BLM111 proje raporu

Ali Uçar
Bilgisayar Mühendisliği
24360859022
2. şube

1.Giriş

- **1.1 Program açıklaması : Bu proje C programlama dili ile yazılmış konsol tabanlı bir uzay simulasyonu programıdır.İstenen deney ve girilen değerlere göre tanımlı fonksiyonlar ile güneş sistemi gezegenlerinin her birinde yapılacak deneyin sonucunu sırayla ekrana yazdırır.Bu proje bireysel olarak geliştirilmiştir.**
- **Projenin github linki: https://github.com/AUCorporation/BLM111_24360859022_AliUcar**
- **1.2 Programın genel akışı**

  1.İsim girişi:Program ilk olarak bilim adamının ismini alır.

  2.Deney menüsü:Program yapılmak istenen deneyi kullanıcıya sorar.

  3.Veri girişi:Program kullanıcıdan deneyde var olan değişkenlerin değerlerini ister.
- **4.Simulasyon uygulaması:Program deneyin her gezegen için sonuçlarını sırayla çıkarır.**

2.Teknik detaylar

2.1 Modüler yapı

Program modüler olarak çalışacak şekilde tasarlanmıştır.Şekil 2.1.1' de görüleceği üzere her deney ayrı bir fonksiyondur.Bu durum hem var olan deneylerin daha kolay ayırt edilmesini sağladı hem de yeni deneyler eklemeyi oldukça kolaylaştırdı.Daha sonra bu fonksiyonlar switch-case yapısı ile çağırılarak main içinde kullanıldı.

```
void serbest_dusme(const double *yercekimi_ptr) {
    double t, h;
printf("\n--- Serbest Dusme Deneyi ---\n");
    printf("Dusus suresini saniye cinsinden giriniz: ");
    scanf("%lf", &t);
t = (t < 0) ? -t : t;
printf("Girilen sure (mutlak deger): %.2f saniye\n\nSimulasyon sonuclari:\n", t);
for (int i = 0; i < gezegen_sayisi; i++) {
double g = *(yercekimi_ptr + i);
h = 0.5 * g * (t * t);
printf("%-10s -> Yukseklik: %.2f metre\n", *(gezegen_isim + i), h);
}}
```

Şekil 2.1.1

- **2.2 Kullanılan diziler ve sabitler**

Program şekil 2.2.2' de görüleceği üzere pi,yerçekimi sabitleri,gezegen isimleri ve sayısını sabit olarak barındırmaktadır.Yer çekimi sabitleri şekil 2.2.1' de de görülen tabloadan alınmıştır.Program deneylerde hesaplama yapacağı zaman bu sabitlere giderek gerekli verileri alır ve sırayla sonuçları yazdırabilir.

Acceleration Due to Gravity Comparison				
Body	**Mass [kg]**	**Radius [m]**	**Acceleration Due to Gravity, "g" [m/s²]**	**g / g-Earth**
Sun	1.99×10^{30}	6.96×10^{8}	274.13	27.95
Mercury	3.18×10^{23}	2.43×10^{6}	3.59	0.37
Venus	4.88×10^{24}	6.06×10^{6}	8.87	0.90
Earth	5.98×10^{24}	6.38×10^{6}	9.81	1.00
Moon	7.36×10^{22}	1.74×10^{6}	1.62	0.17
Mars	6.42×10^{23}	3.37×10^{6}	3.77	0.38
Jupiter	1.90×10^{27}	6.99×10^{7}	25.95	2.65
Saturn	5.68×10^{26}	5.85×10^{7}	11.08	1.13
Uranus	8.68×10^{25}	2.33×10^{7}	10.67	1.09
Neptune	1.03×10^{26}	2.21×10^{7}	14.07	1.43
Pluto	1.40×10^{22}	1.50×10^{6}	0.42	0.04

Şekil 2.2.1

```
const double Pi = 3.14159265;
const double yercekimi_sabit[]={3.59, 8.87, 9.81, 3.77, 25.95, 11.08, 10.67, 14.07};//yer çekimi sabitleri
const char *gezegen_isim[]={"Merkur", "Venus", "Dunya", "Mars", "Jupiter", "Saturn", "Uranus", "Neptun"};
const int gezegen_sayisi = 8;
```

Şekil 2.2.2

- **2.3 Deneylerde hesaplama**

Her deney için oluşturulmuş farklı fonksiyonlar içinde kullanıcıdan farklı değişkenler alındı ve farklı birimlerde sonuçlar üretildi.Programda kullanılan fonksiyonlar,birimler,formüller ve çıktıları:

- **2.3.1 Serbest Düşme Deneyi:Bu deneyde sürtünmesiz ortamda serbest bırakılan bir cismin, girilen süre sonunda ne kadar yol kat edeceği hesaplanmıştır.Kullanıcıdan düşüş süresi alınmış ve h=(g*t^2)/2 formülüyle her gezegen için yükseklik hesaplanmıştır.(Şekil 2.3.1.1 ve şekil 2.3.1.2)**

```c
void serbest_dusme(const double *yercekimi_ptr) {
    double t, h;
printf("\n--- Serbest Dusme Deneyi ---\n");
    printf("Dusus suresini saniye cinsinden giriniz: ");
    scanf("%lf", &t);
t = (t < 0) ? -t : t;
printf("Girilen sure (mutlak deger): %.2f saniye\n\nSimulasyon sonuclari:\n", t);
for (int i = 0; i < gezegen_sayisi; i++) {
double g = *(yercekimi_ptr + i);
h = 0.5 * g * (t * t);
printf("%-10s -> Yukseklik: %.2f metre\n", *(gezegen_isim + i), h);
}}
```

Şekil 2.3.1.1

```
--- Serbest Dusme Deneyi ---
Dusus suresini saniye cinsinden giriniz: 11.7
Girilen sure (mutlak deger): 11.70 saniye

Simulasyon sonuclari:
Merkur     -> Yukseklik: 245.72 metre
Venus      -> Yukseklik: 607.11 metre
Dunya      -> Yukseklik: 671.45 metre
Mars       -> Yukseklik: 258.04 metre
Jupiter    -> Yukseklik: 1776.15 metre
Saturn     -> Yukseklik: 758.37 metre
Uranus     -> Yukseklik: 730.31 metre
Neptun     -> Yukseklik: 963.02 metre
```

Şekil 2.3.1.2

- **2.3.2 Yukarı Atış Deneyi:Bu deneyde belirli bir ilk hızla dikey olarak yukarı fırlatılan bir cismin, gezegenlerin yerçekimi ivmesine bağlı olarak çıkabileceği maksimum yükseklik hesaplanmıştır.Kullanıcıdan atış hızı istenmiş ve h max= (v^2) / 2g formülüyle her gezegen için maksimum yükseklik hesaplanmıştır.(Şekil 2.3.2.1 ve şekil 2.3.2.2)**

```c
void yukari_atis(const double *yercekimi_ptr) {
    double v, h;
printf("\n---  Yukari Atis Deneyi ---\n");
    printf("Atış hızını m/s cinsinden giriniz: ");
    scanf("%lf", &v);
v = (v < 0) ? -v : v;
printf("Girilen hiz (mutlak deger): %.2f m/s\n\nSimulasyon sonuclari:\n", v);
for (int i = 0; i < gezegen_sayisi; i++) {
double g = *(yercekimi_ptr + i);
h = (v*v) / (2*g);
printf("%-10s -> Yukseklik(max): %.2f metre\n", *(gezegen_isim + i), h);
}}
```

Şekil 2.3.2.1

```
---  Yukari Atis Deneyi ---
Atış hızını m/s cinsinden giriniz: 15
Girilen hiz (mutlak deger): 15.00 m/s

Simulasyon sonuclari:
Merkur     -> Yukseklik(max): 31.34 metre
Venus      -> Yukseklik(max): 12.68 metre
Dunya      -> Yukseklik(max): 11.47 metre
Mars       -> Yukseklik(max): 29.84 metre
Jupiter    -> Yukseklik(max): 4.34 metre
Saturn     -> Yukseklik(max): 10.15 metre
Uranus     -> Yukseklik(max): 10.54 metre
Neptun     -> Yukseklik(max): 8.00 metre
```

Şekil 2.3.2.2

- **2.3.3 Ağırlık Deneyi:Bu deneyde kütlesi bilinen bir cismin, farklı gezegenlerdeki yerçekimi ivmelerine bağlı olarak değişen ağırlık değerleri hesaplanmıştır.Kullanıcıdan kütle değeri istenmiş ve G=m*g formülüyle her gezegen için cismin ağırlığı hesaplanmıştır.(Şekil 2.3.3.1 ve şekil 2.3.3.2)**

```
void agirlik(const double *yercekimi_ptr) {
    double m, G;
printf("\n--- Agirlik Deneyi ---\n");
    printf("Cismin kütlesini kilogram cinsinden giriniz: ");
    scanf("%lf", &m);
m = (m < 0) ? -m : m;
printf("Girilen kütle (mutlak deger): %.2f kilogram\n\nSimulasyon sonuclari:\n", m);
for (int i = 0; i < gezegen_sayisi; i++) {
double g = *(yercekimi_ptr + i);
G=m*g;
printf("%-10s -> Agirlik: %.2f Newton\n", *(gezegen_isim + i), G);
}}
```

Şekil 2.3.3.1

```
--- Agirlik Deneyi ---
Cismin kütlesini kilogram cinsinden giriniz: 6.5
Girilen kütle (mutlak deger): 6.50 kilogram

Simulasyon sonuclari:
Merkur     -> Agirlik: 23.34 Newton
Venus      -> Agirlik: 57.65 Newton
Dunya      -> Agirlik: 63.77 Newton
Mars       -> Agirlik: 24.50 Newton
Jupiter    -> Agirlik: 168.67 Newton
Saturn     -> Agirlik: 72.02 Newton
Uranus     -> Agirlik: 69.36 Newton
Neptun     -> Agirlik: 91.45 Newton
```

Şekil 2.3.3.2

- **2.3.4 Kütleçekimsel Potansiyel Enerji Deneyi:Bu deneyde belirli bir yükseklikte bulunan cismin, kütlesine ve gezegenin çekim alanına bağlı olarak sahip olduğu potansiyel enerji hesaplanmıştır.Kullanıcıdan cismin kütlesi ve yerden yüksekliği istenmiş ve Ep=m*g*h formülüyle her gezegendeki potansiyel enerjisi hesaplanmıştır.(Şekil 2.3.4.1 ve şekil 2.3.4.2)**

```c
void k_pot_enerji(const double *yercekimi_ptr) {
    double h,m,Ep;
printf("\n--- Kütlecekimsel Potansiyel Enerji Deneyi ---\n");
    printf("Cismin kütlesini kilogram cinsinden giriniz: ");
    scanf("%lf", &m);
m = (m < 0) ? -m : m;
    printf("Cismin yerden yuksekligini metre cinsinden giriniz: ");
    scanf("%lf", &h);
h = (h < 0) ? -h : h;
printf("Girilen kütle (mutlak deger): %.2f kilogram yukseklik(mutlak deger): %.2f metre\n\nSimulasyon sonuclari:\n", m,h);
for (int i = 0; i < gezegen_sayisi; i++) {
double g = *(yercekimi_ptr + i);
Ep=m*g*h;
printf("%-10s -> Potansiyel enerji: %.2f Joule\n", *(gezegen_isim + i), Ep);
}}
```

Şekil 2.3.4.1

```
--- Kütlecekimsel Potansiyel Enerji Deneyi ---
Cismin kütlesini kilogram cinsinden giriniz: 8
Cismin yerden yuksekligini metre cinsinden giriniz: 14
Girilen kütle (mutlak deger): 8.00 kilogram yukseklik(mutlak deger): 14.00 metre

Simulasyon sonuclari:
Merkur     -> Potansiyel enerji: 402.08 Joule
Venus      -> Potansiyel enerji: 993.44 Joule
Dunya      -> Potansiyel enerji: 1098.72 Joule
Mars       -> Potansiyel enerji: 422.24 Joule
Jupiter    -> Potansiyel enerji: 2906.40 Joule
Saturn     -> Potansiyel enerji: 1240.96 Joule
Uranus     -> Potansiyel enerji: 1195.04 Joule
Neptun     -> Potansiyel enerji: 1575.84 Joule
```

Şekil 2.3.4.2

- **2.3.5 Hidrostatik Basınç Deneyi:Bu deneyde bir sıvının, belirli bir derinlikte yüzeye uyguladığı dik kuvvet yani hidrostatik basınç hesaplanmıştır.Kullanıcıdan sıvının yoğunluğu ve derinliği alınmış ve P=p*g*h formülüyle sıvının her gezegendeki hidrostatik basıncı hesaplanmıştır.(Şekil 2.3.5.1 ve şekil 2.3.5.2)**

```
void h_basinc(const double *yercekimi_ptr) {
    double h,p,P;
printf("\n--- Hidrostatik Basinc deneyi ---\n");
    printf("Sivinin birim hacimdeki kutlesini(yogunlugunu) kg/m³ cinsinden giriniz: ");
    scanf("%lf", &p);
p = (p < 0) ? -p : p;
    printf("Sivinin derinligini metre cinsinden giriniz: ");
    scanf("%lf", &h);
h = (h < 0) ? -h : h;
printf("Girilen birim hacimde kutle (mutlak deger): %.2f kg/m³ derinlik(mutlak deger): %.2f metre\n\nSimulasyon sonuclari:\n", p,h);
for (int i = 0; i < gezegen_sayisi; i++) {
double g = *(yercekimi_ptr + i);
P=p*g*h;
printf("%-10s -> Hidrostatik basinc: %.2f Pascal\n", *(gezegen_isim + i), P);
}}
```

Şekil 2.3.5.1

```
--- Hidrostatik Basinc deneyi ---
Sivinin birim hacimdeki kutlesini(yogunlugunu) kg/m³ cinsinden giriniz: 9
Sivinin derinligini metre cinsinden giriniz: 8
Girilen birim hacimde kutle (mutlak deger): 9.00 kg/m³ derinlik(mutlak deger): 8.00 metre

Simulasyon sonuclari:
Merkur     -> Hidrostatik basinc: 258.48 Pascal
Venus      -> Hidrostatik basinc: 638.64 Pascal
Dunya      -> Hidrostatik basinc: 706.32 Pascal
Mars       -> Hidrostatik basinc: 271.44 Pascal
Jupiter    -> Hidrostatik basinc: 1868.40 Pascal
Saturn     -> Hidrostatik basinc: 797.76 Pascal
Uranus     -> Hidrostatik basinc: 768.24 Pascal
Neptun     -> Hidrostatik basinc: 1013.04 Pascal
```

Şekil 2.3.5.2

- **2.3.6 Arşimet Kaldırma Kuvveti Deneyi:Bu deneyde sıvı içerisindeki bir cisme etki eden kaldırma kuvveti hesaplanmıştır.Kullanıcıdan sıvının yoğunluğu ve cismin batan hacmi alınmış ve Fk=p*g*V formülü ile cisme her gezegende uygulanan kaldırma kuvveti hesaplanmıştır.(Şekil 2.3.6.1 ve şekil 2.3.6.2)**

```
void a_kaldirma_kuvveti(const double *yercekimi_ptr) {
    double V,p,Fk;
printf("\n--- Arsimet Kaldirma Kuvveti Deneyi ---\n");
    printf("Sivinin birim hacimdeki kutlesini(yogunluk) kg/m³ cinsinden giriniz: ");
    scanf("%lf", &p);
p = (p < 0) ? -p : p;
    printf("Cismin batan hacmini m³ cinsinden giriniz: ");
    scanf("%lf", &V);
V = (V < 0) ? -V : V;
printf("Girilen birim hacimde kutle(yogunluk) (mutlak deger): %.2f kg/m³ cismin batan hacmi(mutlak deger): %.2f m³\n\nSimulasyon sonuclari:\n", p,V);
for (int i = 0; i < gezegen_sayisi; i++) {
double g = *(yercekimi_ptr + i);
Fk=p*g*V;
printf("%-10s -> Kaldirma kuvveti: %.2f Newton\n", *(gezegen_isim + i), Fk);
}}
```

Şekil 2.3.6.1

```
--- Arsimet Kaldirma Kuvveti Deneyi ---
Sivinin birim hacimdeki kutlesini(yogunluk) kg/m³ cinsinden giriniz: 18
Cismin batan hacmini m³ cinsinden giriniz: 24
Girilen birim hacimde kutle(yogunluk) (mutlak deger): 18.00 kg/m³ cismin batan hacmi(mutlak deger): 24.00 m³

Simulasyon sonuclari:
Merkur     -> Kaldirma kuvveti: 1550.88 Newton
Venus      -> Kaldirma kuvveti: 3831.84 Newton
Dunya      -> Kaldirma kuvveti: 4237.92 Newton
Mars       -> Kaldirma kuvveti: 1628.64 Newton
Jupiter    -> Kaldirma kuvveti: 11210.40 Newton
Saturn     -> Kaldirma kuvveti: 4786.56 Newton
Uranus     -> Kaldirma kuvveti: 4609.44 Newton
Neptun     -> Kaldirma kuvveti: 6078.24 Newton
```

Şekil 2.3.6.2

- **2.3.7 Basit Sarkaç Periyodu Deneyi:Bu deneyde basit harmonik hareket yapan bir sarkacın, bir tam salınımı tamamlaması için geçen süre hesaplanmıştır.Kullanıcıdan sarkacın ip uzunluğu alınmış ve T=2*pi*√(L/g) formülüyle sarkacın her gezegendeki periyodu hesaplanmıştır:(Şekil 2.3.7.1 ve şekil 2.3.7.2)**

```c
void sarkac_periyodu(const double *yercekimi_ptr) {
    double L,T;
printf("\n--- Basit Sarkac Periyodu Deneyi ---\n");
    printf("Sarkacin ip uzunlugunu metre cinsinden giriniz: ");
    scanf("%lf", &L);
L = (L < 0) ? -L : L;
printf("Girilen sarkac ip uzunlugu (mutlak deger): %.2f m\n\nSimulasyon sonuclari:\n", L);
for (int i = 0; i < gezegen_sayisi; i++) {
double g = *(yercekimi_ptr + i);
T = 2*Pi*sqrt(L / g);
printf("%-10s -> Periyot: %.2f saniye\n", *(gezegen_isim + i), T);
}}
```

Şekil 2.3.7.1

```
--- Basit Sarkac Periyodu Deneyi ---
Sarkacin ip uzunlugunu metre cinsinden giriniz: 17
Girilen sarkac ip uzunlugu (mutlak deger): 17.00 m

Simulasyon sonuclari:
Merkur     -> Periyot: 13.67 saniye
Venus      -> Periyot: 8.70 saniye
Dunya      -> Periyot: 8.27 saniye
Mars       -> Periyot: 13.34 saniye
Jupiter    -> Periyot: 5.09 saniye
Saturn     -> Periyot: 7.78 saniye
Uranus     -> Periyot: 7.93 saniye
Neptun     -> Periyot: 6.91 saniye
```

Şekil 2.3.7.2

- **2.3.8 Sabit İp Gerilmesi Deneyi:Bu deneyde kütlesiz ve esnemez bir ipe asılı duran cismin, ip üzerinde oluşturduğu gerilme kuvveti hesaplanmaktadır.Kullanıcıdan cismin ağırlığı alınmış ve T=m*g ile her gezegen için ip gerilmesi hesaplanmıştır.(Şekil 2.3.8.1 ve şekil 2.3.8.2)**

```c
void ip_gerilmesi(const double *yercekimi_ptr) {
    double m,T;
printf("\n--- Sabit Ip Gerilmesi Deneyi ---\n");
    printf("Cismin kütlesini kilogram cinsinden giriniz: ");
    scanf("%lf", &m);
m = (m < 0) ? -m : m;
printf("Girilen cisim kutlesi (mutlak deger): %.2f kg\n\nSimulasyon sonuclari:\n", m);
for (int i = 0; i < gezegen_sayisi; i++) {
double g = *(yercekimi_ptr + i);
T = m*g;
printf("%-10s -> Ip gerilmesi: %.2f Newton\n", *(gezegen_isim + i), T);
}}
```

Şekil 2.3.8.1

```
--- Sabit Ip Gerilmesi Deneyi ---
Cismin kütlesini kilogram cinsinden giriniz: 4
Girilen cisim kutlesi (mutlak deger): 4.00 kg

Simulasyon sonuclari:
Merkur     -> Ip gerilmesi: 14.36 Newton
Venus      -> Ip gerilmesi: 35.48 Newton
Dunya      -> Ip gerilmesi: 39.24 Newton
Mars       -> Ip gerilmesi: 15.08 Newton
Jupiter    -> Ip gerilmesi: 103.80 Newton
Saturn     -> Ip gerilmesi: 44.32 Newton
Uranus     -> Ip gerilmesi: 42.68 Newton
Neptun     -> Ip gerilmesi: 56.28 Newton
```

Şekil 2.3.8.2

- **2.3.9 Asansör Deneyi:Bu deneyde ivmeli hareket yapan bir asansör içindeki cismin etkin ağırlığı (hissedilen ağırlık) hesaplanmaktadır.Kullanıcıdan cismin kütlesi, asansörün ivmesi ve hareket yönü bilgileri alınmış;eğer asansör yukarı yönde hızlanıyorsa N=m(g+a),aşağı yönde hızlanıyorsa N=m(g-a) ile her gezegen için cismin etkin ağırlışı hesaplanmıştır.(Şekil 2.3.9.1 ve şekil 2.3.9.2)**

```
printf("\n--- Asansör Deneyi ---\n");
    printf("Cismin kütlesini kilogram cinsinden giriniz: ");
    scanf("%lf", &m);
m = (m < 0) ? -m : m;
    printf("Asansorun ivmesini m/s^2 cinsinden giriniz: ");
    scanf("%lf", &a);
a = (a < 0) ? -a : a;
printf("\n1. Yukari yonlu hizlanan (veya asagi yavaslayan)\n2. Asagi yonlu hizlanan (veya yukari yavaslayan)\n");
printf("\nAsansorun hareket yonunu seciniz:");
scanf("%d", &yon);
printf("Girilen cisim kutlesi (mutlak deger): %.2f kg,ivmesi %.2f m/s\n\nSimulasyon sonuclari:\n", m,a);
for (int i = 0; i < gezegen_sayisi; i++) {
double g = *(yercekimi_ptr + i);
N= (yon == 1) ? (m*(g+a)) : (m*(g-a));
printf("%-10s -> Etkin agirlik: %.2f Newton\n", *(gezegen_isim + i), N);
}}
```

Şekil 2.3.9.1

```
--- Asansör Deneyi ---
Cismin kütlesini kilogram cinsinden giriniz: 15
Asansorun ivmesini m/s^2 cinsinden giriniz: 5

1. Yukari yonlu hizlanan (veya asagi yavaslayan)
2. Asagi yonlu hizlanan (veya yukari yavaslayan)

Asansorun hareket yonunu seciniz:1
Girilen cisim kutlesi (mutlak deger): 15.00 kg,ivmesi 5.00 m/s

Simulasyon sonuclari:
Merkur     -> Etkin agirlik: 128.85 Newton
Venus      -> Etkin agirlik: 208.05 Newton
Dunya      -> Etkin agirlik: 222.15 Newton
Mars       -> Etkin agirlik: 131.55 Newton
Jupiter    -> Etkin agirlik: 464.25 Newton
Saturn     -> Etkin agirlik: 241.20 Newton
Uranus     -> Etkin agirlik: 235.05 Newton
Neptun     -> Etkin agirlik: 286.05 Newton
```

Şekil 2.3.9.2

- **2.4 Hata yönetimi ve girdi düzeltme**
- **a.Switch-case yapısında hatalı değer girilmesi halinde default çalışır,hata mesajı gösterilir ve tekrar tablo gelerek seçim yapılması istenir.(Şekil 2.4.1 ve şekil 2.4.4)**
- **b.Her fonksiyonun kendi içinde kullanıcıdan aldığı değerler ternary kullanılarak mutlak değleri alınır.(Şekil 2.4.2)**
- **c.Girdi alırken string girilirse input buffer yüzünden sonsuz döngü olmaması için getchar ile temizleniyor.(Şekil 2.4.3)**

```
switch(secim){//menüde uygun seçime göre fonksiyona yönlendirme
    case -1:printf("Program sonlandiriliyor...\n");
            break;
    case 1:
        serbest_dusme(yercekimi_sabit);
        break;
    case 2:
        yukari_atis(yercekimi_sabit);
        break;
    case 3:
        agirlik(yercekimi_sabit);
        break;
    case 4:
        k_pot_enerji(yercekimi_sabit);
        break;
    case 5:
        h_basinc(yercekimi_sabit);
        break;
    case 6:
        a_kaldirma_kuvveti(yercekimi_sabit);
        break;
    case 7:
        sarkac_periyodu(yercekimi_sabit);
        break;
    case 8:
        ip_gerilmesi(yercekimi_sabit);
        break;
    case 9:
        asansor(yercekimi_sabit);
        break;
    default:printf("Gecersiz secim!Lütfen tekrar deneyin.\n");
    break;
```

Şekil 2.4.1

```
void serbest_dusme(const double *yercekimi_ptr) {
    double t, h;
printf("\n--- Serbest Dusme Deneyi ---\n");
    printf("Dusus suresini saniye cinsinden giriniz: ");
    scanf("%lf", &t);
t = (t < 0) ? -t : t;
printf("Girilen sure (mutlak deger): %.2f saniye\n\nSimulasyon sonuclari:\n", t);
for (int i = 0; i < gezegen_sayisi; i++) {
double g = *(yercekimi_ptr + i);
h = 0.5 * g * (t * t);
printf("%-10s -> Yukseklik: %.2f metre\n", *(gezegen_isim + i), h);
}}
```

Şekil 2.4.2

```
do {//do-while ile tekrarlayan menü ve deney uygulaması
printf("\n--- DENEY LISTESI ---\n");
printf("1. Serbest Dusme Deneyi\n");
printf("2. Yukari Atis Deneyi\n");
printf("3. Agirlik Deneyi\n");
printf("4. Kütlecekimsel 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("-1. Cikis\n");
printf("Seciminiz: ");
scanf("%d", &secim);
while(getchar() != '\n');
```

Şekil 2.4.3

```
9. Asansor Deneyi
-1. Cikis
Seciminiz: 25
Gecersiz secim!Lütfen tekrar deneyin.

--- DENEY LISTESI ---
1. Serbest Dusme Deneyi
```

Şekil 2.4.4

3.Eksiklikler ve geliştirmeler

3.1.1 Verilerin dosyaya kaydı ve geri çağırılması

Simulasyon sonuçlarının bir metin belgesi veya tablo halinde saklanması ve istendiğinde geri dönülebilmesi sağlanabilirdi.Ancak konu hakkında bilgi yetersizliği,anlık gösterimin kısa vaadede yeterli olması ve ödev dökümanında belirtilmemesi sebebiyle eklenmedi.

3.1.2 Grafik kullanıcı arayüzü(GUI) desteği

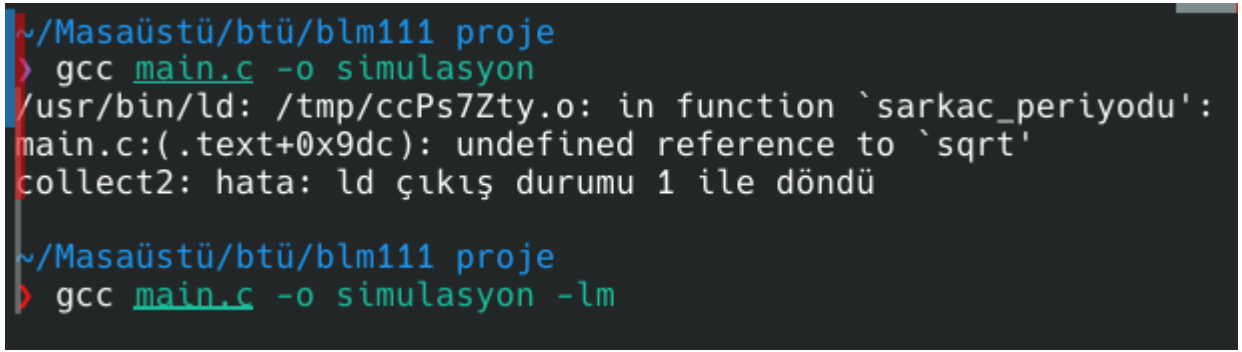
Program bir grafik arayüzüne sahip olması sağlanabilirdi.Bu sayede hem daha kullanıcı dostu bir görünüm hem de basit animasyonlarla yapılan deneyin daha iyi anlaşılması mümkün olurdu.Ancak ödevin amacından çok saptığından ve bilgi eksikliğinden ötürü eklenmedi.

3.1.3 Yeni dinamik gezegen ekleme

Programa kullanıcının güneş sistemiyle sınırlı kalmayıp kendi istediği gezegenleri ekleyebilmesi,bunların bellekte veya daha sonra kullanım için bir dosyada saklanması sağlanabilirdi.Bu sayede kullanıcı güneş sistemi dışındaki veya hayali gezegenler ile de deneyleri gerçekleştirebilirdi.Ancak ödev dökümanında belirtilmemesi sebebiyle eklenmedi.

3.2 Karşılaşılan zorluklar

- Kullanıcı deney seçiminde integer yerine string girmesiyle oluşan input buffer sorunu araştırma sonucuyla çözülmüştür.
- Linux'da gcc'nin otomatik olarak matematik kütüphanesi fonksiyonlarını bağlamaması sebebiyle kodun derlenememesi sorunu komut sonuna veya derleyiciye -lm argümanı eklenmesiyle çözüldü. (Şekil 3.2.1)



Şekil 3.2.1

4.Sonuç

- Bu proje kapsamında programın kodlanmasında klasik dizi mantığı yerine pointer ve pointer aritmetiği ile ilerlenmiştir.Programda gezegen verilerinin ve simülasyon parametrelerini pointer'lar ile alındı ve kullanıldı.Bu da C dilinde pointer ve pointer aritmetiğinin gücünü ortaya koymuştur.Ayrıca program modüler bir şekilde geliştirildiğinden okunma kolaylığı ve kolayca yeni deney eklenebilmesi gibi avantajlar sağlamaktadır.
- Sonuç olarak bu proje pointerlar,ternary ve modüler program yazma konusunda faydalı bir örnek sunmaktadır.Ayrıca ödev süresinin (3 gün) kısa olması ilerde iş hayatında hızlı yapılması gereken acil görevler için güzel bir alıştırmadır.

Kaynakça

Aerospaceweb.org. (n.d.). *Gravity on other planets.*

https://aerospaceweb.org/question/astronomy/q0227.shtml

Arch Linux Forums. (2022). *Undefined reference to sqrt.*

https://bbs.archlinux.org/viewtopic.php?id=275202

Stack Overflow. (2011). *How can I clear an input buffer in C?*

https://stackoverflow.com/questions/7898215/how-can-i-clear-an-input-buffer-in-c