

KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



KÜRESEL AYNALIRIN SİMALASYONU

TASARIM PROJESİ

Hüseyin SARI

Sabri ARAZ

Tugay İRMİŞ

2015-2016 GÜZ DÖNEMİ

KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

KÜRESEL AYNALIRIN SİMALASYONU

TASARIM PROJESİ

Hüseyin SARI

Sabri ARAZ

Tugay İRMİŞ

Bu projenin teslim edilmesi ve sunulması tarafımdan uygundur.

Danışman : Yrd.Doç Hüseyin Pehlivan

.....

ÖNSÖZ

“Küresel aynaların simülasyon” konulu bu çalışma, Karadeniz Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü’nde “Tasarım projesi” olarak hazırlanmıştır.

Çalışmamızın tamamlanması sırasında optik konusunda bize bilgi veren Fizik Bölümü Hocam Prof. Dr. Emin BACAKSIZ’a, yardımlarını esirgemeyen Sayın Hocam Yrd.Doç Hüseyin PEHLİVAN’a, 4 yıl boyunca iyi birer mühendis olarak yetişmemiz için emek veren bütün Karadeniz Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü hocalarımıza teşekkür ederiz.

İlk günden bugüne, Bilgisayar Mühendisi olarak mezun olmamıza kadar olan süreçte hiçbir zaman yalnız bırakmayan öncelikle ailemize sonra da arkadaşlarımıza saygılarımızı ve sevgilerimizi sunarız.

Hüseyin SARI

Tugay İRMİŞ

Sabri ARAZ

İçindekiler

ÖNSÖZ.....	i
İÇİNDEKİLER	ii
GİRİŞ.....	1
1.KÜRESEL AYNALAR	2
1.2.2 ÇUKUR AYNALARDA GÖRÜNTÜ	4
1.2.1TÜMSEK AYNALAR.....	6
1.2.2 TÜMSEK AYNADA GÖRÜNTÜ.....	6
4.İŞINLARIN ÇİZDİRİLMESİ	8
5.CİSMİN HAKERETİ.....	10
6.AYNANIN YARI ÇAPINI DEĞİŞTİRME	11
7.KAYNAKLAR.....	12

GİRİŞ

Bu projede fizik derslerinde anlatılan küresel aynalar konusunun simülasyonu yapıp konunun daha kolay ve akılda kalıcı şekilde olmasını amaçladık. Küresel aynalarda gerçek veya sanal olan görüntü hesaplamaları da görsel olarak oluşturulmuştur. Yansıması olacak cismin ve ya yansıyan cismin de şekli ve boyutu değiştirilebilir niteliklerde olabileceği varsayımı da yapılmıştır. Görsel programlama ile birlikte küresel aynalar konusunun daha çok çizimlerle ve formüller ile hesaplanan bu kısmını, projemizde birleştirmeye çalıştık. Projemizde, anlatılmak istenilen konuda görsellik ve hesaplama ön planda olacağından kullanılacak olan yöntemler ve formüllerin daha kolay ve anlaşılır olacağını düşündüğümüz için programlama dili olarak C++, görsellik açısından da OPENGL kullandık. Daha önce yapılan çalışmalarda bizim rastladığımız, görsellikten çok hesaplamaya önem verilmiş olduğunu gördük ve projemizde hem hesaplamaya hem de görselliğe önem vermeye çalıştık. Hesaplamaları ve çizimleri yaparken gerek fizik notlarımızda gerekse fizik dersi hocalarımızdan yardım aldık.

1-KÜRESEL AYNALAR

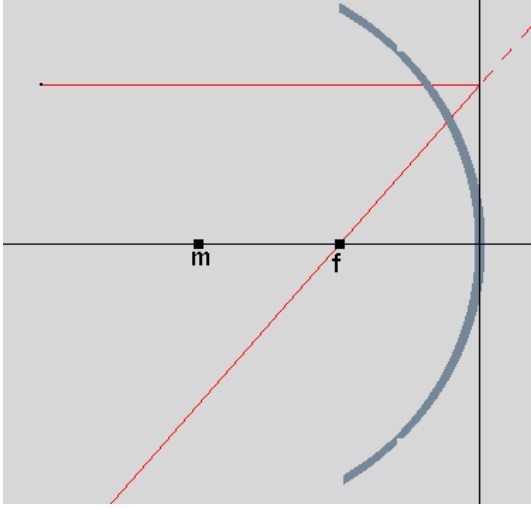
Küresel Aynalar , düz aynadan farklı olarak eğriliğe sahiptirler. Ve bu eğrilik görüntüde değişikliğe sebep olur. Işığı yansıtan bir kürenin iki özdeş parçaya ayrılması ile elde edilen aynalardır. Aynanın dönel simetri eksenine "asal eksen", asal eksen ile çukur aynanın birleştiği noktaya "tepe noktası" adı verilir. Bu aynayı oluşturan yarım kürenin ve aynı zamanda aynanın "eğrilik yarıçapı"; "R", aynanın içinden geçebilecek hayali en büyük yarıçaplı çemberin merkezi; "ayna merkezi"; ve asal eksene paralel ışınların aynadan yansdıktan sonra asal eksen üzerinde kesiştikleri nokta;"odak noktası"; "F" harfleri ile gösterilirler.Odak noktası $F = R/2$ birimdir.

1.1.1 ÇUKUR AYNALAR

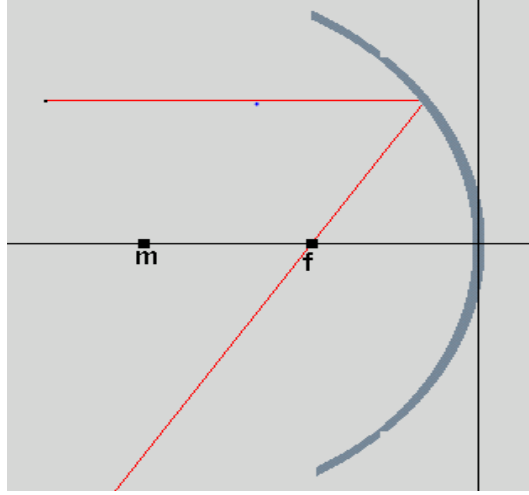
- ✓ Asal eksene paralel gelen herhangi bir ışın, odakta geçecek şekilde yansır.
- ✓ Tepe noktasına gelen ışın, asal eksene geldiği açı ile, asal eksenin diğer tarafına yansır.
- ✓ Merkezden gelen Işın, merkezden geçecek şekilde ve kendisiyle çakışık biçimde yansır.Işığın yansıma ilkesi gereği, kaynak ve yansıyan ışınlar yer değiştirdiklerinde de verilen kurallara uydukları görülür.
- ✓ Aynanın sonsuz küçüklükteki herhangi bir teğeti, kendi yüzeyi içinde bir düz ayna gibi davrandığından üzerine düşen ışını, kendisini merkez ile birleştiren düzlem ile eşit açı yapacağı biçimde yansıtır.

Bir cismin aynada görüntüsünü görmek için; her bir noktadan en az 2 özel ışını aynaya yollamamız gerekir aynadan yansıyan ışınların kesişimi cismin görüntüsünü oluşturur.

Projemizde cismin görüntüsünü bulmak için asal eksene paralel gelen herhangi bir ışın, odakta geçecek şekilde yansır ve tepe noktasına gelen ışın, asal eksene geldiği açı ile, asal eksenin diğer tarafına yansıma özelliğini kullanarak cismin görüntüsünü oluşturduk.



Şekil 1.1

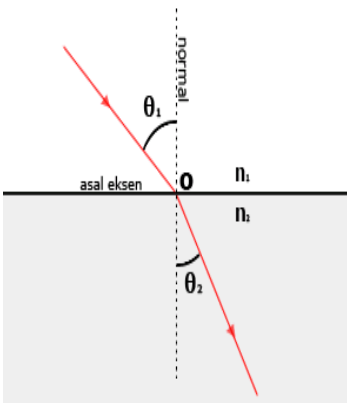


Şekil 1.2

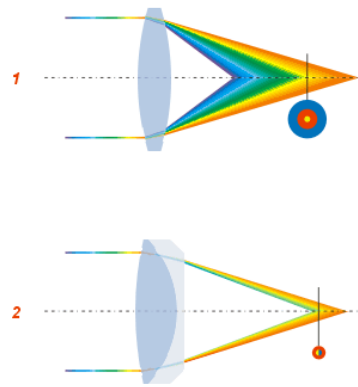
Fizik dersinde bize anlatılan; "aynaya paralel gelen ışın odakta geçecek şekilde yansır(Şekil 2.1)"ifadesinin aslında ideal bir kavram olduğunu ve mümkün olmasının zor olduğunu açıklamaya çalışacağız. Yaptığımızın aslında"asal eksene paralel gelen ışınodaktan geçecek şekilde yansır (Şekil2.2)" ifadesini kullanarak projemizi yaptık.

$$\frac{\sin \theta_1}{\sin \theta_2} = \frac{v_1}{v_2} = \frac{n_1}{n_2}$$

Snell yasasına göre;ışığın geldiği ortamın kırıcılık geliş doğrultusunun normalle yaptığı açının sinüsünün, ışığın gittiği ortamın kırıcılık indisiyile gidiş doğrultusunun normalle yaptığı açının sinüsüyle çarpımına eşitlenmesiyle oluşan formüle dayalı fiziğin optik dalında yer alan bir yasadır.



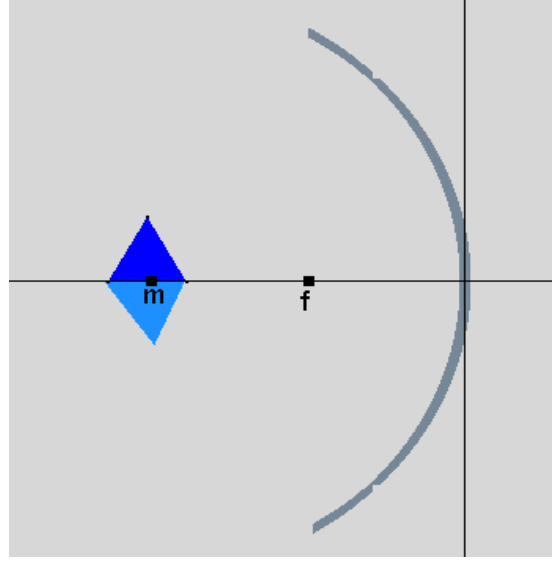
Şekil 2.3



Şekil 2.4

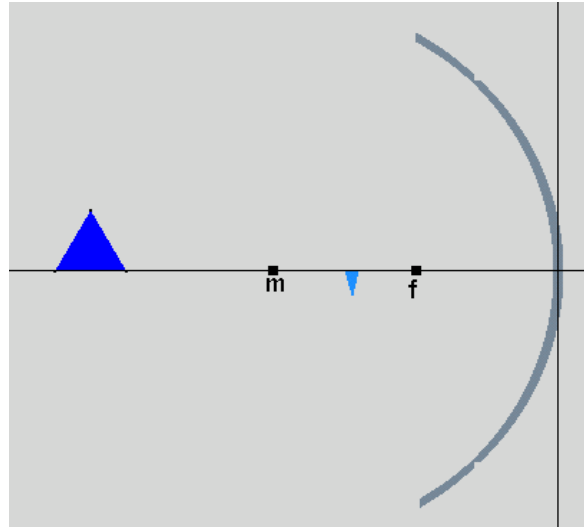
Snell yasasında açının tek bir değerine göre ışın kırılması hesaplanarak; paralel gelen ışın aynadan yansır ifadesini kullanıyoruz fakat bu ifadeyi kullandığımızda paralel gelen ışın tepe noktasından uzaklaştıkça açı değişeceğinden odak noktasının da yerini değişmesi gerekir bu bir mercek kusurudur.(Şekil 2.4)

1.2.2 ÇUKUR AYNALARDA GÖRÜNTÜ



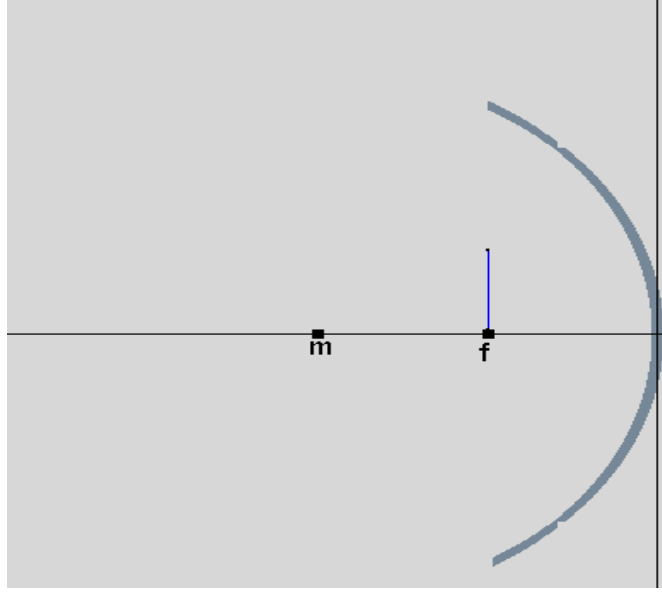
Şekil 1.5

Merkezde bulunan cismin görüntüsü (Şekil2.5) ; Merkezde bulunan cismin görüntüsü gerçek görüntü oluşturur.Oluşan görüntü, cisim ile aynı büyüklükte ve görüntüye terstir.Görüntü merkezde oluşur. Çukur aynaya baktığımızda kendimizi ters ve aynı büyüklükte gördüğümüz nokta, merkezdir.



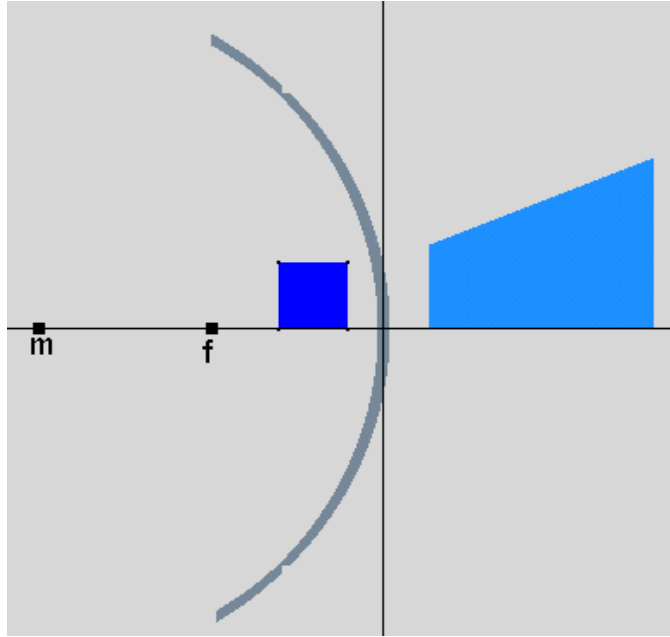
Şekil 2.6

Merkez ile odak arasında bulunan cismin görüntüsü (Şekil 2.6);Merkez ile odak arasında bulunan cismin görüntüsü gerçek, cismin boyundan büyük ve görüntü terstir.Görüntü merkezin dışında oluşur.



Şekil 2.7

Odakta bulunan cismin görüntüsü(Şekil 2.7);Odakta bulunan cismin görüntüsü sonsuzda oluşur. Işınlar kesişmez.



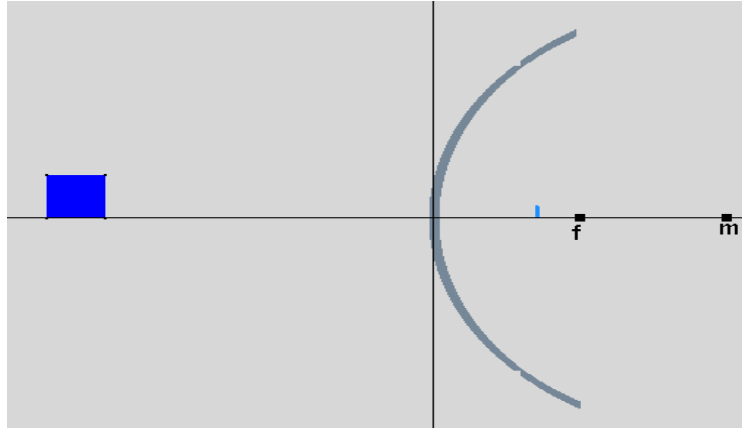
Şekil 2.8

Tepe noktası ile odak arasında bulunan cismin görüntüsü(Şekil2.6);Tepe noktası ile odak arasında bulunan cismin görüntüsü zahiridir(sanaldır,gerçek değildir).Görüntü düz ve boyu cismin boyundan büyüktür.Görüntü aynanın arkasında oluşur.

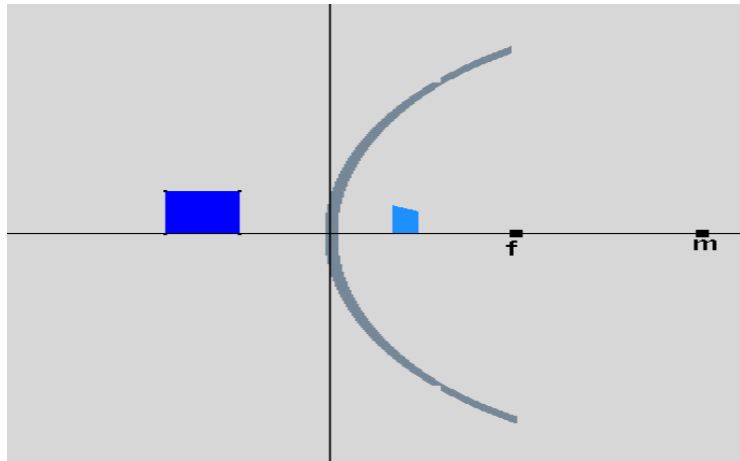
1.2.1 TMSEK AYNALAR

- ✓ Asal eksene paralel gelen ışınlar, uzantısı odaktan geçecek şekilde yansır.
- ✓ Uzantısı odaktan geçecek şekilde gelen ışın, asal eksene paralel yansır.
- ✓ Merkeze gelen ışın, geldiđi dođrultuda geri yansır.
- ✓ Tepe noktasına gelen ışın, yine aynı açı ile yansır.

1.2.2 TMSEK AYNADA GRNT



Şekil 2.9



Şekil 2.10

Tmsek ayna nnde bir cisim her nerede olursa olsun grnts her zaman odak ile tepe noktası arasında oluşur. Grnt daima dz, daima sanal ve daima cismin boyundan kktr.

2.CİSMİ OLUŞTURMA

Click* curr = top;

```
glColor3f(1, 1, 1);
if (stack_count == 2){
    glBegin(GL_LINES);
}
else if (stack_count == 3){
    glBegin(GL_TRIANGLE_FAN);
}
else if (stack_count == 4){
    glBegin(GL_QUADS);
}
else{
    glBegin(GL_POLYGON);
}
while (curr) {
    float x = curr->position[0] + cisX;
    float y = curr->position[1] + cisY;
    glVertex2f(x, y);
    curr = curr->next;
}
glEnd();
```

Listboxdan seçtiğimiz cisim türüne göre önceden yığına eklediğimizi noktaları ekranda çizdirilerek cisim oluşur ayrıca maus tıklamasıyla dinamik olarak cisim oluşur. Tıkladığımız noktaların kordinatları yığına eklenir. Oluşturulan yığın veri yapısından çekilen nokta sayısı 2 ise doğru çizimi, 3 ise üçgen çizimi, 4 ise kare çizimi ve daha fazlaysa poligon çizimi yaptırılır.Yığın alınan noktalar *glVertex2f(x, y)* fonksiyonu verilerek cisim çizdirilir.

3. GÖRÜNTÜYÜ OLUŞTURMA

```
curr = top;
glColor3f(0, 0.74902, 1);
if (stack_count == 2){
    glBegin(GL_LINES);
}
else if (stack_count == 3){
    glBegin(GL_TRIANGLE_FAN);
}
else if (stack_count == 4){
    glBegin(GL_QUADS);
}
else{
    glBegin(GL_POLYGON);
}
}
```

```

while (curr) {
    float x = curr->position[0] + cisX;
    float y = curr->position[1] + cisY;
    Goruntu(x, y);
curr = curr->next
}
glEnd();

```

Oluşturulan yığın veri yapısından çekilen nokta verileri sayısı 2 ise doğru çizimi, 3 ise üçgen çizimi, 4 ise kare çizimi ve daha fazlaysa poligon çizimi yaptırılır. Yığın alınan noktalar **Goruntu(x, y)** fonksiyonu verilerek görüntü çizdirilir.

```

void Goruntu(float x, float y){
    float gorX, gorY;
    gorX = (f*f) / (x - f) + f;
    gorY = -((y*f) / (x - f));

    glVertex2f(gorX, gorY);

}

```

Üçgende benzerlik bağıntıları kullanılarak verilen cismin noktaları için oluşacak cisim noktaları hesaplanır.

4. IŞINLARIN ÇİZDİRİLMESİ

```

curr = top;
glColor3f(1, 0, 0);
for (int i = 0; i < kirmizi_for_iMax; i++) {
    float x = curr->position[0] + cisX;
    float y = curr->position[1] + cisY;
    Kirmizi_Isin(x, y);
    curr = curr->next
}

```

```

void Kirmizi_Isin(float x, float y){
if (kirmizi_isin){
    glEnable(GL_LINE_STIPPLE);
    glLineStipple(1, 0xff);
    glBegin(GL_LINE_STRIP);
    glVertex2f(0, y);
    glVertex2f(sonsuz, -(sonsuz*y - f*y) / f);
    glEnd();
    glDisable(GL_LINE_STIPPLE);
}
}

```

```

if (r < 0){
    glBegin(GL_LINE_STRIP);
    glVertex2f(x, y);
    glVertex2f(0, y);
    glEnd();
    glBegin(GL_LINE_STRIP);
    glVertex2f(0, y);
    glVertex2f(-sonsuz, (sonsuz*y - f*y) / f);
    glEnd();
}

else{
    glBegin(GL_LINE_STRIP);
    glVertex2f(x, y);
    glVertex2f(0, y);
    glVertex2f(f, 0);
    glVertex2f(-sonsuz, (sonsuz*y - f*y) / f);
    glEnd();
}
}
}

```

Yığındaki noktaların sayısına göre kırmızı ışınların sayısını belirlenir. Rdeğeri 0'dan küçük ise tümsek aynadaki r değeri 0'dan büyükse çukur aynanın kırmızı ışınlar çizdirilir. ***Kırmızı Isin(x, y) fonksiyonuyla*** asal eksene paralel gelen ışın, odak noktasından geçecek şekilde çizdirilir. ***glLineStipple(1, 0xff)*** fonksiyonuyla çizilen kırmızı ışının kesikli olması sağlanılır.

```
curr = top;
```

```
glColor3f(1, 0, 0);
```

```
for (int i = 0; i < yesil_for_iMax; i++) {
    float x = curr->position[0] + cisX;
    float y = curr->position[1] + cisY;
    Yesil_Isin(x, y);
    curr = curr->next;
}

```

```
void Yesil_Isin(float x, float y){
```

```
if (yesil_isin){
```

```
glColor3f(1, 1, 0);
glBegin(GL_LINE_STRIP);
glVertex2f(x, y);
glVertex2f(0, 0);
glVertex2f(x, -y);
glVertex2f(100 * x, -100 * y);
glEnd();
}
}

```

```

        glEnable(GL_LINE_STIPPLE);
        glLineStipple(1, 0xff);

        glBegin(GL_LINE_STRIP);
glVertex2f(0, 0);
        glVertex2f(-100 * x, +100 * y);
        glEnd();
        glDisable(GL_LINE_STIPPLE);

    }

```

Yığındaki noktaların sayısına göre yeşil ışın sayısı belirlenir. Rdeğeri 0 dan küçük ise tümsek aynadaki r değeri 0 büyükse çukur aynanın yeşil ışınlar çizdirilir. Yeşil_Işın (x, y) *fonksiyonuyla* tepе noktasına gelen ışın eşit açı yaparak yansıyan ışını çizdirilir. *glLineStipple(1, 0xff)* fonksiyonuyla çizilen yeşil ışının kesikli olması sağlanır.

5.CİSMİN HAKERETİ

```

void SpecialKeys(int key, int x, int y){

switch (key) {

    case GLUT_KEY_RIGHT:
        cisX += 1.5f;
        break;
    case GLUT_KEY_LEFT:
        cisX -= 1.5f;
        break;
    case GLUT_KEY_DOWN:
        cisY -= 1.5f;
        break;
    case GLUT_KEY_UP:
        cisY += 1.5f;
        break;

    }

    puKeyboard(key + PU_KEY_GLUT_SPECIAL_OFFSET, PU_DOWN);
    glutPostRedisplay();
}

```

Cismin x ve y eksenlerin de hareketi için cismin x ve y koordinat ekleme ve çıkarma yapılarak cismin hareketini sağladık.Klavyeden ok tuşlarını kullanarak cismin hareketi sağlanır.

6. AYNANIN YARI ÇAPINI DEĞİŞTİRME

```
void NormalKeys(unsigned char key, int x, int y){  
  
    switch (key)  
    {  
    case 27:  
        exit(0);  
        break;  
    case '+':  
        r += 1.1f;  
  
        break;  
    case '-':  
        r -= 1.1f;  
        break;  
  
    default:  
        break;  
    }  
    puKeyboard(key, PU_DOWN);  
    glutPostRedisplay();  
}
```

Aynanın yarı çapını(r) değerinin artırılmasını + tuşuyla , azaltılmasını - tuşuyla yaptık. Aynanın yarı çapının değişmesi odak noktasının büyüüp-küçülmesini sağlar.

7. KAYNAKLAR

<http://stackoverflow.com/>

https://nccastaff.bournemouth.ac.uk/jmacey/RobTheBloke/www/opengl_programming.html

www3.ntu.edu.sg/home/ehchua/programming/opengl/CG_Introduction.html#zz-1

<http://eryms.com/opengl-kutuphanesi/>

<https://tr.wikipedia.org/>

www.fizik.net.tr