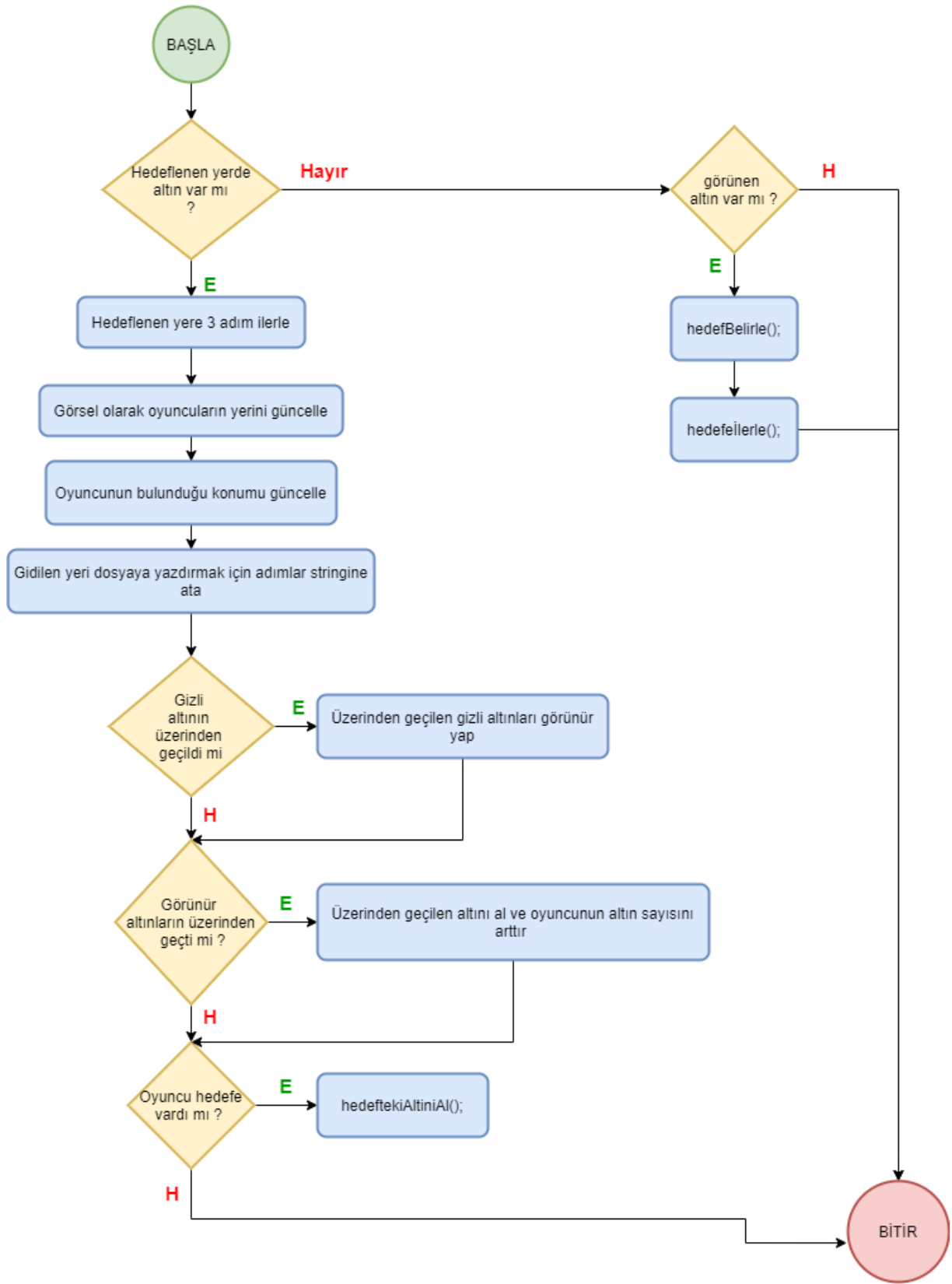
GamerA Classının HedefBelirle Metotunun Akış Diyagramı



Gamer Classının HamleYap Metedonun Akış Diyagramı



GamerD Classının hedefBelirle Metotunun Akış Diyagramı



Gamer Classının Hedeflerle Metotunun Akış Diyagramı