



*Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü
2021-2022 Bahar Dönemi - Bilgisayar Mühendisliğinde Gelişmeler IV*

METEOR VURMA

Levent Buğdaycı
02175076054

Özet	2
Giriş	3
Proje Kapsamı	3
Uygulama Geliştirme	3
4.1 Kullandığım Kütüphaneler	4
Programlama	4
5.1 Arka Plan Hareketi	4
5.2 Uzay Aracının Hareketi	5
5.3 Uzay Aracı İle Meteorların Çarpışması	6
5.4 Lazerin Hareketi	7
5.5 Lazer ile Meteor Çarpışması	8
Oyun İçi Görüntüler	9
6.1 Giriş Ekranı	9
6.2 Hakkımda	10
6.3 Kontroller	11
6.4 Oynanış	12
6.5 Market	13
Kaynakça	13

1. Özet

Oyunun amacı oynayan kişinin boş vaktini eğlenceli bir şekilde geçirmesini sağlamaktır.

Oyunun hedef kitlesi çocuklardır.

Öğretim tekniği hızlı algılama.

Kullanılan Program : IntelliJ IDEA

Kullanılan yazılım dili : Java

2. Giriş

Konusu uzay boşluğunda hareket halindeki aracın karşısına çıkan engelleri aşarak ilerleyen basit 2 Boyutlu atari oyunu.
Oyunun hedef kitlesi çocuklardır.

3. Proje Kapsamı

Oyun projemin amacı Java kullanarak 2 boyutlu bir bilgisayar oyunu tasarlamayı öğrenmektir. Oyun sonsuz döngülü. 2 boyutlu bir uzay oyunudur. Oyunda uzay aracı olarak oynayan kullanıcı meteorların arasından meteorlara çarpmadan geçmenin bir yolunu bulması gerekiyor. Her 2 saniyede bir lazer ateşleyebiliyor ve böylece. engelleri temizleyebiliyor. Her oyunun başında belirli bir can değerine sahip olup her meteor ile çarpıştığında bu can değeri bir azalıyor. Lazer ile vurduğu her meteordan 1 altın kazanıyor ve oyunun sonunda marketten bu altınlar ile can satın alabiliyor.

4. Uygulama Geliştirme

Oyunu açtığımız zaman ekranda 3 buton bulunuyor bunlar Oyna, Hakkında ve Çıkış. Hakkında butonuna basıldığında oyun içinde kullanılması gereken tüm tuşlar yazıyor. Oyna butonuna basıldığında oyun başlıyor. Ekranın sol üst köşesinde Skorumuz Sağ üst köşesinde sağlık ve kazandığımız altının miktarı yazıyor. Oyun başladığında rastgele bir şekilde yukarıdan meteorlar geliyor uzay aracını W, A, S, D tuşları ile hareket ettirebiliyoruz. SPACE tuşuna basıldığında ise uzay aracı her iki saniyede bir olmak üzere Lazer ateşleyebiliyor. Ateşlediğimiz lazer herhangi bir meteora çarptığında ise meteor yok oluyor ve 1 altın kazanmış oluyoruz.

4.1 Kullandığım Kütüphaneler

Jaco MP3 Player
Java Util Timer
Java Awt KeyListener
Java Awt KeyEvent
Java Awt Graphics

5. Programlama

5.1 Arka Plan Hareketi

```
@Override
public void oynatma() {

    hareket = new Timer();

    hareket.scheduleAtFixedRate(new TimerTask() {
        @Override
        public void run() {

            if (GrafikSinifi.arkaplanY1 < 495) {
                GrafikSinifi.arkaplanY1 += 2;
            } else {
                GrafikSinifi.arkaplanY1 = -500;
            }

            if (GrafikSinifi.arkaplanY2 < 495) {
                GrafikSinifi.arkaplanY2 += 2;
            } else {
                GrafikSinifi.arkaplanY2 = -500;
            }

        }
    }, delay: 0, GrafikSinifi.arkaplanHiz);
```

Burada iki tane aynı olan arka plan fotoğrafını Y koordinatında (yani aşağı doğru) hareket etmesi için yazdığım bir fonksiyon. Eğer arka plan fotoğrafı ekran dışına çıkarsa tekrardan yukarı gönderilir ve sanki sonsuzmuş gibi bir görüntü elde ederiz.

5.2 Uzak Aracının Hareketi

Uzak aracı hareket ettirmek için gerekli olan şey basılan tuşu KeyListener ile algılayıp bir boolean değer ile kontrol edebilmemiz gerekiyor. Eğer “W” tuşuna basılırsa aracımız “ileri” boolean değişkeni true olur ve aracın Y koordinatından belirli bir miktar azaltırız bu ekrana araç ileri hareket gerçekleştiriyormuş gibi yansır. Fakat araç ekran dışına çıkmaması için gerekli olan bir şey daha aracın Y koordinatı 0-400 arasında, X koordinatı 10-520 arasında olmalıdır olmaz bu şartları gerçekleştirmezse araç kontrol edilemez. Diğer hareket yönleri için de aynı şeyler yapılır.

```
@Override
public void oynatma() {

    hiz = new Timer();

    hiz.scheduleAtFixedRate(new TimerTask() {
        @Override
        public void run() {
            if (GrafikSinifi.ileri == true && GrafikSinifi.aracY > 0) {
                GrafikSinifi.aracY -= aracinHizi;
            } else if (GrafikSinifi.geni == true && GrafikSinifi.aracY < 400) {
                GrafikSinifi.aracY += aracinHizi;
            } else if (GrafikSinifi.saga == true && GrafikSinifi.aracX < 520) {
                GrafikSinifi.aracX += aracinHizi;
            } else if (GrafikSinifi.sola == true && GrafikSinifi.aracX > 10) {
                GrafikSinifi.aracX -= aracinHizi;
            }
        }
    }, delay: 0, period: 15);
}
```

5.3 Uzak Aracı İle Meteorların Çarpışması

```
@Override
public void oynatma() {

    mtr = new Timer();

    mtr.scheduleAtFixedRate(() -> {
        try {
            for (int i = 0; i < 6; i++) {
                if (sayac == 0) {
                    if (GrafikSinifi.aracX >= GrafikSinifi.meteorX[i] - 40 && GrafikSinifi.aracX <= GrafikSinifi.meteorX[i] + 30
                        && GrafikSinifi.aracY >= GrafikSinifi.meteorY[i] - 40 && GrafikSinifi.aracY <= GrafikSinifi.meteorY[i] + 30) {

                        GrafikSinifi.meteorY[i] = -150;

                        GrafikSinifi.patlamaKontrol = true;

                        if (GrafikSinifi.patlamaKontrol == true) {
                            GrafikSinifi.Saglik = GrafikSinifi.Saglik - 1;
                        }
                    }
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }, delay: 0, period: 10);
}
```

Uzak aracı ile meteorun çarpışmasında bir etkileşim sağlamak için gerekli olan şey her iki cismin de koordinatlarını alabilmek. İkisinin de koordinatlarını aldığımızda bir şart ifadesi ile bunları kontrol edebilir eğer meteor x ve y koordinatı uzak aracının x ve y koordinatı ile kesişiyorsa meteoru sil, patlama efektini göster ve aracın canını bir azalt.

5.4 Lazerin Hareketi

```
@Override
public void oynatma() {

    timer = new Timer();

    timer.scheduleAtFixedRate(new TimerTask() {

        @Override
        public void run() {

            Iterator<Lazer> itr = GrafikSinifi.lazerler.iterator();

            try {
                while (itr.hasNext()) {
                    Lazer lazer = itr.next();
                    if (GrafikSinifi.lazerKontrol == true) {
                        lazer.setAtesY(lazer.getAtesY() - GrafikSinifi.lazerAzalanY);
                        if (lazer.getAtesY() < 0) {
                            itr.remove();
                        }
                    }
                    if (GrafikSinifi.astreoidCarpmaKucuk == true) {
                        itr.remove();
                        GrafikSinifi.astreoidCarpmaKucuk = false;
                        Runtime.getRuntime().gc();
                    }
                }
            } catch (Exception e) {
                System.out.println("Hata");
            }
        }
    }, delay: 0, period: 15);
}
```

Burada yazılan kod şöyle çalışıyor; Oluşturulan lazerler bir ArrayList'e ekleniyor. Eğer "SPACE" tuşuna basılırsa Lazer uzay aracının bulunduğu koordinatta oluşuyor ve belli bir hız ile Y koordinatında ilerliyor. Eğer bir meteora çarpmazsa ve ekranın en üstüne çıktığında siliniyor.

5.5 Lazer ile Meteor Çarpışması

```
@Override
public void oynatma() {

    timer = new Timer();

    timer.scheduleAtFixedRate(new TimerTask() {

        @Override
        public void run() {

            Iterator<Lazer> it = GrafikSinifi.lazerler.iterator();

            while (it.hasNext()) {
                Lazer lazer = it.next();
                for (int i = 0; i < 6; i++) {

                    try {
                        if (lazer.getAtesX() >= GrafikSinifi.meteorX[i] - 30 && lazer.getAtesX() <= GrafikSinifi.meteorX[i] + 40
                            && lazer.getAtesY() >= GrafikSinifi.meteorY[i] - 30 && lazer.getAtesY() <= GrafikSinifi.meteorY[i] + 30) {
                            GrafikSinifi.carpmaDegeri++;
                            GrafikSinifi.astreoidCarpmaKucuk = true;

                            GrafikSinifi.meteorY[i] = -100;

                            if (GrafikSinifi.astreoidCarpmaKucuk == true) {
                                GrafikSinifi.Skor = GrafikSinifi.Skor + 10;
                                GrafikSinifi.para +=1;
                            }
                            if (GrafikSinifi.astreoidCarpmaKucuk == true) {
                                if (GrafikSinifi.Skor >149) {
                                    GrafikSinifi.para += 2;

                                    if (GrafikSinifi.Skor>349) {
                                        GrafikSinifi.para +=1;
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    });
}
```

Burada gerçekleşen şey uzay aracı ile meteor çarpışmasında gerçekleşenin aynısını uygulanıyor. Eğer lazer meteor ile temas ederse skorumuz 10 ve paramız da 1 artar. Skor 150' yi geçerse oyun içi paramız vurulan her meteor için 2 artar.

6. Oyun İçi Görüntüler

6.1 Giriş Ekranı



6.2 Hakkımda

HAKKIMDA

Levent BUĞDAYCI

02175076054

KURALLAR

GERİ

6.3 Kontroller

TUŞLAR

W,A,S,D : Yön Tuşları

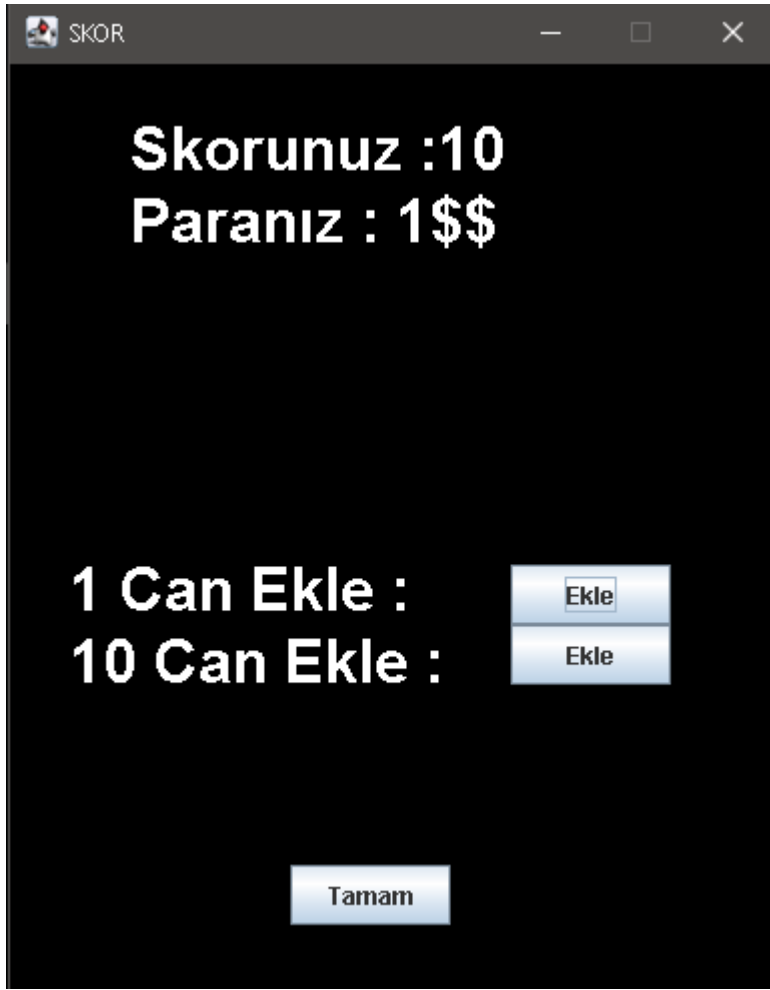
O,P Tuşları : Müzik Başlat ve Durdur

GERİ

6.4 Oynaniř



6.5 Market



7. Kaynakça

The Coders. (2016, February 8). *Java - Timer Kullanımı – The Coders*. The Coders. Retrieved April

25, 2022, from <https://www.okanuzun.com/java-timer-kullanimi/>

Coşkun, M. M. (2020). *(42 Saat) Sıfırdan Komple Java Geliştirici Kursu*. Udemy. Retrieved April 25,

2022, from

<https://www.udemy.com/course/sifirdan-ileri-seviyeye-komple-java-gelistirici-kursu/>

GeeksforGeeks. (2020, November 14). *Java.util.Timer Class in Java*. GeeksforGeeks. Retrieved April

25, 2022, from <https://www.geeksforgeeks.org/java-util-timer-class-java/>

Tutorials Point. (2020). *AWT Graphics Class*. Tutorialspoint. Retrieved April 25, 2022, from

https://www.tutorialspoint.com/awt/awt_graphics_class.htm