



T.C.
SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
PROGRAMLAMA DİLLERİNİN PRENSİPLERİ ÖDEV RAPORU

ÖDEV BAŞLIĞI

G211210061- Ash DÖNGEZ

SAKARYA

Mayıs, 2025

Programlama Dillerinin Prensipleri Dersi

C Dili ile Gezegenler Arası Uzay Yolculuğu Uygulaması

Aslı Döngöz

G211210061-2B

Özet

Bu proje, Programlama Dillerinin Prensipleri dersi kapsamında C programlama dili kullanılarak geliştirilmiş bir uzay simülasyonu sistemidir. Simülasyon başlatıldığında, Kisiler.txt, Araclar.txt ve Gezegenler.txt dosyaları okunarak sistem başlatılır. Bu veriler kullanarak insanlar uzay araçlarıyla bir gezegenden başka bir gezegene yolculuk ederler. Proje çok modüllü bir yapıda tasarlanmış, her bileşen kendi dosyasında soyutlanarak bağımsız şekilde kodlanmıştır. Gezegen, Uzay Aracı, Kişi, Zaman, Dosya Okuma, Simülasyon yapılarının görevleri ayrılmış ve nesne yönelimli programlamanın C dilinde yapılabilecek en yakın simülasyonu olan fonksiyon pointer'ları ve struct kullanımı ile gerçekleştirilmiştir. Her yapı, kendi verisini ve davranışlarını yöneterek simülasyonun farklı bir boyutunu kontrol etmektedir.

© 2017 Sakarya Üniversitesi.

Bu rapor benim özgün çalışmamdır. Faydalanmış olduğum kaynakları içerisinde belirttim. Herhangi bir kopya işleminde sorumluluk bana aittir.

1. GELİŞTİRİLEN YAZILIM

1.1. Gezegen Yapısı

Her gezegenin ismi, bir günün kaç saat olduğu ve tarih bilgisi tutulur. Ayrıca gezegende bulunan kişilerin bilgileri nüfus dizisinde saklanır. Gezegenler kendi zaman yapısına sahiptir ve zamanları gün-saat olarak iletirler. Her gezegenin kendine özgü yaşlanma katsayısı bulunur ve bu oranlarla kişiler yaşlandırılır.

1.2. Alt Gezegen Yapıları

Gezegen yapısı bir üst yapı (base struct) olarak tanımlanmış ve KayacGezegen, GazDeviGezegen, BuzDeviGezegen, CuceGezegen gibi alt türler bu yapıdan derste gösterilen kalıtım benzetimi yöntemi ile türetilmiştir. Kalıtım benzetimi için her gezegen türü, *super* alanı üzerinden temel gezegen yapısına erişebilmekte, *yaslanma()* gibi metotlar kullanılmıştır. Böylece C dilinde soyut sınıf benzetimi başarıyla uygulanmıştır.

1.3. Kisi Yapısı

Kişi bilgileri (isim, yaş, kalan ömür, bulunduğu aracın ismi) tutulur. Uzay aracının durumuna göre yaşlanma katsayısıyla her simülasyon adımında *guncelleKalanOmur* fonksiyonu ile yaş ve ömür güncellenir. Kalan ömür sıfırlandığında kişi ölü kabul edilir ve diziden çıkarılır.

1.4. UzayAraci Yapısı

Her uzay aracının ismi, kalkış/varış gezegeni, kalkış tarihi, yolcuları, hedefe kalan saati ve varış tarihi tutulur. Yolcular hareket etmeden önce, kalkış zamanı kontrol edilir ve yola çıkılır. Yolculuk esnasında zaman azaltılır ve vardığında yolcular gezegene eklenir. Tüm yolcular ölmüşse araç imha edilir.

* Ödev Sorumlusu. Aslı Döngöz G211210061

Mail Adresi: asli.dongez @ogr.sakarya.edu.tr

1.5. Zaman Yapısı

Saat, gün, ay ve yıl bilgilerini içerir. Her gezegenin kendi saat uzunluğuna göre zaman hesaplanır. **hedefTarihiHesapla** fonksiyonu ile bir uzay aracının varış tarihi, çıkış gezegeninin txtteki tarihi baz alınarak hedef gezegene varacağı tarih hesaplanır.

1.6. Simulasyon Yapısı

Tüm simulasyon döngüsünü kontrol eder. **baslat** fonksiyonu ile başlar, her turda zaman ilerletilir, gezegenler güncellenir, uzay araçlarının hareketleri sağlanır ve ekrana yazdırılır. Tüm uzay araçları hedefe vardığında veya imha olduğundan simulasyon biter.

1.7. DosyaOkuma Yapısı

Kisiler.txt, Araclar.txt ve Gezegenler.txt dosyalarındaki veriler okunarak struct yapılarına dönüştürülür. Dinamik olarak kişi, uzay aracı ve gezegen dizileri oluşturulur.

1.8. Exception Yapısı

setjmp/longjmp kullanılarak hatalı durumlarda sistemin temiz şekilde sonlandırılması sağlanır. Null pointer, dosya bulunamadı gibi durumlar kontrol altına alınır.

1.9. Console Yapısı

Her adımda ekran temizlenerek, simulasyonun güncel hali yazdırılır. **clear** veya **cls** komutu sistem türüne göre çalıştırılır.

1.10. Main Yapısı

Programın başlangıç noktasıdır. Gerekli yapılar oluşturulur, dosyalar okunur, simulasyon başlatılır ve sonlandığında tüm bellek temizlenir.

2. PROJE KAPSAMINDA ÖĞRENİLENLER

Bu projeye birlikte C dilinde nesneye dayalı sistemler kurmayı, fonksiyon göstericileri ile soyutlamanın nasıl sağlanabileceğini, karmaşık sistemleri modülerlikle kolayca yönetilebilir hale getirilebileceğini öğrendim.

3. PROJE KAPSAMINDA KARŞILAŞILAN ZORLUKLAR

Projede en çok zorlandığım noktalardan biri, kesinlikle bellek hataları (segmentation fault) ile uğraşmaktı. Sistemin birçok yerine koyduğum debuglar yardımıyla bu sorunu çözebildim. Bu yöntemle bellek adresleri, null kontrolleri ve değişken değerleri sürekli olarak gözlemladım. Yaşlanma katsayısının zamanın ilerlemesiyle doğru hesaplanmasını sağlamayı da sürekli testler yardımıyla hesapladım

4. ÇIKTILAR

Simülasyon tüm uzay araçları varlığında veya imha olduğunda sonlanır. Araçlar vardıktan sonra eğer bütün yolcuları ölürse de imha olabilir.

```
=== Final Durum ===
Gezenler:
    --- A ---    --- B ---    --- C ---    --- D ---    --- E ---
Tarih           01.06.2025    30.05.2025    25.06.2025    14.06.2025    09.06.2025
Nufus            3            8            9            7            0

Uzay Araclari:
Arac Adi        Durum    Cikis    Varis    Hedefe Kalan Saat    Hedefe Varacagi Tarih
X                Vardi    A        B        0                17.05.2025
Y                Vardi    B        C        0                14.06.2025
Z                Vardi    C        A        0                09.05.2025
W                Vardi    A        D        0                14.06.2025
Q                IMHA    D        E        --                --
R                Vardi    E        A        0                29.05.2025
(base) aslidongez@Asl-MacBook-Air pdp2odev %
```

Şekil 1. Programın çalışmasına ve sonlanmasına ait örnek çıktı

```
=== Final Durum ===
Gezenler:
    --- A ---    --- B ---    --- C ---    --- D ---    --- E ---
Tarih           08.05.2025    02.05.2025    15.05.2025    30.04.2025    08.05.2025
Nufus            0           10            5            3            6

Uzay Araclari:
Arac Adi        Durum    Cikis    Varis    Hedefe Kalan Saat    Hedefe Varacagi Tarih
X                Vardi    A        B        0                13.04.2025
Y                Vardi    B        C        0                03.05.2025
Z                Vardi    C        D        0                16.04.2025
Q                IMHA    A        E        --                --
L                Vardi    D        E        0                30.04.2025
M                IMHA    E        A        --                --
N                IMHA    B        E        --                --
(base) aslidongez@Asl-MacBook-Air pdp2odev %
```

Şekil 2. Programın çalışmasına ve sonlanmasına ait başka bir örnek çıktı

5. SONUÇ

Bu proje sayesinde C dilinde nesne benzetimi, zaman tabanlı simülasyon sistemleri ve modüler arası doğru yapılandırmayı öğrendim. Aynı zamanda bir önceki ödevle benzer olduğu için java ve c dillerinin farklılıklarını kavramış oldum. Her yapıya ait bellek temizliği, soyutlama, modülerlik gibi yazılım tasarım ilkeleri uygulamayı öğrendim.