



T.C

**KOCAELİ SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ
MÜHENDİSLİK VE DOęA BİLİMLERİ FAKÜLTESİ
YAZILIM MÜHENDİSLİęİ PROGRAMI**

ANİMASYONLU ÇİZİM EKRANI

**Hazırlayan
İBRAHİM TEKİN
220502037**

<https://github.com/IbrahimTekin03>

**DERS SORUMLUSU
Dr. Öğr. Üyesi
Elif Pınar HACİBEYOęLU**

24.11.2024

TARİH (12 Punto)

İÇİNDEKİLER

1. ÖZET (ABSTRACT)	3
2. GİRİŞ (INTRODUCTION)	3
3. YÖNTEM (METHOD)	3
3.1 Örnek Alt Başlık	3
3.2 Örnek Alt Başlık	5
4. SONUÇ VE ÖĞRENİLEN DERSLER	7
5. KAYNAKÇA	7

1. ÖZET

Bu projede, Python ve Tkinter kullanılarak bir top animasyonu uygulaması geliştirilmiştir. Kullanıcı, farklı boyut ve renk seçenekleri sunularak, ekranda rastgele konumlandırılan toplar oluşturabilmektedir. Uygulama, hareketli nesnelerin animasyonunu ve kullanıcı etkileşimini içermektedir. Projenin amacı, temel grafik ve animasyon işlemlerini anlamak ve Tkinter kütüphanesiyle çalışmayı pekiştirmektir.

2. GİRİŞ

Bu proje, Python'un Tkinter kütüphanesi kullanılarak temel bir animasyon uygulaması geliştirmeyi amaçlamaktadır. Kullanıcıların boyut ve renk seçimi yaparak rastgele hareket eden toplar ekleyebildiği interaktif bir sistem tasarlanmıştır. Proje sayesinde, grafik arayüz tasarımı ve animasyon mantığı üzerine temel bir deneyim kazanılması hedeflenmiştir.

3. YÖNTEM

3.1 Top sınıfı ve hareket fonksiyonları

```
import tkinter as tk
import random

class Ball:
    def __init__(self, canvas, x, y, size, color, dx, dy):
        self.canvas = canvas
        self.size = size
        self.color = color
        self.dx = dx
        self.dy = dy
        self.ball = canvas.create_oval(
            x, y, x + size, y + size, fill=color, outline=color
        )

    def move(self):
        coords = self.canvas.coords(self.ball)
        x1, y1, x2, y2 = coords
        if x1 <= 0 or x2 >= self.canvas.winfo_width():
            self.dx = -self.dx
        if y1 <= 0 or y2 >= self.canvas.winfo_height():
            self.dy = -self.dy
        self.canvas.move(self.ball, self.dx, self.dy)
```

Bu bölümde, Ball sınıfı kullanılarak topun hareketi ve davranışları açıklanmıştır. Sınıf, topların başlangıç özelliklerini (renk, boyut, pozisyon, hız) belirler ve hareket sırasında kenarlara çarpma gibi durumlarda doğru şekilde yön değiştirmelerini sağlar.

Ball sınıfının temel işlevleri şunlardır:

Başlangıç Özellikleri: Her bir topun boyutu, rengi, başlangıç noktası ve hareket yönleri belirlenir.

Hareket Mekaniği: Top, belirtilen hızlarla her bir animasyon karesinde hareket eder.

Çarpma ve Sekme: Tuvalin kenarlarına çarptığında, topun hareket yönü otomatik olarak tersine çevrilir, böylece hareket devam eder.

Bu yaklaşım, topların dinamik ve sürekli bir şekilde tuval üzerinde hareket etmesini sağlar. Ayrıca farklı boyutlarda ve renklerde toplar bağımsız olarak davranış gösterebilir.

3.2 BallSimulator Sınıfı

```
class BallSimulator:
    def __init__(self, root):
        self.root = root
        self.canvas = tk.Canvas(root, bg="lightgray", width=800, height=500)
        self.canvas.pack(fill=tk.BOTH, expand=True)

        self.balls = []
        self.running = False
        self.speed = 1
        self.selected_color = "red"

        self.create_controls()

    def create_controls(self):
        control_frame = tk.Frame(self.root, bg="white")
        control_frame.pack(fill=tk.X, side=tk.BOTTOM, anchor="w")

        size_frame = tk.Frame(control_frame, bg="white")
        size_frame.pack(side=tk.TOP, pady=5, anchor="w")
        sizes = [(20, 2), (40, 4), (60, 6)]
        for size, pad in sizes:
            size_button = tk.Canvas(size_frame, width=60, height=60, bg="white", highlightthickness=0)
            size_button.create_oval(
                30 - size / 2, 30 - size / 2, 30 + size / 2, 30 + size / 2, fill="gray"
            )
            size_button.bind("<Button-1>", lambda event, s=size: self.add_ball(s))
            size_button.pack(side=tk.LEFT, padx=10)
```

Bu bölümde, BallSimulator sınıfının bir parçası olan kontrol oluşturma işlevi açıklanmaktadır. Bu işlev, kullanıcının topların boyutunu seçmesine ve bu seçimle topları tuval üzerinde eklemesine olanak tanır.

Boyut Seçim Kontrolleri (size_frame)

- Amaç: Kullanıcının 3 farklı boyut seçeneğinden birini seçerek topları bu boyutlarda oluşturmalarını sağlamak.
- Yerleşim: Bu kontroller bir çerçeve (size_frame) içinde düzenlenmiştir ve ana kontrol çerçevesine yerleştirilmiştir.

Boyut Çerçevesi ve Boyut Butonları

- Çerçeve Oluşturma: Boyut butonları için bir çerçeve oluşturulur ve ekranın alt kısmında yatay olarak yerleştirilir.
- Boyutlar: Top boyutları ve her butonun görünümünde kullanılan dolgu/padding değerleri bir liste olarak tanımlanmıştır.

Oval (Yuvarlak) Görünüm

- Her boyut için bir Canvas oluşturularak, kullanıcıya o boyutu temsil eden bir yuvarlak çizilir.
- Çizim: Her bir Canvas üzerine bir oval (yuvarlak) eklenir ve size değişkenine göre boyutu dinamik olarak ayarlanır.

Butonlara Tıklama İşlevi

- Kullanıcı herhangi bir boyut butonuna tıkladığında, bir olay tetiklenir ve bu boyutta bir top tuval üzerine eklenir.
- Bu işlem, self.add_ball(size) fonksiyonu aracılığıyla gerçekleştirilir.

3.3 Arayüz Butonları

```
color_frame = tk.Frame(control_frame, bg="white")
color_frame.pack(side=tk.TOP, pady=5, anchor="w")
colors = ["red", "blue", "yellow"]
for color in colors:
    color_button = tk.Button(
        color_frame,
        bg=color,
        width=4,
        height=2,
        command=lambda c=color: self.select_color(c),
    )
    color_button.pack(side=tk.LEFT, padx=10)

action_frame = tk.Frame(control_frame, bg="white")
action_frame.pack(side=tk.TOP, pady=5, anchor="w") # Sola hizalama
tk.Button(action_frame, text="START", command=self.start).pack(side=tk.LEFT, padx=5)
tk.Button(action_frame, text="STOP", command=self.stop).pack(side=tk.LEFT, padx=5)
tk.Button(action_frame, text="RESET", command=self.reset).pack(side=tk.LEFT, padx=5)
tk.Button(action_frame, text="Speed Up", command=self.speed_up).pack(side=tk.LEFT, padx=5)

def select_color(self, color):
    self.selected_color = color
```

Bu kod parçası, kullanıcıya renk seçimi ve çeşitli aksiyonlar (başlatma, durdurma, sıfırlama, hızlandırma) için bir kontrol arayüzü sağlar.

Renk Seçim Kontrolleri (color_frame)

- Amaç: Kullanıcının topların rengini seçmesine olanak tanır.
- Yerleşim: Renk seçim butonları bir çerçevede (color_frame) düzenlenmiştir ve kontrol paneline eklenmiştir.
- Renkler: "Red", "Blue", "Yellow" olmak üzere 3 seçenek mevcuttur.

Butonların Oluşturulması

- Her renk için bir Button (buton) oluşturulur ve arka plan rengi (bg) seçilen renkle ayarlanır.
- Tıklama İşlevi: Bir butona tıklandığında, seçilen renk self.select_color(color) işlevi ile kaydedilir.

Aksiyon Butonları (action_frame)

- Amaç: Kullanıcıya topların hareketi ile ilgili işlemleri başlatma, durdurma, sıfırlama veya hızlandırma fırsatı verir.
- Yerleşim: Bu butonlar, action_frame adlı bir çerçeve içinde yatay olarak düzenlenmiştir.

START Butonu:

- Tüm topların hareket etmesini başlatır. self.start() fonksiyonu ile kontrol edilir.

STOP Butonu:

- Topların hareketini durdurur. self.stop() fonksiyonu çağrılır.

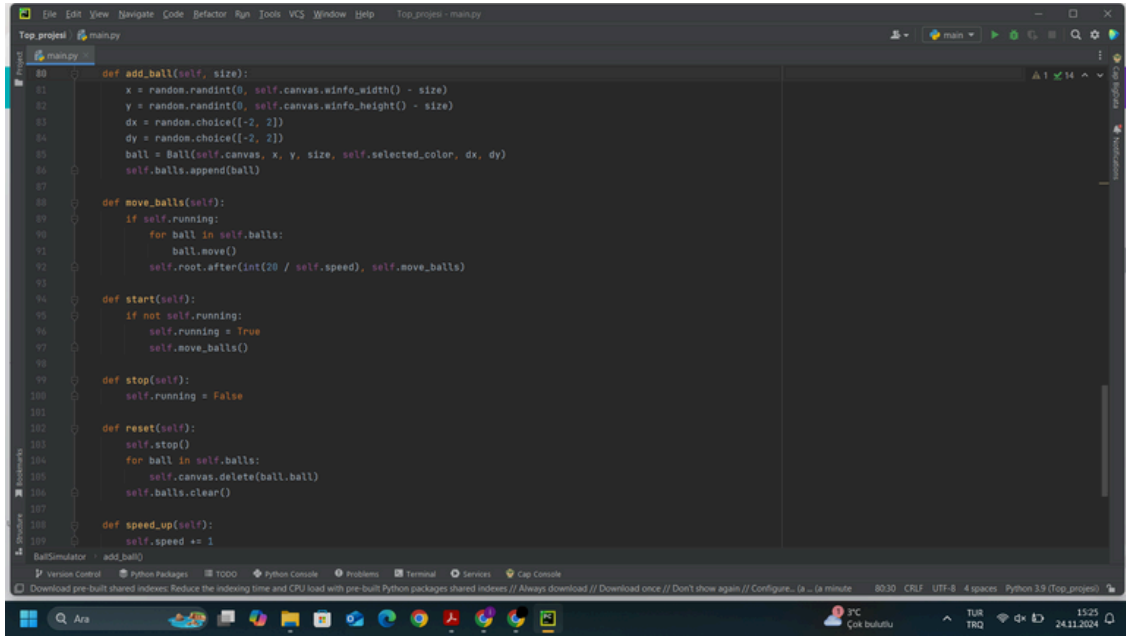
RESET Butonu:

- Tüm topları ve mevcut durumu sıfırlayarak başlangıç konumuna döndürür. self.reset() fonksiyonu kullanılır.

Speed Up Butonu:

- Topların hareket hızını artırır. self.speed_up() fonksiyonuna bağlanmıştır.

3.4 BallSimulator Sınıfı



```
20 def add_ball(self, size):
21     x = random.randint(0, self.canvas.winfo_width() - size)
22     y = random.randint(0, self.canvas.winfo_height() - size)
23     dx = random.choice([-2, 2])
24     dy = random.choice([-2, 2])
25     ball = Ball(self.canvas, x, y, size, self.selected_color, dx, dy)
26     self.balls.append(ball)
27
28 def move_balls(self):
29     if self.running:
30         for ball in self.balls:
31             ball.move()
32         self.root.after(int(20 / self.speed), self.move_balls)
33
34 def start(self):
35     if not self.running:
36         self.running = True
37         self.move_balls()
38
39 def stop(self):
40     self.running = False
41
42 def reset(self):
43     self.stop()
44     for ball in self.balls:
45         self.canvas.delete(ball)
46     self.balls.clear()
47
48 def speed_up(self):
49     self.speed += 1
50
51 BallSimulator.add_ball()
```

Bu kod, kullanıcı etkileşimlerine bağlı olarak topların eklenmesini, hareket ettirilmesini, durdurulmasını, sıfırlanmasını ve hızlandırılmasını sağlar.

`add_ball(size)` fonksiyonu, rastgele bir konum, boyut ve başlangıç hareket yönü (`dx`, `dy`) belirleyerek kullanıcı tarafından seçilen renkte bir top oluşturur ve bu topu tuvale ekler. Eklenen tüm toplar, bir listeye kaydedilir.

`move_balls()`, topların hareketinden sorumludur. Bu işlev, topların her birinin hareket yönünü kontrol eder ve hareketlerini düzenli olarak günceller. Topların çarpma veya sekme davranışları, `Ball` sınıfında tanımlıdır. Eğer `self.running` `True` ise, animasyon devam eder ve hız, `self.speed` değişkenine göre ayarlanır.

`start()` fonksiyonu, animasyonu başlatır ve topların hareket etmeye başlamasını sağlar. Eğer animasyon zaten çalışıyorsa, bu işlev tekrar tetiklenmez.

`stop()`, animasyonu durdurur ve topların hareketini geçici olarak askıya alır.

`reset()`, mevcut topları tuvalden kaldırır, listeyi temizler ve animasyonu sıfırlayarak uygulamayı başlangıç durumuna getirir.

`speed_up()`, animasyon hızını artırır. Bu işlev her çağrıldığında topların hareket hızını bir önceki hızın üzerine çıkarır.

Bu sınıf, tkinter kütüphanesi kullanılarak bir grafik arayüz ile entegre edilmiştir ve uygulama, kullanıcı etkileşimlerine göre dinamik bir şekilde çalışır.

4. SONUÇ VE ÖĞRENİLEN DERSLER

Bu proje, Python ile tkinter kullanılarak grafiksel bir kullanıcı arayüzü oluşturma ve animasyon yönetimi üzerine bir çalışma olmuştur. Proje kapsamında toparların rastgele konumlandırılması, hareket ettirilmesi ve kullanıcı etkileşimlerine göre kontrol edilmesi sağlanmıştır.

Bu süreçte, tkinter'ın farklı bileşenleriyle çalışma, animasyon yönetimi, kullanıcı etkileşimleriyle olay tetikleme ve nesne tabanlı programlama teknikleri üzerinde önemli deneyimler kazanılmıştır. Proje, temel grafik uygulamaları ve olay tabanlı programlama konularında sağlam bir anlayış kazandırmıştır.

5. KAYNAKÇA

<https://chat.openai.com>

<https://stackoverflow.com>

Ödev No: 1	Tarih 24.11.2024	7/8
------------	------------------	-----