

Flutter Animasyonları Hakkında Rapor

Hazırlayan

1030510285 Tunahan Akçay

Amaç

Bu projenin amacı Erciyes Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü, Dr. Öğr. Üyesi FEHİM KÖYLÜ tarafından verilen Mobile Application Development dersi kapsamında, Flutter animasyonlarının nasıl kullanılacağını açıklamak ve basit örneklerle konuyu anlatan bir rapor hazırlamaktır.

Kullanılan Teknolojiler

- Flutter SDK
- Dart dili
- Flutter animasyon widget'ları:
 - AnimatedContainer
 - AnimatedOpacity
 - TweenAnimationBuilder

Animasyon Nedir?

Animasyonlar, kullanıcı arayüzünde geçişleri ve durum değişikliklerini daha akıcı ve estetik hale getiren görsel efektlerdir. Flutter, animasyonlar için hem basit hem de gelişmiş seviyede kontrol sağlayan araçlar sunar. Kullanıcı etkileşimlerini daha anlamlı ve sezgisel hale getirir.

Basit Animasyon Örnekleri

1. AnimatedContainer ile Boyut ve Renk Değişimi

Bu örnekte kullanıcı kutuya tıkladığında, kutunun boyutu ve rengi animasyonla değişmektedir.

Kod Örneği: AnimatedContainer Kullanımı

```
class AnimatedBox extends StatefulWidget {  
  @override  
  _AnimatedBoxState createState() => _AnimatedBoxState();  
}
```

```
class _AnimatedBoxState extends State<AnimatedBox> {  
  double _size = 100;  
  Color _color = Colors.blue;  
  
  void _changeBox() {  
    setState(() {  
      _size = _size == 100 ? 200 : 100;  
      _color = _color == Colors.blue ? Colors.red : Colors.blue;  
    });  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    return GestureDetector(  
      onTap: _changeBox,  
      child: AnimatedContainer(  
        width: _size,  
        height: _size,  
        color: _color,  
        duration: Duration(seconds: 1),  
        curve: Curves.easeInOut,  
      ),  
    );  
  }  
}
```

2. AnimatedOpacity ile Görünürlük Efektı

Butona basıldığında, metin yavaşça kaybolur veya geri görünür. Bu animasyon, öğeleri gizleme/gösterme durumlarında şık geçişler sağlar.

Kod Örneği: AnimatedOpacity Kullanımı

```
class FadeWidget extends StatefulWidget {  
  @override  
  _FadeWidgetState createState() => _FadeWidgetState();  
}  
  
class _FadeWidgetState extends State<FadeWidget> {  
  double _opacity = 1.0;  
  
  void _toggleOpacity() {  
    setState(() {  
      _opacity = _opacity == 1.0 ? 0.0 : 1.0;  
    });  
  }  
}
```

```
@override
Widget build(BuildContext context) {
  return Column(
    children: [
      AnimatedOpacity(
        opacity: _opacity,
        duration: Duration(seconds: 1),
        child: Text('Merhaba Flutter!'),
      ),
      ElevatedButton(
        onPressed: _toggleOpacity,
        child: Text('Gizle/Göster'),
      ),
    ],
  );
}
```

3. TweenAnimationBuilder ile Renk Geçişi

`TweenAnimationBuilder` ile belirli bir başlangıç ve bitiş değeri arasında yumuşak bir geçiş elde edilebilir. Bu örnekte, bir kutunun rengi zamanla değişir.

Kod Örneği: TweenAnimationBuilder Kullanımı

```
class ColorTweenBox extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return TweenAnimationBuilder<Color?>(
      tween: ColorTween(begin: Colors.green, end: Colors.orange),
      duration: Duration(seconds: 2),
      builder: (context, color, child) {
        return Container(
          width: 100,
          height: 100,
          color: color,
        );
      },
    );
  }
}
```

Sonuç

Flutter, animasyonları kolay ve etkili bir şekilde uygulamak için çok sayıda hazır widget sunar. Bu animasyonlar:

- Uygulamanın kullanıcı deneyimini artırır
- Görsel geçişleri daha yumuşak ve doğal hale getirir
- Arayüze profesyonel bir hava katar

Basit animasyonlarla başlamak, daha gelişmiş animasyonlara geçiş için iyi bir adımdır.

`AnimatedContainer` , `AnimatedOpacity` ve `TweenAnimationBuilder` gibi bileşenler, Flutter'da animasyonların temel taşlarını oluşturur.