

Proje Raporu

Proje Adı: Adam Asmaca Oyunu

Grup Üyeleri:

Mert Semih SARIYERLİ - 23181616001

Ramazan KANAT - 23181616003

Ders Adı: Nesne Yönelimli Programlama

Dönem Bilgisi: Güz – 2024

İçindekiler

1.Giriş

2.Teknik Bölüm

2.1. Problemin Tanımı

2.2. Mimari ve Kullanılan Sınıflar/Metotlar

2.3. Kod Parçaları ve Açıklamaları

3.Sonuç ve Değerlendirme

4.Ekler

5.Kaynakça

1. Giriş

Projenin Amacı:

Projemizin temel amacı, kullanıcı dostu bir arayüzle klasik Adam Asmaca oyununu Java programlama dili ve Swing kütüphanesi kullanarak gerçekleştirmektir. Bu süreçte hem kullanıcı etkileşimlerini yönetmek hem de algoritmik düşünme becerilerini geliştirmek hedeflenmiştir.

Kapsam:

Bu proje, Swing ile kullanıcı arayüzü oluşturma, kelime tahmini algoritması geliştirme ve hatalı tahminlerde şekil çizen bir grafiksel bileşen içerir.

Kullanılan Teknolojiler:

- Programlama Dili: Java
- Grafiksel Arayüz: Swing
- IDE: IntelliJ IDEA / Eclipse
- Diğer Araçlar
 - Youtube
 - BTK Akademi
 - ChatGPT

2. Teknik Bölüm

2.1. Problemin Tanımı

Adam Asmaca, oyuncunun gizli bir kelimeyi tahmin etmeye çalıştığı popüler bir kelime oyunudur. Her yanlış tahminde, "adam" figürü bir adım daha tamamlanır. Oyuncunun amacı, tüm tahmin hakları bitmeden kelimeyi doğru şekilde tamamlamaktır.

2.2. Mimari ve Kullanılan Sınıflar/Metotlar

Mimari:

1. **Arayüz Modülü:** Swing bileşenleriyle oluşturulmuş kullanıcı arayüzü.
2. **Oyun Mantığı:** Rastgele harf veya kelime seçimi, doğru ve yanlış tahminleri takip eden algoritma.
3. **Çizim Modülü:** Adam figürünü kalan hak durumuna göre çizen bileşen.

Ana Sınıflar ve Metotlar:

- **Main sınıfı:** Programın çalışmaya başladığı ana sınıftır. Diğer sınıfların bağlantısını sağlar ve uygulamanın başlatılmasını yönetir.
 - **main(String[] args):** Programı başlatır ve giriş ekranını gösterir.

- **Giris sınıfı:** Kullanıcıya giriş ekranı sağlar. Kullanıcıdan ad ve şifre gibi bilgileri alır ve oyuna giriş yapmayı sağlar.
- **Oyun sınıfı:** Oyunun temel mantığını içerir. Oyunun başlatılması, bitirilmesi ve zorluk seviyesinin ayarlanması gibi işlemleri yönetir.
 - **oyunuBaslat():** Oyunu başlatır
 - **oyunuBitir():** Oyunu bitirir ve skor bilgisi gösterilir.
- **Zorluk sınıfı:** Oyunun zorluk seviyesini belirler. Bu sınıf, oyuncunun seçtiği zorluk seviyesine göre oyun dinamiklerini değiştirir.
 - **zorlukSec():** Kullanıcının seçtiği zorluk seviyesini alır ve uygun ayarları yapar.
- **Kelimeyönetimi sınıfı:** KelimeIslemleri sınıfını extend eder ve kelime dosyasındaki kelimelerle işlem yapar.
 - **kelimeEkle(String kelime):** Verilen kelimeyi dosyaya ekler.
 - **kelimeleriListelleEkle(List<String> kelimeListesi):** Listeyi dosyaya ekler.
 - **dosyayiKontrolEtVeOlustur():** Dosya var mı diye kontrol eder ve yoksa oluşturur.
 - **kelimeleriOku():** Dosyadaki kelimeleri okur.
 - **rastgeleKelimeSec():** Rastgele bir kelime seçer.

- **KelimeIslemleri sınıfı** :IkelimeIslemleri arayüzünü implement eder. Bu sınıf, kelimelerle ilgili temel işlemleri içerir.
 - **kelimeEkle(String kelime)**: Soyut metod, alt sınıflarda implement edilmelidir.
 - **kelimeleriOku()**: Dosyadaki kelimeleri okur ve liste olarak döndürür.
- **IkelimeIslemleri interface'i**: Kelime işlemleri için temel metodları tanımlar.
 - **kelimeEkle(String kelime)**: Bir kelimeyi kaynağa ekler.
 - **kelimeleriOku()**: Kelimeleri kaynaktan okur ve bir liste döndürür.
- **DosyaOkumaHatasi sınıfı**: Dosya okuma işlemleri sırasında oluşan hataları yönetmek için kullanılan özel bir exception sınıfıdır.
 - **Constructor**: Hata mesajını alır ve üst sınıfa ileterek exception nesnesi oluşturur.

Genel Mimari:

- **IkelimeIslemleri Arayüzü**: Kelime işlemleri için temel metodları tanımlar.
- **KelimeIslemleri Soyut Sınıfı**: Arayüzü implement eder ve temel dosya okuma işlemleri sağlar.
- **KelimeYonetimi Sınıfı**: Dosya işlemleri ve kelime yönetimini sağlar.
- **Giris Sınıfı**: Kullanıcı girişini yönetir.
- **Oyun Sınıfı**: Oyunun başlatılması ve bitirilmesi işlemlerini içerir.
- **Zorluk Sınıfı**: Zorluk seviyesi yönetimini sağlar.
- **DosyaOkumaHatasi**: Dosya okuma hatalarını yönetir.

2.3. Kod Parçaları ve Açıklamaları

```
public class Main {
    public static void main(String[] args) {

        // Sistem görünümünü ayarla
        try {
            // UIManager sınıfı, Java uygulamalarının görünümünü ayarlamak için kullanılır.
            // Burada, uygulamanın görünümünü sistemin varsayılan görünümüne ayarlıyoruz.
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (Exception e) {
            // Eğer Look and Feel ayarlanırken bir hata olursa, hata mesajı yazdırılır.
            System.err.println("Look and Feel ayarlanırken hata oluştu: " + e.getMessage());
            e.printStackTrace(); // Hata izlerini yazdırır, böylece hata nedenini bulmak kolaylaşır.
        }

        // Swing bileşenlerinin düzgün bir şekilde çalışması için uygulama arka planda başlatılır.
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                // Giriş penceresini oluştur ve göster
                // Burada, Giriş sınıfından bir nesne oluşturuluyor.
                Giriş giris = new Giriş();

                // setVisible(true) çağırısı ile giriş penceresini ekranda görünür hale getiriyoruz.
                giris.setVisible(true);
            }
        });
    }
}
```

```
public void dosyayikontrolEtVeOluştur() { 1 usage
    File dosya = new File(dosyaAdi);
    if (!dosya.exists()) {
        try {
            // Dosya yoksa yeni dosya oluşturulur
            if (dosya.createNewFile()) {
                System.out.println("Kelime dosyası oluşturuldu: " + dosyaAdi);
                // Başlangıç kelimeleri dosyaya eklenir
                kelimelerilisteIleEkle(List.of(
                    ...elements: "ARABA", "BİLGİSAYAR", "TELEFON", "KLAVYE", "EKRAN",
                    "MASA", "SANDALYE", "KİTAP", "KALEM", "DEFTER",
                    "TREN", "UÇAK", "GEMİ", "OTOBÜS", "BİSİKLET",
                    "MÜZİK", "FOTOĞRAF", "VİDEO", "KAMERA", "PRİZ",
                    "ÇEKMECE", "DOLAP", "MUTFAK", "BANYO", "YATAK",
                    "YASTIK", "BATTANİYE", "HALI", "TELEVİZYON", "RADYO",
                    "ÇALARSAAT", "AYDINLATMA", "KOLTUK", "TERLİK", "AYAKKABI",
                    "KÜTÜPHANE", "SİLİKON", "PENCERE", "DUVAR", "BİNA",
                    "GÜNEŞ", "AY", "DÜNYA", "YILDIZ", "UZAY",
                    "GALAKSİ", "ATOM", "MOLEKÜL", "ENERJİ", "KUANTUM"
                ));
            }
        } catch (IOException e) {
            // Dosya oluşturulurken hata olursa, hata mesajı yazdırılır
            System.err.println("Dosya oluşturulurken hata oluştu: " + e.getMessage());
        }
    } else {
        // Dosya zaten varsa, bilgi verilir
        System.out.println("Kelime dosyası zaten mevcut: " + dosyaAdi);
    }
}
```

3. Sonuç ve Değerlendirme

Proje Çıktıları:

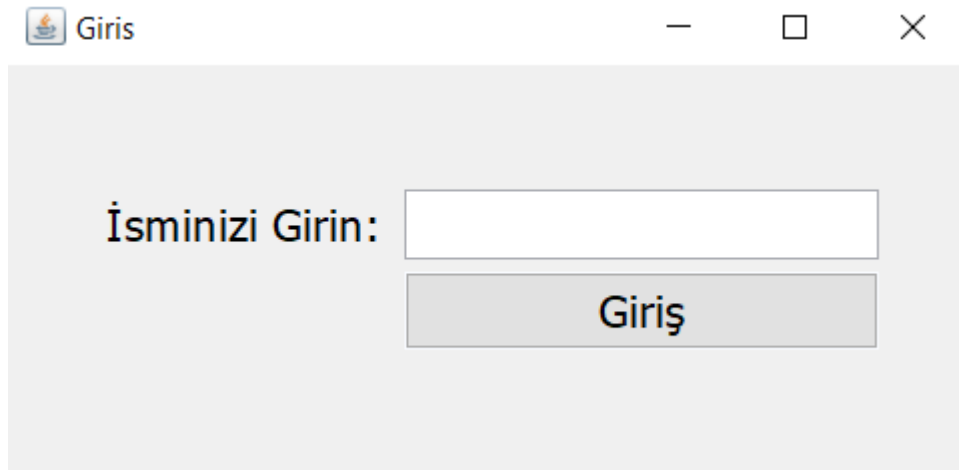
- Kullanıcı dostu bir Swing tabanlı arayüz oluşturuldu.
- Oyun, yanlış tahminlere göre "adam" figürünü dinamik olarak çizebiliyor.
- Tahmin kontrolü ve kelime güncelleme algoritması başarıyla çalışmaktadır.

Karşılaşılan Zorluklar ve Çözümler:

- 1. Arayüz Karmaşıklığı:** Başlangıçta arayüzde istediğimiz değişiklikleri yapmakta zorluk yaşıyordu. Swing UI Designer kullanımını öğrenerek bu sorun ortadan kaldırıldı.
- 2. Kelime Oluşumu:** İlk algoritmada doğru tahminler kelimeye işlenmiyordu; bu, StringBuilder ile güncellenmiştir.
- 3. Ascii Art Sorunu:** Başlangıçta “Adam” figürü istediğimiz gibi ekrana çıktı alınamıyordu. Bu sorun html-pre etiketiyle ortadan kaldırıldı.

4.Ekler

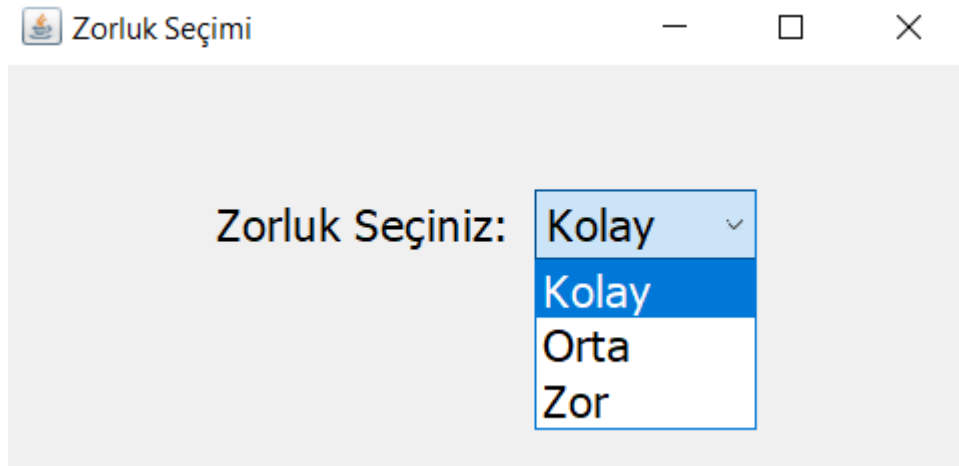
1. Kodlar ve proje dosyaları ZIP formatında teslim edilmiştir.
2. Çalışan uygulamadan ekran görüntüleri;



Giris

İsminizi Girin:

Giriş



Zorluk Seçimi

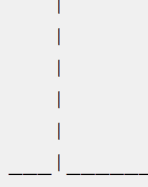
Zorluk Seçiniz:

Kolay
Kolay
Orta
Zor

Oyuncu: Ömer

Süre: 1 saniye

Kalan Hak: 4



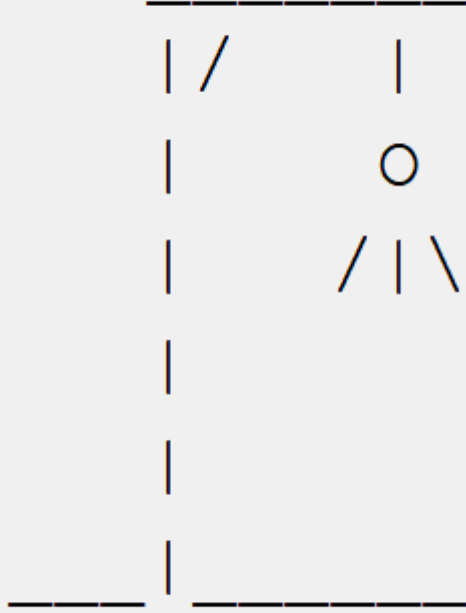
Tahmin Et

Uyarı

Skor Tablosu

1. Semih - Süre: 67 saniye - Puan: 25
2. Ramazan - Süre: 70 saniye - Puan: 0

Kullanılan İfadeler



5. Kaynakça

- Oracle Java SE Documentation, [Çevrim içi]. Erişim: <https://docs.oracle.com/javase/>
- BTK Akademi Java ile Programlamaya Giriş,[Çevrim içi]. Erişim: <https://www.btkakademi.gov.tr/portal/course/java-ile-programlamaya-giris-9617>
- Code Cube, “IntelliJ ile Swing GUI Tasarımı”, Youtube, 9 Nisan 2020, [Çevrim içi].Erişim: <https://www.youtube.com/watch?v=rwksn-BkSpA&t=1014s>
- OpenAI, "ChatGPT Modeli," Kullanılarak oluşturulan içerik, 21 Aralık 2024. [Çevrim içi]. Erişim: <https://chat.openai.com>