



Python ile İngilizce Kelime Öğrenme Oyunu

Yazılım Mühendisliği Ana Bilim Dalı

Bitirme projesisiz Yüksek Lisans / Bitirme Projesinin Raporu

Merve GÜL

Y210234087

Proje Danışmanı: Prof. Dr. Femin Yalçın Küçükbayrak

Haziran 2023

Yazarlık Beyanı

Ben, **Merve GÜL** başlığı **Python ile İngilizce Kelime Öğrenme Oyunu** olan bu Bitirme projesinin ve Bitirme projesinin içinde sunulan bilgilerin şahsıma ait olduğunu beyan ederim. Ayrıca:

- Bu çalışmanın bütünü veya esası bu üniversitede Yüksek Lisans / Doktora derecesi elde etmek üzere çalıştığım süre içinde gerçekleştirilmiştir.
- Daha önce bu Bitirme projesinin herhangi bir kısmı başka bir derece veya yeterlik almak üzere bu üniversiteye veya başka bir kuruma sunulduysa bu açık biçimde ifade edilmiştir.
- Başkalarının yayımlanmış çalışmalarına başvurduğum durumlarda bu çalışmalara açık biçimde atıfta bulundum.
- Başkalarının çalışmalarından alıntıladığımda kaynağı her zaman belirttim. Bitirme projesinin bu alıntılar dışında kalan kısmı tümüyle benim kendi çalışmamdır.
- Kayda değer yardım aldığım bütün kaynaklara teşekkür ettim.
- Bitirme projesinde başkalarıyla birlikte gerçekleştirilen çalışmalar varsa onların katkısını ve kendi yaptıklarımı tam olarak açıkladım.

Tarih:

24.06.2023

Python ile İngilizce Kelime Öğrenme Oyunu

ÖZ

Bu çalışmada günümüzde oldukça önem arz eden bir konu üzerine çalışıldı. Okullarımızda, gündelik hayatımızda veya iş hayatımızda İngilizce bilmek büyük bir avantajdır. İngilizce kelime öğrenme oyunu projesinde kelimeleri öğrenme ve öğrenilen kelimelerin unutulmamasını hedefler. Beynimizde ki tür bellek sistemi vardır. Bunlar; uzamsal bellek sistemi ve ezberleyerek öğrenme sistemidir. Projenin bir diğer hedefi ise kullanıcının karşısına rasgele kelimeler çıkararak ezberleyerek öğrenme sistemi yerine uzamsal bellek sistemini kullanmaktır. Projede programlama dili olarak Python programlama dili kullanılmıştır.

Anahtar Sözcükler: Mühendislik, teknoloji, analitik yöntem, fen bilimleri, araştırma, Kelime Öğrenme, Python, Oyun , Programlama

Teşekkür

Yüksek lisans çalışma sürecimde değerli katkıları ve desteklerini esirgemeyen, değerli hocam ve sayın danışmanım Sayın Prof. Dr. Femin Yalçın Küçükbayrak hocama teşekkürlerimi ve şükranlarımı sunarım.

İçindekiler

Yazarlık Beyanı	2
Öz	3
Teşekkür.....	4
Şekiller Listesi.....	6
1 Giriş	8
2 Yöntem	9
2.1 Kullanılan programlama dili, kütüphaneler ve araçlar.....	9
2.1.1 Python Programlama Dili	9
2.1.2 Projede Kullanılan Kütüphaneler	11
2.1.2.1 Pandas Kütüphanesi	11
2.2.2.2 Random Kütüphanesi	13
2.1.3 Projede Kullanılan Araçlar:Anaconda Navigator.....	14
2.2 Kelimeler veri setinin araştırılması ve hazırlanması.....	16
3.2.2 Proje Kodları	
3 Veri Seti ve Kodlar	18
3.1 Veri Seti	18
3.1.1 Veri Seti İncelemesi.....	18
3.1.2 Veri Setinin Ortama Yüklenmesi ve Okunması	18
3.1.3 Gerekli Kütüphanelerin Ortama Yüklenmesi	19
3.2 Kodlar	19
3.2.1 Proje Kodları Açıklamaları.....	19
3.2.2 Proje Kodları	26
Kaynaklar	36
Özgeçmiş	38

Şekiller Listesi

Şekil 1:Python dili resmi logosu	10
Şekil 2: Jupiter ve Anaconda Navigator [8].....	15
Şekil 3: Veri setinde rasgele 20 kelime.....	17
Şekil 4:Verimizin kaç satır ve sütundan oluştuğunu gösterir. İlk yazan satır sayısı, ikincisi ise sütun sayısıdır.	18
Şekil 5: Veri Setinin Ortama Yüklenmesi.....	18
Şekil 6: Kütüphanelerin İport Edilmesi	19
Şekil 7:Dosya Hata Kontrolü Yapan Kod.....	19
Şekil 8:Tablodaki Boş Gelen Puanları Ekler	19
Şekil 9:Son Puanı Kaydeden Fonksiyon	20
Şekil 10: Test Sonucu	20
Şekil 11:Puanların Sıfırlanmış Olmasını Denetler	21
Şekil 12: Kullanıcının Seçim Yapmasını Sağlayan Döngü ve Çıktısı	21
Şekil 13:Eğer R seçilirse	22
Şekil 14:Girilen Sayı Kadar Kelime Seçer.....	23
Şekil 15: Döngü İçerisindeki Kontroller	23
Şekil 16: Kodların Tam Hali	25

Bölüm 1

Giriş

Bu “Python ile İngilizce Kelime Öğrenme Oyunu” nun amacı, nitelikli dil öğrenmeye yardımcı olmaktır. Dil öğrenme geçmişten günümüze eğitim sisteminde hep üzerinde düşünülen ve önem verilen bir alan olmuştur. Ülkemizde 4. Ve 5. sınıflarda en az bir yabancı dil eğitimine başlanıyor olmasına rağmen sayıları sınırlı bazı okullar dışında, istenilen düzeyde bir yabancı dil öğretimi gerçekleştirilememekte ve bu alanda çeşitli sorunlar yaşanmaktadır. Tabii bu sorunu tek bir nedenle açıklanamayacak kadar önemli tek bir yöntemle çözülmeyecek kadar kapsamlıdır.

Öğrenme her toplumda önemlidir. Öneminden dolayı çoğu insan bu öğrenme sürecin çok ciddi, durgun ve zamana yayılan bir şekilde yürütülmesi gerektiğini düşünmektedir. Oysa psikoloji, fizyoloji, sosyoloji ve iletişim alanlarında yapılan çalışmalar eğlenceli ve keyifli ortamların da yaşamda ve sosyal etkileşimde önemli bir yeri olduğunu ortaya koymuştur. Nitekim eğlenerek öğrenmenin örneklerini günümüz teknolojisi ile görmekteyiz.

Oyunun yabancı dil öğreniminde başarıyı sağladığına ilişkin literatürde çeşitli araştırmalar bulunmaktadır. Aksungur (1997)’un Present Perfect Tense’in öğretiminde oyunu kullanarak yaptığı deneysel çalışmada, oyun ile öğretimde öğrencilerin daha başarılı olduğu belirlenmiştir. Uberman (1998) ile Huyen ve Nga (2003) tarafından yapılan çalışmalarda da İngilizce sözcük öğreniminde, oyun ile öğretimin öğrenci başarısı üzerinde, geleneksel uygulamalara göre daha etkili olduğu belirlenmiş ve daha başarılı bulunmuştur. [1]

Bölüm 2

Yöntem

Bu bölümde,

- 1)Kullanılan programlama dili, kütüphaneler ve araçlar
- 2) Kelime veri setinin hazırlanışı gibi detaylar başlıklar halinde verilecektir.

2.1 Kullanılan programlama dili, kütüphaneler ve araçlar

2.1.1 Python Programlama Dili

Python, genel amaçlı, yüksek seviyeli, kolay okunabilir ve anlaşılır bir programlama dilidir. Python, basit ve temiz sözdizimiyle bilinen bir dil olup, hızlı prototipleme, veri analizi, yapay zekâ, web geliştirme, oyun geliştirme ve bilimsel hesaplama gibi birçok alanda kullanılmaktadır. Python'un bazı özellikleri şunlardır:

Sözdizimi; Python, diğer programlama dillerine kıyasla daha az karmaşık bir sözdizimine sahiptir. İndentasyon (girintileme) kullanarak blokları belirtir ve kodun okunabilirliğini artırır. Kolay Öğrenme: Python, basit ve anlaşılır yapısıyla öğrenmesi ve anlaması kolay bir dildir. Genel olarak, diğer dillere kıyasla daha az kod yazmanız gerekmektedir.

Büyük Standart Kütüphane: Python, zengin bir standart kütüphaneye sahiptir. Bu kütüphane, birçok işlevi hızlı bir şekilde gerçekleştirmenize olanak tanır ve geniş bir

işlevsellik yelpazesini içerir. Python, birçok özel amaçlı kütüphane ve modül içeren geniş bir ekosisteme sahiptir. Örneğin, veri analizi için Pandas, bilimsel hesaplamalar için NumPy, yapay zeka için TensorFlow veya PyTorch gibi kütüphaneleri kullanabilirsiniz. Bu kütüphaneler, önceden yazılmış kodları kullanmanıza ve işleri daha hızlı bir şekilde gerçekleştirmenize olanak sağlar. Çapraz Platform Desteği: Python, çeşitli işletim sistemlerinde (Windows, macOS, Linux) çalışabilir. Bu, geliştirilen uygulamaların farklı platformlarda sorunsuz bir şekilde çalışabilmesini sağlar. Geniş Topluluk Desteği (Taşınabilirlik): Python, büyük bir açık kaynak topluluğuna sahiptir. Bu topluluk, çeşitli sorunları çözmeye, kaynaklara erişmeye ve bilgi alışverişinde bulunmanıza yardımcı olur.



Şekil 1:Python dili resmi logosu

Nesne Yönelimli: Python, nesne yönelimli programlamayı destekler. Bu programlama paradigmalarıyla çalışmanıza olanak tanır ve kodunuzun daha modüler, yeniden kullanılabilir ve sürdürülebilir olmasını sağlar. Dinamik Tip Sistemi: Python, dinamik bir tip sistemine sahiptir. Bu, değişkenlerin türlerinin çalışma zamanında belirlendiği anlamına gelir. Bu, programlama esnekliği sağlar, ancak hataların çalışma zamanında

ortaya çıkmasına da neden olabilir. Hızlı Prototipleme: Python, hızlı prototipleme için ideal bir dil olarak bilinir. Kolay okunabilirlik ve yazılabilirlik özellikleri sayesinde fikirlerinizi hızlı bir şekilde kod haline getirebilir ve deneyebilirsiniz. Bu da geliştirme sürecini hızlandırır. Python, geniş bir kullanıcı tabanına sahip olduğu için çeşitli kaynaklara ve öğrenme materyallerine kolayca erişebilirsiniz. Python ile ilgili belgeler, kitaplar, çevrimiçi kurslar ve topluluk forumları ile oldukça geniş bir bilgi kaynağına sahiptir. Öğrenmesi kolay ve kullanımı esnek bir programlama dilidir. Kapsamlı bir standart kütüphaneye sahip olması, geniş topluluk desteği ve popülerlik gibi faktörler de Python'u tercih edilen bir dil haline getirmiştir. [2] [3] [4]

2.1.2 Projede Kullanılan Kütüphaneler

2.1.2.1 Pandas Kütüphanesi

Pandas, Python programlama dilinde veri manipülasyonu ve veri analizi için kullanılan güçlü bir kütüphanedir. Pandas, veri setlerini işlemek, dönüştürmek, filtrelemek, gruplamak, birleştirmek ve analiz etmek için bir dizi yüksek seviyeli veri yapıları ve veri analizi araçları sağlar. Pandas kütüphanesinin bazı temel özellikleri:

Veri Yapıları: Pandas, veri manipülasyonu için temel veri yapıları olan DataFrame ve Series'i sağlar. DataFrame, tablo benzeri bir yapıdır ve satır ve sütunlardan oluşur. Series ise tek boyutlu etiketli bir veri dizisidir. Bu veri yapıları, verileri kolayca işlemenizi, dönüştürmenizi ve analiz etmenizi sağlar. **Veri Okuma ve Yazma:** Pandas, farklı veri formatlarını okuyabilme ve yazabilme yeteneğine sahiptir. CSV, Excel, SQL veritabanları, JSON, HTML ve daha birçok formatla uyumlu çalışabilir. Bu sayede veri kaynaklarınızı kolayca okuyabilir ve verilerinizi farklı formatlarda kaydedebilmemizi sağlar.

Veri Manipülasyonu: Pandas, veri setlerini filtreleme, dönüştürme, yeniden şekillendirme ve birleştirme gibi işlemleri yapmak için bir dizi işlev ve yöntem sağlar. Sütun ve satırları seçme, ekleme, silme ve güncelleme işlemlerini kolaylıkla

gerçekleştirilebilmektedir. Veri Analizi: Pandas, veri analizi için birçok istatistiksel ve hesaplamalı işlemi kolayca gerçekleştirebileceğiniz işlemlere sahiptir. İstatistiksel özetler, gruplama, birleştirme, sıralama, sorgulama ve pivotlama gibi işlemler yapılabilir. Veri Görselleştirme: Pandas, Matplotlib veya Seaborn gibi görselleştirme kütüphaneleriyle entegre çalışır. Bu sayede verilerinizi grafikler, histogramlar, kutu grafikleri ve diğer görsel öğelerle kolayca görselleştirebilirsiniz. Pandas, veri analizi ve manipülasyonu için güçlü ve esnek bir araçtır. Öğrenmesi kolaydır ve geniş bir topluluk desteği vardır. Pandas'ın dokümantasyonu zengin kaynaklara sahiptir ve birçok örnek kod bulunmaktadır. Proje içerisinde kullanılan bazı pandas kütüphanesinin veri yapılarının detayları aşağıda belirtilmiştir.

DataFrame: Pandas'ın en önemli veri yapısı olan DataFrame, tablo benzeri bir yapıdır. DataFrame, satır ve sütunlardan oluşur ve etiketlenmiş veriyle çalışır. Satırlar genellikle gözlemleri, sütunlar ise değişkenleri temsil eder. DataFrame, veri manipülasyonu için birçok işlem ve yöntem sağlar. Veri setlerini okumak, filtrelemek, dönüştürmek, gruplamak, birleştirmek ve analiz etmek için kullanılabilir. **Series:** Pandas'ın bir diğer önemli veri yapısı olan Series, tek boyutlu bir veri dizisidir. Her bir öge, bir indeksle etiketlenir ve veriye karşılık gelen değeri içerir. Series, DataFrame içinde sütunları temsil etmek için kullanılabilir. Ayrıca, tek boyutlu veri işlemleri için de kullanılabilir. **Veri Okuma ve Yazma:** Pandas, farklı veri formatlarını okuyabilme ve yazabilme yeteneğine sahiptir. `read_csv()`, `read_excel()`, `read_sql()` gibi işlevlerle CSV, Excel, SQL veritabanları ve diğer birçok veri kaynağını okuyabilmekteyiz. Ayrıca, `to_csv()`, `to_excel()`, `to_sql()` gibi işlevlerle verileri farklı formatlarda kaydedebiliriz.

Veri Manipülasyonu: Pandas, veri setlerini kolayca manipüle etmek için bir dizi işlem ve yöntem sağlar. Sütunları seçme, filtreleme, sıralama, yeniden şekillendirme, birleştirme ve dönüştürme gibi işlemleri gerçekleştirebilirsiniz. Örneğin, `df['column_name']` şeklinde bir sütuna erişebilir veya `df[df['column_name'] > 10]` gibi bir filtreleme işlemi yapılabilir.

Veri Analizi: Pandas, veri analizi için birçok istatistiksel ve hesaplamalı işlemi kolaylıkla gerçekleştirebileceğiniz işlemlere sahiptir. Örneğin, `mean()`, `sum()`, `min()`, `max()` gibi işlevlerle sütunlar üzerinde istatistiksel hesaplamalar yapılabilir. Gruplama

işlemleri için groupby() işlevi kullanılabilir. Veri Görselleştirme: Pandas, Matplotlib veya Seaborn gibi görselleştirme kütüphaneleriyle entegre çalışır. Bu sayede verilerinizi grafikler, histogramlar, kutu grafikleri, dağılım grafikleri ve diğer görsel öğelerle kolayca görselleştirilebilir. plot() işlevini kullanarak veriler çizdirilebilir. [5]

2.1.2.2 Random Kütüphanesi

Python'un random kütüphanesi, rasgele sayı üretimi, seçimleri rastgele yapma, örnekleme işlemleri ve diğer rasgele işlemleri gerçekleştirmek için kullanılır. Bu kütüphane, rastgelelikle ilgili çeşitli işlevleri içerir.

İşlevler:

random(): 0 ile 1 arasında rasgele bir ondalık sayı döndürür.

randrange(start, stop, step): Belirli bir aralıktan rasgele bir tamsayı döndürür.

randint(a, b): Belirli bir aralıktan (a ve b dahil) rasgele bir tamsayı döndürür.

choice(seq): Bir dizi veya liste gibi bir diziden rasgele bir öğe seçer.

shuffle(seq): Bir diziyi (listeyi) rasgele sıralar.

sample(population, k): Bir popülasyondan (dizi, liste vb.) belirli sayıda benzersiz örnekleme yapar.

uniform(a, b): Belirli bir aralıktan (a ve b arasında) rasgele bir ondalık sayı döndürür.

Random kütüphanesi, rastgelelikle ilgili çeşitli işlemleri kolaylıkla gerçekleştirebilen basit ve kullanıcı dostu bir arayüze sahiptir. Kütüphane, rasgele sayı üretimi için çeşitli işlevler sunar. Bu sayede rastgele sayılarla çalıştığımız uygulamaları kolayca geliştirebilmemizi sağlar. Random kütüphanesi, bir popülasyondan rasgele örneklem almak için işlevler sağlar. Bu özellik, istatistiksel analizler veya simülasyonlar gibi alanlarda faydalıdır. Kütüphane, çeşitli dağılımlara (örneğin normal dağılım, üçgen dağılım) dayalı rasgele sayı üretimini destekler. Bu, farklı rastgelelik ihtiyaçlarınızı

karşılmak için esneklik sağlar. Fakat bazı arařtırmalar sonucu birkaç dezavantaj bilgilere ulařılmıřtır Bunlar;

Pseudo-rasgelelik: random kütüphanesi, aslında pseudo-rasgele (yani deterministik) sayılar üretir. Bu, belirli bir başlangıç noktası (seed) kullanılarak üretilen rasgele sayı dizilerinin tekrar edilebilir olması anlamına gelir. Gerçek rastgelelik için özel donanımlar veya hizmetlere ihtiyaç duyulabilir.

Güvenlik: random kütüphanesi, kriptografik güvenlik gerektiren uygulamalar için uygun deęildir. Kriptografik amaçlarla güvenilir rastgele sayı üretimi için secrets kütüphanesi kullanılması önerilir.

İstatistiksel özellikler: Bazı durumlarda, random kütüphanesinin ürettięi rasgele sayılar, istatistiksel açıdan tamamen homojen veya dengeli olmayabilir. Bu durum, bazı uygulamalarda dikkate alınması gereken bir husustur. [6]

2.1.3 Projede Kullanılan Araçlar: Anaconda Navigator

Anaconda Navigator, Anaconda dağıtımıyla birlikte gelen bir grafiksel kullanıcı arayüzüdür. Anaconda Navigator, veri bilimi ve makine öğrenmesi projeleri için çeşitli araçlara erişimi kolaylařtıran bir uygulamadır. Jupyter Notebook, Anaconda Navigator üzerinden başlatılabilen bir araçtır ve veri analizi, kod geliştirme, belgeleme ve paylaşım gibi bir dizi işlemi destekler. Jupyter Notebook, interaktif bir kodlama ve veri analizi ortamıdır. Web tabanlı bir araç olan Jupyter Notebook, Python, R, Julia ve dięer programlama dilleriyle çalışabilir. Kullanıcılar, kod, metin, grafikler ve denklemleri bir araya getirerek belgelendirilmiş bir çalışma ortamı oluşturabilir. Jupyter Notebook, kod bloklarını ayrı ayrı çalıştırma, sonuçları görselleştirme, veri analizi yapma ve raporlama gibi işlemleri kolaylařtırır. Anaconda Navigator üzerinden Jupyter Notebook'u başlatmak için ařaęıdaki adımları izleyebilirsiniz:

Anaconda Navigator'ı başlatın.

Sol taraftaki "Home" sekmesini seçin.

"Jupyter Notebook" simgesini tıklayın.

Bu, Jupyter Notebook'un web tabanlı arayüzünü başlatacağı ve bir tarayıcı penceresi açılacaktır. Jupyter Notebook'un kullanıcı dostu bir arayüzü vardır. Bir Jupyter Notebook oturumu açıldığında, yeni bir Python veya başka bir programlama diline dayalı bir not defteri oluşturabilirsiniz. Not defterleri, kod hücrelerini içeren bölümlerden oluşur. Her kod hücresi, kodu çalıştırmaya ve sonuçlarını görmeye olanak tanır. Metin hücreleri ise açıklama, belgeleme veya açıklama metni için kullanılabilir. Jupyter Notebook, veri analizi, veri görselleştirme, makine öğrenmesi modeli eğitimi, raporlama, öğretim materyalleri oluşturma ve daha fazlası gibi birçok kullanım alanına sahiptir. İnteraktif kodlama deneyimi, zengin metin biçimlendirme seçenekleri, grafiklerin etkileşimli olarak görüntülenmesi ve hızlı prototipleme yetenekleri gibi özellikleriyle veri bilimciler ve araştırmacılar arasında yaygın olarak kullanılan bir araçtır. Anaconda Navigator, Jupyter Notebook'u kullanarak veri bilimi projelerini daha erişilebilir hale getirir. Kullanıcılar, Anaconda Navigator üzerinden kolayca Jupyter Notebook'u başlatabilir ve proje geliştirmek, kod yazmak, analiz yapmak ve belgeler oluşturmak için kullanabilir. [7]



Şekil 2: Jupiter ve Anaconda Navigator [8]

2.2 Kelimeler veri setinin araştırılması ve hazırlanması

Kelimeler veri setinin araştırılması ve hazırlanması süreci, güvenilir kaynaklardan veri toplama, veri temizleme ve düzenleme adımlarını içermektedir. Kelime veri setini oluşturmak için güvenilir ve kaynakları belirtilmiş sözlükler, dil kaynakları, akademik çalışmalar ve öğretim materyalleri gibi kaynaklardan yararlanabilirsiniz. Bu kaynakların doğru referanslarını belirtmek ve intihalden kaçınmak önemlidir. Araştırma kaynaklarınızı kullanarak, gerekli olan verileri toplamaya başlayabilirsiniz. Bu süreçte, kaynaklarınızı dikkatlice tarayarak doğru ve güvenilir verileri toplamak önemlidir.

Veri setinizi oluşturduktan sonra, veri temizleme adımına geçebilirsiniz. Bu adımda, veri setinizdeki gereksiz karakterler, boşluklar veya hatalı formatlarda bulunan kelimeler gibi veri hatalarını düzeltebilirsiniz. Veri setinizin tutarlı ve doğru olduğundan emin olmak için dikkatli bir şekilde veri temizleme işlemleri gerçekleştirmeniz önemlidir. Veri setinizi düzenlerken, hangi formatta kullanmak istediğinizi belirlemeniz gerekebilir. CSV (Comma-Separated Values), Excel veya JSON gibi farklı formatlar arasından seçim yapabilirsiniz. Seçtiğiniz formata göre, veri setinizi uygun bir şekilde düzenleyerek okunabilirlik ve erişilebilirlik sağlamalısınız. Veri setinizi oluştururken, kullanmayı düşündüğünüz kaynakların lisanslarını kontrol etmek önemlidir. Bazı kaynaklar telif haklarına tabi olabilir ve izin almadan kullanmak intihal olarak kabul edilebilir. Kaynaklarınızın lisanslarını dikkatlice inceleyerek, kullanım haklarınızı anlamalı ve lisans kurallarına uygun hareket etmelisiniz. [9] [10] [11]


```
In [3]: import pandas as pd
```

```
In [4]: df=pd.read_excel("word_list.xlsx")
```

```
In [9]: df.sample(20)
```

```
Out[9]:
```

	E	T	T_puan	E_puan	Total_Wrong
876	Offend	Kızdırmak, İncitmek (Birini)	NaN	NaN	NaN
557	Liquid	Sıvı	NaN	NaN	NaN
1678	Wall	Duvar	NaN	NaN	NaN
199	Basic	Esas, Temel	NaN	NaN	NaN
1710	Wide	Geniş	NaN	NaN	NaN
993	Pill	Hap	NaN	NaN	NaN
1612	Us	Biz	NaN	NaN	NaN
1543	These	Bunlar	NaN	NaN	NaN
1223	Radio	Radyo	NaN	NaN	NaN
92	Among	Arasında, Arasına	NaN	NaN	NaN
783	Mosquito	Sivrisinek	NaN	NaN	NaN
103	Announce	Duyurmak, Anons Etmek	NaN	NaN	NaN
518	Lawyer	Avukat	NaN	NaN	NaN
1079	Print	Basmak (Harfler vs.), İz	NaN	NaN	NaN
1743	You	Sen	NaN	NaN	NaN
588	Lamppost	Sokak Lambası	NaN	NaN	NaN
573	Lot	Mülk	NaN	NaN	NaN
749	Massacre	Katliam, Katliam Yapmak	NaN	NaN	NaN
1072	Prevent	Korumak	NaN	NaN	NaN
547	Lie	Yalan, Yalan Söylemek	NaN	NaN	NaN

Şekil 3: Veri setinde rasgele 20 kelime

Bölüm 3

Veri Seti ve Kodlar

Bu bölümde, veri setinin içeriğinden bahsedilip projede yer alan kodlar açıklanmıştır.

3.1 Veri Seti

3.1.1 Veri Seti İncelemesi

Projede yer alan veri seti farklı web siteleri ve dokümanlardan yararlanılarak tarafımdan hazırlanmıştır. Veri setim excel formatında olup 5 kolon 1761 satırdan oluşmaktadır.

```
In [11]: df.shape  
Out[11]: (1761, 5)
```

Şekil 4: Verimizin kaç satır ve sütundan oluştuğunu gösterir. İlk yazan satır sayısı, ikincisi ise sütun sayısıdır.

3.1.2 Veri Setinin Ortama Yüklenmesi ve Okunması

Veri seti ve çalışacağımız jupiter notebook dosyası aynı klasör içerisinde olmalıdır. Pandas kütüphanesinin veri formatlarını okuyabilme ve yazabilme yeteneğine sahip olması sayesinde veriyi ortamda inceleyebiliyor ve işleyebiliyoruz.

```
df=pd.read_excel("word_list.xlsx")
```

Şekil 5: Veri Setinin Ortama Yüklenmesi

3.1.3 Gerekli Kütüphanelerin Ortama Yüklenmesi

Kütüphaneler, import komutu ile yüklenmektedir. İport komutundan sonra yüklenmek istenen kütüphane yazılır. sonrasında yazılan as komutu ise yüklemek istediğimiz kütüphaneyi hangi kısaltma ile ifade edeceğimizi belirtmemize yarar. Proje içerisinde kullanacağımız pandas, numpy ve random kütüphanelerini ortama import ediyoruz.

```
import pandas as pd
import numpy as np
import random
```

Şekil 6: Kütüphanelerin İport Edilmesi

3.2 Kodlar

3.2.1 Proje Kodları Açıklamaları

Oyun içerisine yüklenen dosyada hata kontrolü yapan koddur.

```
file_name= "word_list.xlsx"
try:
    df = pd.read_excel(file_name)
except:
    text = "Lütfen dosya adını, yolunu ve kurulumunu kontrol edin! "
    raise ValueError(text)
```

Şekil 7: Dosya Hata Kontrolü Yapan Kod

DataFrame içerisindeki tabloda boş gelen puanlar varsa eklenen kodla doldurulur.

```
df[["t_puan", "e_puan"]] = df[["t_puan", "e_puan"]].fillna(3).astype(int)
```

Şekil 8: Tablodaki Boş Gelen Puanları Ekler

Oyun içerisinde çıkmak istendiğinde mevcut puan tabloya kaydedilir. Son puanı kaydeden fonksiyon şekil 9’da belirtilmiştir.

```
def saveTablo(data=df):  
    data.to_excel(filename, index=False)  
  
    print(f"Tabloya kaydedilir {filename}\n")  
    print("Puanlar")  
    total_puan = data.shape[0] * 3  
    print(f"English puan: {total_puan}/{total_puan - data.e_puan.sum()}")  
    print(f"Turkish puan: {total_puan}/{total_puan - data.t_puan.sum()}")
```

Şekil 9: Son Puanı Kaydeden Fonksiyon

Yazılan fonksiyonun testini yapmak için kodu çalıştırıyoruz. Beklenen puan $1761 \times 3 = 5283$ ’tür.

```
def saveTablo (data=df):  
  
    data.to_excel(file_name, index=False)  
  
    print(f"Tabloya kaydedildi {file_name}\n")  
    print("Puanlar")  
    total_puan = data.shape[0] * 3  
    print(f"English puan: {total_puan}/{total_puan - data.e_puan.sum()}")  
    print(f"Turkish puan: {total_puan}/{total_puan - data.t_puan.sum()}")
```

```
saveTablo()
```

```
Tabloya kaydedildi word_list.xlsx
```

```
Puanlar
```

```
English puan: 5283/0
```

```
Turkish puan: 5283/0
```

Şekil 10: Test Sonucu

Kullanıcının hangi dilde çalışma yapmak istediğini seçtiği alanı While döngüsü içerisinde kuruyoruz. Döngü içerisinde öncelikle tüm Excel dosyasının içerisinde puanların sıfırlanmış olmasını denetleyen bir koşul yazılmalı.

```

while True:
    if df.e_puan.sum() == 0 and df.t_puan.sum() == 0:
        print("Tüm kelimeler başarıyla tamamlandı")
        print("Lütfen Excel dosyasına yeni kelimeler ekleyin")
        break

```

Şekil 11: Puanların Sıfırlanmış Olmasını Denetler

Kullanıcıdan aşağıdaki seçimlerden birini yapmasını isteyen döngüyü yazıyoruz.

```

while True:
    if df.e_puan.sum() == 0 and df.t_puan.sum() == 0:
        print("Tüm kelimeler başarıyla tamamlandı")
        print("Lütfen Excel dosyasına yeni kelimeler ekleyin")
        break

    type = input("== Hangisini seçmek istersiniz?\nEnglish = E\nTurkish = T\nrandom add puan = R\nquit = Q\n--> ")

    if type not in ["E", "T", "R", "Q"]:
        print("Bunlardan birini seçmelisin \nE, T, R or Q\n\n")
        continue

    if type == "Q":
        print("\n\n\n...BYEEEE...\n")
        break

== Hangisini seçmek istersiniz?
English = E
Turkish = T
random add puan = R
quit = Q
--> E
== Hangisini seçmek istersiniz?
English = E
Turkish = T
random add puan = R
quit = Q
--> R
== Hangisini seçmek istersiniz?
English = E
Turkish = T
random add puan = R
quit = Q
--> Q

...BYEEEE...

```

Şekil 12: Kullanıcının Seçim Yapmasını Sağlayan Döngü ve Çıktısı

Eğer kullanıcı “E” seçerse İngilizce-Türkçe alıştırma, “T” seçerse Türkçe-İngilizce alıştırma , “R” Rastgele beş kelimenin puanını artırıyoruz. Eğer “Q” seçerse programdan çıkılıyor.

Kullanıcının “R” seçeneğini seçmesi sonucunda çalışacak döngü şekil 13’de belirtilmiştir. Kod bloğundan da görüleceği üzere puanı düşük olan beş kelime

bulunarak rastgele puanları artırılıyor. İşlem sonunda Excel kaydediliyor. Ardından yine kullanıcıdan ne yapmak istediği ile ilgili input değeri isteniyor.

```
if type == "R":
    exists = []
    x = 0
    while x < 5:
        if len(df[df["e_puan"] == 0]) + len(df[df["t_puan"] == 0]) < 6:
            print("rastgele değeri doldurmak için kırılma noktası")
            break

        r_index = random.randrange(0, len(df), 1)
        if r_index in exists:
            continue

        exists.append(R_index)

        language_puan = random.choice(["e_puan", "t_puan"])
        language_p_value = df.at[R_index, language_puan]
        if language_p_value != 0:
            continue

        df.at[R_index, language_puan] = language_p_value + 1
        print("add puan")
        x += 1

    saveTablo(df)

    continue
```

Şekil 13: Eğer R seçilirse

İngilizce veya Türkçe çalışma alanı seçen kullanıcı karşısına 3 tane seçenek geliyor. Bunlar 1,2, ve Q

1: Rastgele 10 adetten oluşan kelime topluluğu seçilerek sadece bu sözcükler içinde çalışma alanı oluşturulur.

2: Tüm dosya içerisinde rastgele bir adet seçilerek alıştırmaya yapılıyor. Kullanıcı sonlandırına kadar veya puanlar sıfırlanıncaya kadar devam eder.

Q: Çıkış

Şimdi ise DataFrame içerisinde 10 adet satırı seçtirelim ve işaretleyelim.

```
# seçilecek olan kelime sayısı
secilen_kelimler = 10

# datada yer alan toplam satır sayısı
satir_sayisi = df.shape[0]

# seçilmeye elverişli splitlerin listesi
secilmis_liste= []

# toplam satır sayısına kadar tüm sayılar iterasyona alınır
# for döngüsü içerisinde 10'a bölünenler listeye dahil edilir
for i in range(1, satir_sayisi):
    if i % secilen_kelimler == 0:
        secilmis_liste.append(i)

# liste içinde rastgele bir sayı seçilir
son_adım = random.choice(split_list)

# seçilen sayıdan 10 çıkartılarak aralık belirlenmiş olur
ilk_adım = son_adım - split_size
```

Şekil 14: girilen sayı kadar kelime seçer

Kullanıcının kelime seçimi sonrası yapılan kontrolleri görelim.

```
# döngü içerisinde kontroller mevcut
while True:

    # eğer ingilizce ya da türkçe bir alıştırma seçildi ise|
    # kelime gösterim/seçim tarzı soruluyor
    if type in ["E", "T"]:
        print("\nTest için yöntem seçin")
        ans = input(
            f"\n1- Select randomly in {secilen_kelimler} words\n2- Select randomly in all data\nq = Quit\n-->")

        # input olarak beklenen değer dışında bir değer geldiyse uyarı çıkar
        if ans not in ["1", "2", "q"]:
            print("Seçim bunlardan biri olmalıdır \n1, 2 or Q\n-->")
            continue

        # cevap çıkış ise önceki verilen cevap ile işaretlenir
        if ans == "Q":
            type = "Q"
            break

        # cevap bir ise belirtilen değişkenlere atama yapılır
        if ans == "1":
            ilk, son = ilk_adım, son_adım
            break

        # cevap 2 ise belirtilen değişkenlere tüm satırları içerecek atama yapılır
        if ans == "2":
            ilk, son = 0, len(df)
            break

    # kontrol hariç her koşulda döngü sonlandırılır
```

Şekil 15: Döngü içerisindeki kontroller

Oyunun tam kodları jupyter notebook'daki hali.

```
import pandas as pd
import random

def vocabularyTest(filename):
    """
    :param filename: file including path
    """

    try:
        df = pd.read_excel(filename)

    except:
        text = "Lütfen dosya adını, yolunu ve kurulumunu kontrol edin! "

        raise ValueError(text)

    df[["t_puan", "e_puan"]] = df[["t_puan", "e_puan"]].fillna(3).astype(int)

    def saveTablo(data=df):

        data.to_excel(filename, index=False)

        print(f"Tabloya kaydedildi {filename}\n")
        print("Puanlar")
        total_puan = data.shape[0] * 3
        print(f"English score: {total_puan}/{total_puan - data.E_puan.sum()}")
        print(f"Turkish score: {total_puan}/{total_puan - data.T_puan.sum()}")

    while True:

        if df.e_puan.sum() == 0 and df.t_puan.sum() == 0:
            print("Tüm kelimeler başarıyla tamamlandı")
            print("Lütfen Excel dosyasına yeni kelimeler ekleyin")
            break

        type = input("== Hangisini seçmek istersiniz?\nEnglish = E\nTurkish = T\nrandom add puan = R\nquit = Q\n--> ")

        if type not in ["E", "T", "R", "Q"]:
            print("Bunlardan birini seçmelisin \ne, t, r or q\n\n")
            continue

        if type == "q":
            print("\n\n...BYEEEE...\n")
            break

        if type == "R":

            exists = []
            x = 0
            while x < 5:

                if len(df[df["e_puan"] == 0]) + len(df[df["t_puan"] == 0]) < 6:
                    print("rastgele değeri doldurmak için kırılma noktası")
                    break

                r_index = random.randrange(0, len(df), 1)
                if r_index in exists:
                    continue

                exists.append(r_index)

                language_puan = random.choice(["e_puan", "t_puan"])
                language_p_value = df.at[r_index, language_puan]
                if language_p_value != 0:
                    continue

                df.at[r_index, language_puan] = language_p_value + 1
                print("add puan")
                x += 1

            saveTablo(df)

            continue
```



```

"""Generating random range"""
secilen_kelimler = 10
satir_sayisi = df.shape[0]
secilmis_liste = []
for i in range(1, satir_sayisi):
    if i % secilen_kelimler == 0:
        secilmis_liste.append(i)

son_adim = random.choice(secilmis_liste)
ilk_adim = son_adim - secilen_kelimler

while True:

    if type in ["E", "T"]:
        print("\nTest için yöntem seçin")
        ans = input(
            f"\n1- Rastgele seç {secilen_kelimler} \n2- Tüm verilerde rastgele seç\nQ = Quit\n-->")

        if ans not in ["1", "2", "Q"]:
            print("Seçim bunlardan biri olmalıdır \n1, 2 or Q\n-->")
            continue

        if ans == "Q":
            type = "Q"
            break

        if ans == "1":
            ilk, son = ilk_adim, son_adim
            break

        if ans == "2":
            ilk, son = 0, len(df)
            break

```

```

while True:

    r = random.randrange(ilk, son, 1)

    if type == "T":
        p = df.at[r, "t_puan"]
        words = "t"
        puan = "t_puan"

    if type == "E":
        p = df.at[r, "e_puan"]
        words = "e"
        puan = "e_puan"

    english = df.at[r, "e"]

    turkish_string = df.at[r, "t"]

    turkish = df.at[r, "t"].split(",")

    turkish_and_english = turkish + [english]

    if df[puan].sum() == 0:
        saveTablo(df)
        break

    if df[(df.index >= ilk) & (df.index < son)][puan].sum() == 0:
        break

```

```

| if p == 0:
    continue

answer = input(f"\n{df.at[r, words]}\n-->")

if answer == "q":
    saveTablo(df)
    break

if answer in turkish_and_english:

    print("\nTRUE!!!")

    df.at[r, puan] = p - 1

    print(f"{english} = {turkish_string}\n--Önceki Puan: {p}\n--Yeni Puan: {df.at[r, puan]}")
    continue

else:

    print("\nFALSE")
    print(f"English: {english}")
    print(f"Turkish: {turkish_string}")

```

```
vocabularyTest(filename="word_list.xlsx")
```

Şekil 16: Kodların Tam Hali

3.2.2 Proje Kodları

```
import pandas as pd
```

```
import random
```

```
def vocabularyTest(filename):
```

```
    """
```

```
    :param filename: file including path
```

```
    """
```

```
    try:
```

```
        df = pd.read_excel(filename)
```

```
    except:
```

```
        text = "Lütfen dosya adını, yolunu ve kurulumunu kontrol edin! "
```

```
        raise ValueError(text)
```

```
df[["t_puan", "e_puan"]] = df[["t_puan", "e_puan"]].fillna(3).astype(int)

def saveTablo(data=df):

    data.to_excel(filename, index=False)

    print(f"Tabloya kaydedildi {filename}\n")

    print("Puanlar")

    total_puan = data.shape[0] * 3

    print(f"English score: {total_puan}/{total_puan - data.E_puan.sum()}")

    print(f"Turkish score: {total_puan}/{total_puan - data.T_puan.sum()}")

while True:

    if df.e_puan.sum() == 0 and df.t_puan.sum() == 0:

        print("Tüm kelimeler başarıyla tamamlandı")

        print("Lütfen Excel dosyasına yeni kelimeler ekleyin")
```

```
break
```

```
type = input("== Hangisini seçmek istersiniz?\nEnglish = E\nTurkish =  
T\nrandom add puan = R\nquit = Q\n--> ")
```

```
if type not in ["E", "T", "R", "Q"]:
```

```
    print("Bunlardan birini seçmelisin \ne, t, r or q\n\n")
```

```
    continue
```

```
if type == "q":
```

```
    print("\n\n...BYEEEE...\n")
```

```
    break
```

```
if type == "R":
```

```
    exists = []
```

```
    x = 0
```

```
    while x < 5:
```

```
if len(df[df["e_puan"] == 0]) + len(df[df["t_puan"] == 0]) < 6:

    print("rastgele değeri doldurmak için kırılma noktası")

    break

r_index = random.randrange(0, len(df), 1)

if r_index in exists:

    continue

exists.append(r_index)

language_puan = random.choice(["e_puan", "t_puan"])

language_p_value = df.at[r_index, language_puan]

if language_p_value != 0:

    continue

df.at[r_index, language_puan] = language_p_value + 1

print("add puan")

x += 1
```

```
saveTablo(df)

continue

"""Generating random range"""

secilen_kelimler = 10

satir_sayisi = df.shape[0]

secilmis_liste = []

for i in range(1, satir_sayisi):

    if i % secilen_kelimler == 0:

        secilmis_liste.append(i)

son_adım = random.choice(secilmis_liste)

ilk_adım = son_adım - secilen_kelimler

while True:
```

```
if type in ["E", "T"]:
```

```
    print("\nTest için yöntem seçin")
```

```
    ans = input(
```

```
        f"\n1- Rastgele seç {secilen_kelimler} words\n2- Tüm verilerde rastgele  
    seç\nQ = Quit\n-->")
```

```
if ans not in ["1", "2", "Q"]:
```

```
    print("Seçim bunlardan biri olmalıdır \n1, 2 or Q\n-->")
```

```
    continue
```

```
if ans == "Q":
```

```
    type = "Q"
```

```
    break
```

```
if ans == "1":
```

```
    ilk, son = ilk_adım, son_adım
```

```
    break
```

```
if ans == "2":
```

```
ilk, son = 0, len(df)
```

```
break
```

```
while True:
```

```
    r = random.randrange(ilk, son, 1)
```

```
    if type == "T":
```

```
        p = df.at[r, "t_puan"]
```

```
        words = "t"
```

```
        puan = "t_puan"
```

```
    if type == "E":
```

```
        p = df.at[r, "e_puan"]
```

```
        words = "e"
```

```
        puan = "e_puan"
```

```
    english = df.at[r, "e"]
```



```
turkish_string = df.at[r, "t"]
```

```
turkish = df.at[r, "t"].split(",")
```

```
turkish_and_english = turkish + [english]
```

```
if df[puan].sum() == 0:
```

```
    saveTablo(df)
```

```
    break
```

```
if df[(df.index >= ilk) & (df.index < son)][puan].sum() == 0:
```

```
    break
```

```
if p == 0:
```

```
    continue
```

```
answer = input(f"\n{df.at[r, words]}\n-->")
```

```
if answer == "q":
```

```
    saveTablo(df)
```

```
    break
```

```
if answer in turkish_and_english:
```

```
    print("\nTRUE!!!")
```

```
    df.at[r, puan] = p - 1
```

```
    print(f"{english} = {turkish_string}\n--Önceki Puan: {p}\n--Yeni Puan:  
{df.at[r, puan]}")
```

```
    continue
```

```
else:
```

```
    print("\nFALSE")
```

```
    print(f"English: {english}")
```

```
print(f"Turkish: {turkish_string}")
```

```
vocabularyTest(filename="word_list.xlsx")
```

Kaynaklar

- [1] Y. D. D. M. N. GÖMLEKSİZ, «OYUN İLE İNGİLİZCE ÖĞRETİMİNİN UYGULANMASI VE ÖĞRENCİ BAŞARISINA ETKİSİ,» [Çevrimiçi]. Available: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://dergipark.org.tr/en/download/article-file/844602.
- [2] [Çevrimiçi]. Available: <https://www.python.org/>.
- [3] B. Akademi. [Çevrimiçi]. Available: <https://www.btkakademi.gov.tr/portal/course/sifirdan-ileri-seviye-python-programlama-5877>.
- [4] [Çevrimiçi]. Available: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)).
- [5] P. Kütüphanesi. [Çevrimiçi]. Available: <https://pandas.pydata.org/docs/>.
- [6] r. kütüphanesi. [Çevrimiçi]. Available: <https://docs.python.org/3/library/random.html>.
- [7] [Çevrimiçi]. Available: <https://www.anaconda.com/products/navigator>.
- [8] [Çevrimiçi]. Available: <https://newsroom.unl.edu/announce/holland/7344/41179>.

[9] Kaggle. [Çevrimiçi]. Available: (www.kaggle.com): .

[10] WordNet. [Çevrimiçi]. Available: (wordnet.princeton.edu).

]

[11] N. L. T. (NLTK). [Çevrimiçi]. Available: (www.nltk.org).

]

[12] [Çevrimiçi].

Available:

] <https://newsroom.unl.edu/announce/holland/7344/41179>.

Özgeçmiş

Adı Soyadı: Merve GÜL

Eğitim:

2014-2016 Süleyman Demirel Üniversitesi M.Y.O.
Elektronik Haberleşme Teknolojisi

2017-2021 Süleyman Demirel Üniversitesi Teknoloji Fakültesi
Enerji Sistemleri Mühendisliği

2022-2023 Katip Çelebi Üniversitesi Fen Bilimleri Enstitüsü
Yazılım Mühendisliği Bitirme projesisiz Yüksek Lisans

İş Deneyimi:

2023 – Etiya Bilgi Teknolojileri Yazılım Sanayi ve Ticaret A.Ş