



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

PROJE KONUSU: ANİMASYONLU ÇİZİM EKRANI

ÖęRENCİ ADI: Anıl ERDOęAN-Akın TURAN
ÖęRENCİ NUMARASI: 220501006-220501013

DERS SORUMLUSU: DR. ÖęR. ÜYESİ Elif Pınar
HACİBEYOęLU

TARİH:

1 GİRİŞ

1.1 Projenin amacı

Projenin temelinde daire şeklindeki cisimlerin ekrana eklenmesi ardından animasyonlu bir ekranda açılan pencerenin kenarlarına çarparak hareket etmesi bekleniyor. Proje için python dili kullanıldı ve tkinter kütüphanesi ile tasarlandı. Projede olması beklenen özellikler ise şöyle:

- Projede istenen ana menünün tasarlanması.
- Menünün cisim hareketlerini göstermesi.
- Cisimlerin özelleştirilebilmesi.

2 GEREKSİNİM ANALİZİ

2.1 Arayüz gereksinimleri

Projede istenen arayüz gereksinimleri şöyle:

- Sade bir arayüz tasarımı.
- Fonksiyonlara tasarlanan arayüz üzerinden ulaşılabilmesi.
- Animasyon ekranının cisimlerin hareketini göstermesi.

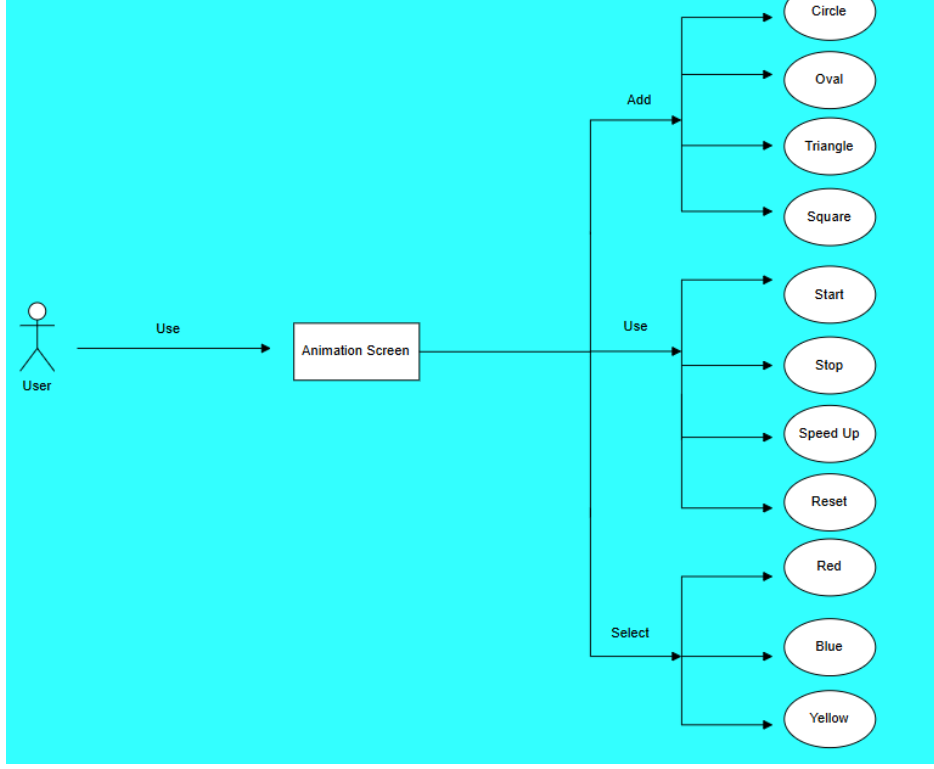
2.2 Fonksiyonel gereksinimler

Projede istenen fonksiyonel gereksinimler şöyle:

- Kullanıcının cisim ekleyebilmesi.
- Eklenen cisimlerin rengini seçebilmesi.
- Eklenen cisimlerin boyutunun belirlenebilmesi.
- Animasyon ekranının kontrol butonlarının olması.
- Cisimlerin istendiği zaman hızının arttırılması.

2.3 Use-Case diyagramı

- Bu projeye ait use- case diyagramı şu şekilde:

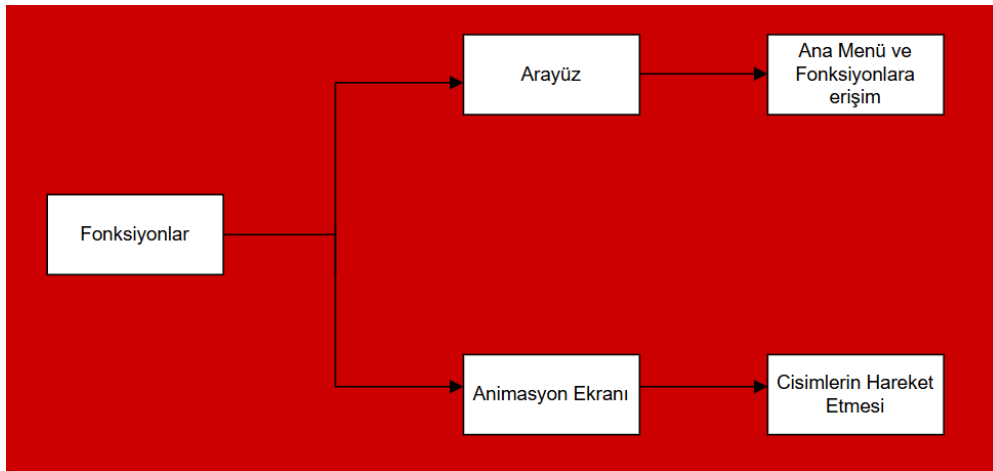


3 TASARIM

3.1 Mimari tasarım

- İlk olarak ana menü tasarlandı ve test edildi.
- İkinci olarak ana isterler çerçevesinde fonksiyon butonları eklendi.
- Son olarak animasyon ekranının görünüşü ve tasarımı istenen hale getirildi.

Modül Diyagramı:



3.2 Kullanılacak teknolojiler

- Yapılan simülasyon için Python dili kullanıldı. Tkinter kütüphanesi aracılığıyla uygulama tasarlandı ve random kütüphanesi ile desteklendi ve tasarlanan program oluşturuldu. Ekstra bir teknoloji kullanılmadı.

3.3 Veri tabanı tasarımı

- Uygulamada herhangi bir veri tabanı tasarımı yapılmadı veya hazır olarak kullanılmadı.

3.4 Kullanıcı arayüzü tasarımı



- Kullanıcı arayüzü isterler çerçevesinde sade bir biçimde tasarlandı oluşturulan menü üzerine gerekli fonksiyonlar eklendi.
- Oluşan küçük hatalar düzeltildi ve son hali test edildi.
- Uygulama kod çalıştırıldıktan sonra direkt olarak açılıyor ve istenilen fonksiyonlar kullanılabilir.

4 UYGULAMA

4.1 Kodlanan bileşenlerin açıklamaları

- **Class-Ball:** Bir topun özelliklerini (boyut, renk, hız) ve hareketlerini tanımlar.
- **__init__(Cisim):** Bir topun başlangıçta nerede, hangi boyutta, hangi renkte ve hızda olacağını ayarlar.
- **animation:** Topun hareket etmesini sağlar ve kenarlara çarpınca yönünü değiştirir.

- **Class-AnimationScreen:** Ekranda topların hareket edeceği alanı ve kullanıcı arayüzünü (düğmeler vb.) tanımlar.
- **__init__(Uygulama Ekranı):** Uygulama ekranını, tuval alanını (canvas), menü düğmelerini ve başlangıç ayarlarını oluşturur.
- **color_button:** Belirli bir renk seçmek için düğme oluşturur.
- **size_button:** Belirli bir boyut seçmek için bir düğme oluşturur.
- **select_size:** Topun boyutunu seçmek için kullanılır.
- **select_color:** Topun rengini seçmek için kullanılır.
- **start(self):** Topların hareketini başlatır.
- **stop(self):** Topların hareketini durdurur.
- **reset(self):** Tüm topları ve hız ayarlarını sıfırlar.
- **speed_up(self):** Tüm topların hızını artırır.
- **add_ball(self):** Tuvale yeni bir top ekler.
- **exit(self):** Programı kapatır.
- **animate(self):** Ekrandaki topların sürekli hareket etmesini sağlar.

4.2 Görev dağılımı

- Kodun ve tasarlanan projenin tamamı birlikte sesli iletişim uygulaması kullanılarak aynı anda yazıldı.
- Rapor da aynı şekilde sesli iletişim uygulaması kullanılarak birlikte yazıldı.

4.3 Karşılaşılan zorluklar ve çözüm yöntemleri

- Projede beklenmeyen bir durum veya hesaplanılamayan herhangi bir zorlukla karşılaşılmadı.

4.4 Proje isterlerine göre eksik yönler

- Projede eksik bir ister bulunmamaktadır.

5 TEST VE DOĞRULAMA

5.1 Yazılımın test süreci

- Yazılımın test sürecinde temel fonksiyonlar yazıldıktan sonra çalışıp çalışmadığı test edildi düzgün çalışmayanlar yeniden yazıldı.
- Her fonksiyonun çalışıp çalışmadığı kontrol edildi.
- Son versiyon için uygun hız ayarlandı.

```
def speed_up(self):  
  
    self.speed += 5  
  
    for ball in self.balls:  
  
        ball.speed_x = ball.speed_x / abs(ball.speed_x) * self.speed  
        ball.speed_y = ball.speed_y / abs(ball.speed_y) * self.speed
```

5.2 Yazılımın doğrulanması

- Yazılımın son hali kontrol edildi ve isterlere uygun olup olmadığına bakıldı. Teslime uygun olduğu karar verildikten sonra ufak bir iki düzeltmeden sonra proje tamamlandı.

