

Ders Adı: **Algoritma ve Programlamaya Giriş**

Proje Adı: **Şehir/Ülke Tahmin Oyunu**

Ad Soyadı: **Hüseyin Dil**

Numara: **360125058**

Tarih: 01.01.2026

Projenin Amacı ve Açıklaması

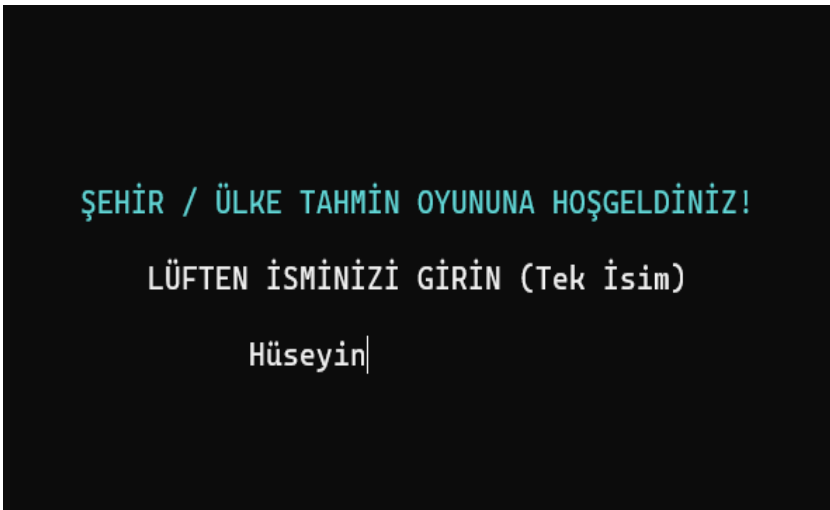
Projem terminalde çalışan görsel olarak şık duran bir şehir/ülke tahmin oyunudur. İki moddan oluşur:

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

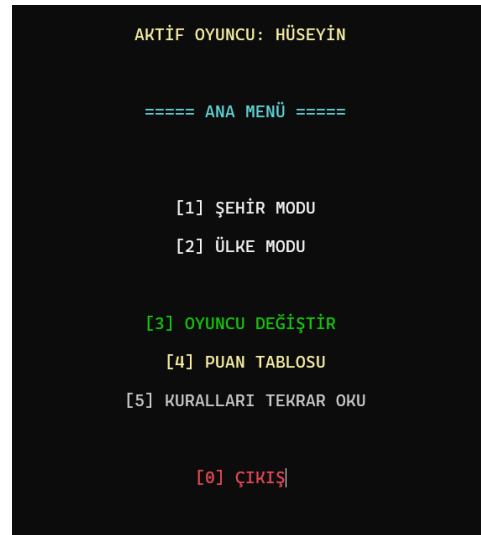
Oyun başlangıçta bir isim ister ve ardından kuralları ekrana yazdırır. Daha sonrasında ise mod seçimi için oyuncuya sorar. Burada istenilirse direk mod seçilip oyuna başlanılabilir veya puan tablosu, oyuncu değişikliği gibi sekmeler açılabilir. Seçilen moda göre toplam hak sayısı değişir. Bunun sebebi 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 ekrana yazdırılır ve kullanıcıdan 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 ipucuyu puan karşılığında kullanabilir.

Aşağıda projenin bazı adımlarının ekran görüntüsü yer almaktadır:

Giriş Ekranı



Ana Menü



Oyun Ekranları

OYUNCU: HÜSEYİN

MOD: ŞEHİR TAHMİNİ

PUAN: 55 | HAK: 5

MAL_TY_ (7 Harf)

(Bölge: Doğu Anadolu)

Tahmininiz (İpucu için '1' yazıp ENTER'a basınız.): |

YANLIŞ TAHMİNLER:
ANTALYA - TRABZON -

TEBRİKLER KAZANDINIZ!

Cevap: Malatya | Kazanılan Puan: 55

Devam etmek için bir tuşa basınız.|

OYUNCU: HÜSEYİN

MOD: ÜLKE TAHMİNİ

PUAN: 150 | HAK: 10

--- (14 Harf)

Tahmininiz (İpucu için '1' yazıp ENTER'a basınız.): |

TEBRİKLER KAZANDINIZ!

Cevap: Çek Cumhuriyeti | Kazanılan Puan: 150

Devam etmek için bir tuşa basınız.|

İpucu Ekranı

--- İPUCU MARKETİ ---

[1] Rastgele Harf Aç (10 Puan)

[2] İlk Harfi Göster (15 Puan)

[3] Bölgeyi Göster (20 Puan)

[4] İptal / Geri Dön|

Puan Tablosu

===== PUAN TABLOSU =====

İSİM	ŞEHİR (Oyun/Puan)	ÜLKE (Oyun/Puan)	TOPLAM PUAN
HÜSEYİN	5 / 450	1 / 115	565
YUSUF	1 / 50	0 / 0	50
EMRE	2 / 165	0 / 0	165
ZORO	10 / 715	0 / 0	715
HALİT	1 / 105	0 / 0	105
DJHDJK	1 / 95	0 / 0	95
ALİ	1 / 120	0 / 0	120

Menüye dönmek için herhangi bir tuşa basın...|

=====

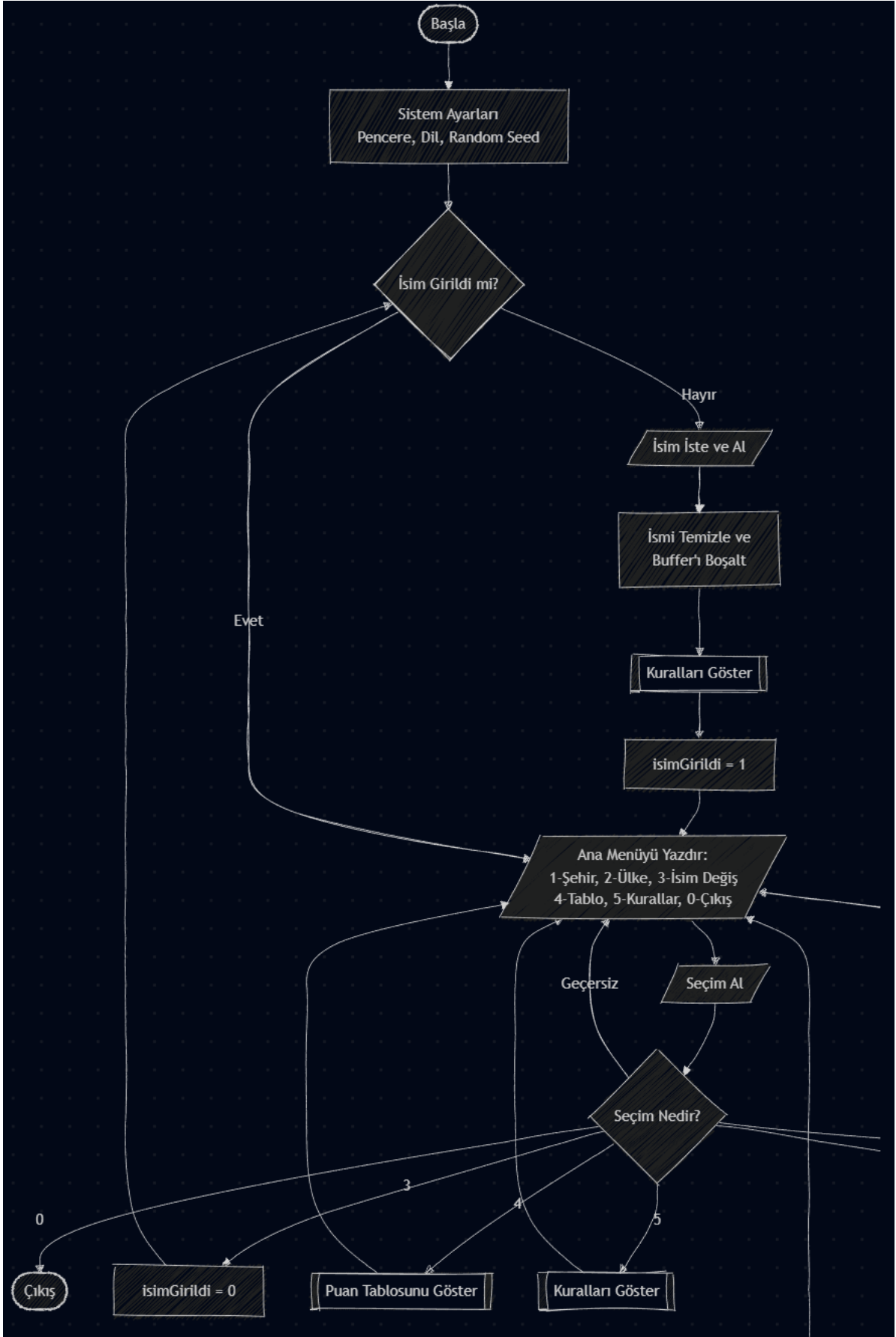
OYUN KURALLARI VE BİLGİLERİ

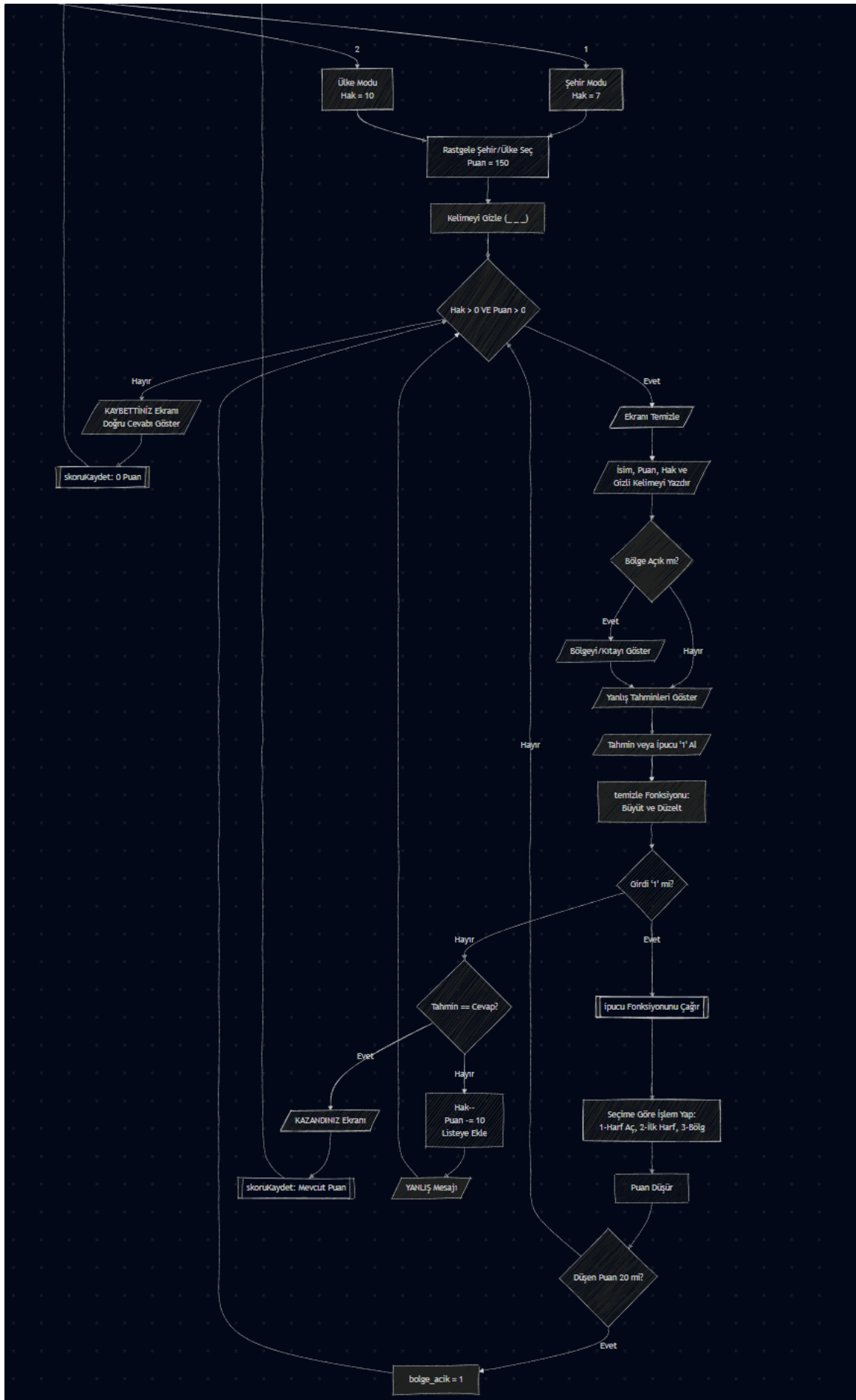
=====

1. Her oyuna 150 PUAN ile başlarsınız.
2. Yanlış tahminlerinizde 10 PUAN ve 1 HAK kaybedersiniz.
3. Şehir modunda 7, Ülke modunda 10 hakkınız vardır.
4. İpucu marketinden puan karşılığı yardım alabilirsiniz.
5. Puanınız veya Hakkınız biterse oyun sona erer.
6. Kazandığınız puanlar mod bazlı olarak Puan Tablosuna kaydedilir.

Devam etmek için bir tuşa basın...|

Akış Diyagramları





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

Değişkenler:

- 1) **char oyuncuismi[50]** : Oyunu oynayacak kişinin ismi burada tutulur.
- 2) **int isimGirildi** : İsim girilip girilmediğini kontrol eder. Ayrıca yeni isim girilmek istenilirse kullanılır.
- 3) **char isimBar[50]** : Ana ekranda ve oyun ekranında en üstte oyuncu ismini göstermek için kullanılır.
- 4) **int secim** : Mod seçimi için kullanılır. Seçilen moda göre bazı değişkenler farklı değer alır. Oyuna geçilir veya farklı bir sekme (puan tablosu, isim değiştirme) açılır.
- 5) **char dogru_cevap[50]** : Rastgele seçilen şehir/ülkenin teknik adını (karşılaştırma için) alır.
- 6) **char dogru[50]** : Rastgele seçilen şehir/ülkenin görsel adını (kullanıcıya göstermek için) alır.
- 7) **char bolge[50]** : Rastgele seçilen şehir/ülkenin bölgesini veya kıtasını alır. (ipucu için)
- 8) **int idx** : Rastgele bir şehir/ülke seçmek için oluşturulan rastgele sayıyı tutar.
- 9) **int mod** : Hangi modun seçildiğini tutar. (1=Şehir, 2=Ülke)
- 10) **int hak** : Kaç hakkın olacağı belli olduktan sonra (7 veya 10) o değeri alır.
- 11) **int puan** : Toplam puanı alır. (150 Puan)
- 12) **int i, j, k** : For döngüsü için kullanılır.
- 13) **int harf** : Seçilen cevap kaç harften oluşuyorsa o değeri alır.
- 14) **int uzunluk** : For döngüsünde koşul için kullanılır. Cevabın uzunluğunu (kaç karakterden oluştuğunu) alır.
- 15) **char gorunen[50]** : Oluşan gizli kelimeyi (Ör: _ _ _ _) alır. (ipucu fonksiyonunda doğrudan müdahale etmek için)
- 16) **char hatali_tahminler[1000]** : Yanlış cevapları tutar. Yanlış cevapları ekrana yazdırmak için kullanılır.
- 17) **int ip1 = 1, ip2 = 1, ip3 = 1** : İpucuların kullanılıp kullanılmadığını belirler. Kullanılan ipucu tükendiğinde 0 değerini alır.
- 18) **int bolge_acik** : bölge ipucusunu ekrana göstermek için kullanılır. Başlangıçta değeri 0'dır.
- 19) **char istatistik[100]** : Kalan puan ve hakkı ekrana yazdırmak için kullanılır.
- 20) **char gizli[100]** : Gizlenen doğru kelimeyi ve harf sayısını (Ör: _ _ _ _ (4)) ekrana yazdırmak için kullanılır.
- 21) **char tahmin[50]** : Oyuncunun tahminini alır. Karşılaştırma kullanmak için kullanılır.

- 22) **int tahmin_uzunluk** : Tahmin kelimesinin uzunluğunu tutar. Temizle fonksiyonunda kullanılır.
- 23) **char final_mesaj[100]** : Doğru cevap ve kazanılan puanı ekrana yazdırmak için kullanılır.
- 24) **char kayip_mesaj[100]** : Hakkın veya puanın bitmesi durumunda doğru cevabı yazdırmak için kullanılır.
- 25) **COORD coord** : COORD tipinde coord isimli bir değişken oluşturur. Bu değişkenin içinde X ve Y adında iki tane tam sayı alanı bulunur. gotoxy(x,y) fonksiyonunda yer alır.
- 26) **coord.X** : Ekranın yatay eksenindeki (soldan sağa) konumunu belirler. x değeri büyüdükçe imleç sağa gider. Genel olarak kodumuzda yazılar ortalı olarak yazdırıldığı için x'e pek fazla müdahale etmeyiz.
- 27) **int uz** : Ekran Genişliğinden (Bizim kodda 160) yazacağımız cümlelerin uzunluğu (uz) çıkarılır ve ikiye bölünür. Böylece x'in başlangıç yeri her cümle için ortalı olur. ortayaYaz() fonksiyonunda yer alır.
- 28) **coord.Y** : Ekranın dikey eksenindeki (yukarıdan aşağıya) konumunu belirler. y değeri büyüdükçe imleç aşağı iner. Yazdırmak istediğimiz cümleyi yazacağımız yere göre ortayaYaz fonksiyonuna bir sayı gönderir.
- 29) **int sayac** : Puan tablosunda kaç kişinin olduğunu bulmak için kullanılır.
- 30) **int bulundu** : Oyuncu isim girdiğinde girilen ismin daha önce oynayıp oynamadığını (.txt dosyasında olup olmadığını) kontrol etmek için kullanılır.
- 31) **char baslik[100]** : Puan tablosundaki başlığı içinde tutar. Burada kategoriler yazılmıştır.
- 32) **int baslikUzunluk** : Başlığın uzunluğuna göre altta gelecek isimlerin ve bilgilerin aynı hizada göstermek için kullanılır. x koordinatının manuel olarak değiştirdiğim tek yerdir.
- 33) **int tabloBaslangicX** : x'in başlayacağı yerin sayısıdır. Diğer bir değişle imlecin bulunacağı konumunun apsisiidir.
- 34) **char isim[50]** : .txt dosyasındaki ismi alır. Puan tablosuna yazdırmak için kullanılır.
- 35) **int sOyun, sPuan, uOyun, uPuan** : .txt dosyasındaki değerleri alırlar. Puan tablosuna yazdırılmak için kullanılırlar..
- 36) **int toplam** : Şehir ve ülke puanının toplamı burada bulunur. Puan tablosuna yazdırmak için kullanılır.
- 37) **int u** : temizle() fonksiyonunun içine gönderilen kelimenin veya cümlelerin uzunluğunu alır. Buradaki işlemler için kullanılır.
- 38) **char buffer[100]** : İpucu fonksiyonunda ekrana yazdırılacak seçenekleri içine alır.
- 39) **int a** : İpucu fonksiyonunda hangi ipucunun seçildiğini anlamak için kullanılır.
- 40) **int p** : İpucu fonksiyonunda harcanan puanı toplam puandan düşürmek için kullanılır. Başlangıçta 0'dır.
- 41) **int var_mi** : İlk ipucuda açılmamış harfin olup olmadığını kontrol eder.

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) `#include <windows.h>` : İmleci istediğim yere götürebilmek ve yazı renklerini ayarlamak için kullanıldı. Ayrıca system() ve Sleep() fonksiyonlarını bu kütüphane sayesinde kullanabildim.
- 8) `#include <conio.h>` : getch() fonksiyonu için kullanıldı.
- 9) `"sehirler_ulkeler.h"` : Şehirlerin ve ülkelerin bulunduğu .h dosyasını kullanmak için kullanıldı.
- 10) `#define EKRAN_GENISLIGI 160` : Ekranın yatay uzunluğunu sabit tutmak için kullanıldı.

Fonksiyonlar:

- 1) `int main(void)` : Kodumuzun çalıştığı ana fonksiyon.
- 2) `system("mode con: cols=160 lines=35")` : Terminalin yatay ve dikey uzunluğunu sabit tutmak için kullanıldı.
- 3) `system("chcp 1254")` : Türkçe karakterlerin terminalde doğru gözükmeleri için kullanıldı.
- 4) `system("cls")` : Belli yerlerde ekranı temizlemek için kullanıldı.
- 5) `setlocale(LC_ALL, "Turkish")` : Türkçe karakterlerin terminalde doğru gözükmeleri için kullanıldı.
- 6) `srand(time(NULL))` : Gerçek rastgele sayı üretmek için kullanıldı.
- 7) `getch()` : Programın hızlı çalışması için kullanıldı.
- 8) `switch()` : Seçime göre işlemlerin devam etmesi için kullanıldı.
`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.)
- 9) `strcmp()` : tahmin değişkenini dogru_cevap ile karşılaştırmak ve isim değişkenininin .txt de dosyasına olup olmadığını karşılaştırmak için kullanıldı.
- 10) `strcpy()` : String kopyalamak için kullanıldı.
- 11) `strlen()` : dogru_cevap değişkeninin ve tahmin değişkeninin uzunluğunu bulmak için kullanıldı.
- 12) `strcat()` : Yanlış tahminleri `hatali_tahminler[1000]` değişkenine eklemek için kullanıldı.

- 13) **sprintf()** : Değişkenli bir diziye bir arada tutmak ve ortayaYaz fonksiyonuna göndermek için kullanıldı. printf() ile farkı printf() değişkenli diziye terminale yazarken sprintf bu diziye değişikene yazar.
- 14) **fscanf()** : .txt dosyasındaki bilgileri Oyuncu dizisine yazdırmak için kullanıldı.
- 15) **void ortayaYaz()** : Girilen metni belli bir yükseklikte (y ekseninde) belli bir renkle ortaya yazmak için kullanıldı.
- 16) **void gotoxy()** : İmlecin ışınlanacağı yeri belirlemek için kullanıldı.Genel olarak ortayaYaz() fonksiyonunda kullanıldı.
- 17) **void renkDegistir()** : Yazının rengini değiştirmek için kullanıldı.Genel olarak ortayaYaz() fonksiyonunda kullanıldı.
- 18) **void kurallariGoster()** : Kurallar sayfasını açmak için kullanıldı.
- 19) **void puanTablosunuGoster()** : Puan tablosunu açmak için kullanıldı.
- 20) **void skoruKaydet()** : Oyun sonunda kullanıcının adını ve kazandığı puanı .txt dosyasına kaydetmek için kullanıldı.
- 21) **rastgele()** : Rastgele bir sayı üretmek için kullanıldı.
- 22) **temizle(tahmin)** : Girilen tahmin kelimesinin sonundaki fgets kaynaklı \n karakteri \0 ile değiştirildi ve kelimenin bütün harfleri büyütülerek karşılaştırma için uygun hale getirildi. Ayrıca isim değişkenindeki bütün harfleri büyütmek için kullanıldı.
- 23) **ipucu()** : İpucu sayfasını açmak ve burada seçilen işlemi yapmak, en sonunda da bir puan döndürerek puanın düşürülmesini sağlamak için kullanıldı.

Diziler:

Projemde 4 adet dizi bulunmaktadır.Bunlar .c kodumun dışında sehirler_ulkeler_h kodunda bulunur. Bunlar :

- 1) **Sehirler[81]** : Şehirlerin tutulduğu dizi.
- 2) **Ulkeler[196]** : Ülkelerin tutulduğu dizi.
- 3) **oyuncular** : Oyunu oynayan kişilerin isimlerinin ve istatistiklerinin .txt de dosyasından kopyalanıp tutulduğu dizi.
- 4) **hatali_tahminler[1000]** : Oyun sırasında yapılan yanlış tahminleri ekrana yazdırmak için kullanıldı.

Ayrıca bu dosyada **struct** veri tipini kullanarak Sehirler ve Ulkeler dizisini Duzen altında, oyuncular dizisini ise Oyuncu altında belli bir yapıya göre tuttum.

Döngüler ve Koşullar:

- 1) **while (1)** : 34. satırda bulunan bu döngüde oyunun sürekli olarak devam edilmesi sağlanmıştır.
- 2) **if (isimGirildi == '0')** : 36. satırda bulunan bu koşulda isim girilip girilmediği kontrol ediliyor.

- 3) **if (secim == '0') :** 66. satırda bulunan ve devamında 68, 69 ve 70. satırda devam eden if koşulları seçilen tercihe göre programı ya tamamen bitirir ya da gerekli fonksiyonlar ekrana gösterilir ve programın ana ekranına yani tercih kısmına geri döner.
- 4) **if (secim == '1') :** 75. satırda bulunan bu koşul ile şehir modu seçilir ve oyunun devamı için bazı değişkenler gerekli değerleri alır.
- 5) **else if (secim == '2') :** 83. satırda bulunan bu koşul ile ülke modu seçilir ve oyunun devamı için bazı değişkenler gerekli değerleri alır.
- 6) **else continue;** 91. satırda bulunan bu koşul farklı bir seçeneğin seçilmesi durumunda ana ekrana geri döner.
- 7) **for(i=0;i<uzunluk;i++) :** 97. satırda bulunan bu döngüde doğru kelime gizlenir ve kelimenin gizli hali görünen değişkenine atanır. Ayrıca toplam harf sayısı da bu değişken içinde hesaplanır.
- 8) **while(hak>0 && puan>0) :** 108. satırda bulunan bu döngü oyunumuzun devam ettiği asıl yerdir. Hak veya puandan biri bitene kadar oyunun devam etmesini sağlar.
- 9) **if(bolge_acik) :** 130. satırda bulunan bu koşulda bölge ipucusu kullanılmışsa bölge veya kıta ekrana yazdırılır.
- 10) **if(strlen(hatali_tahminler) > 0) :** 142. satırda bulunan bu koşulda hatalı bir kelime veya kelimeler girilmişse bunlar ekrana yazdırılır.
- 11) **if(strcmp(tahmin,"1") == 0) :** 156. satırda bulunan bu koşulda kullanıcı tahmin olarak 1 girmiş ve benim kodumda bunun karşılığı ipucu panelinin açılmasıdır. Bu yüzden bu koşul gerçekleştiğinde ipucu fonksiyonu çağrılır ve gerekli işlemler orada yapıldıktan sonra tekrar döngüye geri dönlür.
- 12) **else if(strcmp(tahmin, dogru_cevap) == 0) :** 164. satırda bulunan bu koşulda oyuncu tahmini ile doğru cevap karşılaştırılır. Doğru yapmışsa tebrik mesajı ve kazanılan puan ekrana yazdırılır. Yanlış ise puan ve hakkı düşürülüp tekrar denemesi istenilir.
- 13) **if(hak <= 0 || puan <= 0) :** 188. satırda bulunan bu koşulda hak veya puandan biri bitmiş olup oyuncu kaybetmiştir. Kaybetme mesajı ve doğru cevap yazıldıktan sonra oyuncu ismi ve istatistikleri .txt dosyasına kaydedilir.
- 14) **if (dosya != NULL) :** 285. satırda bulunan bu kodda dosya okuma modunda açıldığı için dosya varsa bütün bilgiler işlem yapılabilmesi için oyuncular dizisine çekilir(280. satırdaki while döngüsü ile).
- 15) **if (bulundu == 0) :** 305. satırda bulunan bu koşulda oyuncu ismi oyuncular dizisinde yoksa kaydedilir.
- 16) **if (dosya != NULL) :** 323. satırda bulunan bu koşulda dosya yazma modunda açılır. Eğer dosya yoksa açılır varsa eğer içindekiler silinir ve oyuncular dizisi buraya baştan yazdırılır.
- 17) **if (dosya == NULL) :** 354. satırda bulunan bu koşulda henüz bir dosya (oyuncu) olmadığı için puan tablosu gösterilmiyor. Eğer varsa hemen ardından gelen else koşulu çalışıyor.

- 18) `while (fscanf(dosya, "%s %d %d %d %d", isim, &sOyun, &sPuan, &uOyun, &uPuan) != EOF) :`
359. satırda bulunan bu döngüde puan tablosuna .txt dosyasındaki bilgiler yazılması için bütün bilgiler belirlenen değişkenlere kaydediliyor ve devamında ekrana belli bir düzende yazdırılıyor.
- 19) `if (x[u- 1] == '\n') :` 384. satırda bulunan bu koşulda `fgets` kaynaklı alınan tahmin kelimesinin sonundaki enter (`\n`) karakteri `'\0'` ile değiştirilir. (Karşılaştırma için)
- 20) `for(j=0;j<u;j++) :` 389. satırda bulunan bu döngüde tahmin kelimesinin her harfi `toupper()` fonk. ile büyütülür. (Karşılaştırma için)

Switch Case:

- 1) 450.satırda bulunan switch case ipucu sayfasında seçilen işlemi gerçekleştirmek için kullanılmıştır.

Karşılaştığım Zorluklar ve Çözümlerim:

- Oyunu belli bir çerçevede ve estetikde durması için `windows.h` kütüphanesi kullandım. Burada yaptığım şeyi oturtmak benim baya bi zamanımı aldı. Yer yer yapay zekadan yardım aldım. Nasıl yaptığıma gelirse `gotoxy()` fonksiyonu ile imleci istediğim yere ışınladım. Bunu yapmak için terminali 160,35 olarak ayarladım ve imleci ışınlamak istediğim yerin koordinatını x,y olarak buraya gönderiyordum. Böylece yazılacak olan yazının başlangıç yeri belli oluyordu. Kodumda genel olarak x'e müdahale edilmemiştir. Ekran genişliği'nden (160) yazılacak olan metnin uzunluğu çıkarılıp ikiye bölünerek o metin için imleci ekranın tam ortasına ortalı olarak yazılacağı yere ışınılıyordu. Burada daha çok y ile metnin yazılacağı kısma karar verdim. Ayrıca her metin aynı renkte yazılmadı. Bunun için yine `windows.h` kütüphanesindeki bir fonksiyonu kullandım. 0'dan 15'e kadar renklerin kodu var ve istediğim rengin kodunu girerek sonraki yazılacak metnin rengine karar verebildim. Bu ikisini bir arada ve daha düzenli daha seri kullanabilmek için ise `ortayaYaz()` fonksiyonunu kullandım.
- `ortayaYaz()` fonksiyonunu açıklayacak olursam, başta yazılacak olan metin veya değişken sonrasında yazılacağı yükseklik en sonda ise yazdırılacağı rengin kodunu girerek daha hızlı ve daha kısa bir kod yazdım.
- Bu projede beni en çok zorlayan şey anadolu lisesi çıkışlı olmam. Ben yazılıma merak saldığım için bu bölümü seçtim 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. Kodlamanın mantığını bana harika bir şekilde anlattı. 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 gerekir.

Kullandığım Siteler

- **Gemini:** h. kodunun büyük bir kısmını ona yazdırdım. Ayrıca döngülerden pointerlara kadar bir çok konuyu öğrenmek ve bunu pekiştirmek için bana ödev vermesini isteyip bu ödevleri çözüp ona tekrar attım. Böylece konuları hızlı bir şekilde öğrenmeme yardımcı oldu.
- **Mermaid Live Editor:** Akış şemasını hazırlamak için kullandım.