

Ders Adı: Algoritma ve Programlamaya Giriş

Proje Adı: Şehir/Ülke Tahmin Oyunu

Ad Soyadı: Hüseyin Dil

Numara: 360125058

Tarih: 11.12.2025

Projenin Amacı ve Açıklaması

Projem terminalde çalışan bir şehir/ülke tahmin oyunudur. İki moddan oluşur:

1. Şehir Modu (7 Hak)
2. Ülke Modu (10 Hak)

Oyun başlangıçta kuralları ekranaya yazdırır. Daha sonrasında ise mod seçimi için oyuncuya sorar. Seçilen moda göre toplam hak sayısı değişir. Bunun sebebi ülke modunda çok daha fazla ülkenin olmasıdır. Mod seçildikten sonra seçilen moda göre rastgele bir şehir/ülke seçilir ve seçilen şehir/ülkenin gizli hali, harf sayısı, toplam puanı, kalan hakkı ekranaya yazdırılır ve kullanıcidan tahmin girilmesi istenilir. Bu noktadan sonra oyun oyuncu doğru bilene kadar veya hakkı/puanı bittiğinde sonlanır. Ayrıca oyuncu tahmin ekranında 1 sayısını tuşlayarak ipucu ekranını açabilir ve buradan istediği bir ipucuya puan karşılığında kullanabilir.

Aşağıda projenin bazı adımlarının ekran görüntüsü yer almaktadır:

---Şehir Tahmin Oyununa Hoşgeldiniz---

Oyuna başlamadan önce genel kuralları görmek için ENTER'a basınız.

Oyuna 150 puan ile başlıyorsunuz.
Her bilmediğiniz turda 10 puan kaybedersiniz.
Ayrıca puan kullanarak ipucu alabilirsiniz.
Her modda yalnızca 3 ipucu vardır.
Puanınız veya hakkınız bittiğinde oyun sonlanır.

Tahmin yapacağınız kelimelerde lütfen Türkçe karakter(ç,ğ,ı,ö,ş,ü,Ç,Ğ,İ,Ö,Ş,Ü) kullanmayınız.
İZMİR yerine İzmir, Çin halk cumhuriyeti yerine Cin Halk Cumhuriyeti vb.

Oyun modu seçmek için ENTER'a basın.

Şehir Modu:

```
Şehir modu için '1' tuşlayınız.          Ülke modu için '2' tuşlayınız.  
1  
  
Şehir modunda 7 hakkınız vardır.  
Şehir Modu Başlatılıyor...  
  
*****  
  
Şehir:___(3)          Puan: 150      Kalan Hak: 7      İpucu kullanmak için '1' yazıp ENTER'a basınız.  
  
Lütfen tahmininizi yazınız: |
```

```
Lütfen tahmininizi yazınız:      1  
Seçmek istediğiniz ipucunun numarasını giriniz:  
  
-1-      (10 Puan)      Rastgele bir harf göster.  
-2-      (15 Puan)      İlk harfi göster.          (Bir kez kullanılabilir)  
-3-      (20 Puan)      Şehrin bulunduğu bölgeyi göster.      (Bir kez kullanılabilir)  
-4-      Geri dön.  
|
```

Ülke Modu:

```
Şehir modu için '1' tuşlayınız.          Ülke modu için '2' tuşlayınız.  
2  
  
Ülke modunda 10 hakkınız vardır.  
Ülke Modu Başlatılıyor...  
  
*****  
  
Ülke:_____(5)          Puan: 150      Kalan Hak: 10      İpucu kullanmak için '1' yazıp ENTER'a basınız.  
  
Lütfen tahmininizi yazınız: |
```

```
Lütfen tahmininizi yazınız:      1  
Seçmek istediğiniz ipucunun numarasını giriniz:  
  
-1-      (10 Puan)      Rastgele bir harf göster.  
-2-      (15 Puan)      İlk harfi göster.          (Bir kez kullanılabilir)  
-3-      (20 Puan)      Ülkenin bulunduğu kıtayı göster.      (Bir kez kullanılabilir)  
-4-      Geri dön.  
|
```

Örnek Oyun Ekranı:

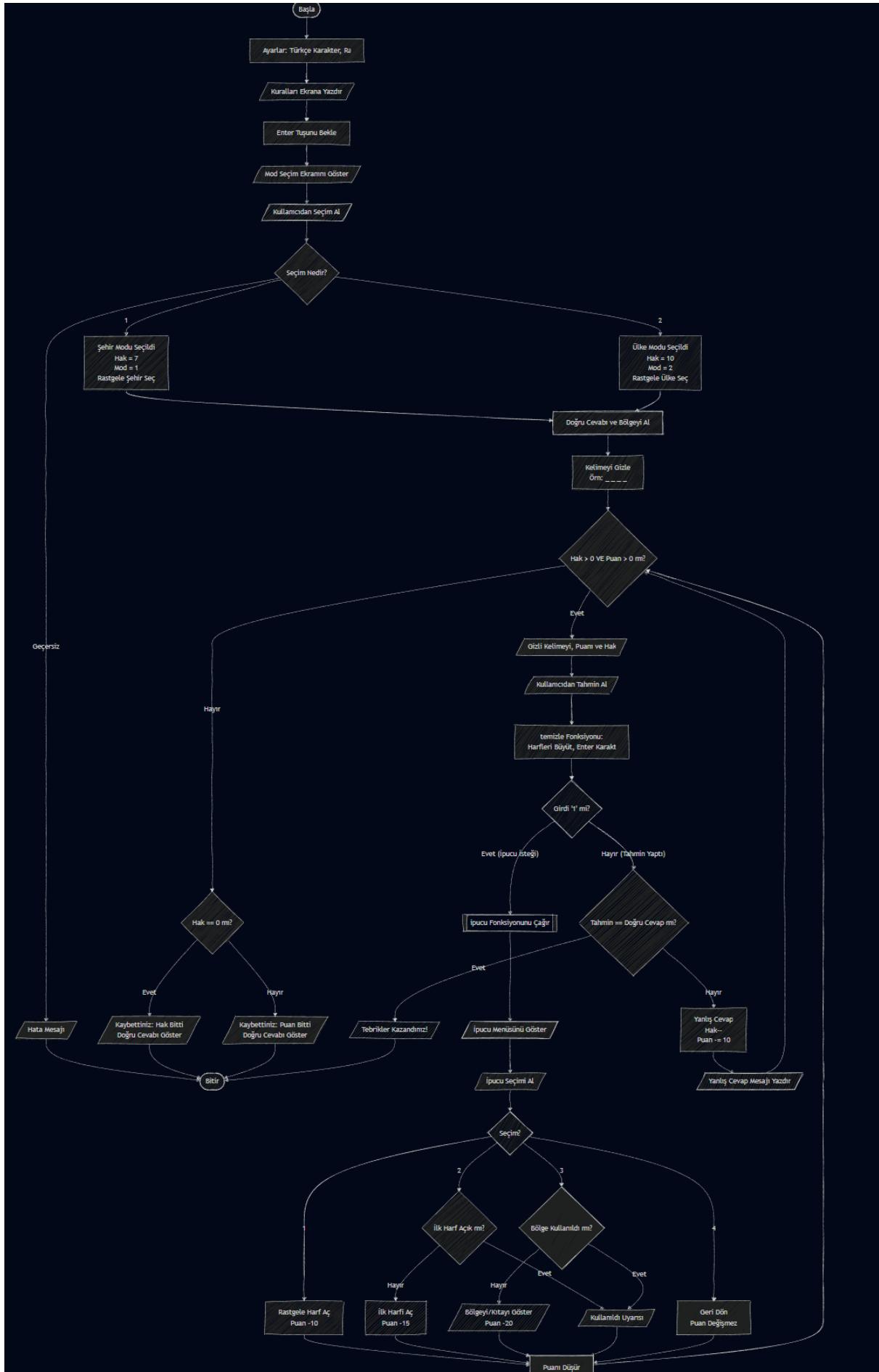
Şehir:_____ (6) Puan: 150 Kalan Hak: 7 İpucu kullanmak için '1' yazıp ENTER'a basınız.

Lütfen tahmininizi yazınız: mardin Yanlış cevap. Tekrar deneyiniz. Puan: 140 Kalan Hak: 6

samsun
Yanlış cevap. Tekrar deneyiniz. Puan: 130 Kalan Hak: 5

ankara
Tebrikler. Doğru cevabı 3. denemede buldunuz. Kazanılan Puan: 130

Akış Diyagramı



Akış diyagramı linki (net hali): <https://bit.ly/4j9FuKI>

- Projemde kullandığım değişkenler, kütüphaneler, fonksiyonlar, diziler, koşullar, döngüler vb. şeyler:

Değişkenler:

- 1) **int secim** : Mod seçimi için kullanılır. Seçilen moda göre bazı değişkenler farklı değer alır.
- 2) **char dogru_cevap[50]** : Rastgele seçilen şehir/ülkenin teknik adını (karşılaştırma için) alır.
- 3) **char dogru[50]** : Rastgele seçilen şehir/ülkenin görsel adını (kullanıcıya göstermek için) alır.
- 4) **char bolge[50]** : Rastgele seçilen şehir/ülkenin bölgesini veya kıtasını alır. (ipucu için)
- 5) **int indeks** : Rastgele bir şehir/ülke seçmek için oluşturulan rastgele sayıyı tutar.
- 6) **int mod** : Hangi modun seçildiğini tutar. (1=Şehir, 2=Ülke)
- 7) **int hak** : Kaç hakkın olacağı belli olduktan sonra (7 veya 10) o değeri alır.
- 8) **int puan** : Toplam puanı alır. (150 Puan)
- 9) **int i** : For döngüsü için kullanılır.
- 10) **int harf** : Seçilen cevap kaç harften oluşuyorsa o değeri alır.
- 11) **int uzunluk** : For döngüsünde koşul için kullanılır. Cevabın uzunluğunu (kaç karakterden oluştuğunu) alır.
- 12) **char gorunen[50]** : Oluşan gizli kelimeyi (Ör: ___) alır. (ipucu fonksiyonunda doğrudan müdahale etmek için)
- 13) **char tahmin[50]** : Oyuncunun tahminini alır. Karşılaştırma kullanmak için kullanılır.
- 14) **int tahmin_uzunluk** : Tahmin kelimesinin uzunluğunu tutar. Temizle fonksiyonunda kullanılır.
- 15) **int a** : Hangi ipucunun seçileceğine karar vermek için kullanılır.
- 16) **int sayı** : 0'dan doğru cevabın karakter sayısına kadar rastgele bir sayı seçilir. Seçilen sayıyı tutar.
- 17) **int p** : Kullanılan ipucunun kaç puan olduğunu belirler. İpucu fonksiyonu çağrııldığında p değişkeni döndürülür ve harcanan puan düşürülür.
- 18) **static int ipucu_2=1** : Bir kere kullanılan ipucu içindir. Kullanıldıktan sonra 0 (sıfır) değerini alır ve bir daha değişmez.
- 19) **static int ipucu_3=1** : Bir kere kullanılan ipucu içindir. Kullanıldıktan sonra 0 (sıfır) değerini alır ve bir daha değişmez.

Kütüphaneler:

- 1) `<stdio.h>` : Temel giriş çıkış işlemleri için kullanıldı.
- 2) `<time.h>` : Srand fonksiyonu için kullanıldı.
- 3) `<stdlib.h>` : Srand fonksiyonu için kullanıldı.
- 4) `<locale.h>` : `setlocale(LC_ALL, "Turkish")` fonksiyonu için kullanıldı.
- 5) `<string.h>` : Dizilerdeki işlemler için kullanıldı.
- 6) `<ctype.h>` : `toupper()` fonksiyonu için kullanıldı.(Tahmin kelimesinin harflerini büyütmek için)
- 7) `"sehirler_ulkeler.h"` : Şehirlerin ve ülkelerin bulunduğu .h dosyasını kullanmak için kullanıldı.

Fonksiyonlar:

- 1) `int main(void)` : Kodumuzun çalıştığı ana fonksiyon.
- 2) `setlocale(LC_ALL, "Turkish")` : Türkçe karakterlerin terminalde doğru gözükmesi için kullanıldı.
- 3) `srand(time(NULL))` : Gerçek rastgele sayı üretmek için kullanıldı.
- 4) `getchar()` : Programın adım adım çalışması için kullanıldı.
- 5) `switch()` : Seçime göre işlemlerin devam etmesi için kullanıldı.
- 6) `fgets(tahmin,50,stdin)` : Oyuncunun tahminini almak için kullanıldı.(scanf yerine bunu kullanmamın sebebi scanf'in boşluk gördüğü yere kadar okuyabilmesi fgets'in ise entera basana kadar olan kısmı alması yüzünden. Çünkü iki ve üç kelimeli ülkeler de var.)
- 7) `strcmp()` : tahmin değişkenini doğru_cevap ile karşılaştırmak için kullanıldı.
- 8) `strcpy()` : String kopyalamak için kullanıldı.
- 9) `strlen()` : doğru_cevap değişkeninin ve tahmin değişkeninin uzunluğunu bulmak için kullanıldı.
- 10) `rastgele()` : Rastgele bir sayı üretmek için kullanıldı.
- 11) `temizle(tahmin)` : Girilen tahmin kelimesinin sondaki fgets kaynaklı \n karakteri \0 ile değiştirildi ve kelimenin bütün harfleri büyütülerek karşılaştırma için uygun hale getirildi.
- 12) `ipucu()` : İpucu sayfasını açmak ve burada seçilen işlemi yapmak, en sonunda da bir puan döndürerek puanın düşürülmesini sağlamak için kullanıldı.

Diziler:

Projemde 2 adet dizi bulunmaktadır.Bunlar .c kodumun dışında sehirler_ulkeler_h kodunda bulunur. Bunlar :

- 1) Sehirler[81]

2) Ulkeler[196]

Ayrıca bu dosyada **struct** veri tepini kullanarak her iki dizimide belli bir düzene göre oluştururdum.

Döngüler ve Koşullar:

- 1) **if(mod == 0) :** 83. satırda bulunan bu koşulda seçim ekranında 1 veya 2 den başka bir seçenek seçilmesi durumunda programın direk sonlandırılmasını sağlıyor.
- 2) **for(i=0;i<uzunluk;i++) :** 92. satırda bulunan bu döngüde doğru kelime gizlenir, alt tire ile ekrana yazdırılır ve kelimenin gizli hali gorunen değişkenine atanır. Ayrıca toplam harf sayısı da bu değişken içinde hesaplanır.
- 3) **while (getchar() != '\n' && getchar() != EOF) :** 109. satırda bulunan bu döngüyü yapay zekadan almış olup kullanma amacım ise önceki getchar kullanımılarından dolayı bellekte biriken enterları silmek. Çünkü bu kod olmadığından direkt 1. tahmini atlayıp puan ve hakkı düşürdükten sonra 2. tahmine direkt olarak geçiyordu.
- 4) **while(hak>0 && puan>0) :** 113. satırda bulunan bu döngü oyunumuzun devam ettiği asıl yerdir. Hak veya puandan biri bitene kadar oyunun devam etmesini sağlar.
- 5) **if(strcmp(tahmin,"1") == 0) :** 117. satırda bulunan bu koşulda kullanıcı tahmin olarak 1 girmiş ve benim kodumda bunun karşılığı ipucu panelinin açılmasıdır. Bu yüzden bu koşul gerçekleştiğinde ipucu fonk. çağrılır ve gerekli işlemler orada yapıldıktan sonra tekrar döngüye geri dönülür.
- 6) **if(strcmp(tahmin,dogruc_cevap) == 0) :** 123. satırda bulunan bu koşulda oyuncu tahmini ile doğru cevap karşılaştırılır. Doğru yapmışsa tebrik mesajı ve kazanılan puan ekrana yazdırılır. Yanlış ise puan ve hakkı düşürülp tekrar denenmesi istenilir.
- 7) **if (x[tahmin_uzunluk - 1] == '\n') :** 168. satırda bulunan bu koşulda fgets kaynaklı alınan tahmin kelimesinin sondaki enter ('\n') karakteri '\0' ile değiştirilir. (Karşılaştırma için)
- 8) **for(j=0;j<tahmin_uzunluk;j++) :** 174. satırda bulunan bu döngüde tahmin kelimesinin her harfi toupper() fonk. ile büyütülür. (Karşılaştırma için)
- 9) **if(ipucu_2 == 1) ve if(ipucu_3 == 1) :** 197. satırda ve 205. satırda bulunan bu koşullar bu ipucunun daha önce kullanılmış kullanılmadığını gösterir.

Switch Case:

- 1) 46. satırda bulunan switch case mod seçimi için kullanılmış bazı değişkenler gereken değerlere atanmıştır. Böylece oyun başlanır.
- 2) 227. satırda bulunan switch case ipucu sayfasında seçilen işlemleri gerçekleştirmek için kullanılmıştır.

Karşılaştığım Zorluklar ve Çözümlerim:

- 1) C dilinin Türkçe harfleri doğru okuyamaması. Özellikle de oyuncudan tahmin almaya çalıştığımda Türkçe karakterleri rastgele sembollerle değiştirmesi idi. Bunun çözümü için kullanıcının tahmin girerken Türkçe karakter kullanmamasını kural olarak ekledim. Ayrıca output ekranında Türkçe karakterlerin doğru gözükmesi için locale.h kütüphanesinden setlocale() fonksiyonunu kullandım. En son sonuç ekranında ise doğru cevabı olması gereği gibi doğru[50] (görsel_ad) değişkeni ile yazdırıldım.
- 2) Projeyi adım adım çalıştmak istiyordum. Bu yüzden getchar fonksiyonunu kullandım ama bu bellekte \n birikmesine neden oluyordu ve ilk tahminde direkt oyuncuyu atlayıp doğru kelime ile \n karakterini karşılaştırıyordu ve bu haliyle yanlış cevap olmasına sebep oluyordu. Bunu düzeltmek için yapay zekadan **while (getchar() != '\n' && getchar() != EOF)** kodunu aldım.
- 3) İpucu fonksiyonu beni biraz zorladı. Bir çok değişkeni çağırırmam gerekiyordu ve haliyle en sonda bir puan düşmesi gerekiyordu. Burada oyuncunun seçeceği ipucu için gerekli kodları yazarken zorlandım. Değişken adlarını olabildiğince koruyarak kodları yazdım. 2. ve 3. ipucu tek kullanımlığı bu yüzden bunları sadece birer kez kullanabilememi sağlayan değişkenleri (ipucu_2, ipucu_3) static int olarak atadım. Böylece ipucu döngüsü bir daha çağrıldığında bu değişkenler önceden kullanılmış ise değeri 0 kalıyor ve oyuncuya bu ipucu kullanılmıştır diyor.
- 4) Bu projede beni en çok zorlayan şey anadolu lisesi çıkışlı olmam. Ben yazılıma merak saldığını için bu bölüm séctim ve ilk kez burada yazılımı görüyorum. Ama okulda öğrenmemiz çok yavaş oluyordu. Haftada yalnızca 2 saatimiz vardı ve benim bir an önce fonksiyonları, dizileri ve stringleri öğrenmem gerekiyordu. Bu yüzden okul dışında gemini ile c dilini öğrenmeye çalıştım. Öğrendim de o bana konuyu anlatıp çeşitli ödevler veriyordu. Daha sonra ben bu ödevleri çözüp ona geri gönderiyordum. Bana hatalarımı ve doğrularımı söylüyordu. Kodlamamın mantığını bana harika bir şekilde anlattı. Projemi ona önceden söylemiştim özellikle stringleri çok iyi öğrenmiştim. Bana lazım olacak fonksiyonların çoğunu kendim yazarak kendimi geliştirdim.

Öğrendiklerim ve Önerilerim:

- Döngüler, koşullar, fonksiyonlar, kütüphaneler, diziler ve stringler başta olmak üzere bir çok şey öğrendim.
- Kod düzenine dikkat ederek, gerektiğinde yorum satırları kullanarak kodun okunabilirliğini yüksek seviyede tuttum.
- Kesinlikle yapay zeka doğru kullanıldığında çok iyi bir eğitici olabiliyor. Bizler yazılımcı olmak istiyoruz. Bu yolda yapay zekayı bir düşman olarak görebilirsiniz ama onu doğru kullanırsanız size zaman kazandırdığına, size bir düşman değil çok iyi bir yardımcı olduğunu gösterecektir.

Gelecekte bir gün yapay zeka tek başına ileri düzey kodlamaları tek başına yapabilir ama bunun için önce birileri tarafından kodlanması gereklidir.