

# App-Programmierung Mit Dart und Flutter

Dart Syntax  
Google Material Design  
Flutter Layouts  
Native Features  
Plugins z.B. Flame Game Engine



## Dart?

- Ähnlich wie Java, C#, C++
- (V 2)
- Compiliert Ahead of Time oder Just in Time (für schnelle Entwicklung)

```
void main() {  
    for (int i = 0; i < 5; i++) {  
        print('hello ${i + 1}');  
    }  
}
```

main.dart

```
> dart main.dart
```

Von der Konsole aus startet man das Main-Programm mit dem Kommando „dart“

```
void main() {
    int aNumber = 4;
    double otherNumber = 4.0;
    bool isTrue = true;
    String hallo = '''
Multiline
String
''';
    Symbol sym = #metoo;
    Function foo = (double x) => x*x;
    Function bar = (x) {
        assert(x is double);
        double val = x*x;
        return val;
    };
    var list1 = [1,2,3, 'Foo'];
    print(list1.runtimeType);
    List<double> list2 = [1.0, 2.0, 3.0];
    Map map1 = {
        'name': 'Hans',
        'alter': 33,
        'square': (x) => x*x
    };
    print(map1["square"](3.0));
    print(map1.runtimeType);
}

typedef MyFunction = int Function(int);
MyFunction foo = (int x) {
    return x*x;
}
```

Nur zwei Zahlentypen: int und double!

Multiline Strings mit drei '''

Datentyp Symbol als Identifier

Lambda-Funktionen mit Fat Arrow => (Fart) oder Rumpf in Klammern {}

Liste: Dynamisches Array

Map: Dictionary Datentyp. Nicht wie Javascript Object!  
(Kann nicht mit this.alter in einer Funktion auf sich selbst zugreifen)

Typedef zur Spezifikation eigenen Typen in Form von von Funktions-Signaturen

- Beispiel ähnlich wie Java, C#

```
class Person {
```

Klasse mit Feldern, einem Konstruktor und einer Methode...

```
    String vname;  
    String nname;  
    int alter;
```

```
    Person(String vname, String nname, int alter) {  
        this.vname = vname;  
        this.nname = nname;  
        this.alter = alter;  
    }
```

```
    void info() {  
        print("Person "+this.vname+" "+ this.nname+" ist "+(this.alter.toString())+" Jahre alt.");  
    }
```

```
}
```

Person.dart

```
import 'Person.dart';
```

Import Instantiierung und  
Methodenaufruf...

```
void main() {  
    Person p = new Person("Hans", "Meier", 33);  
    p.info();  
}
```

main.dart

- Kompakter

```
class Person {
```

```
    String _vname;  
    String _nname;  
    int _alter;
```

Es gibt kein public und private. \_ vor Variable = Package local;  
(im gleichen File z.B).

```
    Person(this._vname, this._nname, this._alter);
```

Man muss die Variablen nicht  
mehr umkopieren im Konstruktor

```
    void info() {  
        print("Person $_vname $_nname ist $_alter Jahre alt.");  
    }
```

Variablen in Strings mit \$ oder \${}

```
}
```

Coding Style: DON'T type annotate  
initializing formals.

Person.dart

```
import 'Person.dart';
```

```
void main() {  
    var p = Person("Hans", "Meier", 33);  
    p.info();  
}
```

main.dart

- Fat Arrow (Fart) definiert Funktionen (s. Lambda)

```
class Person {
```

```
    String _vname;  
    String _nname;  
    int _alter;
```

Optionale Parameter mit Default-  
Werten. Kein Überladen nötig!

```
    Person(this._nname, [this._vname = 'Unbekannt', this._alter = 18]);
```

```
    get info => "Person $_vname $_nname ist $_alter Jahre alt.";
```

Getter und Setter.

```
    printInfo() => print(info);
```

Fat Arrow Syntax für Funktionen (Farts)

```
}
```

Person.dart

```
import 'Person.dart';
```

```
void main() {  
    Person p = Person("Meier");  
    print(p.info);  
    p.printInfo(); // gleiches Ergebnis
```

```
}
```

main.dart

```
import 'package:meta/meta.dart';  
class Person {
```

Definiert Annotationen wie @required

```
  final String vname;  
  final String nname;  
  final int alter;
```

Immutable (unveränderliche) Klasse: Felder sind final, Konstruktor const

Optionale benannte Parameter kommen in geschweifte Klammern

```
  const Person(this.nname, {this.vname: 'Unbekannt', @required this.alter});
```

```
  get info => "Person $vname $nname ist $alter Jahre!!! alt."  
  printInfo() => print(info);
```

Annotation @required kann ausgewertet werden mit Dartanalyzer

```
}
```

```
> dartanalyzer hello.dart
```

## Person.dart

Man kann Annotationen selbst definieren

```
name: Persons  
description: A Test  
  
dependencies:  
  meta: ^1.1.6}
```

```
import 'Person.dart';  
void main() {  
  Person p = Person("Meier",  
    vname: 'Hans', alter: 23);  
  p.printInfo();  
}
```

```
const required =  
  const Required("No Arg");  
class Required {  
  final String _msg;  
  const Required([this._msg]);  
} //So ähnlich...
```

pubspec.yaml

main.dart

meta.dart

```
> pub get
```

Download von Paketen mit pub get

- Mehrfachvererbung mit Mixins

```
class Person {
  final String vname;
  final String nname;
  final int alter;

  const Person(this.nname, this.vname, int this.alter);

  get info => "Person $vname $nname ist $alter Jahre!!! alt.";
  printInfo() => print(info);
}
abstract class Learner {
  void learn() => print("thinking...");
}
class Student extends Person with Learner{
  int matrNr;
  Student (nname, vname, alter, this.matrNr)
    : assert(nname != null), assert(nname != ""),
      super(nname, vname, alter);
}
```

Bug: Keine optionalen Parameter bei Verwendung von Mixins!  
<https://github.com/dart-lang/sdk/issues/31767>

Es gibt auch abstrakte Klassen

Elternklasse: Person, es kann Superkonstruktor aufgerufen werden

: assert(...), ist optional

## Person.dart

```
import 'Person.dart';
void main() {
  Student s = Student("Meier", 'Hans', 23, 123);
  s.printInfo();
  s.learn();
  print (s is Learner);
}
```

Learner: ebenfalls Elternklasse, Mixin

true

## main.dart



- Asynchrone Operationen

```
import 'dart:async';
```

```
import 'dart:io';
```

```
Future<String> getPath(String name) async {
```

Future bietet ein `.then(..)`

```
  final file = new File("$name");
```

```
  await file.exists();
```

Es wird auf etwas gewartet, Zeile danach erst ausgeführt.

```
  return file.path;
```

```
}
```

```
void foo() {
```

```
  getPath('Foo.txt').then((String path) {
```

```
    print(path);
```

```
  });
```

```
  print("Davor oder danach");
```

Wird gleich ausgeführt, `print(path)` evtl. erst danach

```
}
```

## Quellen

Dart Homepage: <https://www.dartlang.org/>

Sample Code: <https://www.dartlang.org/samples>

For Java Developers:

<https://codelabs.developers.google.com/codelabs/from-java-to-dart/#0>

Test IDE (Browser): <https://dartpad.dartlang.org/>

DartDoc: <https://github.com/dart-lang/dartdoc>

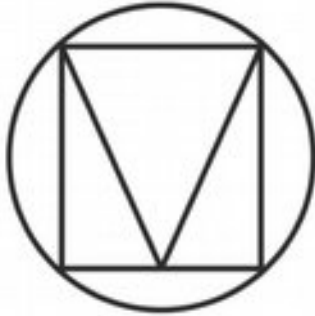
Generics: <https://www.dartlang.org/guides/language/language-tour#generics>

Generatoren: <https://www.dartlang.org/guides/language/language-tour#generators>

Streams:

<https://medium.com/@takahiom/how-to-write-flutter-asynchronous-processing-22f845204f30>

Coding Style: <https://www.dartlang.org/guides/language/effective-dart/style>



## Google Material Design

- Karten-Metapher
- Dicke
- Höhe
- Schatten
- Animation



Skeuomorphismus



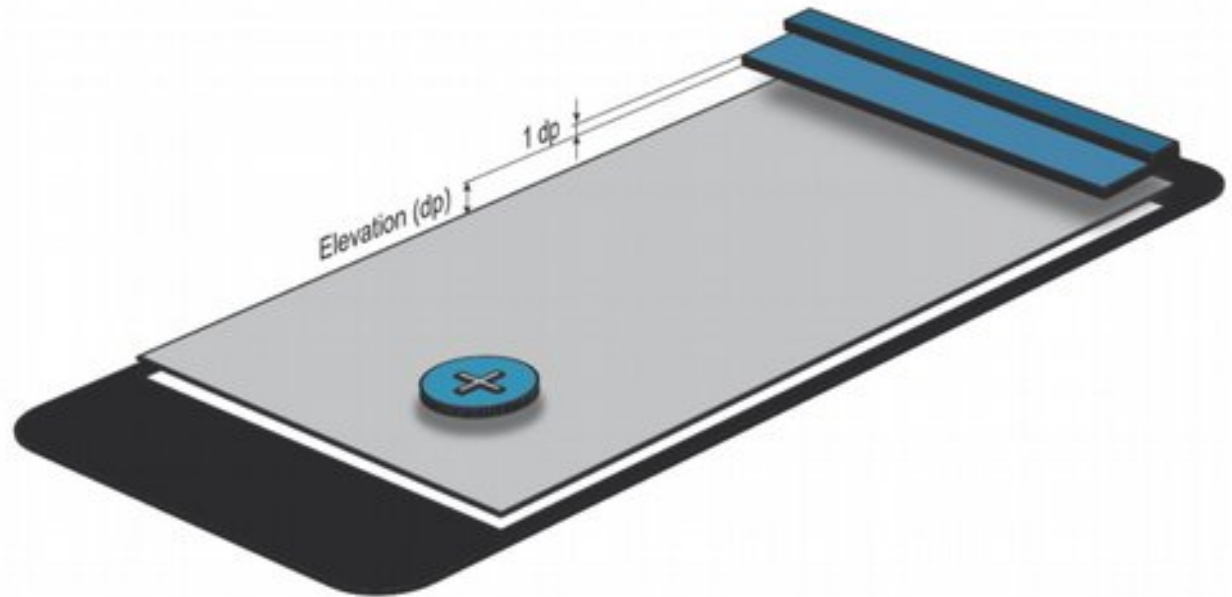
Flat-Design



Material Design



## Metapher





## Flutter?

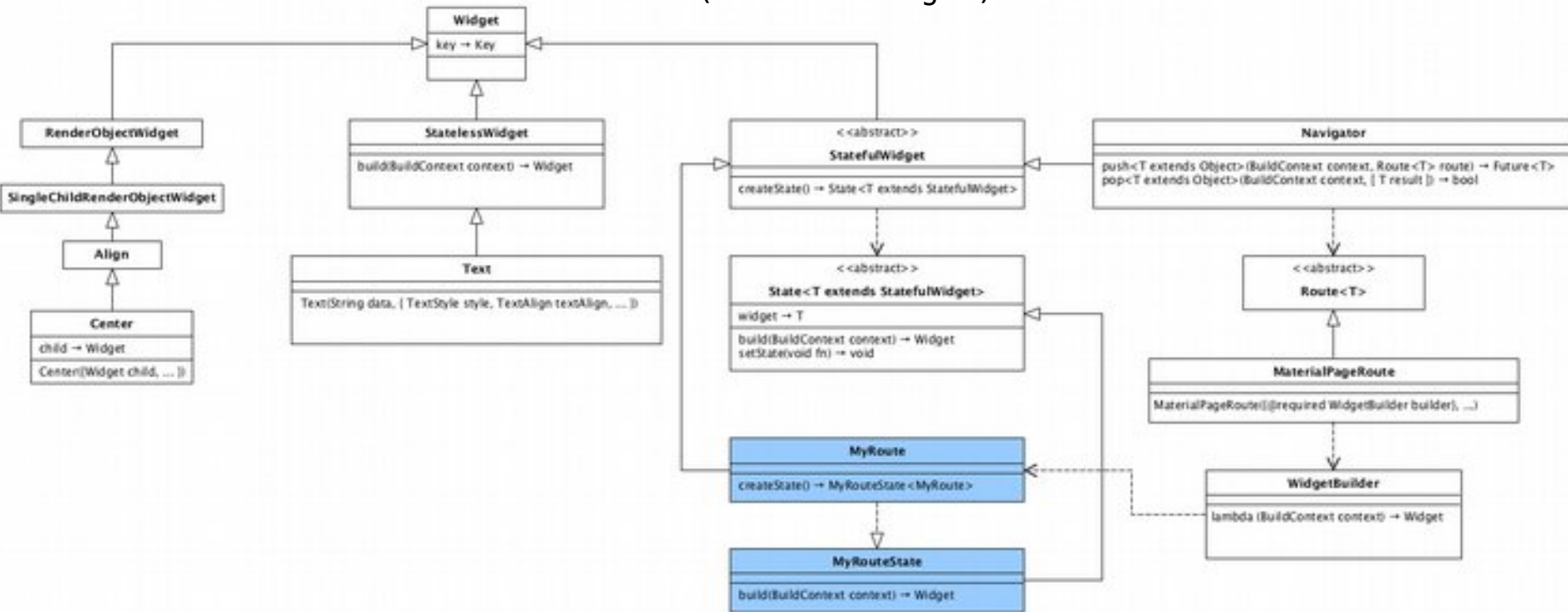
- UI-Toolkit für experimentelles Betriebssystem Fuchsia
- Anwendungs-Programmierung mit Flutter
- Plattform-Übergreifend für Android und IOS
- Performant, da eigene Render-Engine, kompiliert zu ARM-Code
- Hübsch (Google Material Design)
- Zukunft: Flutter für Web
- Zukunft: Flutter für Desktop
- Zukunft: Embedded Systeme
- Modern (Reactive) und so wie man es sich wünscht...

# Installation

- **Flutter für Android Studio:**
- (siehe auch: <https://flutter.io/get-started/editor/>)
- Android Studio installieren und öffnen. (<https://developer.android.com/studio/>)
- File>Settings>Plugins
- Browse repositories..., suche Flutter, Clique install.
- (Yes bei „Dart“ Plugin; bestätigen bei der Installation)
- Dann muss man noch den Pfad zum fehlenden Flutter SDK angeben.
- Also: Runterladen: <https://flutter.io/get-started/install/>
- und in einen Ordner entpacken, users/stud/...
- (Das SDK ist der größte Brocken)
- Nach der Installation neues Flutter Projekt anlegen
- Dort von Flutter den Pfad eintragen, erst wenn dieser auf den Ordner „flutter“ im entpackten Ordner verweist, kann man „weiter“ klicken.
- Projekt nicht löschen, sonst Einstellungen wieder weg.

# Klassen: Alles Widgets

Widget: Dingsbums für Bildschirm  
(Windows-Gadget)



## Layouts in Dart

- Keine weitere Sprache wie XML
- Kein Layout-Builder
- Denn: Dart ist deklarativ

```
Padding(  
  padding:  
  EdgeInsets.all(8.0),),  
  child:  
  Center(child:  
    Text("Hallo"),  
  )  
)  
,
```

```
<div style='padding: 8px'>  
  <div style='text-align: center'>  
    <span>Hallo</span>  
  </div>  
</div>
```



## Eine App: World Cup



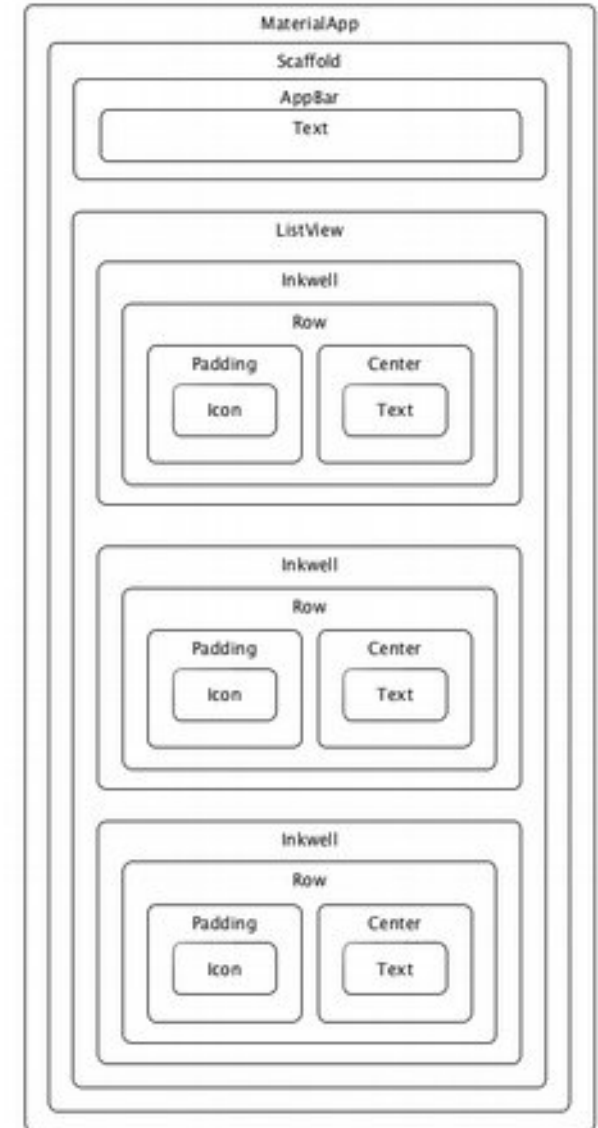
CC0 1.0

```
import 'package:flutter/material.dart';

void main() => runApp(new MyApp());

class MyApp extends StatelessWidget { Statisch, ohne State

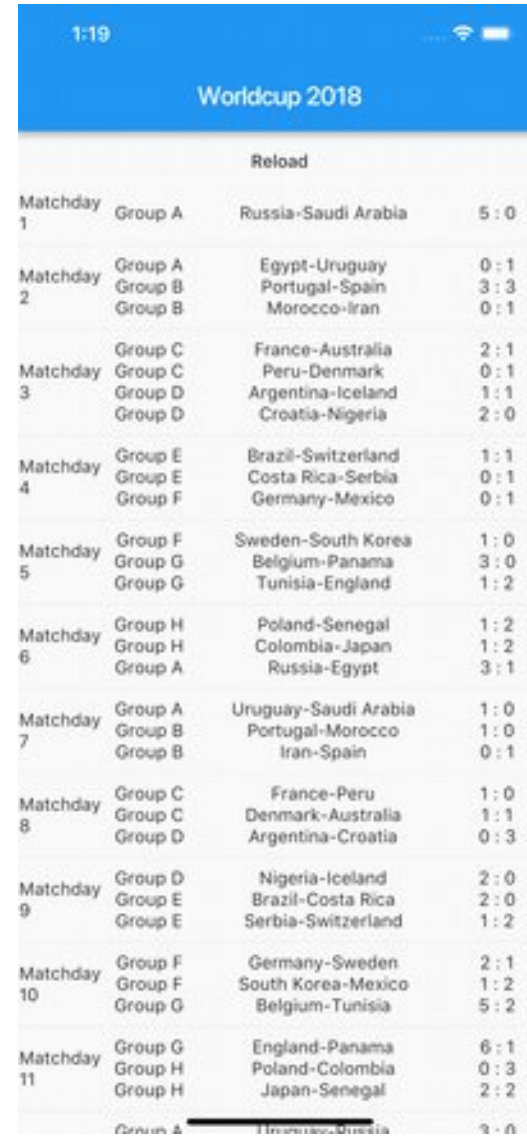
  @override
  Widget build(BuildContext context) {
    List<Widget> elems = <Widget>[];
    Color color = Colors.green;
    for (int i = 0; i < 30; i++) {
      var el = InkWell(
        highlightColor: color,
        splashColor: color,
        onTap: () => print("Hallo"),
        borderRadius: BorderRadius.circular(10.0),
        child: Row(children: <Widget>[
          Padding(child:
            Icon(Icons.cake), padding: EdgeInsets.all(8.0)),
          Center(child:
            Text("Hallo $i"), )
        ]),
      );
      elems.add(el);
    }
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'List',
      home: Scaffold(
        appBar: AppBar(title: Text('List4')),
        body: ListView(children: elems));
    }
  }
}
```



## Dynamische Layouts

```
import 'dart:convert';  
import 'package:flutter/material.dart';  
import 'package:http/http.dart' as http;  
  
void main() => runApp(new FootballApp());  
  
String uri =  
  "https://raw.githubusercontent.com/openfootball/world-  
cup.json/master/2018/worldcup.json";
```

Web-Service

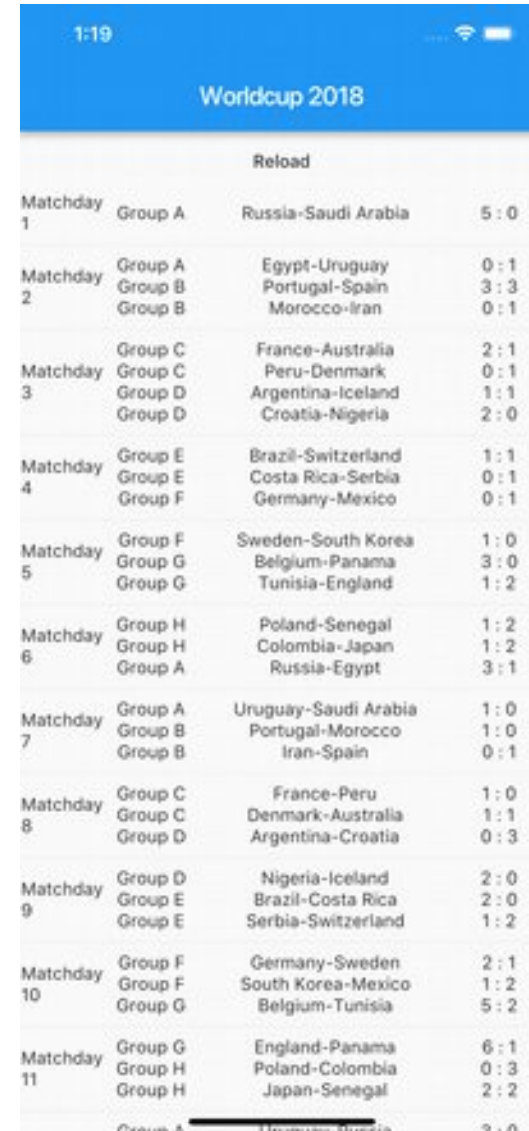


1:19 Worldcup 2018

Reload

Matchday 1	Group A	Russia-Saudi Arabia	5 : 0
Matchday 2	Group A	Egypt-Uruguay	0 : 1
	Group B	Portugal-Spain	3 : 3
	Group B	Morocco-Iran	0 : 1
Matchday 3	Group C	France-Australia	2 : 1
	Group D	Peru-Denmark	0 : 1
	Group D	Argentina-Iceland	1 : 1
Matchday 4	Group D	Croatia-Nigeria	2 : 0
	Group E	Brazil-Switzerland	1 : 1
	Group E	Costa Rica-Serbia	0 : 1
Matchday 5	Group F	Germany-Mexico	0 : 1
	Group G	Sweden-South Korea	1 : 0
	Group G	Belgium-Panama	3 : 0
Matchday 6	Group G	Tunisia-England	1 : 2
	Group H	Poland-Senegal	1 : 2
	Group H	Colombia-Japan	1 : 2
Matchday 7	Group A	Russia-Egypt	3 : 1
	Group A	Uruguay-Saudi Arabia	1 : 0
	Group B	Portugal-Morocco	1 : 0
Matchday 8	Group B	Iran-Spain	0 : 1
	Group C	France-Peru	1 : 0
	Group C	Denmark-Australia	1 : 1
Matchday 9	Group D	Argentina-Croatia	0 : 3
	Group D	Nigeria-Iceland	2 : 0
	Group E	Brazil-Costa Rica	2 : 0
Matchday 10	Group E	Serbia-Switzerland	1 : 2
	Group F	Germany-Sweden	2 : 1
	Group F	South Korea-Mexico	1 : 2
Matchday 11	Group G	Belgium-Tunisia	5 : 2
	Group G	England-Panama	6 : 1
	Group H	Poland-Colombia	0 : 3
Matchday 11	Group H	Japan-Senegal	2 : 2
	Group A	Tunisia-Croatia	3 : 0

```
class FootballApp extends StatefulWidget {  
  @override  
  State<StatefulWidget> createState() {  
    var fbs = FootballAppState();  
    return fbs;  
  }  
}  
  
class FootballAppState extends State<FootballApp> {  
  
  List<Widget> _gamesList = <Widget>[]; Teil des State!  
  ...  
  void _getFootballResults() async { Liste dynamisch aufbauen  
    setState(() {  
      _gamesList = ... State ändern!  
    });  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp( ... );  
  }  
  
  @override  
  void initState() { Start nach Initialisierung  
    super.initState();  
    _getFootballResults();  
  }  
}
```



Worldcup 2018			
Reload			
Matchday 1	Group A	Russia-Saudi Arabia	5 : 0
Matchday 2	Group A	Egypt-Uruguay	0 : 1
	Group B	Portugal-Spain	3 : 3
	Group B	Morocco-Iran	0 : 1
Matchday 3	Group C	France-Australia	2 : 1
	Group C	Peru-Denmark	0 : 1
	Group D	Argentina-Iceland	1 : 1
	Group D	Croatia-Nigeria	2 : 0
Matchday 4	Group E	Brazil-Switzerland	1 : 1
	Group E	Costa Rica-Serbia	0 : 1
	Group F	Germany-Mexico	0 : 1
Matchday 5	Group F	Sweden-South Korea	1 : 0
	Group G	Belgium-Panama	3 : 0
Matchday 6	Group G	Tunisia-England	1 : 2
	Group H	Poland-Senegal	1 : 2
	Group H	Colombia-Japan	1 : 2
Matchday 7	Group A	Russia-Egypt	3 : 1
	Group A	Uruguay-Saudi Arabia	1 : 0
	Group B	Portugal-Morocco	1 : 0
Matchday 8	Group B	Iran-Spain	0 : 1
	Group C	France-Peru	1 : 0
	Group C	Denmark-Australia	1 : 1
Matchday 9	Group D	Argentina-Croatia	0 : 3
	Group D	Nigeria-Iceland	2 : 0
	Group E	Brazil-Costa Rica	2 : 0
Matchday 10	Group E	Serbia-Switzerland	1 : 2
	Group F	Germany-Sweden	2 : 1
	Group F	South Korea-Mexico	1 : 2
Matchday 11	Group G	Belgium-Tunisia	5 : 2
	Group G	England-Panama	6 : 1
	Group H	Poland-Colombia	0 : 3
	Group H	Japan-Senegal	2 : 2

```
class FootballAppState extends State<FootballApp> {
```

```
...
```

```
@override
```

```
Widget build(BuildContext context) {
```

```
  return MaterialApp(
```

```
    debugShowCheckedModeBanner: false,
```

```
    home: Scaffold(
```

Scaffold

```
      appBar: AppBar(
```

```
        title: Text("Worldcup 2018"),
```

```
      ),
```

```
      body: Center(
```

```
        child: Column(children: <Widget>[
```

```
          MaterialButton(
```

```
            child: Text("Reload"),
```

```
            onPressed: _getFootballResults, Reload Button
```

```
          ),
```

```
          Expanded(
```

```
            child: ListView(children: _gamesList),
```

```
          ),
```

```
        ]),
```

```
      ),
```

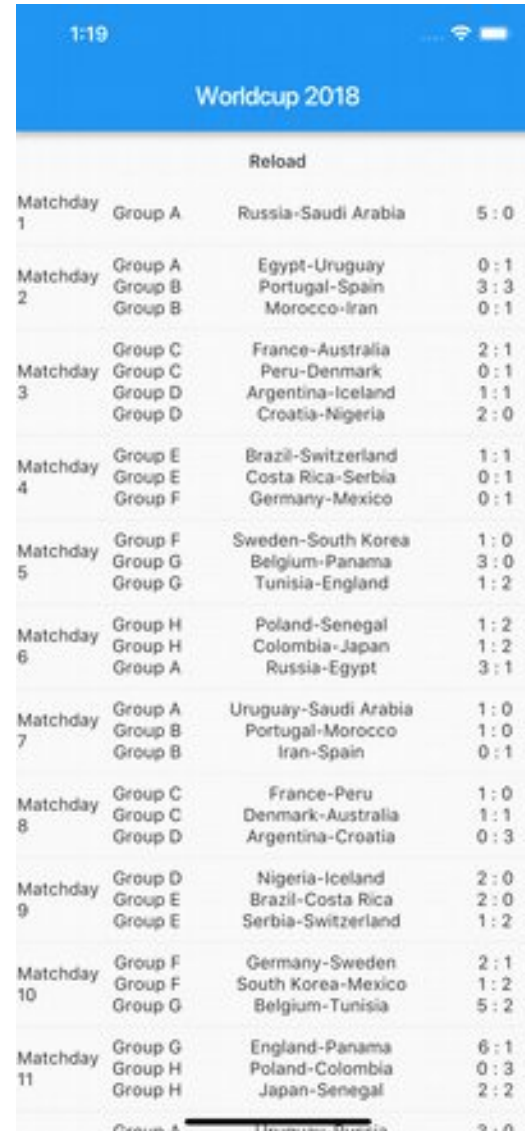
```
    );
```

```
  }
```

```
}
```

```
...
```

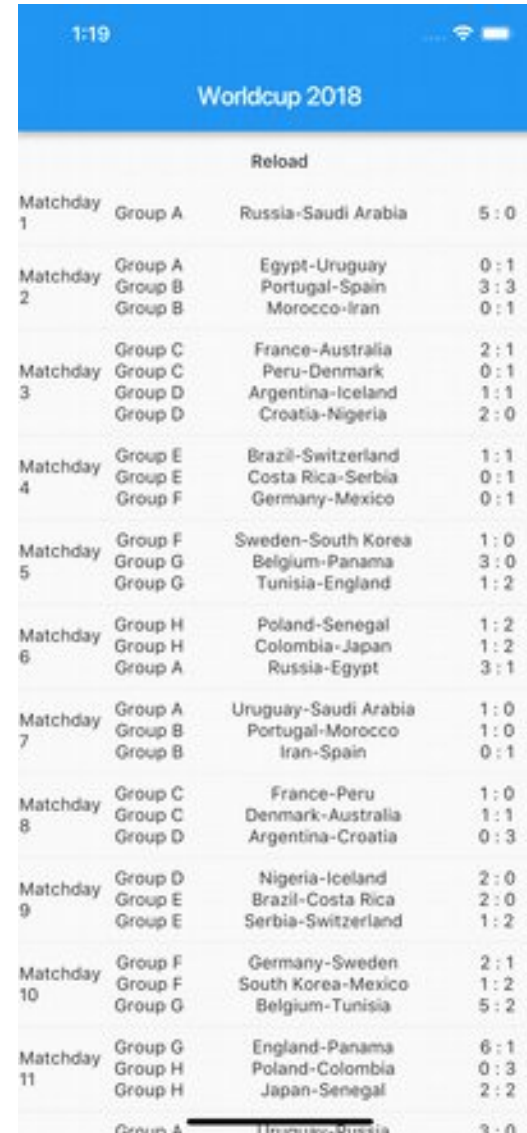
```
}
```



Worldcup 2018			
Reload			
Matchday 1	Group A	Russia-Saudi Arabia	5 : 0
Matchday 2	Group A	Egypt-Uruguay	0 : 1
	Group B	Portugal-Spain	3 : 3
Matchday 3	Group B	Morocco-Iran	0 : 1
	Group C	France-Australia	2 : 1
Matchday 4	Group C	Peru-Denmark	0 : 1
	Group D	Argentina-Iceland	1 : 1
Matchday 5	Group D	Croatia-Nigeria	2 : 0
	Group E	Brazil-Switzerland	1 : 1
Matchday 6	Group E	Costa Rica-Serbia	0 : 1
	Group F	Germany-Mexico	0 : 1
Matchday 7	Group F	Sweden-South Korea	1 : 0
	Group G	Belgium-Panama	3 : 0
Matchday 8	Group G	Tunisia-England	1 : 2
	Group H	Poland-Senegal	1 : 2
Matchday 9	Group H	Colombia-Japan	1 : 2
	Group A	Russia-Egypt	3 : 1
Matchday 10	Group A	Uruguay-Saudi Arabia	1 : 0
	Group B	Portugal-Morocco	1 : 0
Matchday 11	Group B	Iran-Spain	0 : 1
	Group C	France-Peru	1 : 0
Matchday 12	Group C	Denmark-Australia	1 : 1
	Group D	Argentina-Croatia	0 : 3
Matchday 13	Group D	Nigeria-Iceland	2 : 0
	Group E	Brazil-Costa Rica	2 : 0
Matchday 14	Group E	Serbia-Switzerland	1 : 2
	Group F	Germany-Sweden	2 : 1
Matchday 15	Group F	South Korea-Mexico	1 : 2
	Group G	Belgium-Tunisia	5 : 2
Matchday 16	Group G	England-Panama	6 : 1
	Group H	Poland-Colombia	0 : 3
Matchday 17	Group H	Japan-Senegal	2 : 2
	Group A	Russia-Saudi Arabia	3 : 0

```
List<Widget> _gamesList = <Widget>[];
```

```
void _getFootballResults() async { Async-Await
  var httpClient = http.Client();
  var response = await httpClient.read(uri);
  setState(() {
    _gamesList = <Widget>[];
  });
  Map<String, dynamic> wc = json.decode(response);
  print(wc);
  for (var r in wc['rounds']) { JSON-Decode
    print(r['name']);
    var day = <String>[];
    var teams = <String>[];
    var scores = <String>[];
    for (var m in r['matches']) {
      day.add(m['group'] ?? ' ');
      teams.add("${m['team1']['name']}-${m['team2']
['name']}");
      var score = m['score1'] == null || m['score2'] ==
null ? '? : ?' :
        "${m['score1']} : ${m['score2']}";
      scores.add(score);
    }
    try {
      _addResult(r['name'], day, teams, scores);
    } finally {
      //Nothing
    }
  }
}
```



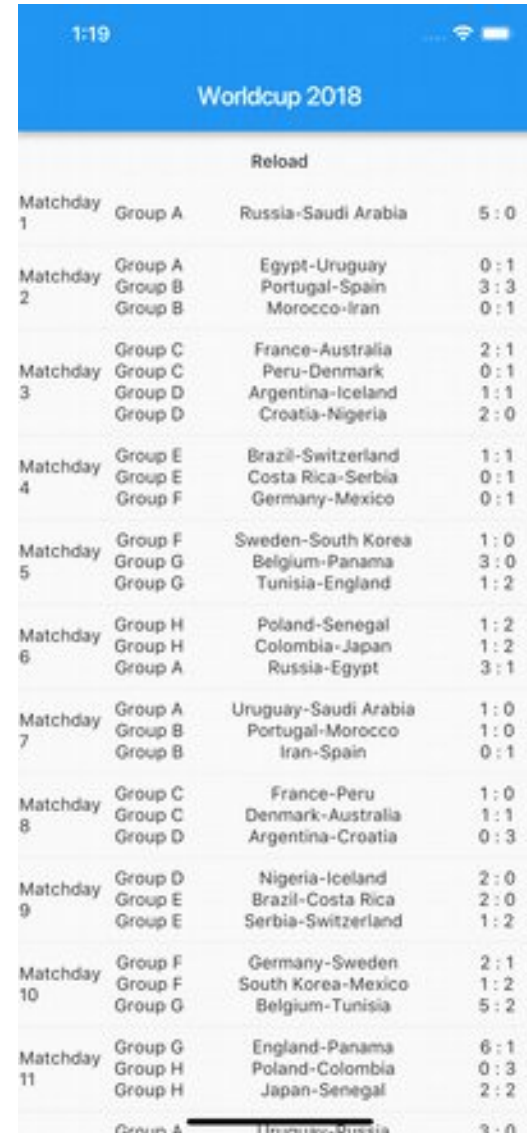
1:19

Worldcup 2018

Reload

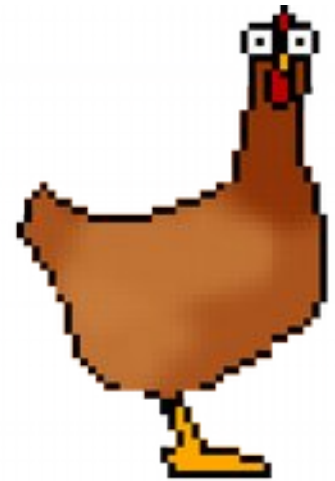
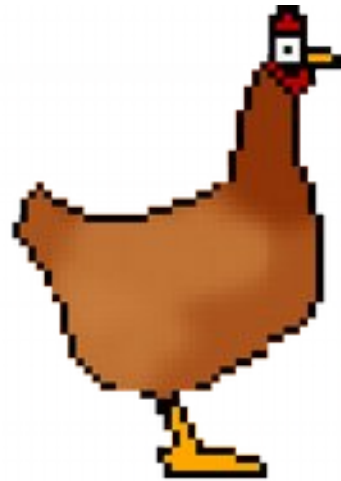
Matchday	Group	Match	Score
1	Group A	Russia-Saudi Arabia	5 : 0
2	Group A	Egypt-Uruguay	0 : 1
	Group B	Portugal-Spain	3 : 3
	Group B	Morocco-Iran	0 : 1
3	Group C	France-Australia	2 : 1
	Group C	Peru-Denmark	0 : 1
	Group D	Argentina-Iceland	1 : 1
	Group D	Croatia-Nigeria	2 : 0
4	Group E	Brazil-Switzerland	1 : 1
	Group E	Costa Rica-Serbia	0 : 1
	Group F	Germany-Mexico	0 : 1
5	Group F	Sweden-South Korea	1 : 0
	Group G	Belgium-Panama	3 : 0
	Group G	Tunisia-England	1 : 2
6	Group H	Poland-Senegal	1 : 2
	Group H	Colombia-Japan	1 : 2
	Group A	Russia-Egypt	3 : 1
7	Group A	Uruguay-Saudi Arabia	1 : 0
	Group B	Portugal-Morocco	1 : 0
	Group B	Iran-Spain	0 : 1
8	Group C	France-Peru	1 : 0
	Group C	Denmark-Australia	1 : 1
	Group D	Argentina-Croatia	0 : 3
9	Group D	Nigeria-Iceland	2 : 0
	Group E	Brazil-Costa Rica	2 : 0
	Group E	Serbia-Switzerland	1 : 2
10	Group F	Germany-Sweden	2 : 1
	Group F	South Korea-Mexico	1 : 2
	Group G	Belgium-Tunisia	5 : 2
11	Group G	England-Panama	6 : 1
	Group H	Poland-Colombia	0 : 3
	Group H	Japan-Senegal	2 : 2

```
void _addResult(String day, List<String> groups,  
                List<String> teams, List<String> scores) {  
    List<Widget> groupList = <Widget>[];  
    for (String s in groups) {  
        groupList.add(Text(s));  
    }  
    List<Widget> teamList = <Widget>[];  
    for (String s in teams) {  
        teamList.add(Text(s));  
    }  
    ...//ScoreList  
    Column c = Column( children: <Widget>[ Row(  
        mainAxisAlignment: MainAxisAlignment.spaceBetween,  
        children: <Widget>[  
            Expanded( child: Column(  
                children: <Widget>[Text(day)],  
            ), flex: 1,  
            ),  
            Expanded( child: Column(  
                children: groupList,  
            ), flex: 1,  
            ),  
            Expanded(child: Column(  
                children: teamList,  
            ), flex: 3,  
            ),  
        ...],  
    ),  
    Divider(height: 16.0, ),  
]);  
    setState(() {  
        _gamesList.add(c);  
    });  
}
```



Worldcup 2018			
Reload			
Matchday 1	Group A	Russia-Saudi Arabia	5 : 0
Matchday 2	Group A	Egypt-Uruguay	0 : 1
	Group B	Portugal-Spain	3 : 3
Matchday 3	Group B	Morocco-Iran	0 : 1
	Group C	France-Australia	2 : 1
Matchday 4	Group C	Peru-Denmark	0 : 1
	Group D	Argentina-Iceland	1 : 1
Matchday 5	Group D	Croatia-Nigeria	2 : 0
	Group E	Brazil-Switzerland	1 : 1
Matchday 6	Group E	Costa Rica-Serbia	0 : 1
	Group F	Germany-Mexico	0 : 1
Matchday 7	Group F	Sweden-South Korea	1 : 0
	Group G	Belgium-Panama	3 : 0
Matchday 8	Group G	Tunisia-England	1 : 2
	Group H	Poland-Senegal	1 : 2
Matchday 9	Group H	Colombia-Japan	1 : 2
	Group A	Russia-Egypt	3 : 1
Matchday 10	Group A	Uruguay-Saudi Arabia	1 : 0
	Group B	Portugal-Morocco	1 : 0
Matchday 11	Group B	Iran-Spain	0 : 1
	Group C	France-Peru	1 : 0
Matchday 12	Group C	Denmark-Australia	1 : 1
	Group D	Argentina-Croatia	0 : 3
Matchday 13	Group D	Nigeria-Iceland	2 : 0
	Group E	Brazil-Costa Rica	2 : 0
Matchday 14	Group E	Serbia-Switzerland	1 : 2
	Group F	Germany-Sweden	2 : 1
Matchday 15	Group F	South Korea-Mexico	1 : 2
	Group G	Belgium-Tunisia	5 : 2
Matchday 16	Group G	England-Panama	6 : 1
	Group H	Poland-Colombia	0 : 3
Matchday 17	Group H	Japan-Senegal	2 : 2
	Group A	Russia-Saudi Arabia	3 : 0

## Ein Spiel: Chicken Maze







## Flame

**dependencies:**

**flutter:**

**sdk:** flutter

...

**flame:** ^0.11.2 **Plugin**

**flutter:**

**assets:** **Grafik und Sound-Ressourcen**

- assets/images/chicken.png
- assets/images/enemy.png
- assets/images/tiles.png
- assets/images/grain.png
- assets/tiles/map1.tmx **Tilemap**
- assets/audio/music.mp3

- Animierte Sprites
- Tilemaps
- Parallaxe
- Physics
- Audio
- Input

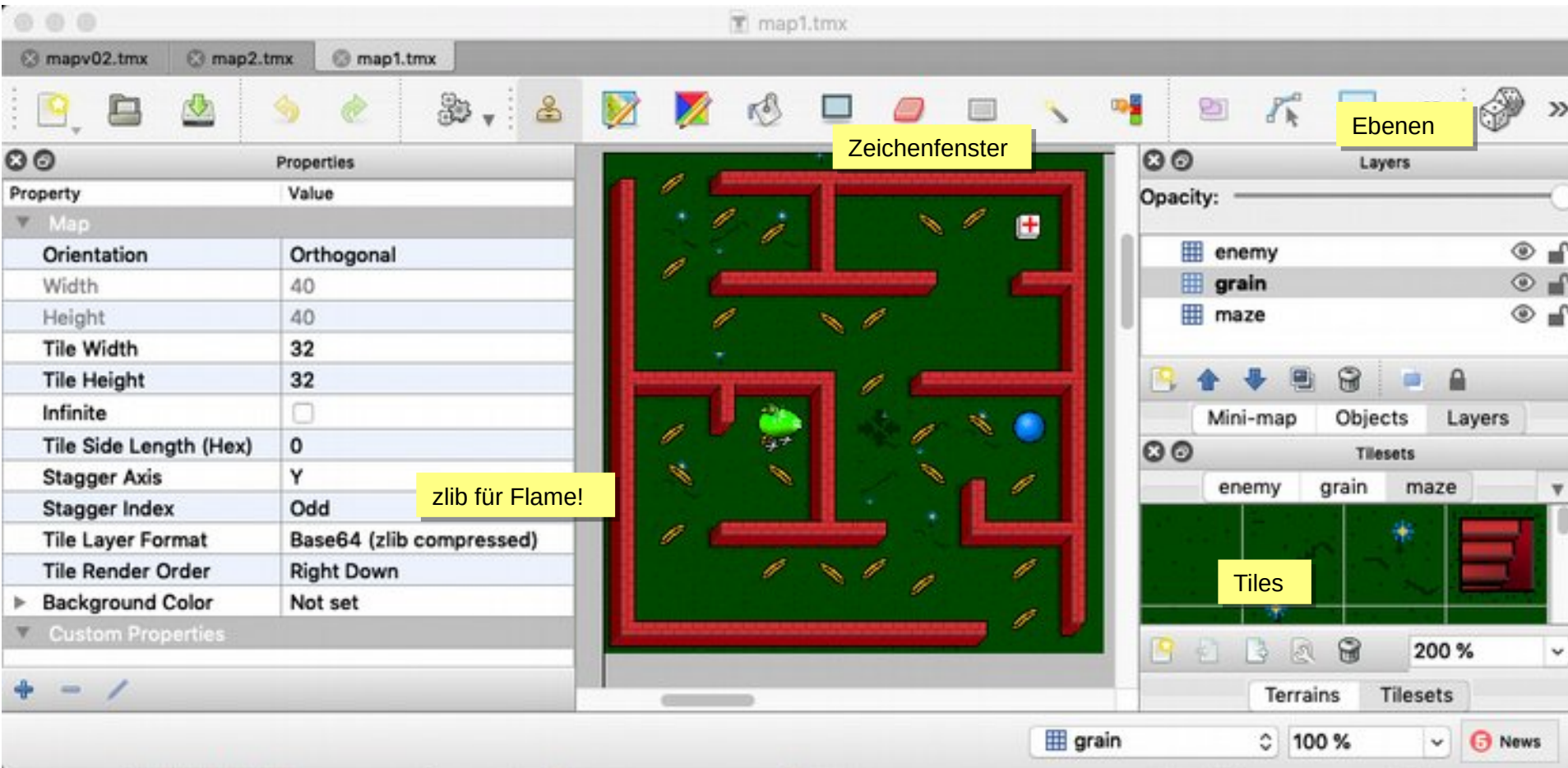
pubspec.yaml

```
> flutter packages get
```

Download von Flutter-Paketen



# Tilemaps



Tiled  
Editor

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<map version="1.2" tiledversion="1.2.2" orientation="orthogonal"
renderorder="right-down" width="40" height="40" tilewidth="32"
tileheight="32" infinite="0" nextlayerid="2" nextobjectid="1">
```

Globale Eigenschaften

```
<tileset firstgid="1" name="maze" tilewidth="32" tileheight="32"
tilecount="24" columns="4">
```

Tilesets

```
<image source="tiles.png" width="128" height="192"/>
</tileset>
```

```
<tileset firstgid="25" name="grain" tilewidth="32" tileheight="32"
tilecount="10" columns="10">
```

```
<image source="grain.png" width="320" height="32"/>
</tileset>
```

```
<tileset firstgid="35" name="enemy" tilewidth="32" tileheight="32"
tilecount="12" columns="4">
```

```
<image source="enemy.png" width="128" height="96"/>
</tileset>
```

Layers

```
<layer id="1" name="maze" width="10" height="10">
<data encoding="base64" compression="zlib">
```

```
eJyFkFEKwCAMQ53CdLDCQDfvf9Nl0EI0HX48IoWkqS0ltIEJKthVmQYEFJBVRX2M0NzyHtV08/
PHH+Udrldz+dXlXeTlvtyvuB1Z32NxR1fvHexd3WGeCPubr/sLbsoDFQ==
```

```
</data>
```

```
</layer>
```

```
<layer id="2" name="grain" width="10" height="10">
```

```
<data encoding="base64" compression="zlib">
```

```
eJxjYGBgkGYgHkiiqZeBiimgqSHFPBCQIkEflvdKo9GyUDbIjYpI6mTQaJj96GaTEi4gIMmA8Ic0k
hixekEAAESeAvE=
```

```
</data>
```

```
</layer>
```

```
<layer id="3" name="enemy" width="10" height="10">
```

```
<data encoding="base64" compression="zlib">
```

```
eJxjYBh+QG2gHUABAAadeAAAn
```

```
</data>
```

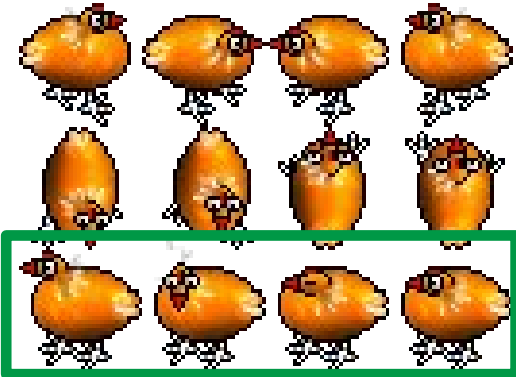
```
</layer>
```

```
</map>
```

Zlib (+Base64) Kompression  
der csv-Daten:  
20,1,23,7,6,7,7,7,7,8,13,3,2, ...

```
import 'package:tilde/tilde.dart';  
class Maze {  
  ...  
  TiledComponent tiles;  
  bool _initialized;  
  bool get initialized => _initialized; Nur lesen  
  
  Maze(...) {  
    tiles = new TiledComponent("map${game.level}.tmx", );  
    _initialized = false;  
    tiles.future.then((t) { Wenn Tiles geladen,  
dann...  
      _initialized = true;  
      tiles.map.layers[2].visible = false;  
    });  
    ...  
  }  
  void render(Canvas canvas) { Auf Canvas rendern  
    tiles.render(canvas);  
  }  
  ...  
}
```

```
class AssetLoader {  
static const chickenpath = "chicken.png";  
static animation.Animation get chickenAnimationIdle {  
    var c = animation.Animation.variableSequenced(chickenpath, 4,  
    [20,0.5,1,5],  
        textureX: 0.0, Dauer jedes Frames in Sekunden  
        textureY: raster * 2.0,  
        textureWidth: 0.0 + raster,  
        textureHeight: 0.0 + raster,  
    );  
    c.loop = true;  
    return c;  
}  
}
```



```
...  
void render(Canvas canvas) {  
    ...  
    currentAnimation= animationIdle;  
    currentAnimation.update(0.1); Update in Sekunden  
    currentAnimation.getSprite().renderPosition(canvas,  
        Position( 100.0, 100.0));  
    ...  
}
```

```
import 'package:flame/game.dart';
import 'package:chicken_maze/Chicken.dart';
import 'package:chicken_maze/Maze.dart';

class ChickenGame extends BaseGame {
  @override
  void render(Canvas canvas) {
    if (paused || !maze.initialized) return;
    canvas.scale(scaleFactor);
    canvas.translate(maze.bgrPos.x, maze.bgrPos.y);
    maze.render(canvas);
    chicken.render(canvas);
  }
  @override
  void update(double t) {
    assert(context != null);
    if (!paused && chicken.lives <= 0) {
      Navigator.of(context).pushReplacementNamed('ro
ute);
    }
  }
}
```

Vorher definierte Klassen

Basisklasse bietet Game-Funktionalitäten

Übergibt Canvas

Position updaten

Screen wechseln bei Game Over

# Einstieg

- Schnellüberblick: <https://flutter.io/get-started/codelab/>
- Udacity Kurs:  
<https://www.udacity.com/course/build-native-mobile-apps-with-flutter--ud905>
- Kochbuch (mit Standard-Problemen): <https://flutter.io/cookbook/>

## Quellen

Homepage Flutter: <https://flutter.io/>

Docs: <https://flutter.io/docs/>

Install: <https://flutter.io/get-started/install/>

API: <https://docs.flutter.io/>

Tutorial: Flutter Layouts Alignment:

<https://flutter.io/tutorials/layout/#alignment>

Tiled Editor: <https://www.mapeditor.org/>

Flame Github: <https://github.com/luanpotter/flame>