



Casual Games mit Flutter

*Joachim Böhmer
Petra Langenbacher*



Joachim Böhmer

 kaptnkoala



Petra Langenbacher

 developpetra



Agenda = {

- 1: Warum Flutter?
- 2: Game Design
- 3: Layout System
- 4: Layer Links
- 5: Animationen
- 6: Partikel System
- 7: Widgetbook
- 8: A/B Tests
- 9: Rückblick / Ausblick

}

{ Warum Flutter? }



Historie von 4 Pics



2012



zwei
separate Apps



2020
Rewrite



4 Bilder 1 Wort

Made with ❤ in

Bad Nauheim

(Hessen)





Rewrite: Probleme im Web



Offline first

Langsamer Datentransport über Bridge



Visuelle Probleme

Endlos scrollende Listen, Darstellung Lottie Animationen,
Page Transitions & PageViews



Viele Workarounds für natives Erlebnis

The interface includes a top bar with a brain icon, IQ 105, a star icon, 279 points, a coin icon, 999 coins, and a plus sign. Below the images is a word puzzle grid with letters A, U, F, G, T, E, B, N, Y, E, and a green checkmark icon with '60' next to it. At the bottom is an orange button labeled 'Frag deine Freunde' and a small 'X' icon.



Anforderungen

- Native first (mit Option auf Web)
- Mehr “normale App” als Spiel
- Routing zu verschiedenen Unterseiten







Anforderungen

- Native first (mit Option auf Web)
- Mehr “normale App” als Spiel
- Routing zu verschiedenen Unterseiten
- Viele Animationen





Warum Flutter?



Cross-
Plattform
Support



Dart: viel
Funktionalität
out of the box



Flutter
Framework &
Eco System



{ Game Design }



Game Design mit Flutter

- Flutter ideal für Umsetzung von Casual Games
- Spezielle Optik entsteht durch viele Verläufe
- Die meisten Elemente sind Decorated Boxes
- Für komplexe Widgets nutzen wir Canvas API

Spielen



```
class Button extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Container(  
      width: 260,  
      height: 160,  
      color: const Color(0xff7cb400),  
    );  
  }  
}
```

Spielen



```
class Button extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Container(  
      width: 260,  
      height: 160,  
      decoration: BoxDecoration(  
        borderRadius: BorderRadius.circular(20),  
        border: Border.all(color: const Color(0xffb8d700), width: 8),  
        gradient: const VerticalGradient(  
          colors: [Color(0xffb8d700), Color(0xff7cb400)],  
        ),  
        boxShadow: const [  
          BoxShadow(color: Color(0xff49660b), offset: Offset(0, 6)),  
        ],  
      ),  
    );  
  }  
}
```

Spielen



```
class Button extends StatefulWidget {}

class ButtonState extends State<Button> {
    bool _pressed = false;

    @override
    Widget build(BuildContext context) {
        return GestureDetector(
            onTapDown: (_) => setState(() => _pressed = true),
            onTapUp: (_) => setState(() => _pressed = false),
            child: Transform.translate(
                offset: Offset(0, _pressed ? 3 : 0),
                child: Container(
                    decoration: BoxDecoration(
                        boxShadow: [
                            BoxShadow(offset: Offset(0, _pressed ? 3 : 6)),
                        ],
                    ),
                ),
            ),
        );
    }
}
```

Spielen



```
class _ButtonState extends State<Button> {
  bool _pressed = false;

  @override
  Widget build(BuildContext context) {
    return GestureDetector(
      onTapDown: (_) => setState(() => _pressed = true),
      onTapUp: (_) => setState(() => _pressed = false),
      behavior: HitTestBehavior.opaque,
      child: Transform.translate(
        offset: Offset(0, _pressed ? 3 : 0),
        child: Container(
          width: 260,
          height: 160,
          margin: const EdgeInsets.all(20),
          decoration: BoxDecoration(...),
        ),
      ),
    );
  }
}
```





Decorated Text

- Dekorieren von Text mit Rand und Verlauf
- Text Widget unterstützt dieses nicht direkt
- Eigene DecoratedText Implementierung

Normaler Text

Text mit Outline

Text mit Verlauf



```
class DecoratedText extends StatelessWidget {  
    final String text;  
    final TextStyle? style;  
    final Color strokeColor;  
    final double strokeWidth;  
  
    @override  
    Widget build(BuildContext context) {  
        return Stack(  
            children: [  
                Text(  
                    text,  
                    style: style?.copyWith(  
                        // Use the foreground paint of TextStyle to draw an text outline  
                        foreground: Paint()  
                            ..style = PaintingStyle.stroke  
                            ..strokeJoin = StrokeJoin.round  
                            ..color = strokeColor  
                            ..strokeWidth = strokeWidth * 2,  
                    ),  
                ),  
                Text(text, style: style.copyWith(shadows: [])),  
            ],  
        );  
    }  
}
```

Text mit Outline

Text mit Outline

Text mit Outline



```
class DecoratedText extends StatelessWidget {  
    final String data;  
    final TextStyle? style;  
    final Gradient? gradient;  
  
    @override  
    Widget build(BuildContext context) {  
        return Stack(  
            children: [  
                if (gradient == null)  
                    Text(data, style: style)  
                else  
                    // Wrap the text with shader mask to blend a gradient with the text  
                    ShaderMask(  
                        shaderCallback: (bounds) => gradient!.createShader(bounds.size),  
                        child: Text(  
                            data,  
                            style: style?.copyWith(color: Colors.white),  
                        ),  
                    ),  
            ],  
        );  
    }  
}
```

Text mit Verlauf

Text mit Verlauf



Gradient Border

- Rand mit Verlauf **nicht direkt** unterstützt
- **Workaround** über verschachtelte Container
- GradientBorder Implementierung
- Verschiedene Packages auf **pub.dev**





```
class GradientBorder extends Border {  
  const GradientBorder({ this.width = 0.0, required this.gradient});  
  
  final Gradient gradient;  
  final double size;  
  
  @override  
  void paint(Canvas canvas, Rect rect, {BorderRadius? borderRadius}) {  
    final paint = Paint()  
      ..style = PaintingStyle.stroke  
      ..strokeWidth = size  
      ..shader = gradient.createShader(rect);  
  
    if (borderRadius != null) {  
      canvas.drawRRect(borderRadius.toRRect(rect).deflate( size / 2), paint);  
    } else {  
      canvas.drawRect(rect.deflate( size / 2), paint);  
    }  
  }  
}
```





Shape Decoration

- Zeichnen und Dekorieren von **Formen**
- ShapeDecoration und ShapeBorder
- Nutzung für eigene Tooltip Komponente

Dieser Tooltip ist
überhalb des Buttons

Spielen

Spielen

Dieser Tooltip ist
unterhalb des Buttons

```
● ● ●  
  
class TooltipStyle {  
    final Color? color;  
    final Gradient? gradient;  
    final List<BoxShadow>? shadows;  
    final double borderRadius;  
    final double borderSize;  
    final Color borderColor;  
}  
  
class Tooltip extends StatelessWidget {  
    @override  
    Widget build(BuildContext context) {  
        return DecoratedBox(  
            decoration: ShapeDecoration(  
                color: style.color,  
                gradient: style.gradient,  
                shadows: style.shadows,  
                shape: _TooltipShape(placement: placement, style: style, offset: offset),  
            ),  
            child: content,  
        );  
    }  
}
```

Dieser Tooltip ist
überhalb des Buttons

Spiele

Spiele

Dieser Tooltip ist
unterhalb des Buttons



```
class _TooltipShape extends ShapeBorder {  
  @override  
  Path getOuterPath(Rect rect, {TextDirection? textDirection}) {  
    return Path()  
      ..moveTo(rect.left, rect.bottom - borderRadius)  
      ..lineTo(rect.left, rect.top + borderRadius)  
      ..arcToPoint(Offset(rect.left + borderRadius, rect.top), radius:  
        Radius.circular(borderRadius))  
    // More path operations to draw the tooltip pointing up or down  
    ..lineTo(rect.left + borderRadius, rect.bottom)  
    ..arcToPoint(Offset(rect.left, rect.bottom - borderRadius), radius:  
      Radius.circular(borderRadius))  
    ..close();  
  }  
  
  @override  
  void paint(Canvas canvas, Rect rect, {TextDirection? textDirection}) {  
    final paint = Paint()  
      ..style = PaintingStyle.stroke  
      ..color = style.borderColor  
      ..strokeWidth = style.borderWidth;  
    canvas.drawPath(getOuterPath(rect), paint);  
  }  
}
```

Dieser Tooltip ist
überhalb des Buttons

Spielen

Spielen

Dieser Tooltip ist
unterhalb des Buttons

{ Layout System }



iPhone SE



iPhone 15 Pro



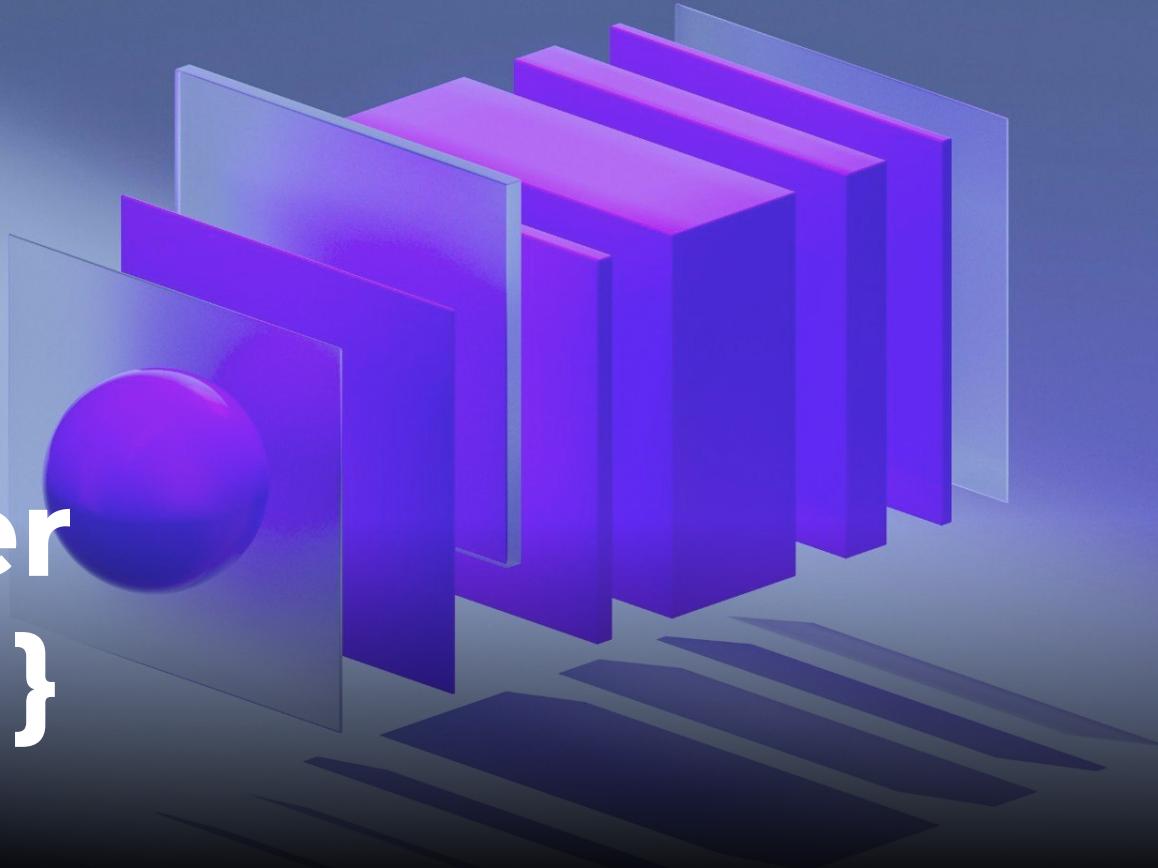
iPad Air 13-inch



```
class NativeApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      builder: (context, child) {
        initLayout(context);
        return child!;
      },
    );
  }
}

class AppIcon extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Transform.rotate(
      angle: radians(10),
      child: Assets.common.image.appicons.appiconDe //.
        .fitted(width: dp(54), height: dp(54)),
    );
  }
}
```

{ Layer Links }





LayerLink Handling

- Positionierung von Komponenten über Layer hinweg
- Widgets mit Hilfe von LayerLinks relativ zueinander positionieren





guess_page.dart:

```
CompositedTransformTarget(  
  link: shareButtonLink,  
  child: ShareButton(  
    onTap: () {},  
  ),  
) ;
```



share_tutorial.dart:

```
CompositedTransformFollower(  
  link: shareButtonLink,  
  child: TappingAnimation(  
    animation: controller,  
    builder: (context, tapping) {  
      return ShareButton(  
        onTap: () {},  
        pressed: tapping,  
      );  
    },  
  ),  
) ;
```

The background of the slide features a dynamic, abstract liquid flow. A large, translucent blue and purple blob is moving from the bottom left towards the top right, creating a wake of smaller bubbles and ripples. The overall effect is organic and fluid, suggesting movement and energy.

{ Animationen }



Animationen

Implicit
Animations

- Automatische Animation von veränderten Eigenschaften
- Interner AnimationController

AnimatedAlign
AnimatedContainer
AnimatedDefaultTextStyle
AnimatedScale
AnimatedRotation
AnimatedSlide
AnimatedOpacity
AnimatedPadding
AnimatedPhysicalModel
AnimatedPositioned
AnimatedPositionedDirectional
AnimatedTheme
AnimatedCrossFade
AnimatedSize
AnimatedSwitcher
...



Animationen

Implicit
Animations

- Automatische Animation von veränderten Eigenschaften
- Interner AnimationController

Explicit
Animations

- Manuelles Verwalten des AnimationControllers
- Mehr Kontrolle über die Animation

```
AlignTransition
DecoratedBoxTransition
DefaultTextStyleTransition
PositionedTransition
RelativePositionedTransition
RotationTransition
ScaleTransition
SizeTransition
SlideTransition
FadeTransition
AnimatedModalBarrier
...
```



Animationen

Implicit
Animations

- Automatische Animation von veränderten Eigenschaften
- Interner AnimationController

Explicit
Animations

- Manuelles Verwalten des AnimationControllers
- Mehr Kontrolle über die Animation

Animated
Builder

- Allzweck-Widget zur Erstellung von Animationen
- Nützlich für komplexere Widgets





Key Transitions

- Widget zwischen zwei Positionen animieren
- Global Keys → Berechnung Offset relativ zum Ancestor/ globalen Koordinatensystem
- PositionedTransition zwischen Start- und Target-Offset



M A N G O



```
KeyTransition (  
    startKey: startKey,  
    targetKey: targetKey,  
    layerKey: layerKey,  
    startSize: Size.square(20),  
    targetSize: Size.square(40),  
    animation: animation,  
    child: Assets.common.image.checkmark.fitted(width: 30),  
)
```



Timeline Animations

IQ 82 221 1015 +

X G L O B Z A ✓
E T N E L Y 🗑 150

Frag deine Freunde

lorum



Timeline Animations

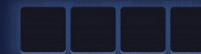
```
final controller = AnimationController(duration:  
    const Duration(seconds: 3)), vsync: this);
```

The image shows a mobile game interface. At the top right are icons for a brain (IQ 82), a red badge with 221, a star, and a coin (1015). Below are four cards with images: a woman in a field, a person in a yellow vest, bees on honeycomb, and a woman standing next to a car. At the bottom are four empty square slots, a word puzzle grid with letters X, G, L, O, B, Z, E, T, N, E, L, Y, and a button that says "Frag deine Freunde".



Timeline Animations

```
final controller = AnimationController(duration:  
    const Duration(seconds: 3), vsync: this);  
  
final animation = Tween<Offset>(begin:  
    Offset.zero, end: const Offset(1,  
    1)).animate(controller);
```



Frag deine Freunde



Timeline Animations

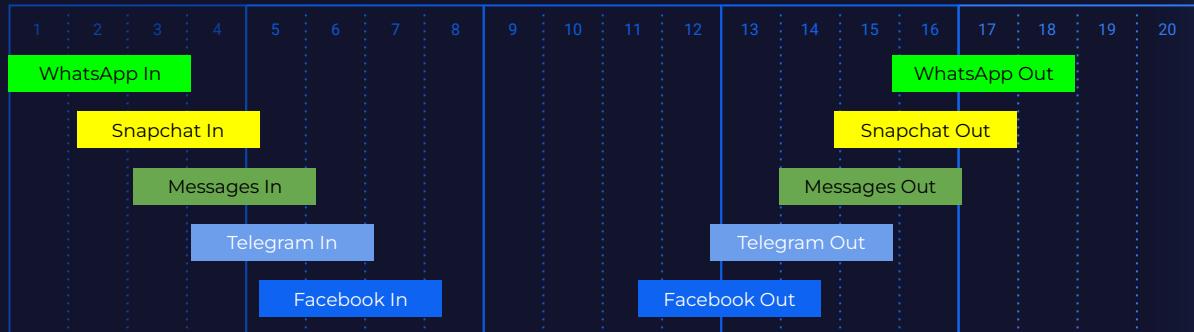
```
final animation = TweenSequence<double>(  
    <TweenSequenceItem<double>>[  
        TweenSequenceItem<double>(  
            tween: Tween<double>(begin: 0, end: 1),  
            weight: 4,  
        ),  
        TweenSequenceItem<double>(  
            tween: ConstantTween<double>(1),  
            weight: 2,  
        ),  
        TweenSequenceItem<double>(  
            tween: Tween<double>(begin: 1, end: 0),  
            weight: 4,  
        ),  
    ],  
).animate(controller);
```





Timeline Animations

Design Übergabe: Animation Spec





Timeline Animations

- Eigenes Animationssystem mit einem "Timeline"-Konzept
- Kontrolle über genaue Timings
- Strukturiertere Verwaltung von Animationen





```
final controller = TimelineController(to: 12, vsync: this);

final animation = controller.animationSections([
    AnimationSection(0.5, 1.9, tween: Tween(begin: 0, end: 1)),
    AnimationSection(2.1, 4.7, tween: ConstantTween<double>(1)),
    AnimationSection(4.9, 6.8, tween: Tween(begin: 1, end: 0)),
]) ;
```



Scene Animations

- Framework zur Verwaltung komplexer Animationen und Partikelsysteme
- SceneController mit verschiedenen Scenes, die wiederum Sceneltems enthalten (Animationen, Partikelanimationen, TimedActions)
- Scenes kann man einzeln abspielen





Scene Animations

Weiter

IQ 83 222 1031 +

The interface shows a top bar with user stats: IQ 83, 222 notifications, 1031 coins, and a plus sign. Below are four square cards: a path through a green field, a snowy landscape with tracks, a close-up of a finger with a fingerprint, and a bunch of red matches. At the bottom is a grid of letters (W, K, Z, Ü, U, A, V, P, E, U, P, P) and a keyboard icon. An orange button at the bottom right says "Frag deine Freunde".



Scene Animations

SceneController

Scene enter



Scene exit





```
void initState() {
    super.initState();
    controller = sceneController([
        Scene(
            Scene.enter,
            items: [
                SimpleAnimation(Animation.opacity, 0.0, 1.5, curve: Curves.easeOut),
                SimpleAnimation(Animation.scale, 0.5, 2.0, curve: Curves.easeOut),
            ],
        ),
        Scene(
            Scene.exit,
            items: [
                SimpleAnimation.reverse(Animation.scale, 0.0, 1.5, curve: Curves.easeOut),
                SimpleAnimation.reverse(Animation.opacity, 0.5, 2.0, curve: Curves.easeOut),
            ],
        ),
    ]).forward();
}
```



```
Widget build(BuildContext context) {  
  return ScaleTransition(  
    scale: controller.animation(Animation.scale),  
    child: FadeTransition(  
      opacity: controller.animation(Animation.opacity),  
      child: Button(),  
    ),  
  );  
}
```

{ Partikel System }



Partikel System

- Für **Games** sind Partikelsysteme essentiell
- **Flame Engine** zu sehr limitiert
- Eigene Implementierung auf **Canvas API**
- **Web-basierter Editor** für HTML5 Games





```
class Emitter {  
    final EmitterLocation location;  
    final double rate;  
    final ParticleSize size;  
    final int max;  
    final DurationValue lifetime;  
    final SpinValue? spin;  
    final ParticleValue? speed;  
    final ParticleValue? direction;  
    final ParticleValue? scale;  
    final List<ParticleShape> shapes;  
}  
  
class ParticleValue {  
    final double start;  
    final double range;  
    final List<double> changes;  
}  
  
class ImageShape extends ParticleShape {  
    final Image image;  
}
```





```
abstract class PhysicsField {
    void affect(Particle particle, int milliseconds, double progress);
}

class GravitationField implements PhysicsField {
    final double direction;
    final double strength;

    @override
    void affect(Particle particle, int milliseconds, double progress) {
        particle.applyForce(direction, strength, milliseconds);
    }
}

class GravitationCenterField implements PhysicsField {
    final Position center;
    final double strength;

    @override
    void affect(Particle particle, int milliseconds, double progress) {
        // calculate how the particle is affected by gravitation center
    }
}
```





400

```
class ConfettiParticleAnimation extends StatelessWidget {  
  const ConfettiParticleAnimation({ this.controller, this.amount = 150});  
  
  final ParticleController? controller;  
  final int amount;  
  
  @override  
  Widget build(BuildContext context) {  
    return ParticleSystem(  
      Assets.common.animation.particlesConfetti,  
      controller: controller,  
      reviver: (emitters) async {  
        for (final emitter in emitters) {  
          emitter.birth = emitter.birth.copyWith(  
            max: amount,  
            rate: 200 + amount * 2,  
          );  
        }  
      },  
    );  
  }  
}
```



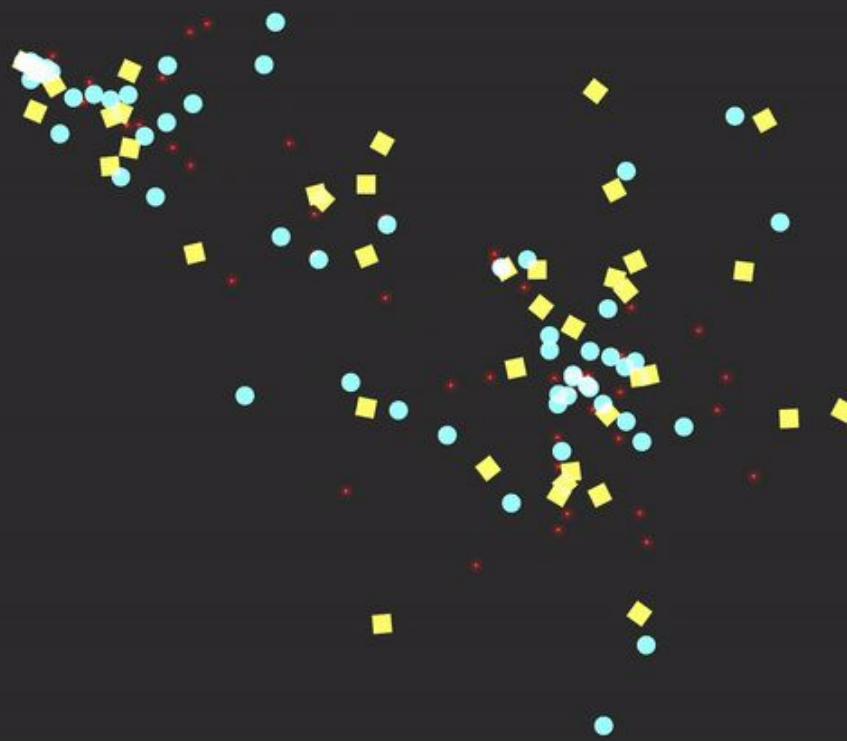
400

```
class _ParticleSystemState extends State<ParticleSystem> {
    final ParticleRuntime _runtime;
    final ParticleController _controller;

    @override
    Widget build(BuildContext context) {
        return CustomPaint(
            painter: _ParticlePainter(_controller, runtime: _runtime),
            size: widget.size,
        );
    }
}

class _ParticlePainter extends CustomPainter {
    @override
    void paint(Canvas canvas, Size size) {
        final time = _controller.lastElapsedDuration!.inMilliseconds;
        _runtime.tick(time, size);

        for (final particle in _runtime.particles) {
            _drawParticle(canvas, size, particle, time);
        }
    }
}
```



Particle

Birth

- Rate: 1 P/s
- Constant:

Amount of particles

- Particles: 0 #
- Restart:

Lifetime

- duration: 1 ms
- range: 0 ms



Aktuelle Überlegungen

- Aktuelles Partikel System ist **zu unflexibel**
- HTML5 Teams nutzen **weiterentwickelte Version**
- Nutzen **dynamische Ausführung** von Javascript
- Programmierung **mit Dart** größte Flexibilität
- Entwicklung von Partikel Animation **in DartPad?**



The background of the slide features a wide-angle photograph of a grand, traditional library. The room is filled with floor-to-ceiling bookshelves packed with books. A large, ornate chandelier hangs from the ceiling, casting light down. In the center, there's a long wooden table with several lamps and some papers or books on it. A balcony level with more bookshelves is visible in the background.

{ Widgetbook }



Entwicklung in Isolation

- Iterative Entwicklung **im Use Case**
- Abbildung von **allen UI Zuständen**
- Identifizierung von **minimaler Widget API**





Tweaken von Animationen

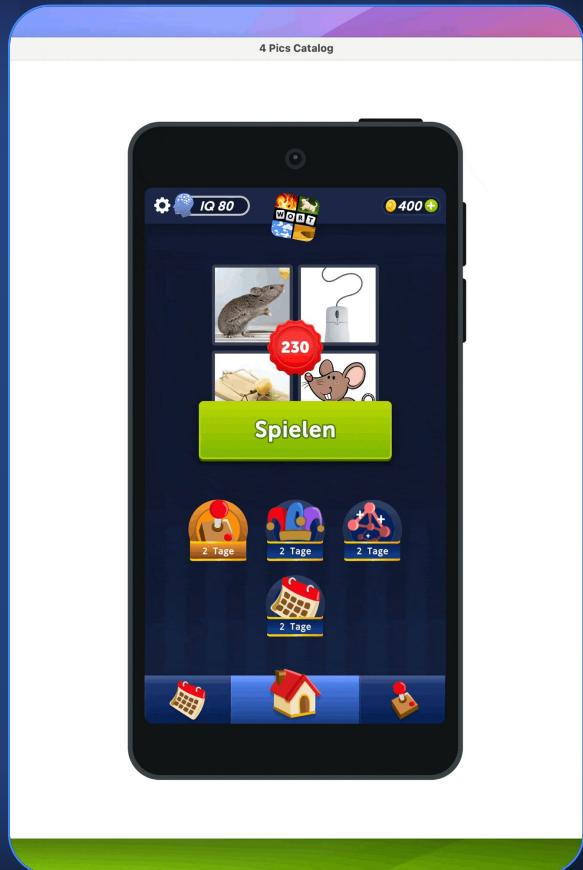
- **TimeDilationAddon** von Widgetbook
- Geschwindigkeit von Animationen justierbar
- Koordinierung von nebenläufigen Animationen
- Abstimmung von Animationen mit Design





Layout Validierung

- **DeviceFrameAddon** von Widgetbook
- Direkte Umschaltung zwischen Gerätegrößen
- Eigene Gerätegrößen konfigurierbar
- Layout bei Entwicklung einfacher validierbar



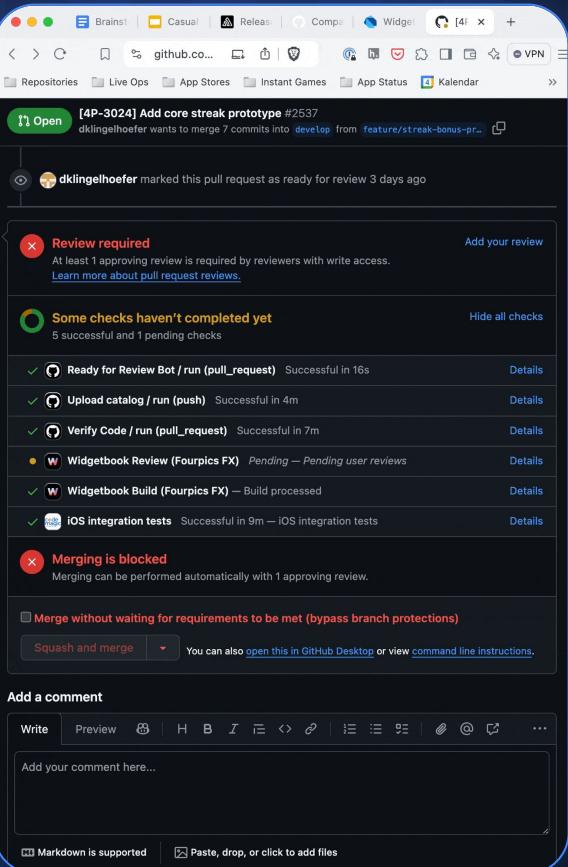


Review der Use Cases

Review der Use Cases

- Abbildung von **neuen Widgets** über Use Cases
- **Alle UI Zustände** der Widgets testbar
- Review der UI Aspekte werden vereinfacht

The screenshot shows a GitHub pull request page for a repository named 'Widget'. The pull request is titled '[4P-3024] Add core streak prototype #2537' and is from the user 'dklingelhoefer'. It has been marked as ready for review 3 days ago. The review status indicates that at least one approving review is required by reviewers with write access. There are several check status items: 'Some checks haven't completed yet' (5 successful and 1 pending), 'Ready for Review Bot / run (pull_request)' (Successful in 16s), 'Upload catalog / run (push)' (Successful in 4m), 'Verify Code / run (pull_request)' (Successful in 7m), 'Widgetbook Review (Fourpicks FX)' (Pending - Pending user reviews), 'Widgetbook Build (Fourpicks FX)' (Build processed), and 'iOS Integration tests' (Successful in 9m — iOS integration tests). A note states that merging is blocked due to requirements not being met. The 'Merge without waiting for requirements to be met (bypass branch protections)' button is visible. At the bottom, there is a comment input field with a rich text editor and a note that Markdown is supported.

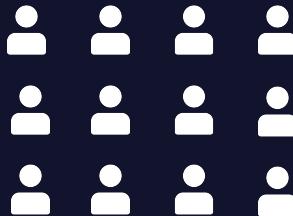
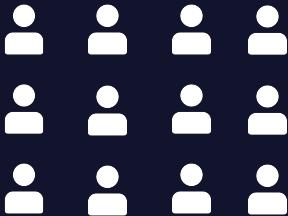


{ A/B Tests }



A/B Tests

Vergleich von zwei (oder mehr) Versionen eines Features





A/B Tests



Anwenden



Neues Feature



Varianten



Analysieren



Test



ab_testing_core 1.0.0

Published 10 months ago • [programmier.bar](#) Dart 3 compatible

SDK | DART | FLUTTER | PLATFORM | ANDROID | IOS | LINUX | MACOS | WEB | WINDOWS

1 7

[Readme](#) [Changelog](#) [Installing](#) [Versions](#) [Scores](#)



AB Testing - Core

A Dart package that helps to implement multiple a/b tests.

Features

This package simplifies the creation and management of multiple a/b tests. It comes with a local adapter for local a/b tests and can be extended with further adapters. It supports test values of the types `bool`, `int`, `String` and `Enum`.

The package `ab_testing_firebase` provides an additional firebase adapter that simplifies remote a/b tests.

7 LIKES | 140 PUB POINTS | 45% POPULARITY

Publisher

[programmier.bar](#)

Metadata

A/B testing for user experience research through randomized experiments that involve two or more variants.

[Repository \(GitHub\)](#)

[View/report issues](#)

Documentation

[API reference](#)

License

[MIT \(license\)](#)

More

[Packages that depend on ab_testing_core](#)





A/B Tests

Best Practices



Anpassbar
zum Testen



Ein Feature
nach dem
anderen



Statistischen
Signifikanz
auswerten



{ Rückblick / Ausblick }



Herausforderungen

- Performance in Flutter Web
- Animation Jank
- Aktuelle Issues in Impeller
- Junges Ökosystem





Ausblick

- Einführung Design System
 - Import von Variablen über Figma API
 - Eigener Transformer





Ausblick

- Einführung Design System
 - Import von Variablen über Figma API
 - Eigener Transformer
- Verstärkete Nutzung von Rive Animationen
 - Performance Probleme mit Lottie
 - Zustandsmaschine aus Code beeinflussbar





Vorteile für unser Team



Multi Plattform
Support inkl.
Web



Schnelle
Entwicklungs-
zyklen



Gute
Developer
Experience

The game interface includes:

- IQ 105
- 550 points
- 1 star achievement
- 999 coins

Game screenshots:

- A hand pointing at a smiley face icon.
- A group of hands giving thumbs up.
- A chef in a white uniform holding a spoon.
- A scenic landscape with rolling hills and vineyards.

Bottom row buttons:

- Blank
- Blank
- Blank

Letter grid:

H	Ä	P	K	T	U	A ✓ 60
G	X	X	Z	C	V	trash 150

Frag deine Freunde



Flutter

Multi-Platform Development Ecosystem Showcase Docs Get started

Flutter Casual Games Toolkit

Free & open source
multiplatform 2D game
development in Flutter



Free & Open Source

Multiplatform Games

Packages & Integrations

No code changes required for native

Write once, run everywhere and develop

Speed up development with native built-in



Multi-Platform Development Ecosystem Showcase Docs Q Get started

Flutter Casual Toolkit

Free & open source multiplatform 2D game development in Flutter

Free & Open Source

No cost charges or usage fees



Multi-Platform Development Ecosystem Showcase Docs Q Get started

Flutter Games

Case studies



PUBG Mobile



Google I/O Pinball



Super TicTacToe



Wallace & Gromit AR



4 Pics 1 Word



Doodle Dash



Danke
für eure
Aufmerksamkeit. ❤