App.xaml.cs File Reference

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

Data Structures

class **Game.App**

The main application class that inherits from MAUI Application. More...

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

Data Structures

Here are the data structures with brief descriptions:

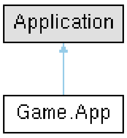
[detail level 1 2 3]

N **Game**	
C **App**	The main application class that inherits from MAUI Application
C **AppShell**	The main shell class that provides navigation framework for the application
C **CardDeckPage**	Page class for managing the card deck interface
C **CursesPage**	Page for displaying active curses
C **InfoPage**	Page class for displaying game information
C **JoinSessionPage**	Page class for handling session joining functionality
C **LiveMapPage**	Page class for displaying and managing the live map functionality
C **MainPage**	The main page of the application
C **QuestionsPage**	Page class for managing game questions
C **QuestionAnswer**	Model class for question-answer pairs
C **RoomPage**	Class managing the game room interface
C **SuchePage**	Page class for the stop search functionality
C **WuerfelPage**	Page class for the dice rolling interface

Game.App Class Reference

The main application class that inherits from MAUI Application. More...

Inheritance diagram for Game.App:



Public Member Functions

App ()

Constructor for the **App** class.

Detailed Description

The main application class that inherits from MAUI Application.

Handles the initialization of the application and sets up the initial navigation

Constructor & Destructor Documentation

◆ App()

Game.App.App ( ) inline

Constructor for the **App** class.

Initializes the application and sets the main page to **AppShell**

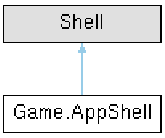
The documentation for this class was generated from the following file:

- **App.xaml.cs**

Game.AppShell Class Reference

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

Inheritance diagram for Game.AppShell:



Public Member Functions

AppShell ()
Constructor for the **AppShell** class.

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

Constructor & Destructor Documentation

◆ AppShell()

Game.AppShell.AppShell ( ) inline

Constructor for the **AppShell** class.

Initializes the shell components and sets up navigation

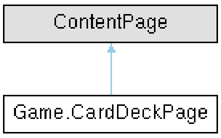
The documentation for this class was generated from the following file:

- **AppShell.xaml.cs**

Game.CardDeckPage Class Reference

Page class for managing the card deck interface. More...

Inheritance diagram for Game.CardDeckPage:



Public Member Functions

CardDeckPage ()
Constructor for **CardDeckPage**.

Detailed Description

Page class for managing the card deck interface.

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

Constructor & Destructor Documentation

◆ CardDeckPage()

Game.CardDeckPage.CardDeckPage ( ) inline

Constructor for **CardDeckPage**.

Initializes the card deck page components

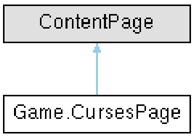
The documentation for this class was generated from the following file:

- **CardDeckPage.xaml.cs**

Game.CursesPage Class Reference

Page for displaying active curses. More...

Inheritance diagram for Game.CursesPage:



Public Member Functions

CursesPage ()

Constructor for **CursesPage**.

Detailed Description

Page for displaying active curses.

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

Constructor & Destructor Documentation

◆ CursesPage()

Game.CursesPage.CursesPage ( ) inline

Constructor for **CursesPage**.

Initializes the page with hardcoded example curses

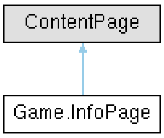
The documentation for this class was generated from the following file:

- **CursesPage.xaml.cs**

Game.InfoPage Class Reference

Page class for displaying game information. More...

Inheritance diagram for Game.InfoPage:



Public Member Functions

InfoPage ()
Constructor for **InfoPage**.

Detailed Description

Page class for displaying game information.

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

Constructor & Destructor Documentation

◆ InfoPage()

Game.InfoPage.InfoPage ( ) inline

Constructor for **InfoPage**.

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

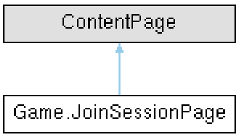
The documentation for this class was generated from the following file:

- **InfoPage.xaml.cs**

Game.JoinSessionPage Class Reference

Page class for handling session joining functionality. More...

Inheritance diagram for Game.JoinSessionPage:



Public Member Functions

JoinSessionPage ()

Constructor for the **JoinSessionPage**.

Detailed Description

Page class for handling session joining functionality.

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

Constructor & Destructor Documentation

◆ JoinSessionPage()

Game.JoinSessionPage.JoinSessionPage ( ) inline

Constructor for the **JoinSessionPage**.

Initializes the components of the join session page

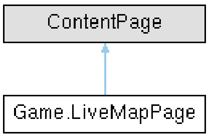
The documentation for this class was generated from the following file:

- **JoinSessionPage.xaml.cs**

Game.LiveMapPage Class Reference

Page class for displaying and managing the live map functionality. More...

Inheritance diagram for Game.LiveMapPage:



Public Member Functions

LiveMapPage ()
Constructor for **LiveMapPage**.

Protected Member Functions

override async void **OnAppearing** ()

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

Constructor & Destructor Documentation

◆ LiveMapPage()

Game.LiveMapPage.LiveMapPage ( ) inline

Constructor for **LiveMapPage**.

Initializes the map components and sets up navigation buttons

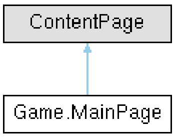
The documentation for this class was generated from the following file:

- **LiveMapPage.xaml.cs**

Game.MainPage Class Reference

The main page of the application. More...

Inheritance diagram for Game.MainPage:



Public Member Functions

MainPage ()

Constructor for the **MainPage** class.

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

Constructor & Destructor Documentation

◆ MainPage()

Game.MainPage.MainPage ( ) inline

Constructor for the **MainPage** class.

Initializes the components and sets up the user interface

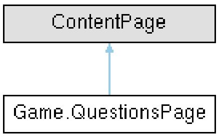
The documentation for this class was generated from the following file:

- **MainPage.xaml.cs**

Game.QuestionsPage Class Reference

Page class for managing game questions. More...

Inheritance diagram for Game.QuestionsPage:



Data Structures

class **QuestionAnswer**

Model class for question-answer pairs. More...

Public Member Functions

QuestionsPage ()

Constructor for **QuestionsPage**.

Detailed Description

Page class for managing game questions.

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

Constructor & Destructor Documentation

◆ 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**

Game.QuestionsPage.QuestionAnswer Class Reference

Model class for question-answer pairs. More...

Properties

string **Question** = string.Empty `[get, set]`

The question text.

string **Answer** = string.Empty `[get, set]`

The answer text.

Detailed Description

Model class for question-answer pairs.

Represents a single question and its corresponding answer

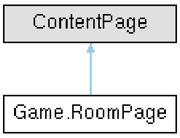
The documentation for this class was generated from the following file:

- **QuestionsPage.xaml.cs**

Game.RoomPage Class Reference

Class managing the game room interface. More...

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.

Detailed Description

Class managing the game room interface.

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

Constructor & Destructor Documentation

◆ RoomPage()

Game.RoomPage.RoomPage (List< string > players) inline

Constructor for **RoomPage**.

Parameters

players List of player names to initialize the room with

Initializes the room components and sets up player list

Property Documentation

◆ Players

IReadOnlyList<string> Game.RoomPage.Players get

Gets the list of players in the room.

Read-only collection of player names currently in the room

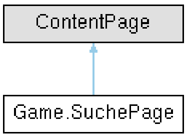
The documentation for this class was generated from the following file:

- **RoomPage.xaml.cs**

Game.SuchePage Class Reference

Page class for the stop search functionality. More...

Inheritance diagram for Game.SuchePage:



Public Member Functions

SuchePage (Dictionary< string, string > haltestellen, TaskCompletionSource< string?> tcs)
Constructor for **SuchePage**.

Detailed Description

Page class for the stop search functionality.

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

Constructor & Destructor Documentation

◆ SuchePage()

Game.SuchePage.SuchePage (Dictionary< string, string > haltestellen,
TaskCompletionSource< string?> tcs) inline

Constructor for **SuchePage**.

Parameters

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

Initializes the search page with available stops and binds the view model

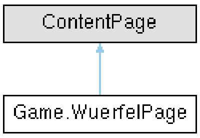
The documentation for this class was generated from the following file:

- **SuchePage.xaml.cs**

Game.WuerfelPage Class Reference

Page class for the dice rolling interface. More...

Inheritance diagram for Game.WuerfelPage:



Public Member Functions

WuerfelPage (WuerfelViewModel viewModel)
Constructor for **WuerfelPage**.

Detailed Description

Page class for the dice rolling interface.

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

Constructor & Destructor Documentation

◆ WuerfelPage()

Game.WuerfelPage.WuerfelPage (WuerfelViewModel viewModel) inline

Constructor for **WuerfelPage**.

Parameters

viewModel 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**

Data Structure Index

A | C | I | J | L | M | Q | R | S | W

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

Here is a list of all documented struct and union fields with links to the struct/union documentation for each field:

- Answer : **Game.QuestionsPage.QuestionAnswer**
- App() : **Game.App**
- AppShell() : **Game.AppShell**
- CardDeckPage() : **Game.CardDeckPage**
- CursesPage() : **Game.CursesPage**
- InfoPage() : **Game.InfoPage**
- JoinSessionPage() : **Game.JoinSessionPage**
- LiveMapPage() : **Game.LiveMapPage**
- MainPage() : **Game.MainPage**
- Players : **Game.RoomPage**
- Question : **Game.QuestionsPage.QuestionAnswer**
- QuestionsPage() : **Game.QuestionsPage**
- RoomPage() : **Game.RoomPage**
- SuchePage() : **Game.SuchePage**
- WuerfelPage() : **Game.WuerfelPage**

Here is a list of all documented functions with links to the struct/union documentation for each field:

- App() : **Game.App**
- AppShell() : **Game.AppShell**
- CardDeckPage() : **Game.CardDeckPage**
- CursesPage() : **Game.CursesPage**
- InfoPage() : **Game.InfoPage**
- JoinSessionPage() : **Game.JoinSessionPage**
- LiveMapPage() : **Game.LiveMapPage**
- MainPage() : **Game.MainPage**
- QuestionsPage() : **Game.QuestionsPage**
- RoomPage() : **Game.RoomPage**
- SuchePage() : **Game.SuchePage**
- WuerfelPage() : **Game.WuerfelPage**

Here is a list of all documented properties with links to the struct/union documentation for each field:

- Answer : **Game.QuestionsPage.QuestionAnswer**
- Players : **Game.RoomPage**
- Question : **Game.QuestionsPage.QuestionAnswer**

Class Hierarchy

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

[detail level 1 2]

C Application	
C Game.App	The main application class that inherits from MAUI Application
C ContentPage	
C Game.CardDeckPage	Page class for managing the card deck interface
C Game.CursesPage	Page for displaying active curses
C Game.InfoPage	Page class for displaying game information
C Game.JoinSessionPage	Page class for handling session joining functionality
C Game.LiveMapPage	Page class for displaying and managing the live map functionality
C Game.MainPage	The main page of the application
C Game.QuestionsPage	Page class for managing game questions
C Game.RoomPage	Class managing the game room interface
C Game.SuchePage	Page class for the stop search functionality
C Game.WuerfelPage	Page class for the dice rolling interface
C Game.QuestionsPage.QuestionAnswer	Model class for question-answer pairs
C Shell	
C Game.AppShell	The main shell class that provides navigation framework for the application

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
~~~

Windows-spezifisch

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

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

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.

5. Ordnerstruktur

```
/Game
├── App.xaml
│   └── App.xaml.cs
 MainPage.xaml
│   └── MainPage.xaml.cs
├── Styles.xaml
├── WürfelPage.xaml
│   └── WuerfelPage.xaml.cs
├── BushalteSuchenPage.xaml
│   └── BushalteSuchenPage.xamll.cs
├── ViewModels
│   ├── WürfelViewModel.cs
│   └── BushalteViewModel.cs
└── Resources
    └── Images
        ├── dice1.png
        ├── dice2.png
        ├── dice3.png
        ├── dice4.png
        ├── dice5.png
        └── dice6.png
```

6. Icons und Lizenz

Die Würfel-Grafiken stammen von **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
```

Game Namespace Reference

Data Structures

class **App**

The main application class that inherits from MAUI Application. More...

class **AppShell**

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

class **CardDeckPage**

Page class for managing the card deck interface. More...

class **CursesPage**

Page for displaying active curses. More...

class **InfoPage**

Page class for displaying game information. More...

class **JoinSessionPage**

Page class for handling session joining functionality. More...

class **LiveMapPage**

Page class for displaying and managing the live map functionality. More...

class **MainPage**

The main page of the application. More...

class **MauiProgram**

Static class that configures and creates the MAUI application.

class **QuestionsPage**

Page class for managing game questions. More...

class **RoomPage**

Class managing the game room interface. More...

class **SuchePage**

Page class for the stop search functionality. More...

class **WuerfelPage**

Page class for the dice rolling interface. More...

Namespace List

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

N	**Game**

Related Pages

Here is a list of all related documentation pages:

README