

# Vorarlberg the Game

Generated by Doxygen 1.14.0



1	README	1
1.0.0.1	Windows-spezifisch	1
1.0.1	4. Projektaufbau	1
1.0.2	5. Ordnerstruktur	2
1.0.3	6. Icons und Lizenz	2
2	Namespace Index	3
2.1	Namespace List	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Data Structure Index	7
4.1	Data Structures	7
5	File Index	9
5.1	File List	9
6	Namespace Documentation	11
6.1	Game Namespace Reference	11
7	Data Structure Documentation	13
7.1	Game.App Class Reference	13
7.1.1	Detailed Description	13
7.1.2	Constructor & Destructor Documentation	13
7.1.2.1	App()	13
7.2	Game.AppShell Class Reference	13
7.2.1	Detailed Description	14
7.2.2	Constructor & Destructor Documentation	14
7.2.2.1	AppShell()	14
7.3	Game.CardDeckPage Class Reference	14
7.3.1	Detailed Description	14
7.3.2	Constructor & Destructor Documentation	14
7.3.2.1	CardDeckPage()	14
7.4	Game.CursesPage Class Reference	14
7.4.1	Detailed Description	15
7.4.2	Constructor & Destructor Documentation	15
7.4.2.1	CursesPage()	15
7.5	Game.InfoPage Class Reference	15
7.5.1	Detailed Description	15
7.5.2	Constructor & Destructor Documentation	15
7.5.2.1	InfoPage()	15
7.6	Game.JoinSessionPage Class Reference	16
7.6.1	Detailed Description	16
7.6.2	Constructor & Destructor Documentation	16

---

7.6.2.1 JoinSessionPage()	16
7.7 Game.LiveMapPage Class Reference	16
7.7.1 Detailed Description	16
7.7.2 Constructor & Destructor Documentation	17
7.7.2.1 LiveMapPage()	17
7.8 Game.MainPage Class Reference	17
7.8.1 Detailed Description	17
7.8.2 Constructor & Destructor Documentation	17
7.8.2.1 MainPage()	17
7.9 Game.QuestionsPage.QuestionAnswer Class Reference	17
7.9.1 Detailed Description	17
7.10 Game.QuestionsPage Class Reference	18
7.10.1 Detailed Description	18
7.10.2 Constructor & Destructor Documentation	18
7.10.2.1 QuestionsPage()	18
7.11 Game.RoomPage Class Reference	18
7.11.1 Detailed Description	19
7.11.2 Constructor & Destructor Documentation	19
7.11.2.1 RoomPage()	19
7.11.3 Property Documentation	19
7.11.3.1 Players	19
7.12 Game.SuchePage Class Reference	19
7.12.1 Detailed Description	19
7.12.2 Constructor & Destructor Documentation	19
7.12.2.1 SuchePage()	19
7.13 Game.WuerfelPage Class Reference	20
7.13.1 Detailed Description	20
7.13.2 Constructor & Destructor Documentation	20
7.13.2.1 WuerfelPage()	20
8 File Documentation	21
8.1 App.xaml.cs File Reference	21
8.1.1 Detailed Description	21

# Chapter 1

## README

```
# MAUI-Demo: WürfelPage & BushalteSuchen
```

```
---
```

```
## 1. Übersicht
```

Dieses kleine MAUI-Projekt enthält zwei einfache Seiten:

1. **WürfelPage**
  - Simuliert das Werfen eines sechsseitigen Würfels.
2. **BushalteSuchen**
  - Eine Demonstrationsseite zur (fiktiven) Suche nach Bushaltestellen.

Beide Komponenten sind bewusst minimal gehalten, um die Grundlagen von MVVM und SkiaSharp in MAUI kennenzulernen.

```
---
```

```
## 2. Voraussetzungen
```

- [.NET 8 SDK](<https://dotnet.microsoft.com/>)
- MAUI-Workload installiert ('dotnet workload install maui')

```
---
```

```
## 3. Abhängigkeiten
```

Führe im Projektordner folgende Befehle aus, um die benötigten NuGet-Pakete hinzuzufügen:

```
```bash
dotnet add package CommunityToolkit.Mvvm --version 8.4.0
dotnet add package CommunityToolkit.Maui --version 8.0.1
dotnet add package SkiaSharp.Views.Maui.Controls --version 3.119.0
```

### 1.0.0.1 Windows-spezifisch

Um Versionskonflikte mit dem Windows SDK zu vermeiden, ergänze in Deiner .csproj:

```
<PropertyGroup>
  <WindowsSdkPackageVersion>10.0.19041.53</WindowsSdkPackageVersion>
</PropertyGroup>
```

---

### 1.0.1 4. Projektaufbau

#### 1. Restore

```
dotnet restore
```

#### 2. Build

```
dotnet build -p:WindowsSdkPackageVersion=10.0.19041.53
```

Unter Windows genügt in der Regel auch:

```
dotnet build
```

Nach erfolgreichem Build liegen die Assemblies im Ordner bin/Debug.

---

## 1.0.2 5. Ordnerstruktur

```
/Game
|- App.xaml
|   |- App.xaml.cs
MainPage.xaml
|   |- MainPage.xaml.cs
|- Styles.xaml
|- WürfelPage.xaml
|   |- WuerfelPage.xaml.cs
|- BushalteSuchenPage.xaml
|   |- BushalteSuchenPage.xaml.cs
|- ViewModels
|   |- WürfelViewModel.cs
|   |- BushalteViewModel.cs
|- Resources
|   |- Images
|       |- dice1.png
|       |- dice2.png
|       |- dice3.png
|       |- dice4.png
|       |- dice5.png
|       |- dice6.png
```

---

## 1.0.3 6. Icons und Lizenz

Die Würfel-Grafiken stammen von [game-icons.net](https://game-icons.net) und stehen unter der Lizenz CC BY 3.0. Bitte achte auf die Namensnennung, wenn Du die Bilder wiederverwendest.

- Quelle: <https://game-icons.net/tags/dice.html>
- Lizenz: Creative Commons Attribution 3.0 Unported (CC BY 3.0) – Erlaubt: Teilen und Bearbeiten, auch kommerziell – Bedingung: Namensnennung des Urhebers

Im Projekt enthaltene Dateien:

```
Resources/Images/dice1.png
Resources/Images/dice2.png
Resources/Images/dice3.png
Resources/Images/dice4.png
Resources/Images/dice5.png
Resources/Images/dice6.png
```

## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

[Game](#) . . . . . 11





# Chapter 3

## Hierarchical Index

### 3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Application	
Game.App . . . . .	13
ContentPage	
Game.CardDeckPage . . . . .	14
Game.CursesPage . . . . .	14
Game.InfoPage . . . . .	15
Game.JoinSessionPage . . . . .	16
Game.LiveMapPage . . . . .	16
Game.MainPage . . . . .	17
Game.QuestionsPage . . . . .	18
Game.RoomPage . . . . .	18
Game.SuchePage . . . . .	19
Game.WuerfelPage . . . . .	20
Game.QuestionsPage.QuestionAnswer . . . . .	17
Shell	
Game.AppShell . . . . .	13



# Chapter 4

## Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">Game.App</a>	The main application class that inherits from MAUI Application . . . . .	13
<a href="#">Game.AppShell</a>	The main shell class that provides navigation framework for the application . . . . .	13
<a href="#">Game.CardDeckPage</a>	Page class for managing the card deck interface . . . . .	14
<a href="#">Game.CursesPage</a>	Page for displaying active curses . . . . .	14
<a href="#">Game.InfoPage</a>	Page class for displaying game information . . . . .	15
<a href="#">Game.JoinSessionPage</a>	Page class for handling session joining functionality . . . . .	16
<a href="#">Game.LiveMapPage</a>	Page class for displaying and managing the live map functionality . . . . .	16
<a href="#">Game.MainPage</a>	The main page of the application . . . . .	17
<a href="#">Game.QuestionsPage.QuestionAnswer</a>	Model class for question-answer pairs . . . . .	17
<a href="#">Game.QuestionsPage</a>	Page class for managing game questions . . . . .	18
<a href="#">Game.RoomPage</a>	Class managing the game room interface . . . . .	18
<a href="#">Game.SuchePage</a>	Page class for the stop search functionality . . . . .	19
<a href="#">Game.WuerfelPage</a>	Page class for the dice rolling interface . . . . .	20



# Chapter 5

## File Index

### 5.1 File List

Here is a list of all documented files with brief descriptions:

[App.xaml.cs](#)

Main application class that serves as the entry point for the MAUI application . . . . [21](#)



# Chapter 6

## Namespace Documentation

### 6.1 Game Namespace Reference

#### Data Structures

- class [App](#)  
The main application class that inherits from MAUI Application.
- class [AppShell](#)  
The main shell class that provides navigation framework for the application.
- class [CardDeckPage](#)  
Page class for managing the card deck interface.
- class [CursesPage](#)  
Page for displaying active curses.
- class [InfoPage](#)  
Page class for displaying game information.
- class [JoinSessionPage](#)  
Page class for handling session joining functionality.
- class [LiveMapPage](#)  
Page class for displaying and managing the live map functionality.
- class [MainPage](#)  
The main page of the application.
- class MauiProgram  
Static class that configures and creates the MAUI application.
- class [QuestionsPage](#)  
Page class for managing game questions.
- class [RoomPage](#)  
Class managing the game room interface.
- class [SuchePage](#)  
Page class for the stop search functionality.
- class [WuerfelPage](#)  
Page class for the dice rolling interface.



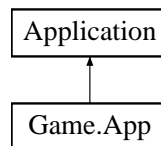


## Chapter 7

# Data Structure Documentation

### 7.1 Game.App Class Reference

The main application class that inherits from MAUI Application.  
Inheritance diagram for Game.App:



Public Member Functions

- [App \(\)](#)  
Constructor for the [App](#) class.

#### 7.1.1 Detailed Description

The main application class that inherits from MAUI Application.  
Handles the initialization of the application and sets up the initial navigation

#### 7.1.2 Constructor & Destructor Documentation

##### 7.1.2.1 App()

Game.App.App () [inline]

Constructor for the [App](#) class.

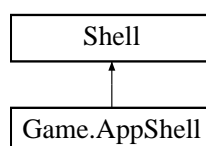
Initializes the application and sets the main page to [AppShell](#)

The documentation for this class was generated from the following file:

- [App.xaml.cs](#)

### 7.2 Game.AppShell Class Reference

The main shell class that provides navigation framework for the application.  
Inheritance diagram for Game.AppShell:



## Public Member Functions

- [AppShell](#) ()  
Constructor for the [AppShell](#) class.

### 7.2.1 Detailed Description

The main shell class that provides navigation framework for the application.

This class inherits from Shell and provides the navigation structure and routing capabilities for the application

### 7.2.2 Constructor & Destructor Documentation

#### 7.2.2.1 AppShell()

Game.AppShell.AppShell () [inline]

Constructor for the [AppShell](#) class.

Initializes the shell components and sets up navigation

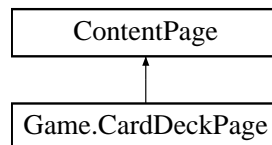
The documentation for this class was generated from the following file:

- AppShell.xaml.cs

## 7.3 Game.CardDeckPage Class Reference

Page class for managing the card deck interface.

Inheritance diagram for Game.CardDeckPage:



## Public Member Functions

- [CardDeckPage](#) ()  
Constructor for [CardDeckPage](#).

### 7.3.1 Detailed Description

Page class for managing the card deck interface.

Handles the display and interaction with the game's card deck system

### 7.3.2 Constructor & Destructor Documentation

#### 7.3.2.1 CardDeckPage()

Game.CardDeckPage.CardDeckPage () [inline]

Constructor for [CardDeckPage](#).

Initializes the card deck page components

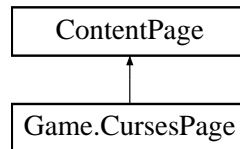
The documentation for this class was generated from the following file:

- CardDeckPage.xaml.cs

## 7.4 Game.CursesPage Class Reference

Page for displaying active curses.

Inheritance diagram for Game.CursesPage:



#### Public Member Functions

- [CursesPage \(\)](#)  
Constructor for [CursesPage](#).

#### 7.4.1 Detailed Description

Page for displaying active curses.

Shows all currently active curses affecting the Seekers, including their effects and duration

#### 7.4.2 Constructor & Destructor Documentation

##### 7.4.2.1 CursesPage()

Game.CursesPage.CursesPage () [inline]

Constructor for [CursesPage](#).

Initializes the page with hardcoded example curses

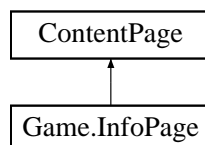
The documentation for this class was generated from the following file:

- CursesPage.xaml.cs

## 7.5 Game.InfoPage Class Reference

Page class for displaying game information.

Inheritance diagram for Game.InfoPage:



#### Public Member Functions

- [InfoPage \(\)](#)  
Constructor for [InfoPage](#).

#### 7.5.1 Detailed Description

Page class for displaying game information.

Manages and displays detailed information about the current game session, including player roles and scores

#### 7.5.2 Constructor & Destructor Documentation

##### 7.5.2.1 InfoPage()

Game.InfoPage.InfoPage () [inline]

Constructor for [InfoPage](#).

Initializes the page with example game information including role, start time, and current score

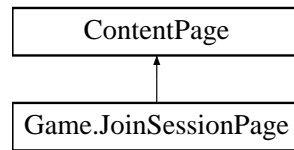
The documentation for this class was generated from the following file:

- InfoPage.xaml.cs

## 7.6 Game.JoinSessionPage Class Reference

Page class for handling session joining functionality.

Inheritance diagram for Game.JoinSessionPage:



Public Member Functions

- [JoinSessionPage \(\)](#)  
Constructor for the [JoinSessionPage](#).

### 7.6.1 Detailed Description

Page class for handling session joining functionality.

This class manages the user interface and logic for joining an existing game session

### 7.6.2 Constructor & Destructor Documentation

#### 7.6.2.1 JoinSessionPage()

`Game.JoinSessionPage.JoinSessionPage () [inline]`

Constructor for the [JoinSessionPage](#).

Initializes the components of the join session page

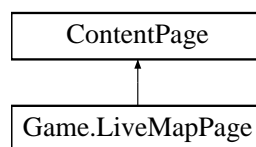
The documentation for this class was generated from the following file:

- `JoinSessionPage.xaml.cs`

## 7.7 Game.LiveMapPage Class Reference

Page class for displaying and managing the live map functionality.

Inheritance diagram for Game.LiveMapPage:



Public Member Functions

- [LiveMapPage \(\)](#)  
Constructor for [LiveMapPage](#).

Protected Member Functions

- `override async void OnAppearing ()`

### 7.7.1 Detailed Description

Page class for displaying and managing the live map functionality.

This class handles the map display, location tracking, and navigation controls for the live map feature

## 7.7.2 Constructor & Destructor Documentation

### 7.7.2.1 LiveMapPage()

Game.LiveMapPage.LiveMapPage () [inline]

Constructor for [LiveMapPage](#).

Initializes the map components and sets up navigation buttons

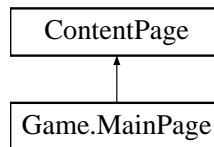
The documentation for this class was generated from the following file:

- LiveMapPage.xaml.cs

## 7.8 Game.MainPage Class Reference

The main page of the application.

Inheritance diagram for Game.MainPage:



### Public Member Functions

- [MainPage](#) ()  
Constructor for the [MainPage](#) class.

### 7.8.1 Detailed Description

The main page of the application.

This class handles the main user interface and navigation logic for the primary page of the application

## 7.8.2 Constructor & Destructor Documentation

### 7.8.2.1 MainPage()

Game.MainPage.MainPage () [inline]

Constructor for the [MainPage](#) class.

Initializes the components and sets up the user interface

The documentation for this class was generated from the following file:

- MainPage.xaml.cs

## 7.9 Game.QuestionsPage.QuestionAnswer Class Reference

Model class for question-answer pairs.

### Properties

- string Question = string.Empty [get, set]  
The question text.
- string Answer = string.Empty [get, set]  
The answer text.

### 7.9.1 Detailed Description

Model class for question-answer pairs.

Represents a single question and its corresponding answer

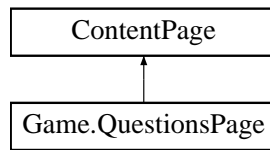
The documentation for this class was generated from the following file:

- QuestionsPage.xaml.cs

## 7.10 Game.QuestionsPage Class Reference

Page class for managing game questions.

Inheritance diagram for Game.QuestionsPage:



### Data Structures

- class [QuestionAnswer](#)  
Model class for question-answer pairs.

### Public Member Functions

- [QuestionsPage](#) ()  
Constructor for [QuestionsPage](#).

#### 7.10.1 Detailed Description

Page class for managing game questions.

Handles the display and interaction with available questions, current questions, and past questions with answers

#### 7.10.2 Constructor & Destructor Documentation

##### 7.10.2.1 QuestionsPage()

Game.QuestionsPage.QuestionsPage () [inline]

Constructor for [QuestionsPage](#).

Initializes the page with example questions and answers

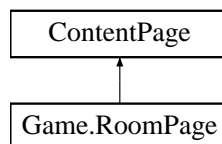
The documentation for this class was generated from the following file:

- QuestionsPage.xaml.cs

## 7.11 Game.RoomPage Class Reference

Class managing the game room interface.

Inheritance diagram for Game.RoomPage:



### Public Member Functions

- [RoomPage](#) (List< string > players)  
Constructor for [RoomPage](#).

### Properties

- IReadOnlyList< string > [Players](#) [get]  
Gets the list of players in the room.

### 7.11.1 Detailed Description

Class managing the game room interface.

Handles the display and interaction of players in a game room, including room management and navigation

### 7.11.2 Constructor & Destructor Documentation

#### 7.11.2.1 RoomPage()

Game.RoomPage.RoomPage (  
     List< string > players) [inline]  
 Constructor for [RoomPage](#).

Parameters

pp	players	pp	List of player names to initialize the room with
----	---------	----	--------------------------------------------------

Initializes the room components and sets up player list

### 7.11.3 Property Documentation

#### 7.11.3.1 Players

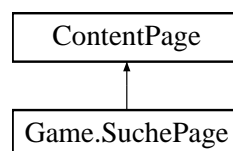
ReadOnlyList<string> Game.RoomPage.Players [get]  
 Gets the list of players in the room.  
 Read-only collection of player names currently in the room  
 The documentation for this class was generated from the following file:

- RoomPage.xaml.cs

## 7.12 Game.SuchePage Class Reference

Page class for the stop search functionality.

Inheritance diagram for Game.SuchePage:



Public Member Functions

- [SuchePage](#) (Dictionary< string, string > haltestellen, TaskCompletionSource< string?> tcs)  
 Constructor for [SuchePage](#).

### 7.12.1 Detailed Description

Page class for the stop search functionality.

Manages the search interface for transit stops, including search functionality and selection handling

### 7.12.2 Constructor & Destructor Documentation

#### 7.12.2.1 SuchePage()

Game.SuchePage.SuchePage (  
     Dictionary< string, string > haltestellen,  
     TaskCompletionSource< string?> tcs) [inline]  
 Constructor for [SuchePage](#).

## Parameters

pp	haltestellen	pp	Dictionary of available stops
pp	tcs	pp	Task completion source for handling selection

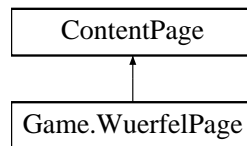
Initializes the search page with available stops and binds the view model  
 The documentation for this class was generated from the following file:

- SuchePage.xaml.cs

## 7.13 Game.WuerfelPage Class Reference

Page class for the dice rolling interface.

Inheritance diagram for Game.WuerfelPage:



## Public Member Functions

- [WuerfelPage](#) (WuerfelViewModel viewModel)  
 Constructor for [WuerfelPage](#).

### 7.13.1 Detailed Description

Page class for the dice rolling interface.

This class manages the user interface and interactions for the dice rolling feature

### 7.13.2 Constructor & Destructor Documentation

#### 7.13.2.1 WuerfelPage()

Game.WuerfelPage.WuerfelPage (  
     WuerfelViewModel viewModel) [inline]  
 Constructor for [WuerfelPage](#).

## Parameters

pp	viewModel	pp	The view model instance to use
----	-----------	----	--------------------------------

Initializes the page components and sets up data binding

The documentation for this class was generated from the following file:

- WuerfelPage.xaml.cs



# Chapter 8

## File Documentation

### 8.1 App.xaml.cs File Reference

Main application class that serves as the entry point for the MAUI application.

#### Data Structures

- class [Game.App](#)

The main application class that inherits from MAUI Application.

#### 8.1.1 Detailed Description

Main application class that serves as the entry point for the MAUI application.

This class initializes the application and sets up the main navigation shell



# Index

## App

Game.App, [13](#)

App.xaml.cs, [21](#)

## AppShell

Game.AppShell, [14](#)

## CardDeckPage

Game.CardDeckPage, [14](#)

## CursesPage

Game.CursesPage, [15](#)

## Game, [11](#)

Game.App, [13](#)

App, [13](#)

Game.AppShell, [13](#)

AppShell, [14](#)

Game.CardDeckPage, [14](#)

CardDeckPage, [14](#)

Game.CursesPage, [14](#)

CursesPage, [15](#)

Game.InfoPage, [15](#)

InfoPage, [15](#)

Game.JoinSessionPage, [16](#)

JoinSessionPage, [16](#)

Game.LiveMapPage, [16](#)

LiveMapPage, [17](#)

Game.MainPage, [17](#)

MainPage, [17](#)

Game.QuestionsPage, [18](#)

QuestionsPage, [18](#)

Game.QuestionsPage.QuestionAnswer, [17](#)

Game.RoomPage, [18](#)

Players, [19](#)

RoomPage, [19](#)

Game.SuchePage, [19](#)

SuchePage, [19](#)

Game.WuerfelPage, [20](#)

WuerfelPage, [20](#)

## InfoPage

Game.InfoPage, [15](#)

## JoinSessionPage

Game.JoinSessionPage, [16](#)

## LiveMapPage

Game.LiveMapPage, [17](#)

## MainPage

Game.MainPage, [17](#)

## Players

Game.RoomPage, [19](#)

## QuestionsPage

Game.QuestionsPage, [18](#)

## README, [1](#)

## RoomPage

Game.RoomPage, [19](#)

## SuchePage

Game.SuchePage, [19](#)

## WuerfelPage

Game.WuerfelPage, [20](#)