BURSA TEKNİK ÜNİVERSİTESİ MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

DERS ADI: Algoritma ve Programlama
PROJE KONUSU: Konsol Tabanlı Uzay Simülasyonu
ADI SOYADI: Emin Eren KADIOĞLU
NUMARA: 25360859011
BÖLÜM: Bilgisayar Mühendisliği
GITHUB: https://github.com/emin-eren-kadioglu/25360859011_EminErenKadioglu_AVP_Proje

1. GİRİŞ

Bu proje, Algoritmalar ve Programlama dersi kapsamında C programlama dili kullanılarak dokümanda belirtilen kurallar çerçevesinde bireysel olarak geliştirilmiş, konsol tabanlı bir uzay simülasyonu uygulamasıdır. Projenin temel amacı; bir bilim insanı senaryosu üzerinden, kullanıcıdan alınan verilerle Güneş Sistemi'ndeki farklı gezegenlerde gerçekleşen temel fiziksel olayların (serbest düşme, potansiyel enerji, hidrostatik basınç vb.) simüle edilmesi ve sonuçların karşılaştırmalı olarak sunulmasıdır. Programın çalışma akışı, kullanıcı deneyimini merkeze alacak şekilde tasarlanmıştır. Simülasyon başladığında ilk olarak kullanıcıdan (bilim insanı) isim bilgisi istenir. Ardından, kullanıcının hesaplama yapmak istediği deneyi seçebilmesi için 9 farklı fizik deneyinden oluşan dinamik bir menü sunulur. Kullanıcı bir deney seçtiğinde, ilgili hesaplama için gerekli olan fiziksel büyüklükler (kütle, uzunluk, süre vb.) parametre olarak talep edilir. Girilen veriler, arka planda her gezegenin yerçekimi ivmesine göre işlenerek sonuçlar ekrana alt alta listelenir. Kullanıcı çıkış yapmak isteyene kadar ("-1" değeri girilene kadar) simülasyon döngüsü devam eder.

2. TEKNİK DETAYLAR

2.1 Program Akışı ve Modüler Yapı

Programın omurgasını oluşturan main fonksiyonu, kullanıcı etkileşimini başlatmak ve simülasyonun sürekliliğini sağlamakla görevlidir. Şekil 1.1'de gösterilen kod yapısında görüleceği üzere; program ilk açılışta kullanıcıdan (bilim insanı) isim bilgisini fgets fonksiyonu ile alır.

Ardından program, sonsuz bir döngü (while) içerisine girerek kullanıcıya dinamik bir deney menüsü sunar. Kullanıcının yaptığı seçime göre, switch-case yapısı devreye girer ve ilgili deney fonksiyonunu (örneğin serbest_dusme, yukari_atis vb.) çağırır. Bu fonksiyonlara, gezegen verilerini içeren g_sabitleri ve gezegen_isimleri dizileri parametre olarak gönderilir. Kullanıcı "-1" girene kadar bu döngü kırılamaz ve simülasyon devam eder.

Programın ana kontrol bloğuna ait C kodu Şekil 1.1'de, bu kodun çalıştırılmasıyla oluşan kullanıcı arayüzü ve menü yapısı ise Şekil 1.2'de gösterilmiştir.

```c
int main() {
    char bilim_insani[50];
    int secim = 0;

    printf("-----------------------------------------------\n");
    printf("BURSA TEKNIK UNIVERSITESI UZAY SIMULATORU\n");
    printf("-----------------------------------------------\n");
    printf("Lutfen bilim insani adinizi giriniz: ");
    fgets(bilim_insani, 50, stdin);

    while (secim != -1) {
        printf("\n===============================================\n");
        printf("Sn. %s Lutfen Deney Seciniz:\n", bilim_insani);
        printf("===============================================\n");
        printf("1. Serbest Dusme Deneyi\n");
        printf("2. Yukari Atis Deneyi\n");
        printf("3. Agirlik Deneyi\n");
        printf("4. Kutlecekimsel Potansiyel Enerji Deneyi\n");
        printf("5. Hidrostatik Basinc Deneyi\n");
        printf("6. Arsimet Kaldirma Kuvveti Deneyi\n");
        printf("7. Basit Sarkac Periyodu Deneyi\n");
        printf("8. Sabit Ip Gerilmesi Deneyi\n");
        printf("9. Asansor Deneyi\n");
        printf("CIKIS ICIN -1 GIRINIZ\n");
        printf("Seciminiz: ");
        scanf("%d", &secim);

        switch (secim) {
            case 1: serbest_dusme(g_sabitleri, gezegen_isimleri); break;
            case 2: yukari_atis(g_sabitleri, gezegen_isimleri); break;
            case 3: agirlik_hesapla(g_sabitleri, gezegen_isimleri); break;
            case 4: potansiyel_enerji(g_sabitleri, gezegen_isimleri); break;
            case 5: hidrostatik_basinc(g_sabitleri, gezegen_isimleri); break;
            case 6: arsimet_kaldirma(g_sabitleri, gezegen_isimleri); break;
            case 7: basit_sarkac(g_sabitleri, gezegen_isimleri); break;
            case 8: ip_gerilmesi(g_sabitleri, gezegen_isimleri); break;
            case 9: asansor_deneyi(g_sabitleri, gezegen_isimleri); break;
            case -1: printf("Simulasyon sonlandiriliyor...\n"); break;
            default: printf("Gecersiz secim! Lutfen tekrar deneyiniz.\n");
        }
```

Şekil 1.1 Programın ana kontrol döngüsü ve main fonksiyonu kodu.

```
-----------------------------------------------
BURSA TEKNIK UNIVERSITESI UZAY SIMULATORU
-----------------------------------------------
Lutfen bilim insani adinizi giriniz: Emin Eren

===============================================
Sn. Emin Eren
 Lutfen Deney Seciniz:
===============================================
1. Serbest Dusme Deneyi
2. Yukari Atis Deneyi
3. Agirlik Deneyi
4. Kutlecekimsel Potansiyel Enerji Deneyi
5. Hidrostatik Basinc Deneyi
6. Arsimet Kaldirma Kuvveti Deneyi
7. Basit Sarkac Periyodu Deneyi
8. Sabit Ip Gerilmesi Deneyi
9. Asansor Deneyi
CIKIS ICIN -1 GIRINIZ
Seciminiz:
```

Şekil 1.2 Programın açılış ekranı ve ana menü yapısı.

2.2 Gezegen Verileri ve Kullanılan Sabitler

Simülasyon kapsamında Merkür, Venüs, Dünya, Mars, Jüpiter, Satürn, Uranüs ve Neptün olmak üzere 8 gezegenin yerçekimi ivmeleri (g) const double tipinde bir dizi içerisinde tutulmuştur. Proje isterlerine uygun olarak; Güneş'e en yakın gezegen olan Merkür dizinin 0. indisli elemanı, Neptün ise son elemanı olacak şekilde sıralanmıştır. Ayrıca gezegen isimleri de paralel bir char dizisinde saklanarak çıktıların anlamlı olması sağlanmıştır.

```c
const double g_sabitleri[GEZEGEN_SAYISI] = {3.70, 8.87, 9.807, 3.71, 24.79, 10.44, 8.69, 11.15};
const char gezegen_isimleri[GEZEGEN_SAYISI][10] = {"Merkur", "Venus", "Dunya", "Mars", "Jupiter", "Saturn", "Uranus", "Neptun"};
```

Şekil 2.1. Gezegen verilerini tutan sabit diziler.

2.3. Deneylerin Hesaplama Mantığı

Bu bölümde, simülasyonda yer alan deneylerin çalışma mantığı ve örnek çıktıları sunulmaktadır

2.3.1. Serbest Düşme Deneyi

Bu deneyde ortamın hava direnci ihmal edilerek, cismin belirli bir süre sonunda ne kadar yol kat ettiği simüle edilir. Kullanıcıdan düşüş süresini (t) saniye cinsinden girmesi istenir. Program, Şekil 3.1'de gösterilen kod bloğunda yer aldığı üzere h = (1/2)gt^2 formülünü kullanarak düşülen yüksekliği hesaplar. Elde edilen sonuçlar metre (m) biriminde Şekil 3.2'deki gibi ekrana yansıtılır.

```c
// 1. Serbest Düşme Deneyi
void serbest_dusme(const double *g_ptr, const char (*isim_ptr)[10]) {
    double t, h;
    printf("\n--- Serbest Dusme Deneyi ---\n");
    printf("Sureyi (t) saniye cinsinden giriniz: ");
    scanf("%lf", &t);
    t = mutlak_deger_al(t);

    printf("\nSonuclar (t = %.2f saniye icin):\n", t);
    for (int i = 0; i < GEZEGEN_SAYISI; i++) {
        double g = *(g_ptr + i);
        h = 0.5 * g * t * t; // Formül (1)
        printf("%-10s -> Dusulen Yukseklik (h): %.2f m\n", (char *)(isim_ptr + i), h);
    }
}
```

Şekil 3.1. Serbest düşme deneyi C kod bloğu.

```
==========================================
Sn. Emin Eren
 Lutfen Deney Seciniz:
==========================================
1. Serbest Dusme Deneyi
2. Yukari Atis Deneyi
3. Agirlik Deneyi
4. Kutlecekimsel Potansiyel Enerji Deneyi
5. Hidrostatik Basinc Deneyi
6. Arsimet Kaldirma Kuvveti Deneyi
7. Basit Sarkac Periyodu Deneyi
8. Sabit Ip Gerilmesi Deneyi
9. Asansor Deneyi
CIKIS ICIN -1 GIRINIZ
Seciminiz: 1

--- Serbest Dusme Deneyi ---
Sureyi (t) saniye cinsinden giriniz: 7

Sonuclar (t = 7.00 saniye icin):
Merkur     -> Dusulen Yukseklik (h): 90.65 m
Venus      -> Dusulen Yukseklik (h): 217.31 m
Dunya      -> Dusulen Yukseklik (h): 240.27 m
Mars       -> Dusulen Yukseklik (h): 90.89 m
Jupiter    -> Dusulen Yukseklik (h): 607.36 m
Saturn     -> Dusulen Yukseklik (h): 255.78 m
Uranus     -> Dusulen Yukseklik (h): 212.91 m
Neptun     -> Dusulen Yukseklik (h): 273.18 m
```

Şekil 3.2. Serbest düşme deneyi ekran çıktısı

2.3.2. Yukarı Atış Deneyi

Sürtünmesiz ortamda dikey olarak yukarı fırlatılan bir cismin çıkabileceği maksimum yükseklik hesaplanır. Girdi olarak cismin ilk hızı (v_0) m/s cinsinden talep edilir. Hesaplama mantığını içeren kod bloğu Şekil 3.3'te verilmiştir. Burada h_max = (v_0^2)2g formülü kullanılarak cismin ulaşabileceği zirve noktası hesaplanır ve Şekil 3.4'te görüldüğü üzere metre (m) cinsinden listelenir.

```c
// 2. Yukarı Atış Deneyi
void yukari_atis(const double *g_ptr, const char (*isim_ptr)[10]) {
    double v0, h_max;
    printf("\n--- Yukari Atis Deneyi ---\n");
    printf("Firlatma hizini (v0) m/s cinsinden giriniz: ");
    scanf("%lf", &v0);
    v0 = mutlak_deger_al(v0);

    printf("\nSonuclar (v0 = %.2f m/s icin):\n", v0);
    for (int i = 0; i < GEZEGEN_SAYISI; i++) {
        double g = *(g_ptr + i);
        h_max = (v0 * v0) / (2 * g); // Formül (2)
        printf("%-10s -> Maksimum Yukseklik (h_max): %.2f m\n", (char *)(isim_ptr + i), h_max);
    }
}
```

Şekil 3.3. Yukarı atış deneyi C kod bloğu.

```
==========================================
Sn. Emin Eren
 Lutfen Deney Seciniz:
==========================================
1. Serbest Dusme Deneyi
2. Yukari Atis Deneyi
3. Agirlik Deneyi
4. Kutlecekimsel Potansiyel Enerji Deneyi
5. Hidrostatik Basinc Deneyi
6. Arsimet Kaldirma Kuvveti Deneyi
7. Basit Sarkac Periyodu Deneyi
8. Sabit Ip Gerilmesi Deneyi
9. Asansor Deneyi
CIKIS ICIN -1 GIRINIZ
Seciminiz: 2

--- Yukari Atis Deneyi ---
Firlatma hizini (v0) m/s cinsinden giriniz: 7

Sonuclar (v0 = 7.00 m/s icin):
Merkur     -> Maksimum Yukseklik (h_max): 6.62 m
Venus      -> Maksimum Yukseklik (h_max): 2.76 m
Dunya      -> Maksimum Yukseklik (h_max): 2.50 m
Mars       -> Maksimum Yukseklik (h_max): 6.60 m
Jupiter    -> Maksimum Yukseklik (h_max): 0.99 m
Saturn     -> Maksimum Yukseklik (h_max): 2.35 m
Uranus     -> Maksimum Yukseklik (h_max): 2.82 m
Neptun     -> Maksimum Yukseklik (h_max): 2.20 m
```

Şekil 3.4. Yukarı atış deneyi ekran çıktısı.

2.3.3. Ağırlık Deneyi

Kütle ve ağırlık arasındaki farkı ortaya koymak amacıyla, kullanıcıdan cismin kütlesi (m) kg biriminde istenir. Yerçekimi ivmesine bağlı olarak değişen ağırlık, Şekil 3.5'teki fonksiyonda görüleceği üzere G = mg formülü ile hesaplanır. Sonuçlar, kuvvet birimi olan Newton (N) cinsinden Şekil 3.6'daki çıktıda sunulmuştur.

```c
// 3. Ağırlık Deneyi
void agirlik_hesapla(const double *g_ptr, const char (*isim_ptr)[10]) {
    double m, G;
    printf("\n--- Agirlik Deneyi ---\n");
    printf("Cismin kutlesini (m) kg cinsinden giriniz: ");
    scanf("%lf", &m);
    m = mutlak_deger_al(m);

    printf("\nSonuclar (m = %.2f kg icin):\n", m);
    for (int i = 0; i < GEZEGEN_SAYISI; i++) {
        double g = *(g_ptr + i);
        G = m * g; // Formül (3)
        printf("%-10s -> Agirlik (G): %.2f N\n", (char *)(isim_ptr + i), G);
    }
}
```

Şekil 3.5. Ağırlık deneyi C kod bloğu.

```
==========================================
Sn. Emin Eren
 Lutfen Deney Seciniz:
==========================================
1. Serbest Dusme Deneyi
2. Yukari Atis Deneyi
3. Agirlik Deneyi
4. Kutlecekimsel Potansiyel Enerji Deneyi
5. Hidrostatik Basinc Deneyi
6. Arsimet Kaldirma Kuvveti Deneyi
7. Basit Sarkac Periyodu Deneyi
8. Sabit Ip Gerilmesi Deneyi
9. Asansor Deneyi
CIKIS ICIN -1 GIRINIZ
Seciminiz: 3

--- Agirlik Deneyi ---
Cismin kutlesini (m) kg cinsinden giriniz: 7

Sonuclar (m = 7.00 kg icin):
Merkur     -> Agirlik (G): 25.90 N
Venus      -> Agirlik (G): 62.09 N
Dunya      -> Agirlik (G): 68.65 N
Mars       -> Agirlik (G): 25.97 N
Jupiter    -> Agirlik (G): 173.53 N
Saturn     -> Agirlik (G): 73.08 N
Uranus     -> Agirlik (G): 60.83 N
Neptun     -> Agirlik (G): 78.05 N
```

2.3.4. Kütleçekimsel Potansiyel Enerji Deneyi

Bir cismin konumundan dolayı sahip olduğu enerjiyi modellemek için kütle (m) kg ve yükseklik (h) metre bilgileri kullanıcıdan alınır. Program, Şekil 3.7'de gösterilen algoritma ile E_p = mgh formülünü işler. Hesaplanan potansiyel enerji değerleri, Şekil 3.8'de görüldüğü gibi Joule (J) biriminde kullanıcıya gösterilir.

```c
// 4. Kütleçekimsel Potansiyel Enerji Deneyi
void potansiyel_enerji(const double *g_ptr, const char (*isim_ptr)[10]) {
    double m, h, Ep;
    printf("\n--- Kutlecekimsel Potansiyel Enerji Deneyi ---\n");
    printf("Cismin kutlesini (m) kg cinsinden giriniz: ");
    scanf("%lf", &m);
    m = mutlak_deger_al(m);

    printf("Yukseklig (h) metre cinsinden giriniz: ");
    scanf("%lf", &h);
    h = mutlak_deger_al(h);

    printf("\nSonuclar (m = %.2f kg, h = %.2f m icin):\n", m, h);
    for (int i = 0; i < GEZEGEN_SAYISI; i++) {
        double g = *(g_ptr + i);
        Ep = m * g * h; // Formül (4)
        printf("%-10s -> Potansiyel Enerji (Ep): %.2f J\n", (char *)(isim_ptr + i), Ep);
    }
}
```

Şekil 3.7. Kütleçekimsel potansiyel enerji deneyi C kod bloğu.

```
==========================================
Sn. Emin Eren
 Lutfen Deney Seciniz:
==========================================
1. Serbest Dusme Deneyi
2. Yukari Atis Deneyi
3. Agirlik Deneyi
4. Kutlecekimsel Potansiyel Enerji Deneyi
5. Hidrostatik Basinc Deneyi
6. Arsimet Kaldirma Kuvveti Deneyi
7. Basit Sarkac Periyodu Deneyi
8. Sabit Ip Gerilmesi Deneyi
9. Asansor Deneyi
CIKIS ICIN -1 GIRINIZ
Seciminiz: 4

--- Kutlecekimsel Potansiyel Enerji Deneyi ---
Cismin kutlesini (m) kg cinsinden giriniz: 61
Yukseklig (h) metre cinsinden giriniz: 7

Sonuclar (m = 61.00 kg, h = 7.00 m icin):
Merkur     -> Potansiyel Enerji (Ep): 1579.90 J
Venus      -> Potansiyel Enerji (Ep): 3787.49 J
Dunya      -> Potansiyel Enerji (Ep): 4187.59 J
Mars       -> Potansiyel Enerji (Ep): 1584.17 J
Jupiter    -> Potansiyel Enerji (Ep): 10585.33 J
Saturn     -> Potansiyel Enerji (Ep): 4457.88 J
Uranus     -> Potansiyel Enerji (Ep): 3710.63 J
Neptun     -> Potansiyel Enerji (Ep): 4761.05 J
```

Şekil 3.8. Kütleçekimsel potansiyel enerji deneyi ekran çıktısı.

2.3.5. Hidrostatik Basınç Deneyi

Sıvıların belirli bir derinlikte yüzeye uyguladığı basıncı hesaplamak adına, sıvının yoğunluğu (rho) kg/m^3 ve derinlik (h) metre parametreleri istenir. Şekil 3.9'daki kod yapısında P = (rho)gh formülü kullanılmıştır. Elde edilen hidrostatik basınç değerleri Pascal (Pa) biriminde Şekil 3.10'daki ekrana basılır.

```c
// 5. Hidrostatik Basınç Deneyi
void hidrostatik_basinc(const double *g_ptr, const char (*isim_ptr)[10]) {
    double rho, h, P;
    printf("\n--- Hidrostatik Basinc Deneyi ---\n");
    printf("Sivinin yogunlugunu (rho) kg/m^3 cinsinden giriniz: ");
    scanf("%lf", &rho);
    rho = mutlak_deger_al(rho);

    printf("Derinligi (h) metre cinsinden giriniz: ");
    scanf("%lf", &h);
    h = mutlak_deger_al(h);

    printf("\nSonuclar (rho = %.2f kg/m^3, h = %.2f m icin):\n", rho, h);
    for (int i = 0; i < GEZEGEN_SAYISI; i++) {
        double g = *(g_ptr + i);
        P = rho * g * h; // Formül (5)
        printf("%-10s -> Hidrostatik Basinc (P): %.2f Pa\n", (char *)(isim_ptr + i), P);
    }
}
```

Şekil 3.9. Hidrostatik basınç deneyi C kod bloğu.

```
==========================================
Sn. Emin Eren
 Lutfen Deney Seciniz:
==========================================
1. Serbest Dusme Deneyi
2. Yukari Atis Deneyi
3. Agirlik Deneyi
4. Kutlecekimsel Potansiyel Enerji Deneyi
5. Hidrostatik Basinc Deneyi
6. Arsimet Kaldirma Kuvveti Deneyi
7. Basit Sarkac Periyodu Deneyi
8. Sabit Ip Gerilmesi Deneyi
9. Asansor Deneyi
CIKIS ICIN -1 GIRINIZ
Seciminiz: 5

--- Hidrostatik Basinc Deneyi ---
Sivinin yogunlugunu (rho) kg/m^3 cinsinden giriniz: 61
Derinligi (h) metre cinsinden giriniz: 3

Sonuclar (rho = 61.00 kg/m^3, h = 3.00 m icin):
Merkur     -> Hidrostatik Basinc (P): 677.10 Pa
Venus      -> Hidrostatik Basinc (P): 1623.21 Pa
Dunya      -> Hidrostatik Basinc (P): 1794.68 Pa
Mars       -> Hidrostatik Basinc (P): 678.93 Pa
Jupiter    -> Hidrostatik Basinc (P): 4536.57 Pa
Saturn     -> Hidrostatik Basinc (P): 1910.52 Pa
Uranus     -> Hidrostatik Basinc (P): 1590.27 Pa
Neptun     -> Hidrostatik Basinc (P): 2040.45 Pa
```

Şekil 3.10. Hidrostatik basınç deneyi ekran çıktısı.

2.3.6. Arşimet Kaldırma Kuvveti Deneyi

Sıvı içerisindeki bir cisme etki eden kaldırma kuvvetini (F_k) bulmak için, sıvının yoğunluğu (rho) kg/m^3 ve cismin batan hacmi (V)m^3 cinsinden kullanıcıdan alınır. Şekil 3.11'de detayları verilen fonksiyonda F_k = (rho)gV formülü uygulanır ve sonuçlar Newton cinsinden Şekil 3.12'de gösterildiği gibi listelenir.

```c
// 6. Arşimet Kaldırma Kuvveti Deneyi
void arsimet_kaldirma(const double *g_ptr, const char (*isim_ptr)[10]) {
    double rho, V, Fk;
    printf("\n--- Arsimet Kaldirma Kuvveti Deneyi ---\n");
    printf("Sivinin yogunlugunu (rho) kg/m^3 cinsinden giriniz: ");
    scanf("%lf", &rho);
    rho = mutlak_deger_al(rho);

    printf("Cismin batan hacmini (V) m^3 cinsinden giriniz: ");
    scanf("%lf", &V);
    V = mutlak_deger_al(V);

    printf("\nSonuclar (rho = %.2f kg/m^3, V = %.4f m^3 icin):\n", rho, V);
    for (int i = 0; i < GEZEGEN_SAYISI; i++) {
        double g = *(g_ptr + i);
        Fk = rho * g * V; // Formül (6)
        printf("%-10s -> Kaldirma Kuvveti (Fk): %.2f N\n", (char *)(isim_ptr + i), Fk);
    }
}
```

Şekil 3.11. Arşimet kaldırma kuvveti deneyi C kod bloğu.

```
==========================================
Sn. Emin Eren
 Lutfen Deney Seciniz:
==========================================
1. Serbest Dusme Deneyi
2. Yukari Atis Deneyi
3. Agirlik Deneyi
4. Kutlecekimsel Potansiyel Enerji Deneyi
5. Hidrostatik Basinc Deneyi
6. Arsimet Kaldirma Kuvveti Deneyi
7. Basit Sarkac Periyodu Deneyi
8. Sabit Ip Gerilmesi Deneyi
9. Asansor Deneyi
CIKIS ICIN -1 GIRINIZ
Seciminiz: 6

--- Arsimet Kaldirma Kuvveti Deneyi ---
Sivinin yogunlugunu (rho) kg/m^3 cinsinden giriniz: 7
Cismin batan hacmini (V) m^3 cinsinden giriniz: 61

Sonuclar (rho = 7.00 kg/m^3, V = 61.0000 m^3 icin):
Merkur     -> Kaldirma Kuvveti (Fk): 1579.90 N
Venus      -> Kaldirma Kuvveti (Fk): 3787.49 N
Dunya      -> Kaldirma Kuvveti (Fk): 4187.59 N
Mars       -> Kaldirma Kuvveti (Fk): 1584.17 N
Jupiter    -> Kaldirma Kuvveti (Fk): 10585.33 N
Saturn     -> Kaldirma Kuvveti (Fk): 4457.88 N
Uranus     -> Kaldirma Kuvveti (Fk): 3710.63 N
Neptun     -> Kaldirma Kuvveti (Fk): 4761.05 N
```

Şekil 3.12. Arşimet kaldırma kuvveti deneyi ekran çıktısı.

2.3.7. Basit Sarkaç Periyodu Deneyi

Basit harmonik hareket yaptığı varsayılan bir sarkacın periyodunu (T) hesaplamak için kullanıcıdan ipin uzunluğu (L) metre cinsinden istenir. Şekil 3.13'teki hesaplama bloğunda $T = 2\pi\sqrt{(L/g)}$ formülü kullanılmıştır. Üretilen periyot süreleri (saniye), Şekil 3.14'teki konsol çıktısında görülmektedir.

```c
// 7. Basit Sarkaç Periyodu Deneyi
void basit_sarkac(const double *g_ptr, const char (*isim_ptr)[10]) {
    double L, T;
    printf("\n--- Basit Sarkac Periyodu Deneyi ---\n");
    printf("Sarkacin uzunlugunu (L) metre cinsinden giriniz: ");
    scanf("%lf", &L);
    L = mutlak_deger_al(L);

    printf("\nSonuclar (L = %.2f m icin):\n", L);
    for (int i = 0; i < GEZEGEN_SAYISI; i++) {
        double g = *(g_ptr + i);
        T = 2 * PI * sqrt(L / g); // Formül (7)
        printf("%-10s -> Periyot (T): %.2f s\n", (char *)(isim_ptr + i), T);
    }
}
```

Şekil 3.13. Basit sarkaç periyodu deneyi C kod bloğu.

```
==========================================
Sn. Emin Eren
 Lutfen Deney Seciniz:
==========================================
1. Serbest Dusme Deneyi
2. Yukari Atis Deneyi
3. Agirlik Deneyi
4. Kutlecekimsel Potansiyel Enerji Deneyi
5. Hidrostatik Basinc Deneyi
6. Arsimet Kaldirma Kuvveti Deneyi
7. Basit Sarkac Periyodu Deneyi
8. Sabit Ip Gerilmesi Deneyi
9. Asansor Deneyi
CIKIS ICIN -1 GIRINIZ
Seciminiz: 7

--- Basit Sarkac Periyodu Deneyi ---
Sarkacin uzunlugunu (L) metre cinsinden giriniz: 61

Sonuclar (L = 61.00 m icin):
Merkur     -> Periyot (T): 25.51 s
Venus      -> Periyot (T): 16.48 s
Dunya      -> Periyot (T): 15.67 s
Mars       -> Periyot (T): 25.48 s
Jupiter    -> Periyot (T): 9.86 s
Saturn     -> Periyot (T): 15.19 s
Uranus     -> Periyot (T): 16.65 s
Neptun     -> Periyot (T): 14.70 s
```

Şekil 3.14. Basit sarkaç deneyi ekran çıktısı.

2.3.8. Sabit İp Gerilmesi Deneyi

Düşey doğrultuda asılı, kütlesiz ve esnemez bir ipin ucundaki kütlenin (m) oluşturduğu gerilme kuvveti (T) hesaplanır. Kullanıcıdan kütle bilgisi istenir ve Şekil 3.15'te gösterildiği gibi T = mg formülü ile ipteki gerilme bulunur. Sonuçlar Newton cinsinden Şekil 3.16'da verilmiştir.

```c
// 8. Sabit İp Gerilmesi Deneyi
void ip_gerilmesi(const double *g_ptr, const char (*isim_ptr)[10]) {
    double m, T;
    printf("\n--- Sabit Ip Gerilmesi Deneyi ---\n");
    printf("Cismin kutlesini (m) kg cinsinden giriniz: ");
    scanf("%lf", &m);
    m = mutlak_deger_al(m);

    printf("\nSonuclar (m = %.2f kg icin):\n", m);
    for (int i = 0; i < GEZEGEN_SAYISI; i++) {
        double g = *(g_ptr + i);
        T = m * g; // Formül (8)
        printf("%-10s -> Ip Gerilmesi (T): %.2f N\n", (char *)(isim_ptr + i), T);
    }
}
```

Şekil 3.15. Sabit ip gerilmesi deneyi C kod bloğu.

```
==========================================
Sn. Emin Eren
 Lutfen Deney Seciniz:
==========================================
1. Serbest Dusme Deneyi
2. Yukari Atis Deneyi
3. Agirlik Deneyi
4. Kutlecekimsel Potansiyel Enerji Deneyi
5. Hidrostatik Basinc Deneyi
6. Arsimet Kaldirma Kuvveti Deneyi
7. Basit Sarkac Periyodu Deneyi
8. Sabit Ip Gerilmesi Deneyi
9. Asansor Deneyi
CIKIS ICIN -1 GIRINIZ
Seciminiz: 8

--- Sabit Ip Gerilmesi Deneyi ---
Cismin kutlesini (m) kg cinsinden giriniz: 7

Sonuclar (m = 7.00 kg icin):
Merkur     -> Ip Gerilmesi (T): 25.90 N
Venus      -> Ip Gerilmesi (T): 62.09 N
Dunya      -> Ip Gerilmesi (T): 68.65 N
Mars       -> Ip Gerilmesi (T): 25.97 N
Jupiter    -> Ip Gerilmesi (T): 173.53 N
Saturn     -> Ip Gerilmesi (T): 73.08 N
Uranus     -> Ip Gerilmesi (T): 60.83 N
Neptun     -> Ip Gerilmesi (T): 78.05 N
```

Şekil 3.16. Sabit ip gerilmesi deneyi ekran çıktısı.

2.3.9. Asansör Deneyi

Bu deneyde, eylemsiz referans sistemlerinde cismin hissettiği "etkin ağırlık" (N) simüle edilir. Kullanıcıdan cismin kütlesi, asansörün ivmesi ve hareket yönü istenir. **Şekil 3.17'deki** kod yapısında görüleceği üzere; asansör yukarı hızlanıyorsa N = m(g+a), aşağı hızlanıyorsa N = m(g-a) formülleri kullanılır.

- Yukarı yönde hızlanma durumu için örnek çıktı **Şekil 3.18'de**,
- Aşağı yönde hızlanma durumu için örnek çıktı **Şekil 3.19'da** gösterilmiştir.

```c
// 9. Asansör Deneyi
void asansor_deneyi(const double *g_ptr, const char (*isim_ptr)[10]) {
    double m, a, N;
    int durum;

    printf("\n--- Asansor Deneyi ---\n");
    printf("Cismin kutlesini (m) kg cinsinden giriniz: ");
    scanf("%lf", &m);
    m = mutlak_deger_al(m);

    printf("Asansorun ivmesini (a) m/s^2 cinsinden giriniz: ");
    scanf("%lf", &a);
    a = mutlak_deger_al(a);

    printf("Hareket yonunu seciniz:\n");
    printf("1. Yukari Hizlanan veya Asagi Yavaslayan (g + a)\n");
    printf("2. Asagi Hizlanan veya Yukari Yavaslayan (g - a)\n");
    printf("Secim: ");
    scanf("%d", &durum);

    printf("\nSonuclar:\n");
    for (int i = 0; i < GEZEGEN_SAYISI; i++) {
        double g = *(g_ptr + i);

        // Formül (9) ve (9.1)
        if (durum == 1) {
            N = m * (g + a);
        } else {
            N = m * (g - a);
        }

        printf("%-10s -> Etkin Agirlik (N): %.2f N\n", (char *)(isim_ptr + i), N);
    }
}
```

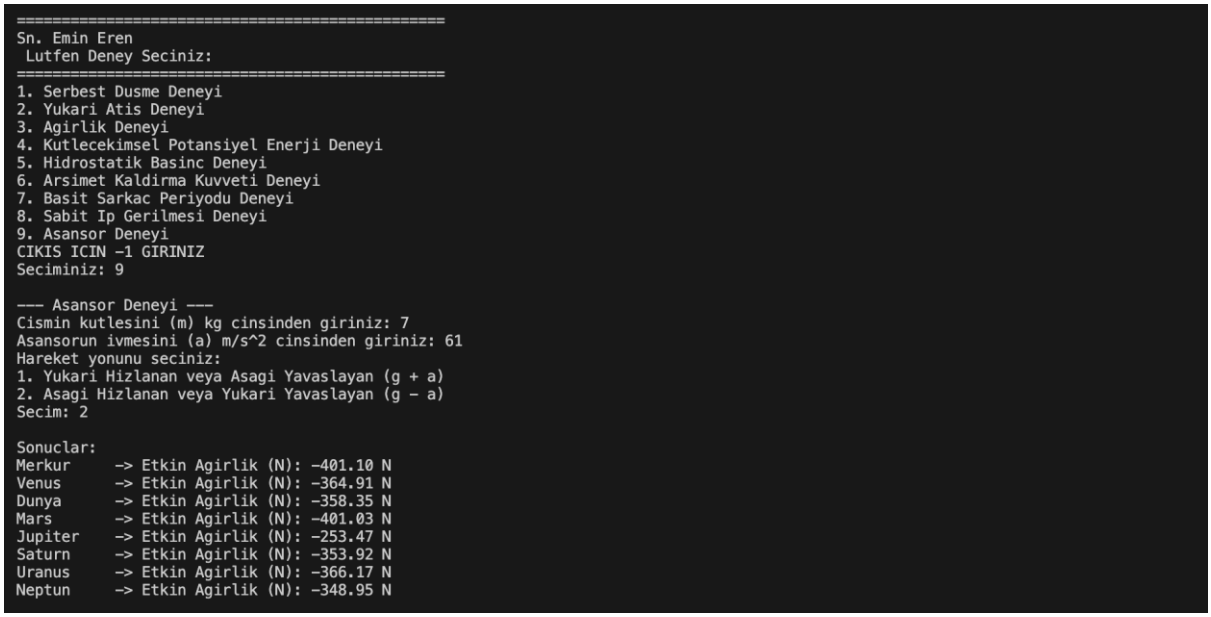
Şekil 3.17. Asansör deneyi C kod bloğu.

```
================================================
Sn. Emin Eren
 Lutfen Deney Seciniz:
================================================
1. Serbest Dusme Deneyi
2. Yukari Atis Deneyi
3. Agirlik Deneyi
4. Kutlecekimsel Potansiyel Enerji Deneyi
5. Hidrostatik Basinc Deneyi
6. Arsimet Kaldirma Kuvveti Deneyi
7. Basit Sarkac Periyodu Deneyi
8. Sabit Ip Gerilmesi Deneyi
9. Asansor Deneyi
CIKIS ICIN -1 GIRINIZ
Seciminiz: 9

--- Asansor Deneyi ---
Cismin kutlesini (m) kg cinsinden giriniz: 7
Asansorun ivmesini (a) m/s^2 cinsinden giriniz: 61
Hareket yonunu seciniz:
1. Yukari Hizlanan veya Asagi Yavaslayan (g + a)
2. Asagi Hizlanan veya Yukari Yavaslayan (g - a)
Secim: 1

Sonuclar:
Merkur     -> Etkin Agirlik (N): 452.90 N
Venus      -> Etkin Agirlik (N): 489.09 N
Dunya      -> Etkin Agirlik (N): 495.65 N
Mars       -> Etkin Agirlik (N): 452.97 N
Jupiter    -> Etkin Agirlik (N): 600.53 N
Saturn     -> Etkin Agirlik (N): 500.08 N
Uranus     -> Etkin Agirlik (N): 487.83 N
Neptun     -> Etkin Agirlik (N): 505.05 N
```

Şekil 3.18. Asansör deneyi ekran çıktısı - 1.



Şekil 3.19. Asansör deneyi ekran çıktısı - 2.

2.4 Girdi Doğrulama ve Hata Yönetimi

Proje gereksinimleri doğrultusunda kütle, uzunluk, süre gibi fiziksel büyüklüklerin negatif girilmesi durumunda programın çökmemesi sağlanmıştır. Şekil 4.1'de gösterilen mutlak_deger_al fonksiyonunda, if yapısı yerine ternary operatörü (kosul ? dogru : yanlis) kullanılarak negatif değerler pozitife çevrilmiştir.

Şekil 4.2'de görüldüğü üzere, kullanıcı ağırlık deneyi için hatalı bir giriş (-50 kg) yapsa dahi, program bu değeri arka planda mutlak değere çevirerek işlemi doğru şekilde (pozitif ağırlık sonucu vererek) tamamlamaktadır.

Ayrıca dizilere erişimde standart indeksleme yerine pointer aritmetiği kullanılmıştır. Şekil 4.3'te yer alan kod parçasında, dizi[i] yerine *(dizi + i) yöntemiyle bellek adreslerine doğrudan erişim sağlandığı görülmektedir.

```c
double mutlak_deger_al(double deger) {
    return (deger < 0) ? -deger : deger;
}
```

Şekil 4.1. Ternary operatörü ile mutlak değer fonksiyonu.

```
Seciminiz: 3

--- Agirlik Deneyi ---
Cismin kutlesini (m) kg cinsinden giriniz: -50

Sonuclar (m = 50.00 kg icin):
Merkur     -> Agirlik (G): 185.00 N
Venus      -> Agirlik (G): 443.50 N
Dunya      -> Agirlik (G): 490.35 N
Mars       -> Agirlik (G): 185.50 N
Jupiter    -> Agirlik (G): 1239.50 N
Saturn     -> Agirlik (G): 522.00 N
Uranus     -> Agirlik (G): 434.50 N
Neptun     -> Agirlik (G): 557.50 N
```

Şekil 4.2. Negatif girdi (-50) hatasının yönetildiği örnek çıktı.

Ayrıca dizilere erişimde dizi[i] yerine pointer aritmetiği *(dizi + i) kullanılarak bellek yönetimi kurallarına uyulmuştur.

```
    for (int i = 0; i < GEZEGEN_SAYISI; i++) {
        double g = *(g_ptr + i);
        Ep = m * g * h; // Formül (4)
        printf("%-10s -> Potansiyel Enerji (Ep): %.2f J\n", (char *)(isim_ptr + i), Ep);
    }
}
```

Şekil 4.3. Pointer aritmetiği ile dizi erişimi kod örneği.

3. EKSİKLİKLER VE GELİŞTİRMELER

Proje geliştirme sürecinde aşağıdaki geliştirmeler planlanmış ancak zaman kısıtı veya proje kapsamı nedeniyle sonraki sürümlere bırakılmıştır:

- **Verilerin Dosyaya Kaydedilmesi:** Şu anki sürümde sonuçlar sadece konsolda gösterilmektedir. Gelecekte, yapılan deneylerin bir .txt veya .csv dosyasına loglanması sağlanabilirdi. Bu özellik eklenirse bilim insanı geçmiş deneylerine erişebilir .
- **Grafik Arayüz (GUI):** Proje kısıtlamaları gereği konsol tabanlı çalışılmıştır. İleride basit bir grafik kütüphanesi ile gezegenlerin görselleri eklenebilir.
- **Hava Sürtünmesi:** Şu anki hesaplamalarda hava sürtünmesi ihmal edilmiştir. Daha gerçekçi sonuçlar için gezegenlerin atmosfer yoğunlukları da formüllere dahil edilebilir.

4. SONUÇ

Bu proje ile C programlama dilinde pointer kullanımı, bellek yönetimi ve modüler programlama teknikleri pekiştirilmiştir. Özellikle switch-case menü yapısı ve pointer

aritmetiği ile dizi erişimi konularında pratik yapılmıştır. Hazırlanan simülasyon, temel fizik kurallarının farklı yerçekimi ortamlarında nasıl değiştiğini başarılı bir şekilde göstermektedir. Proje, dokümanda belirtilen tüm isterleri (ternary kullanımı, pointer ile erişim, formatlı çıktı) karşılayacak şekilde tamamlanmıştır.

5. KAYNAKÇA

[1] Bursa Teknik Üniversitesi. (2025). Algoritmalar ve Programlama Dersi Dönem Projesi Dokümanı.

[2] Google. (2026). Gemini (Large Language Model). [Yapay Zeka Asistanı]. https://gemini.google.com/ (Not: Bu projede kod hata ayıklama ve rapor düzenleme süreçlerinde yardımcı araç olarak kullanılmıştır.)

[3] GeeksforGeeks. (2024). C Pointers and Array-Pointer Interchangeability. https://www.geeksforgeeks.org/pointer-vs-array-in-c/ (Not: Pointer aritmetiği ve bellek yönetimi konuları için referans alınmıştır.)

[4] NASA Glenn Research Center. (2024). Free Fall and Gravitational Forces. https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/free-fall/ (Not: Serbest düşme ve yerçekimi formülleri için referans alınmıştır.)

[5] Tutorialspoint. (2024). C Programming - Ternary Operator. https://www.tutorialspoint.com/cprogramming/c_operators.htm (Not: Koşul operatörleri kullanımı için referans alınmıştır.)