# PYTHON DEMO PROJESİ

Bu proje, Python programlama dili ve Kivy kütüphanesi kullanılarak dokunmatik ekran uyumlu bir çizim uygulaması geliştirmek amacıyla hazırlanmıştır. Kivy'nin çoklu dokunma (multi-touch) desteği ve platform bağımsız yapısı, dokunmatik tabanlı uygulamalar geliştirmek için önemli avantajlar sunmaktadır.

Bu avantajlar göz önünde bulundurularak,
kullanıcının fare veya dokunmatik ekran aracılığıyla serbest çizim yapabildiği,
renk ve fırça kalınlığı seçimi yapılabilen bir demo uygulama tasarlanmıştır.

## 2. Kullanılan Teknolojiler

- Python 3.11
- Kivy Framework (2.3.1)
- Kivy KV Language
- Virtual Environment (venv)

## 3. Uygulama Genel Görünümü


Uygulama üç ana bölümden oluşmaktadır:

1. Üst bilgi alanı
2. Çizim alanı
3. Kontrol paneli (renk, fırça kalınlığı ve temizleme)

## 4. KV Dosyası (dokunmatikcizim.kv) Açıklaması

Bu bölümde uygulamanın arayüz tasarımı, Kivy KV Language kullanılarak oluşturulmuştur. KV dili, arayüz bileşenlerinin Python kodundan ayrılmasını sağlayarak daha düzenli ve okunabilir bir yapı sunmaktadır.

**#:kivy 2.2.0** // Bu satır, KV dosyasının Kivy 2.2.0 ve üzeri sürümlerle uyumlu olduğunu belirtir. Sürüm tanımı, olası uyumsuzlukların önüne geçmek amacıyla kullanılmıştır.

### Ana Yerleşim (MainLayout)

Uygulamanın ana yapısı MainLayout adlı bir BoxLayout üzerinden oluşturulmuştur. Tüm bileşenler dikey (vertical) olarak sıralanmıştır.

```
<MainLayout>:
    orientation: "vertical"
    padding: 20
    spacing: 12
```

### Arka Plan Tasarımı

// Bu bölüm, uygulamanın arka planına açık ve sade bir renk vermek amacıyla
kullanılmıştır. canvas.before kullanılarak arka plan,
arayüz bileşenlerinin altında çizdirilmiştir.

```
canvas.before:
    Color:
        rgba: 0.97, 0.97, 0.99, 1
    Rectangle:
        pos: self.pos
        size: self.size
```

### Üst Başlık Bölümü

// Bu bölüm, uygulamanın başlık alanını temsil eder.

```
BoxLayout:
    size_hint_y: 0.13
```

### Çizim Alanı (DrawingArea)

// Bu alan, kullanıcının fare veya dokunmatik ekran ile çizim yaptığı ana bölümdür.

```
DrawingArea:
    id: cizim_alani
```

**line_color: root.current_color**

**line_width: root.brush_width**

**size_hint_y: 0.6**

### Renk Seçim Alanı

//Bu bölümde kullanıcı, çizim rengi seçebilmektedir.

**Button:**

**text: "Siyah"**

**background_color: 0.1, 0.1, 0.1, 1**

**on_release: root.set_color(0, 0, 0, 1)**

### Fırça Kalınlığı Ayarı (Slider)

// Bu bölümde kullanıcı, çizim kalınlığını ayarlayabilmektedir.

**Slider:**

**min: 1**

**max: 15**

**value: root.brush_width**

**on_value: root.update_brush_width(self.value)**

### Seçili Renk Göstergesi

// Bu bölüm, kullanıcının seçtiği rengin anlık olarak görsel biçimde gösterilmesini sağlar.

**RoundedRectangle:**

**rgba: root.current_color**

### Temizle Butonu

// Bu buton ile çizim alanı tamamen temizlenmektedir. Kullanıcı yeni bir çizime sıfırdan başlayabilmektedir.

**Button:**

**text: "Temizle"**

**background_color: 0.85, 0.2, 0.2, 1**

**on_release: root.clear_canvas()**

## DrawingArea Tanımı

**<DrawingArea>:**

# Python tarafında arka plan zaten beyaz ayarlandı

## Genel Değerlendirme

Bu KV dosyası sayesinde:

- Arayüz ile uygulama mantığı ayrılmıştır
- Kod okunabilirliği artırılmıştır
- Dokunmatik uyumlu, sade ve kullanıcı dostu bir tasarım elde edilmiştir

## Python Kodu (main.py) Açıklaması

## 1. Kütüphane İçe Aktarımları (Imports)

**from kivy.app import App** // Kivy uygulamasının ana sınıfıdır.

**from kivy.uix.boxlayout import BoxLayout** // Arayüzde dikey/yatay yerleşim için kullanılır.

**from kivy.uix.widget import Widget** // Özel çizim alanını oluşturmak için temel sınıftır.

**from kivy.graphics import Color, Line, Rectangle** // Canvas üzerine çizim yapmak için kullanılır.

**from kivy.properties import ListProperty, NumericProperty, StringProperty** // KV dosyasıyla otomatik veri bağlamak (binding) için kullanılır.

## 2. DrawingArea Sınıfı (Çizim Alanı)

**class DrawingArea(Widget):**// Bu sınıf, kullanıcının dokunmatik ekran/fare ile çizim yaptığı özel çizim alanıdır.

// Çizim Rengi ve Kalınlık Property'leri

**line_color = ListProperty([0, 0, 0, 1])**

**line_width = NumericProperty(3)**

// Arka Planın Beyaz Yapılması

**with self.canvas.before:**

**Color(1, 1, 1, 1)**

**self.bg_rect = Rectangle(pos=self.pos, size=self.size)**

**self.bind(pos=self._update_bg, size=self._update_bg)**

// Arka Plan Güncelleme Fonksiyonu

```
def _update_bg(self, *args):
    self.bg_rect.pos = self.pos
    self.bg_rect.size = self.size
```

// Çizime Başlama (on_touch_down)

```
def on_touch_down(self, touch):
    if not self.collide_point(*touch.pos):
        return False
    with self.canvas:
        Color(*self.line_color)
        touch.ud["line"] = Line(points=[touch.x, touch.y], width=self.line_width)
    return True
```

// Çizerken Devam Etme (on_touch_move)

```
def on_touch_move(self, touch):
    if "line" in touch.ud:
        touch.ud["line"].points += [touch.x, touch.y]
```

// Temizleme İşlemi (clear_canvas)

```
def clear_canvas(self):
    self.canvas.clear()
    with self.canvas.before:
        Color(1, 1, 1, 1)
        self.bg_rect = Rectangle(pos=self.pos, size=self.size)
    self.bind(pos=self._update_bg, size=self._update_bg)
```

### 3.MainLayout Sınıfı (Uygulama Kontrol Merkezi)

**class MainLayout(BoxLayout):**// Bu sınıf, KV arayüzündeki buton/slider işlemlerini yönetir.

// Uygulama Durum Property'leri

**current_color = ListProperty([0, 0, 0, 1])**

**brush_width = NumericProperty(3)**

**info_text = StringProperty("Parmağınızla veya fareyle alanda çizim yapabilirsiniz.")**

// Renk Değiştirme

**def set_color(self, r, g, b, a=1):** // KV dosyasındaki renk butonları bu fonksiyonu çağırır ve çizim rengi güncellenir.

**self.current_color = [r, g, b, a]**

// Temizleme Butonu

**def clear_canvas(self):**

**self.ids.cizim_alani.clear_canvas()**

//Slider ile Kalınlık Ayarı

**def update_brush_width(self, value):** // Slider hareket ettikçe kalınlık güncellenir. `int()` ile değer tam sayıya çevrilir

**self.brush_width = int(value)**

### 4. Uygulama Sınıfı (App)

//Kivy uygulamasının başlangıç noktasıdır.build() fonksiyonu ana arayüzü döndürür (MainLayout).

**class DokunmatikCizimApp(App):**

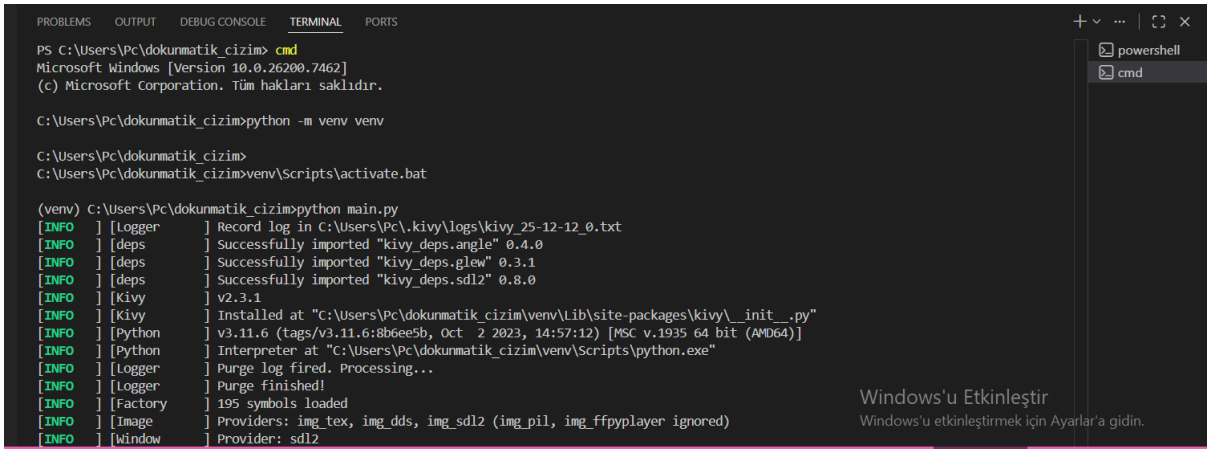
**def build(self):**

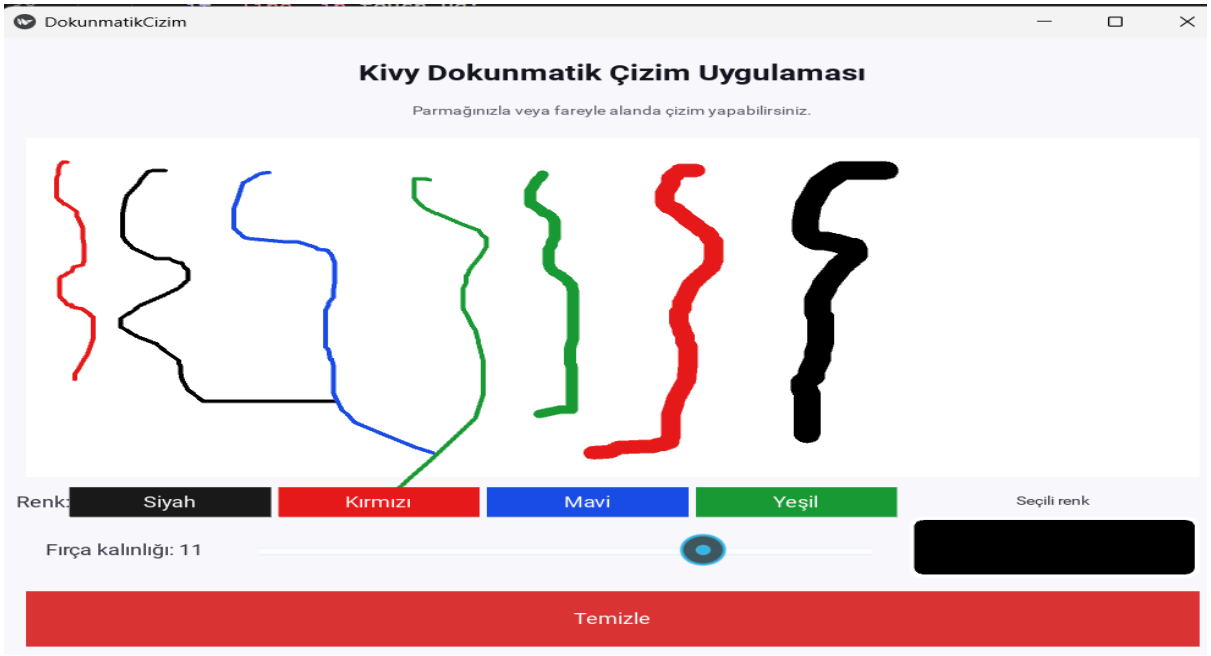
**return MainLayout()**

### 5. Programın Çalıştırılması

**if name == "main": DokunmatikCizimApp().run()**

//Dosya doğrudan çalıştırıldığında uygulama başlatılır.


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS C:\Users\Pc\dokunmatik_cizim> cmd
Microsoft Windows [Version 10.0.26200.7462]
(c) Microsoft Corporation. Tüm hakları saklıdır.
C:\Users\Pc\dokunmatik_cizim>python -m venv venv
C:\Users\Pc\dokunmatik_cizim>
C:\Users\Pc\dokunmatik_cizim>venv\Scripts\activate.bat
(venv) C:\Users\Pc\dokunmatik_cizim>python main.py
[INFO ] [Logger ] Record log in C:\Users\Pc\.kivy\logs\kivy_25-12-12_0.txt
[INFO ] [deps ] Successfully imported "kivy_deps.angle" 0.4.0
[INFO ] [deps ] Successfully imported "kivy_deps.glew" 0.3.1
[INFO ] [deps ] Successfully imported "kivy_deps.sdl2" 0.8.0
[INFO ] [Kivy ] v2.3.1
[INFO ] [Kivy ] Installed at "C:\Users\Pc\dokunmatik_cizim\venv\Lib\site-packages\kivy__init__.py"
[INFO ] [Python ] v3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)]
[INFO ] [Python ] Interpreter at "C:\Users\Pc\dokunmatik_cizim\venv\Scripts\python.exe"
[INFO ] [Logger ] Purge log fired. Processing...
[INFO ] [Logger ] Purge finished!
[INFO ] [Factory ] 195 symbols loaded
[INFO ] [Image ] Providers: img_tex, img_dds, img_sdl2 (img_pil, img_ffpyplayer ignored)
[INFO ] [Window ] Provider: sdl2
powershell
cmd
Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.


DokunmatikCizim
Kivy Dokunmatik Çizim Uygulaması
Parmağınızla veya fareyle alanda çizim yapabilirsiniz.
Renk:
Siyah
Kırmızı
Mavi
Yeşil
Seçili renk
Fırça kalınlığı: 11
Temizle