



T.C
KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ BİLGİSAYAR
MÜHENDİSLİĞİ PROGRAMI

ANİMASYONLU ÇİZİM EKRANI

Hazırlayan
UMUTCAN OĞURLU/BERKAY ÖMER AYDEMİR
230501031/230501032

DERS SORUMLUSU
Prof. Dr. Hüseyin Tarık DURU

24.11.2024

İÇİNDEKİLER

1. ÖZET (ABSTRACT)	3
2. GİRİŞ (INTRODUCTION)	3,4,5,6,7
2.1 Çizim ekranının özellikleri	
2.2 Top ekleme butonu	
2.3 Boyut seçme menüsü	
2.4 Renk seçme menüsü	
2.5 Start butonu	
2.6 Stop butonu	
2.7 Reset butonu	
2.8 Hızlan(Speed up) butonu	
2.9 Hareket animasyonları	
3. YÖNTEM (METHOD)	8
4. SONUÇ VE ÖĞRENİLEN DERSLER	8
5. KAYNAKÇA	8
6. GİTHUB LİNKLERİ	8

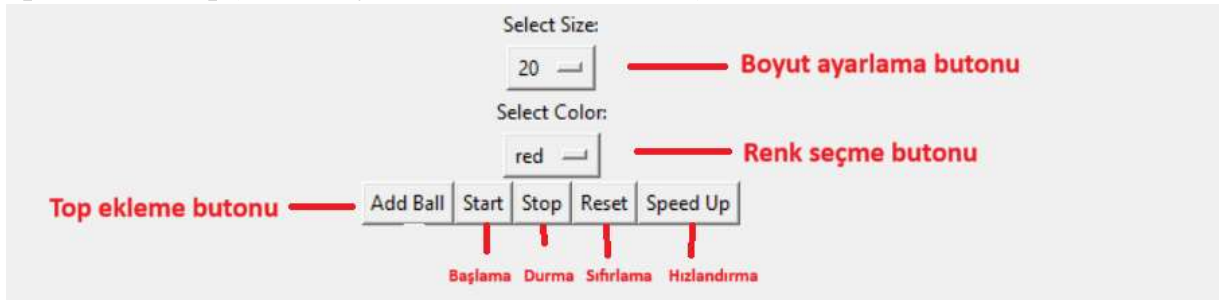
1. ÖZET

Bu projede, çizim ekranında topların boyut, renk ve hareket özelliklerini kontrol eden bir animasyon uygulaması geliştirdik. Kullanıcı Start, Stop, Reset ve Hızlandırma butonları ile topları yönetebilir.

2. GİRİŞ

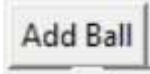
2.1 Çizim Ekranının Özellikleri

Çizim ekranına istenilen sayıda top eklenebilir. 20,30 ve 50 olmak üzere 3 farklı boyut seçeneği; kırmızı, mavi, sarı, yeşil ve gri olmak üzere 5 farklı renk seçeneği ile toplar istenilen şekilde ekrana eklenebilir. Eklenen toplar çizim ekranında random konumlara yerleşirler. Ekranda Add Ball(Top ekleme), Start(Animasyonu başlatma), Stop(Animasyonu durdurma), Reset(Sıfırlama) ve Speed Up(Animasyonu hızlandırma) butonu bulunmaktadır.



2.2 Top Ekleme Menüsü

İstenilen özelliklerde n tane sayıda top animasyona eklenebilir

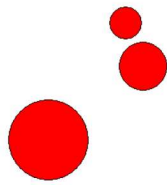
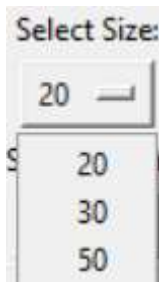


```
def add_ball():
    size = size_var.get()
    color = color_var.get()
    x = random.randint(size, 800 - size)
    y = random.randint(size, 500 - size)
    ball = canvas.create_oval(x - size, y - size, x + size, y + size, fill=color)
    balls.append({"id": ball, "dx": random.choice([-3, 3]), "dy": random.choice([-3, 3])})
```

```
add_button = tk.Button(frame, text="Add Ball", command=add_ball)
add_button.pack(side=tk.LEFT)
```

2.3 Boyut Seçme Menüsü

20,30 ve 50 olmak üzere 3 farklı boyut seçimi yapılabilir.



```
size_label = tk.Label(root, text="Select Size:")
size_label.pack()
size_var = tk.IntVar(value=20)
size_menu = tk.OptionMenu(root, size_var, value=20, *values: 30, 50)
size_menu.pack()
```

2.4 Renk Seçme Menüsü

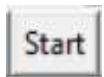
Kırmızı, mavi, sarı, yeşil ve gri olmak üzere 5 farklı renk seçeneği ile toplar istenilen şekilde ekrana eklenebilir.



```
color_label = tk.Label(root, text="Select Color:")
color_label.pack()
color_var = tk.StringVar(value=ball_colors[0])
color_menu = tk.OptionMenu(root, color_var, *values: *ball_colors)
color_menu.pack()
```

2.5 Start Butonu

Start butonuna basıldığında eklenen toplar hareket etmeye başlar.



```
def start():
    global continue_animation
    continue_animation = True
    move_balls()
```

```
start_button = tk.Button(frame, text="Start", command=start)
start_button.pack(side=tk.LEFT)
```

2.6 Stop Butonu

Stop butonuna basıldığında animasyon durur.



```
def stop():  
    global continue_animation  
    continue_animation = False
```

```
stop_button = tk.Button(frame, text="Stop", command=stop)  
stop_button.pack(side=tk.LEFT)
```

2.7 Reset Butonu

Reset butonuna basıldığında animasyon sıfırlanır, program başlangıç durumuna geri döner.

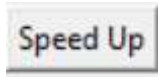


```
def reset():  
    global balls  
    for ball in balls:  
        canvas.delete(ball["id"])  
    balls = []
```

```
reset_button = tk.Button(frame, text="Reset", command=reset)  
reset_button.pack(side=tk.LEFT)
```

2.8 Hızlandırma(Speed up) Butonu

Speed up butonuna basıldığında animasyon hızlanır. Her basıldığında animasyonun hızı artar.



```
def speed_up():  
    for ball in balls:  
        ball["dx"] *= 1.5  
        ball["dy"] *= 1.5
```

```
speed_button = tk.Button(frame, text="Speed Up", command=speed_up)  
speed_button.pack(side=tk.LEFT)
```

2.9 Hareket Animasyonları

Çizim ekranına eklenen n adet top ekranda random konum alır ve start tuşuna basıldıktan sonra animasyon başlar.

```
def move_balls():  
    global continue_animation  
    if not continue_animation:  
        return  
    for ball in balls:  
        x1, y1, x2, y2 = canvas.coords(ball["id"])  
        if x1 <= 0 or x2 >= 800:  
            ball["dx"] = -ball["dx"]  
        if y1 <= 0 or y2 >= 500:  
            ball["dy"] = -ball["dy"]  
        canvas.move(*args: ball["id"], ball["dx"], ball["dy"])  
    root.after(ms: 30, move_balls)
```

Ödev No: 1	Tarih 24.11.2024	7/9
------------	------------------	-----

3. YÖNTEM

Bizden istenilen kriterler doğrultusunda projemizi hazırladık. İstenilen özellikleri ve animasyonları gerçekleştirdik. Kodumuza açıklama satırları ekleyerek daha anlaşılır olmasını sağladık.

4. SONUÇ VE ÖĞRENİLEN DERSLER

Umutcan Oğurlu

Kod yazma yeteneğimi geliştiren bir ödev oldu. Grafik arayüz tasarımı, animasyonlu nesne hareketleri ve buton işlevselliği gibi konularda kendimi geliştirdim.

Berkay Ömer Aydemir

Python hakkında yeni kütüphaneler öğrendim, hem oyun hem de görsel anlamda kendimi geliştirdim.

5. KAYNAKÇA

<https://learn.microsoft.com/tr-tr/>

<https://github.com/>

6. GİTHUB LİNKLERİ

Umutcan Oğurlu

<https://github.com/umutcanogurlu> (profil linki)

Berkay Ömer Aydemir

<https://github.com/Lorenn01> (profil linki)

Ödev No: 1	Tarih 24.11.2024	8/9
------------	------------------	-----