



T.C

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

**YAZILIM LAB1 ÖDEV
Animasyonlu Çizim Ekranı**

**Mert Bülbül
220502006**

<https://github.com/Mertb2627>

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

24/11/2024

1. GİRİŞ

1.1 Projenin Amacı

Bu projenin amacı, Python programlama dili ve Tkinter kütüphanesi kullanılarak bir top animasyonu simülasyonu geliştirmektir.

Kullanıcılar, animasyon alanında topların hareketlerini izleyebilir, renklerini değiştirebilir, hızlarını değiştirebilir, yeni toplar ekleyebilir veya animasyonu sıfırlayabilir. Proje, temel fizik kurallarını simüle ederek çarpışma ve hareket dinamiklerini olabildiğince uygulamaktadır.

Projenin ana hedefleri:

- Basit bir animasyon uygulaması geliştirmek.
- Çarpışma algılama ve hareket simülasyonunu en iyi şekilde kullanma.
- Kullanıcı seçimleriyle (renk, boyut, hız) değişken bir deneyim sunmak.
- Mini bir oyun denilebilir

2. GEREKSİNİM ANALİZİ

2.1 Arayüz Gereksinimleri

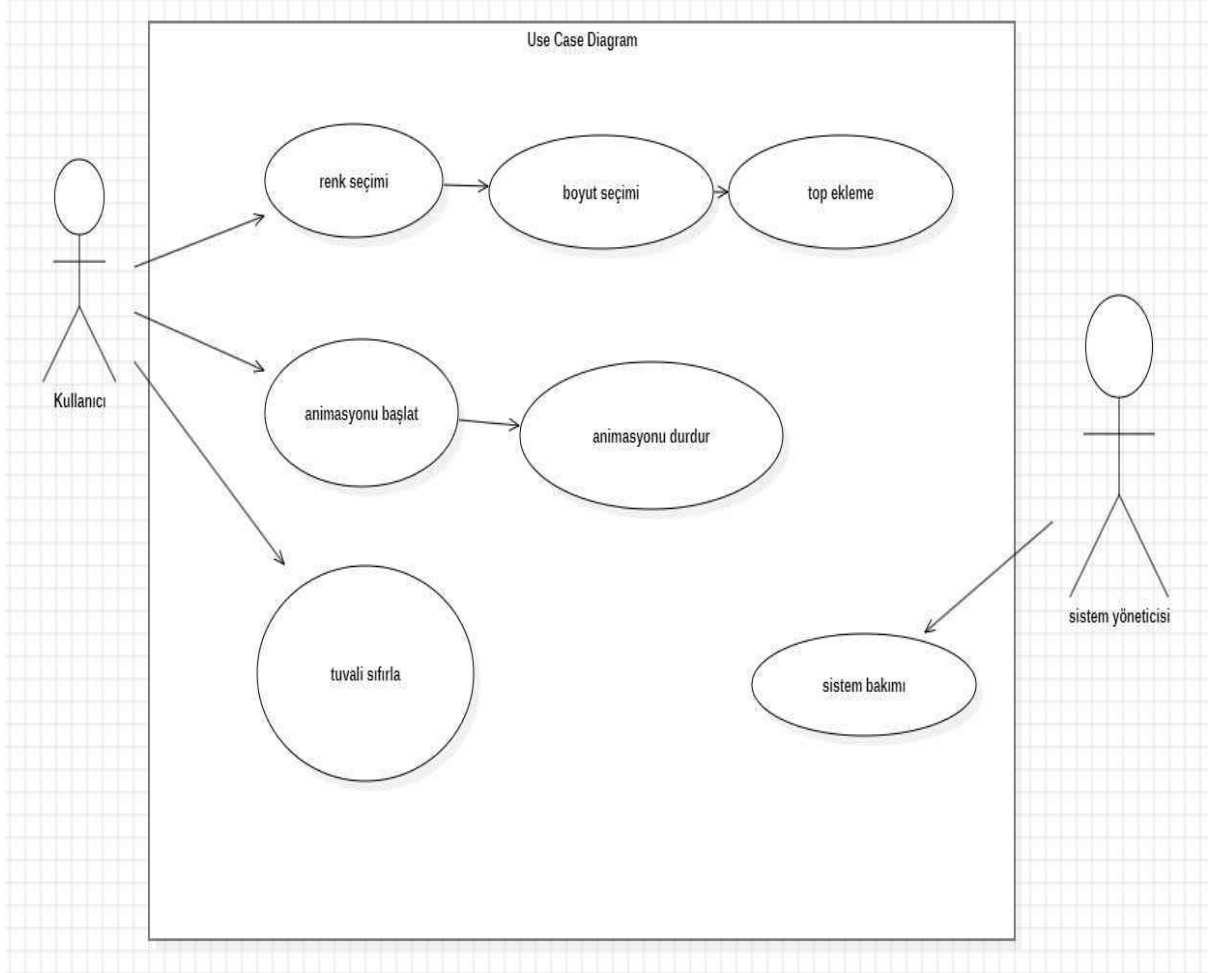
- Kullanıcı arayüzü Tkinter kütüphanesi ile tasarlanmıştır.
- Renk ve boyut seçimleri için butonlar, hız ve diğer işlem kontrolü için ayrı butonlar bulunmaktadır.
- Çarpışmaların ve hareketlerin vb görselleştirilmesi Canvas bileşeni üzerinden yapılmakta.

2.2 Fonksiyonel Gereksinimler

- Topların rastgele hareket ettirilmesi.
- Toplar arasında çarpışma tespiti.
- Kullanıcıların topların renk ve boyutlarını seçebilmesi.

- Kullanıcıların animasyonu başlatma, durdurma ve sıfırlama işlemlerini gerçekleştirmesi.
- Animasyon hızının kullanıcı tarafından ayarlanabilmesi.

2.3 Use-Case Diyagramı



3. TASARIM

3.1 Mimari Tasarım

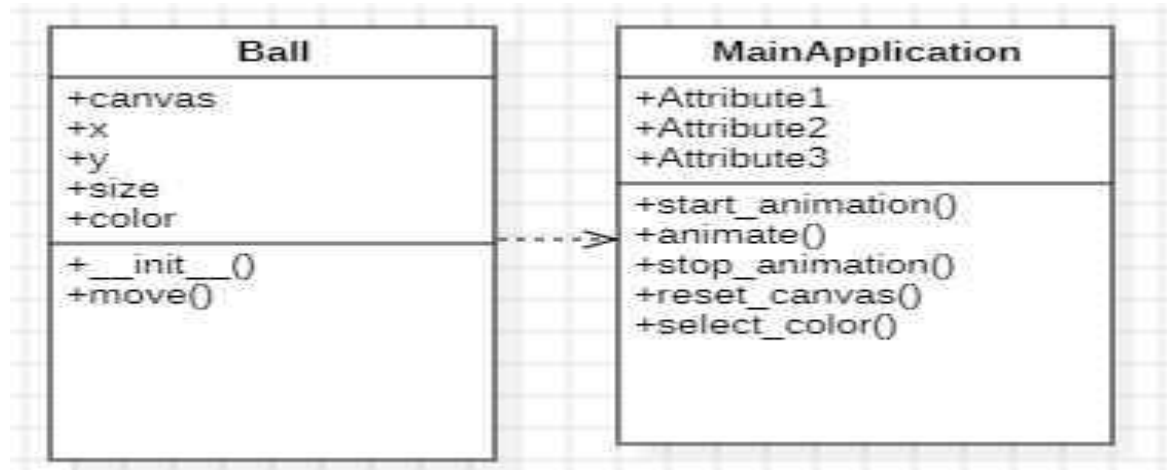
Proje, modüler bir yapıda tasarlanmıştır. Başlıca bileşenler:

1. **Ball Sınıfı:** Her top bir Ball sınıfı nesnesi olarak oluşturulur. Hareket ve çarpışma davranışlarını içerir.
2. **Canvas (Tual):** Topların çizildiği ve animasyonun gerçekleştiği alan.
3. **Arayüz:** Renk, boyut ve hız seçimi gibi kullanıcı etkileşimlerini sağlayan butonlardan oluşur.

3.2 Kullanılacak Teknolojiler

- **Programlama Dili:** Python
- **Kütüphaneler:**
 - **Tkinter:** Kullanıcı arayüzü tasarımı ve animasyon için.
 - **random:** Topların başlangıç koordinatlarını ve hızlarını rastgele belirlemek için.

3.3 Sınıf Diyagramı



3.4 Kullanıcı Arayüzü Tasarımı

- Kullanıcı arayüzü basit bir şekilde tasarlanmıştır.
 - Renk seçimi için farklı renk butonları.
 - Boyut seçimi için belirli boyutlara atanmış butonlar.
 - Animasyonu kontrol etmek için "START", "STOP", "RESET" ve hız ayarı butonları.

4. UYGULAMA

4.1 Kodlanan Bileşenlerin Açıklamaları

- 1. Ball Sınıfı:**
 - a. Topun hareketini (move) ve sınır çarpışmalarını kontrol eder.
 - b. Topun Canvas üzerinde çizilmesini sağlar.
- 2. Çarpışma Algılama:**
 - a. İki top arasındaki mesafeyi hesaplayarak çarpışmayı algılar.
 - b. Çarpışma durumunda topoların hareket yönlerini tersine çevirir.
- 3. Animasyon Kontrolleri:**
 - a. Kullanıcıların animasyonu başlatması, durdurması ve sıfırlaması sağlanmıştır.
 - b. Animasyon hızının artırılıp azaltılmasına olanak tanınmıştır.
- 4. Kullanıcı Etkileşimleri:**
 - a. Renk ve boyut seçimleri için butonlarla kullanıcı girişi sağlanmıştır.
 - b. Topoların özellikleri kullanıcı tarafından belirlenebilir.
 - c. Kullanıcı hız sınırını kendi belirleyebilir

4.2 Karşılaşılan Zorluklar ve Çözüm Yöntemleri

- 1. Çarpışma Algoritması:**
 - a. Topoların merkezleri arasındaki mesafenin, yarıçapların toplamından küçük olup olmadığı kontrol edilmiştir.
 - b. Çarpışma sonrası hareket yönlerini dinamik olarak değiştirmeye göre geliştirilmiştir.
- 2. Hız Kontrolü:**
 - a. Kullanıcı tarafından hız ayarı yapılırken animasyonun akıcılığını korumak için hız katsayısı (speed_multiplier) kullanılmıştır.
- 3. Topoların Çizim Alanını Aşması:**
 - a. Topoların Canvas sınırlarından dışarı çıkmasını engellemek için hareket koordinatları sınır değerlerle kontrol edilmiştir.

5. TEST VE DOĞRULAMA

5.1 Yazılımın Test Süreci

- **Unit Testler:**
 - `Ball.move()` fonksiyonu: Topların hareketinin doğru şekilde işlendiği doğrulanmıştır.
 - Çarpışma algılama fonksiyonu: Çarpışma tespiti ve hareket değişiklikleri test edilmiştir.
- **Kullanıcı Testleri:**
 - Renk ve boyut seçimi işlevleri manuel olarak test edilmiştir.
 - Animasyon kontrol butonlarının doğru çalıştığı gözlemlenmiştir.

5.2 Yazılımın Doğrulanması

- Animasyon akıcı bir şekilde çalışmakta ve çarpışma algoritması doğru şekilde işlemektedir.
- Kullanıcı girişleri hatasız olarak işlenmekte, ancak renk ve boyut seçiminden önce kullanıcıya geri bildirim sağlanması iyileştirilebilir.

Kaynakça:

<https://chatgpt.com/>

<https://www.udemy.com/>

<https://trinket.io/python>

<https://www.w3schools.com/python/>