



**T.C**  
**KOCAELİ SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ**  
**MÜHENDİSLİK VE DOęA BİLİMLERİ FAKÜLTESİ**  
**BİLGİSAYAR/YAZILIM MÜHENDİSLİęİ**

**PROJE KONUSU:**  
**ANİMASYONLU ÇİZİM EKRANI**

**ÖęRENCİ ADI:**  
**ÖęRENCİ NUMARASI:**

Alperen Akın Işgın  
220502034  
Kerem Savaş  
220502034

**DERS SORUMLUSU:**  
**PROF. DR./DR. ÖęR. ÜYESİ**  
**ELİF PINAR HACİBEYOęLU**

**TARİH:23..11.2024**

# **1 GİRİŞ**

## **1.1 Projenin amacı**

- Bu projenin amacı, kullanıcıların belirli bir boyut ve renk seçerek bir animasyon ekranında hareket eden toplar oluşturmalarını sağlamaktır. Kullanıcılar topların hızını artırabilir, animasyonu başlatıp durdurabilir ve tüm topları sıfırlayabilir. Proje, hem kullanıcı dostu bir arayüz hem de temel animasyon prensiplerini gösteren bir Python uygulaması geliştirmeyi hedefler.
- Kullanıcıların farklı boyut ve renklerde toplar oluşturabilmesi.
- Topların ekranda rastgele yönlerde hareket etmesi.
- Kullanıcıların animasyonu başlatma, durdurma ve hızını artırma kontrolüne sahip olması.
- Topların ekrandaki sınırlarla çarpışarak yön değiştirebilmesi.
- Kullanıcı dostu bir arayüz ile kolay kullanım sunulması.

# **2 GEREKSİNİM ANALİZİ**

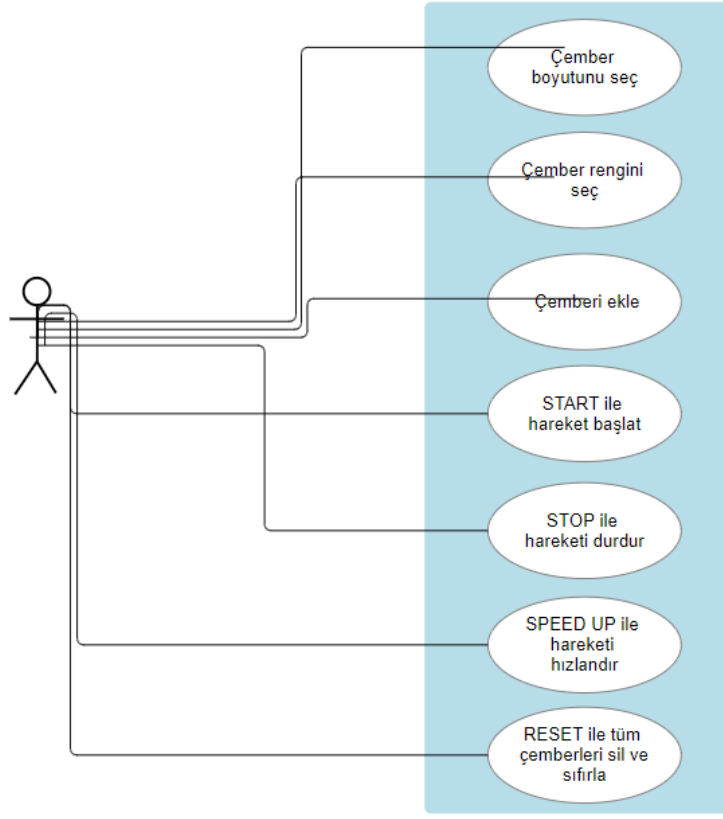
## **2.1 Arayüz gereksinimleri**

- Kullanıcı arayüzü:
  - Bir animasyon alanı (Canvas).
  - Top boyutlarının seçimi için grafiksel arayüz (dairesel seçenekler).
  - Top renklerinin seçimi için grafiksel arayüz (kare şeklinde seçenekler).
  - Başlat, durdur, sıfırla ve hız artırma işlevlerini sağlayan butonlar.
  - Donanım arayüzü:
    - Standart bir fare veya dokunmatik ekran kullanılabilir olmalı.
    - Klavye gereksinimi bulunmamaktadır.

## **2.2 Fonksiyonel gereksinimler**

- Kullanıcının bir boyut ve renk seçimi yaparak animasyona yeni bir top ekleyebilmesi.
- Topların rastgele bir başlangıç konumunda eklenmesi ve belirli bir hızda hareket etmesi.
- Topların ekran sınırlarıyla çarpışma durumunda yön değiştirebilmesi.
- Animasyonu başlatma, durdurma ve hız artırma işlemlerinin yapılabilmesi.
- "Sıfırla" butonuyla tüm topların ekrandan kaldırılması.

## **2.3 Use-Case diyagramı**



## 3 TASARIM

### 3.1 Mimari tasarım

- Ana sınıf: BallAnimationApp sınıfı, uygulamanın temel işlevlerini kontrol eder.
- Canvas: Topların çizildiği ve hareket ettiği ana alan.
- Kontrol Paneli: Kullanıcı seçimleri için boyut ve renk seçeneklerini içeren alan.
- Buton Çerçevesi: Animasyonu kontrol eden butonların bulunduğu çerçeve.
- Animasyon Mekanizması: move\_balls fonksiyonu, topların sürekli hareketini ve çarpışma kontrolünü sağlar.

### 3.2 Kullanılacak teknolojiler

- Yazılım dili: Python 3.
- Harici kütüphaneler: Harici bir kütüphane kullanılmamaktadır, yalnızca Python'un standart tkinter modülü ile yapılmıştır.
- Diğer teknolojiler: Herhangi bir ek teknoloji gereksinimi yoktur.

### 3.3 Veri tabanı tasarımı

-

### 3.4 Kullanıcı arayüzü tasarımı

- Kullanıcı Arayüzü:
  - Canvas üzerinde topların hareketini gösteren bir animasyon alanı.
  - Seçim alanı:
  - Boyutlar için ölçeklendirilmiş daireler.
  - Renkler için kareler.
  - Kontrol butonları: Animasyonu başlatma, durdurma, sıfırlama ve hız artırma işlevleri için.
  - Uygulamanın nasıl çalıştırılacağı:
    - 1. Uygulama, Python'un standart tkinter modülü ile çalıştırılabilir.
    - 2. python <dosya\_adi>.py komutu ile başlatılır.
    - 3. Kullanıcı, boyut ve renk seçerek toplar ekler, ardından "Başlat" butonu ile animasyonu izleyebilir.

## 4 UYGULAMA

### 4.1 Kodlanan bileşenlerin açıklamaları

- - BallAnimationApp sınıfı:
  - Uygulamanın temel sınıfıdır ve tüm ana işlevleri kontrol eder. Kullanıcı arayüzü bileşenlerini oluşturur ve olay yönetimini sağlar.
  - Fonksiyonlar:
  - `_init_` : Sınıfın başlatıcısıdır. Kullanıcı arayüzünü (Canvas, kontrol paneli) oluşturur ve başlangıç ayarlarını yapar.
  - `select_size(size)`: Kullanıcının seçtiği boyutu belirler ve gerektiğinde bir top oluşturur.
  - `select_color(color, rectangle)`: Kullanıcının seçtiği rengi belirler. Seçilen renk karesini animasyonlu olarak vurgular.
  - `animate_rectangle(rectangle)`: Seçilen renk karesi üzerinde yanıp sönen bir animasyon oluşturur.
  - `add_ball()`: Belirlenen boyut ve renge sahip bir topu rastgele bir konuma ekler.
  - `reset_selection()`: Seçim ekranındaki boyut ve renk seçimlerini sıfırlar.
  - `move_balls()`: Topların hareket etmesini ve ekran sınırlarına çarptıklarında yön değiştirmesini sağlar.
  - `start_animation()`: Animasyonu başlatır.
  - `stop_animation()`: Animasyonu durdurur.
  - `reset_canvas()`: Tüm topları ve animasyonu sıfırlar.
  - `speed_up()`: Topların hareket hızını artırır.

## 4.2 Görev dağılımı

- Tasarım ve Geliştirme:
  - 
  - Kullanıcı arayüzü tasarımı: Seçim panelleri ve kontrol butonları.(Alperen A.I.)
  - 
  - Hareket algoritması: move\_balls fonksiyonu ile top hareketlerinin ve çarpışma kontrollerinin geliştirilmesi.(Kerem S. - Alperen A.I.)
  - 
  - Kullanıcı etkileşimleri: Boyut ve renk seçimi mekanizmalarının geliştirilmesi.(Alperen A.I.)
  - 
  - 
  - Rapor Hazırlanması:
  - 
  - Kodun analiz edilmesi ve bileşenlerin açıklanması.
  - 
  - Test sürecinin hazırlanması ve sonuçlarının raporlanması.

## 4.3 Karşılaşılan zorluklar ve çözüm yöntemleri

- Zorluk:
  - Seçilen renk karesinin animasyonu sırasında önceki animasyonun düzgün bir şekilde durdurulması.
  - 
  - Çözüm:
  - self.root.after\_cancel fonksiyonu kullanılarak önceki animasyon döngüsünün iptal edilmesi sağlandı.
  - 
  - 
  - Zorluk:
  - Hareket eden topların ekran sınırlarını doğru algılayarak yön değiştirmesi.
  - 
  - Çözüm:
  - Her topun koordinatları self.canvas.coords ile kontrol edilerek sınır ihlali durumunda hız vektörünün ters çevrilmesi sağlandı.

## 4.4 Proje isterlerine göre eksik yönler

-

## 5 TEST VE DOĞRULAMA

-

