

24 KASIM 2022



NECMETTİN ERBAKAN
ÜNİVERSİTESİ
HAVACILIK VE UZAY
BİLİMLERİ FAKÜLTESİ

UZAY VE UYDU MÜHENDİSLİĞİ BÖLÜMÜ
PROGRAMLAMAYA GİRİŞ-III
YAZILIM PROJESİ ÖDEVİ

DERSİN SORUMLUSU: DR. ÖĞR. ÜYESİ TARIK ÜNLER

HAZIRLAYAN: İREM DUMAN 20030031003

Problemin Tanımı

Bu projede ebeveyn kontrollü basit kuralları olan bir oyun tasarlamak amaçlandı. Oyun başlatıldıktan sonra zaman akmaya başlayacak ve belli bir süre sonra oyun kendiliğinden kapanacak. Böylece ekran başında fazla zaman geçirme engellenmiş olacak. Oyunda amaç, dış yüzleri aynı renkte fakat iç yüzlerinin ikisi aynı biri farklı renkte olan üç karttan dış yüzleri çevriliyken farklı olanı bulabilmek. Farklı olan kart her bulunduğu anda skor artacak ve kullanıcıya gösterilecek.

Problemin Aşamaları

Süreyi tutan butona basıldığı anda oyun başlayacak ve süre akmaya başlayacak. Kullanıcının süre akmadan oyuna başlayamaması için buton oyun alanını kapatıyor olacak. Oyunun amacı ve süresi başlama butonunun yanında görünüyordur olacak. Başlangıçta kullanıcı farklı olan kartı görecektir ve skorla oyuna başlayacaktır. Ardından kartların dış yüzü çevrilecek ve kullanıcı farklı olduğunu düşündüğü karta tıklayacaktır. Kullanıcı her tıkladığında kartın iç yüzü açılacaktır. Eğer farklı olan kartı bulamadıysa kartlar tekrar karışacak ve tekrar tahmin etmesi beklenecektir. Eğer farklı olan kartı bulduysa skor sayacı bir artacak ve kartlar yeniden karışacak ve oyun bu şekilde süre sonuna kadar devam edecektir.

Problemin Çözümü

Burada oyun penceresini açabilmek, açıklamayı gösterebilmek, süreyle skoru gösteren ve kartları temsil eden butonları oluşturabilmek için tkinter modülü kullanılacaktır.

Zaman sayacının çalışması için time modülü kullanılacaktır.

Oyunun otomatik olarak kapanması ve skor butonunun gösterilmesi için donanım kontrolüne yarayan pyautogui modülü kullanılacaktır.

Kartların rastgele karışması için de random modülünden yararlanılacaktır.

Aynı renkteki butonlara tıklandığında iç yüzdeki renklerin gösterilmesi ve kartların karıştırılması için bir fonksiyon tanımlanacaktır. Bu fonksiyonda konumları barındıran bir liste oluşturulup elemanları .shuffle karıştırılacaktır ve butonların yeni yerleri .place ile belirtilecektir. Tıklanan butonun iç rengi ve dış rengi .config ile belirtilecektir.

Farklı renkteki butona tıklandığında ne olacağı da başka bir fonksiyonla tanımlanacaktır. Bu fonksiyon iç renklerin gösterilmesi yanında skor butonuna imleci götürecektir ve bu butona tıklandığında kartlar karışacaktır. Bu fonksiyon çalıştığında skoru tutan değişken bir artacaktır ve skor butonunda değişken text olarak görünecektir.

Yazılım Satır ve Bloklarının Açıklanması

```
import time
from tkinter import*
import random
import pyautogui
```

Kullanılacak fonksiyonları içeren modüller içe aktarıldı.

```
sayac=0
```

Skoru tutacak olan değişken tanımlandı. Skor oyunun başında sıfır olacaktır.

```
def fonk():
    x_konum_list = [1500, 700, 1100]
    random.shuffle(x_konum_list)
    x1=x_konum_list[0]
    x2=x_konum_list[1]
    x3=x_konum_list[2]
    buton1.place(x=x1,y=100)
    buton2.place(x=x2, y=100)
    butonK.place(x=x3, y=100)
    buton1.config(bg="orange",activebackground="blue")
    buton2.config(bg="orange",activebackground="blue")
```

Mavi butonlara tıklandığında çalışacak fonksiyon:

Butonların x konumları bir listeye yerleştirildi ve konumlar rastgele karıştırıldı. Daha sonra listenin birinci elemanı buton1'in x değerini ikinci elemanı buton2'nin x değerini ve üçüncü ögesi de butonK'nın x değerini temsil edecek şekilde x1, x2 ve x3 değişkenleri tanımlandı. (Buradaki butonK kırmızı diğerleri mavi butonlar). Listedenden elde edilen x1 değerleriyle ve y=100 olacak şekilde butonların konumları belirlendi. config içinde butonun tıklandığında mavi tıklanmadığında turuncu olacak şekilde activebackground ve bg değerleri atandı.

```
def kirmizi():
    global sayac
    x_konum_list = [1500, 700, 1100]
    random.shuffle(x_konum_list)
    x1 = x_konum_list[0]
    x2 = x_konum_list[1]
    x3 = x_konum_list[2]
    buton1.place(x=x1, y=100)
    buton2.place(x=x2, y=100)
    butonK.place(x=x3, y=100)
    sayac=sayac+1
    butonK.config(activebackground="red",bg="orange")
    pyautogui.moveTo(69,104)
    pyautogui.doubleClick(button="left")
    sayacButon["text"] = sayac
```

Kırmızı butona tıklandığında çalışacak olan fonksiyon:

Yukarıda tanımlanan sayaç değişkeninin fonksiyon içinde ve dışında geçerli olması için global sayaç yazıldı. Mavi butonlara tıklandığında gerçekleşecek butonların yerinin değişmesi olayı burada da olacağından aynı satırları bu fonksiyon da içeriyor. Kırmızı butona tıklandığında skor artacağından sayaç değişkeninin değeri artırıldı. Yukarıdaki gibi burada da kırmızıya tıklandığında buton kırmızı gözükecek. moveTo ile imleç parantez içindeki konuma gidecek. Bu konum sayacın konumu. Ve skoru gösteren butondaki metin, skoru saklayan sayac değişkeni olacak.

```
pencere = Tk()
pencere.title("Kırmızı Kartı Bul")
pencere.geometry("1919x1079")
aciklaEtiket=Label(pencere)
buton1 = Button(pencere)
buton2 = Button(pencere)
butonK = Button(pencere)
sayacButon= Button(pencere)
zaman=Button(pencere)
```

Pencere açılacak ve başlığı “Kırmızı Kartı Bul” olacak. Pencerenin büyüklüğü 1919x1079 olacak. Oyun penceresinde açıklamanın olacağı etiket, kartları temsil eden butonlar, zamanı ve skoru gösteren butonlar oluşturuldu.

```
buton1.config(bg="blue", padx="120", pady="215", command=fonk)
buton2.config(bg="blue", padx="120", pady="215", command=fonk)
butonK.config(bg="red", padx="120", pady="215", command=kirmizi)
sayacButon.config(text=" ", bg="orange", fg="red", padx="25", pady="15",
command=fonk)
aciklaEtiket= Label(pencere, text="" kartlardan
ikisinin
ön yüzü mavi
diğeri kırmızı.
kırmızı olanı
bulmaya çalış!
bu oyun 100
saniye sürecek"", bg="blue", fg="white", padx="150", pady="380",
font=("Vertana ",20))
```

Başlangıçta kartları temsil eden butonların renkleri(bg), boyutları(padx, pady) ve basıldığında ne olacağı(command); skor butonunun metni rengi(fg), metnin rengi, boyutları ve basıldığında ne olacağı; etiketteki açıklama, etiketin ve yazının rengi, tipi ve boyutu(font) belirtildi. Burada kırmızı butona basıldığında imlecin tıklayacağı sayacButon, fonk() fonksiyonunu gerçekleştirecek. Yani kartlar kırmızı karta basıldıktan sonra tekrar karılacak. Etiketin text kısmında üç tane çift tırnak metnin kesildiği yerden alt satıra geçmesi amacıyla kullanıldı.

```
i=0
def fzaman()
    global i
    i=i+1
    time.sleep(1)
    zaman.config(text=str(i),padx="25", pady="15")
    pencere.after(100,fzaman)
    if i==100:
        pyautogui.moveTo(1885, 15)
        pyautogui.doubleClick(button="left")
```

Ne kadar zaman geçtiğini tutması için i değişkeni tanımlandı ve başlangıçta sıfıra eşit. Değişkenin fonksiyon içinde kullanılabilmesi için global değişken olarak tanımlandı ve değeri 1 artırılıp bir saniye beklenecek. Zaman butonunun metni zamanı tutan i değişkenini gösterecek. pencere.after(100,fzaman) bunun pencere açık olduğu süre boyunca devam etmesini sağlayacak. Yani her 1 saniyede bir i değişkeni 1 artacak ve butondaki satı her saniye 1 artacak ve bu pencere kapanana kadar sürecek. i, yani zaman, 100 saniye olduğunda imleç kapatma butonuna gidip tıklayacak ve oyun kapanmış olacak.