

**222180510 Yağmur Umutlu**

**222080558 Alperen Çevik**

**222080548 Emre Kurt**

## **Python Nedir?**

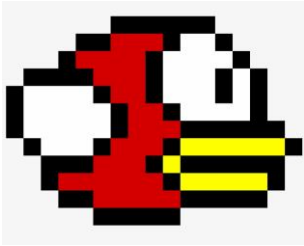
Python yüksek seviyeli, yorumlanmış ve genel kullanıma uygun bir programlama dilidir.



## **Flappy Bird Nedir?**

Flappy bird Vietnamlı programcı Dong Nguyen tarafından yapılmış ve zamanında ciddi derece ünlemiş bir mobil oyundur. Oyuncu ekrana her tıkladığında kuşun yukarı hareket etmesi ve tıklanmadığı sürede aşağı düşmesini temel alan, tüplerin arasından geçtiğimizde skor kazandığımız basit bir oyundur.

Biz de bu oyunu kendimizden birkaç şeyler katarak mekaniğini azcık değiştirerek bilgisayar versiyonunu yaptık.



## **Kullanılan programlar ve kütüphaneler?**

Python: Kullandığımız programlama dili.

Paint 3D:Oyun içinde kullanılan görsellerin oluşturulması veya düzenlenmesi için kullanılan program

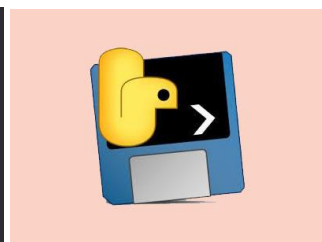
Visual Studio Code 2019: Python için Kullandığımız IDE.

auto-py-to-exe : Python programları .exe dosyasına dönüştüren program.

pygame :Pythonun oyun kütüphanesi.

Random: Rastgele sayılar elde etmek için kullanılan Python kütüphanesi.

os : Pythonun türlü türlü işletim sistemi öğeleri içeren kütüphanesi.



## Yardım Aldığımız Kaynaklar?

Youtube çeşitli tutorial videoları

pygame ve python tutorialları

[docs.python.org](https://docs.python.org)

[icons8.org](https://icons8.org)

[pygame.org](https://pygame.org)

[freesound.org](https://freesound.org)

[rapidtables.org](https://rapidtables.org)

[geeksforgeeks.org](https://geeksforgeeks.org)

[freecodecamp.org](https://freecodecamp.org)

[github.com](https://github.com)

[learnsaylor.org](https://learnsaylor.org)

[pythonprogramming.net](https://pythonprogramming.net)

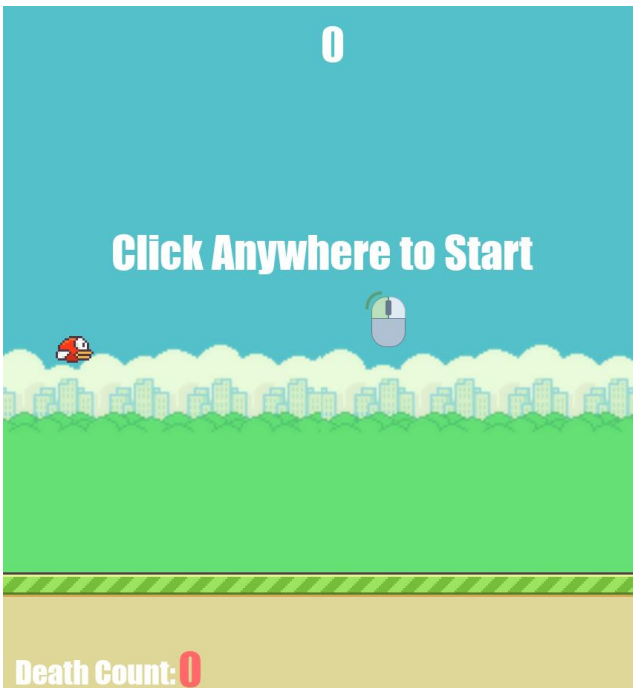
[opengameart.org](https://opengameart.org)

[stackoverflow.com](https://stackoverflow.com)

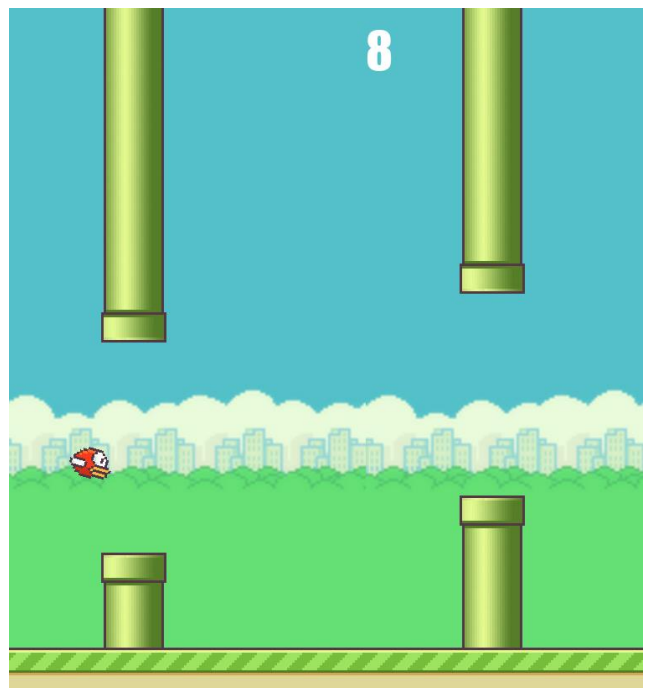
[opensource.com](https://opensource.com)

## Oyundan Kesitler

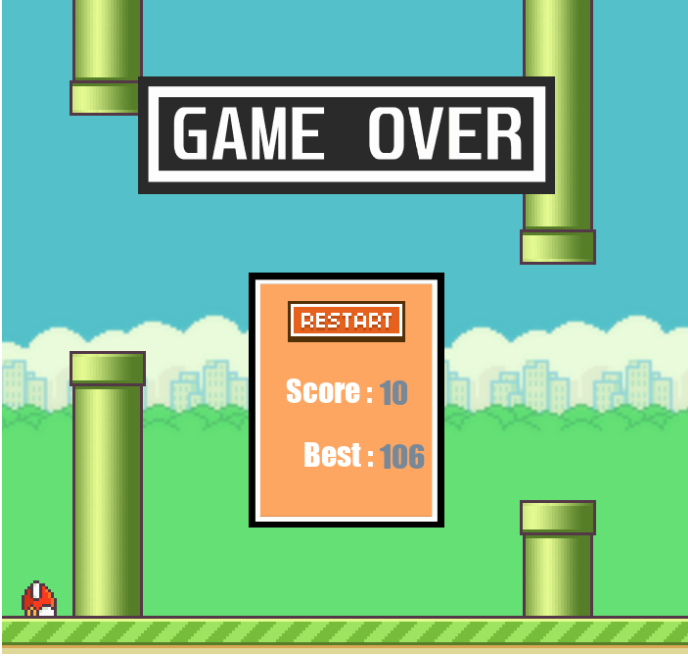
Başlangıç ekranı



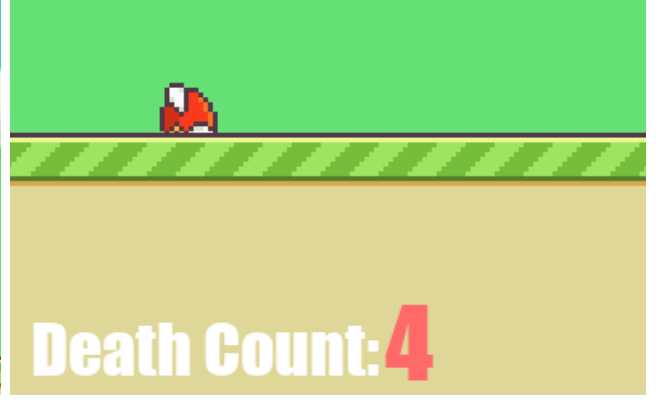
Oyun içi görüntü



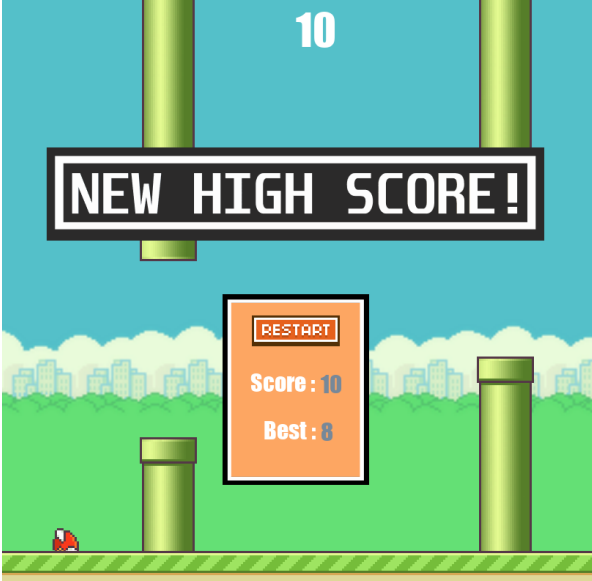
Yanma ekranı



Ölüm sayısı



Yeni yüksek skor ekranı



Yüksek skorun kaydedildiği  
dosya konumu

|                             |                    |                      |
|-----------------------------|--------------------|----------------------|
| img                         | 5/16/2022 1:05 AM  | File folder          |
| pygame                      | 5/16/2022 1:05 AM  | File folder          |
| pyinstaller-5.0.1.dist-info | 5/16/2022 1:05 AM  | File folder          |
| saves                       | 5/16/2022 1:05 AM  | File folder          |
| setuptools-57.4.0.dist-info | 5/16/2022 1:05 AM  | File folder          |
| sound                       | 5/16/2022 1:05 AM  | File folder          |
| _asyncio.pyd                | 5/16/2022 12:53 AM | Python Extension ... |
| _bz2.pyd                    | 5/16/2022 12:53 AM | Python Extension ... |
| _ctypes.pyd                 | 5/16/2022 12:53 AM | Python Extension ... |
| decimal.pyd                 | 5/16/2022 12:53 AM | Python Extension ... |

## Programın Kodları ve Kodların işlevleri?

### # Kütüphanelerin import komutuyla kod alanına eklenmesi

```
import pygame
from pygame.locals import *
import random
import os
```

### # pygame'e ait genel ve ses için başlatma fonksiyonları

```
pygame.init()
pygame.mixer.init()
```

### # Oyunun kare hızının ayarlanması

```
clock = pygame.time.Clock()
fps = 60
```

### # Pencere boyutunun piksel olarak belirlenmesi

```
screen_width = 864
screen_height = 936
```

### # Girilen piksellerin pencereye tanımlanması ve pencerenin isminin atanması

```
screen = pygame.display.set_mode((screen_width, screen_height))
pygame.display.set_caption('Flappy Bird by Omer Faruk Bilir & Alperen Cevik')
```

### # Yazı tiplerinin ve boyutlarının ayarlanması

```
font = pygame.font.SysFont('Impact', 60)
text_font = pygame.font.SysFont('Impact', 42)
menu_font = pygame.font.SysFont('Impact', 32)
```

### # Renklere enum atama

```
white = (255, 255, 255)
black = (0, 0, 0)
coral = (255, 105, 105)
orange = (255, 105, 30)
gray = (119, 136, 153)
```

# Oyun içinde kullanılacak veya yazdırılacak değişkenlerin başlangıç değerleri ile tanımlanması

```
flying = False
game_over = False
score = 0
death_count = 0
death_count_text = "Death Count:"
score_text = "Score :"
best_text = "Best :"
start_text = "Click Anywhere to Start"
scroll_speed = 4
pipe_gap = 250
pipe_frequency = 1500 #milliseconds
ground_scroll = 0
last_pipe = pygame.time.get_ticks() - pipe_frequency
pass_pipe = False
```

# En yüksek skorumuzun kaydedilen kayıt dosyasından okunması

```
savefile = open('saves/data.txt', 'r+')
high_score = int(savefile.read())
savefile.close()
```

# Ses klasörünü ve içerdiği ses dosyalarını tanımlama

```
s = 'sound'

rst_snd = pygame.mixer.Sound(os.path.join(s, 'reset.wav'))
jmp_snd = pygame.mixer.Sound(os.path.join(s, 'jump.wav'))
hscore_snd = pygame.mixer.Sound(os.path.join(s, 'highscore.mp3'))
death_snd = pygame.mixer.Sound(os.path.join(s, 'death.wav'))
```

# Kullanılacak resimlerin tanımlanması

```
bg = pygame.image.load('img/bg.png')
ground_img = pygame.image.load('img/ground.png')
button_img = pygame.image.load('img/restart.png')
gameover_img = pygame.image.load('img/gameover.png')
high_score_img = pygame.image.load('img/highscore.png')
menu_img = pygame.image.load('img/menu.png')
leftclick = pygame.image.load('img/left2.png')
```

# Ekrana yazı yazdırmak için kullanılması gereken fonksiyon

```
def draw_text(text, font, text_col, x, y):
    img = font.render(text, True, text_col)
    screen.blit(img, (x, y))
```

# Oyunu baştan başlatmak için gereken sıfırlama komutları

```
def reset_game():
    pipe_group.empty()
    flappy.rect.x = 100
    flappy.rect.y = int(screen_height / 2)
    score = 0
    return score
```

## # Kuş sınıfının tanımı, özelliklerin ve animasyonların oluşturulması

```
class Bird(pygame.sprite.Sprite):  
  
    def __init__(self, x, y):  
        pygame.sprite.Sprite.__init__(self)  
        self.images = []  
        self.index = 0  
        self.counter = 0  
        for num in range(1, 4):  
            img = pygame.image.load(f"img/bird{num}.png")  
            self.images.append(img)  
        self.image = self.images[self.index]  
        self.rect = self.image.get_rect()  
        self.rect.center = [x, y]  
        self.vel = 0  
        self.clicked = False
```

Animasyon görsellerini tutacak self.images isimli listeyi oluşturur

Index , counter, velocity ve clicked'in başlangıç değerleriyle tanımlanması.

Animasyonun for döngüsüyle ayarlanması.

Koordinat girişi için hazırlık.

```
def update(self):  
  
    if flying == True:  
        #apply gravity  
        self.vel += 0.5  
        if self.vel > 8:  
            self.vel = 8  
        if self.rect.bottom < 768:  
            self.rect.y += int(self.vel)
```

Eğer Kuş uçuyorsa vel değerini 0.5 artırarak aşağı düşmesini yani yerçekimini sağlar.

Maximum hızı 8

Vel değerine göre yukarı aşağı hareketi sağlayan komut satırı.

```
if game_over == False:  
    #jump  
    if pygame.mouse.get_pressed()[0] == 1 and self.clicked == False:  
        self.clicked = True  
        pygame.mixer.Sound.play(jmp_snd)  
        self.vel = -10  
    if pygame.mouse.get_pressed()[0] == 0:  
        self.clicked = False
```

Eğer yanmadıysak mouse basılışını algılar, clicked değerini günceller

Kuşun vel değerini değiştirerek zıplamayı sağlar ve zıplama sesini oynatır

```
#handle the animation
flap_cooldown = 5
self.counter += 1

if self.counter > flap_cooldown:
    self.counter = 0
    self.index += 1
    if self.index >= len(self.images):
        self.index = 0
    self.image = self.images[self.index]
```

Görsellerin belirli zaman aralıklarıyla listeye eklenip çıkarılmasıyla animasyonun oluşması

```
#rotate the bird
self.image = pygame.transform.rotate(self.images[self.index], self.vel * -2)
else:
    #point the bird at the ground
    self.image = pygame.transform.rotate(self.images[self.index], -90)
```

Eğer yanmadıysak kuşun vel değerine göre kafasının yukarı veya aşağı bakmasını sağlar

Eğer yandıysak kuşun kafası yere doğru bakar.

**# Boruların çok sayıda spawn edilmesi için sınıf olarak oluşturulması ve ayarları**

```
class Pipe(pygame.sprite.Sprite):

    def __init__(self, x, y, position):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.image.load("img/pipe.png")
        self.rect = self.image.get_rect()
        #position variable determines if the pipe is coming from the bottom or top
        #position 1 is from the top, -1 is from the bottom
        if position == 1:
            self.image = pygame.transform.flip(self.image, False, True)
            self.rect.bottomleft = [x, y - int(pipe_gap / 2)]
        elif position == -1:
            self.rect.topleft = [x, y + int(pipe_gap / 2)]

    def update(self):
        self.rect.x -= scroll_speed
        if self.rect.right < 0:
            self.kill()
```

Self.image ile boruya görsellik sağlar

Pozisyon 1 ise boru yukardan , -1 ise aşağıdan spawn olur.

Update fonksiyonu ile ekranın kayma hızına göre yeni borular oluşturabilir ve ekrandan çıkanları silebilir

### # Kullanılacak butonun sınıf ayarları ve özellikleri

```
class Button():
    def __init__(self, x, y, image):
        self.image = image
        self.rect = self.image.get_rect()
        self.rect.topleft = (x, y)

    def draw(self):
        action = False

        #get mouse position
        pos = pygame.mouse.get_pos()

        #check mouseover and clicked conditions
        if self.rect.collidepoint(pos):
            if pygame.mouse.get_pressed()[0] == 1:
                action = True

        #draw button
        screen.blit(self.image, (self.rect.x, self.rect.y))

        return action
```

Butonun görsel ve koordinat ayarının yapılması.

Pos değişkenine mousenin lokasyonunun atanması.

Collidepoint fonksiyonu ile mouse butonun üzerinde mi kontrol edilir.

Screen.blit ile butonu çizdirir

### # Oluşturduğumuz sınıflar türünden yeni objeler oluşturma özellik verme ve gruplandırma

```
pipe_group = pygame.sprite.Group()
bird_group = pygame.sprite.Group()

flappy = Bird(100, int(screen_height / 2))

bird_group.add(flappy)
```

Çarpışmayı ayarlayabilmek için oluşturduğumuz sınıflar farklı gruplar içine alınır.

### # Restart butonunun oluşturulması

```
#create restart button instance
button = Button(screen_width // 2 - 60, screen_height // 2 - 21, button_img)
```



### # Oyun çalışır vaziyette iken çalışacak olan sonsuz döngü

```
run = True
while run:

    clock.tick(fps)

    #draw background
    screen.blit(bg, (0,0))

    pipe_group.draw(screen)
    bird_group.draw(screen)
    bird_group.update()

    #draw and scroll the ground
    screen.blit(ground_img, (ground_scroll, 768))

    #check the score
    if len(pipe_group) > 0:
        if bird_group.sprites()[0].rect.left > pipe_group.sprites()[0].rect.left\
            and bird_group.sprites()[0].rect.right < pipe_group.sprites()[0].rect.right\
            and pass_pipe == False:
            pass_pipe = True
        if pass_pipe == True:
            if bird_group.sprites()[0].rect.left > pipe_group.sprites()[0].rect.right:
                score += 1
                pass_pipe = False
    draw_text(str(score), font, white, int(screen_width / 2), 20)

    draw_text(str(death_count_text), text_font, white, 20, int(screen_height) - 60)
    draw_text(str(death_count), font, coral, 240, int(screen_height) - 75)

    #click here to start
    if pygame.mouse.get_pressed()[0] == 0 and flying == False and game_over == False:
        draw_text(str(start_text), font, white, 150, 300)
        screen.blit(leftclick, ( (screen_width / 2) + 50 , (screen_height / 2) - 80 ))
```

Screen.blit fonksiyonu ile arka planın yer yüzünün başlangıç ekranındaki yazı ve resimlerin menü resimleri ve textleri vb şeylerin çizdirilmesi.

Draw\_text ile yazı yazdırılması.

Kuşun borunun en sol kordinatı ile en sağ kordinatı arasındaki geçiş kontrol edilerek skor kazanma işleminin yapılması.

Eğer mouse basılı değilse, kuş halihazırda uçmuyorsa ve yanmadıysak başlangıç ekranının oluşturulması

## # Zorluk ayarları

```
#difficulty adjustment
if score == 7:
    scroll_speed = 4.5
    pipe_gap = 240
if score == 17:
    pipe_gap = 230
    pipe_frequency = 1450
if score == 27:
    scroll_speed = 5
    pipe_gap = 220
    pipe_frequency = 1400
if score == 37:
    scroll_speed = 5.5
    pipe_gap = 200
    pipe_frequency = 1350
if score == 47:
    scroll_speed = 6
    pipe_gap = 190
    pipe_frequency = 1200
if score == 57:
    pipe_gap = 180
    pipe_frequency = 1200
if score == 67:
    pipe_gap = 170
    pipe_frequency = 1100
if score == 77:
    scroll_speed = 6.5
    pipe_gap = 170
    pipe_frequency = 1050
if score == 87:
    scroll_speed = 7
    pipe_gap = 160
    pipe_frequency = 1000
if score == 97:
    pipe_gap = 150
    pipe_frequency = 1000
if score == 107:
    pipe_gap = 150
    pipe_frequency = 900
if score == 117:
    scroll_speed = 7
    pipe_gap = 150
    pipe_frequency = 800
```

10 skor aralıklarla kayma hızının artırılması, boru aralıklarının azaltılması ve boru spawn sıklığının artırılması ile ilerledikçe oyunun zorlaşmasını sağlamak.

### # Çarpışma kontrolünün yapılması

```
#look for collision
if pygame.sprite.groupcollide(bird_group, pipe_group, False, False) or flappy.rect.top < 0:
    game_over = True
```

Eğer kuş ve boru grupları çarpırsa yanmamızı sağlayan blok

### # Kuş yere düştüğünde gameover oluyor ve tekrar uçamıyor ve ölüm sesini oynatıyor

```
#once the bird has hit the ground it's game over and no longer flying
if flappy.rect.bottom >= 768:
    game_over = True
    flying = False
    if high_score >= score:
        pygame.mixer.Sound.play(death_snd)
```

### # Yeni borular oluşturma

```
if flying == True and game_over == False:
    #generate new pipes
    time_now = pygame.time.get_ticks()
    if time_now - last_pipe > pipe_frequency:
        pipe_height = random.randint(-100, 100)
        btm_pipe = Pipe(screen_width, int(screen_height / 2) + pipe_height, -1)
        top_pipe = Pipe(screen_width, int(screen_height / 2) + pipe_height, 1)
        pipe_group.add(btm_pipe)
        pipe_group.add(top_pipe)
        last_pipe = time_now

    pipe_group.update()

    ground_scroll -= scroll_speed
    if abs(ground_scroll) > 35:
        ground_scroll = 0
```

Anlık tick değerini time\_now değişkenine atar

Boru uzunluğu random kütüphanesi ile rastgele seçilir.

Boru classını kullanarak btm ve top olarak 2 değişken tanımlar, bu değişkenleri boru grubuna ekler

O anki en son boru oluşturulduktan sonra tick değerine göre boru frekansından büyük olduğu zaman yeni boru spawn eder.

Update ile güncellenmesini sağlar(bu komut olmadığı zaman hareket olmuyor)

# Oyun bitti mi kontrol eden, yandıysak menüyü çıkaran ve değerleri hesaplayan daha sonrasında reset atmaya yarayan komutlar

```
#check for game over and reset
if game_over == True:

    #highest score check
    if high_score >= score:
        screen.blit(gameover_img, ( (screen_width / 2) - (426 / 2) , (screen_height / 2) - 250 ))
    else:
        screen.blit(high_score_img, ( (screen_width / 2) - (679 / 2) , (screen_height / 2) - 250 ))
        pygame.mixer.Sound.play(hscore_snd)

    screen.blit(menu_img, ( (screen_width / 2) - 100 , (screen_height / 2) - 50 ))
    draw_text(str(score_text), menu_font, white, (screen_width / 2)-62 , (screen_height / 2)+ 50 )
    draw_text(str(score), menu_font, gray, (screen_width / 2)+34 , (screen_height / 2) + 52)
    draw_text(str(best_text), menu_font, white, (screen_width / 2)-44, (screen_height / 2)+ 115)
    draw_text(str(high_score), menu_font, gray, (screen_width / 2)+34, (screen_height / 2) + 117)

    #reset button click event
    if button.draw():
        death_count += 1
        game_over = False

        pygame.mixer.Sound.play(rst_snd)

        #highest score assignment
        if high_score < score:
            savefile = open('saves/data.txt','r+')
            high_score = score
            savefile.write(str(high_score))
            savefile.close()

        #difficulty reset
        scroll_speed = 4
        pipe_gap = 250
        pipe_frequency = 1500

        score = reset_game()

for event in pygame.event.get():
    if event.type == pygame.QUIT:
        run = False
    if event.type == pygame.MOUSEBUTTONDOWN and flying == False and game_over == False:
        flying = True

pygame.display.update()
```

Skorumuz en iyi skorumuzdan yüksek ise yeni rekor ekranını getirir rekor müziğini çalmaya başlar değilse game over ekranı gelir game over müziği çalar

Restart butonuna basıldığında ölüm sayacını artırır zorluğu sıfırlar ve reset\_game fonksiyonunu çalıştırarak oyunu başlangıç konumuna getirir.

# Elektrik kesintisi, güç kaybı vb. hata durumlarında yüksek skor kaydını quit fonksiyonundan sonra yaparak daha hatasız kayıt sistemi oluşturur.

```
pygame.quit()

#highest score assignment save backup for power loss!
if high_score < score:
    savefile = open('saves/data.txt','r+')
    high_score = score
    savefile.write(str(high_score))
    savefile.close()
```

222180510 Yağmur Umutlu

222080558 Alperen Çevik      222080548 Emre Kurt