**Отчет по Главе 3: Компоновки (Layouts)**

**1. Введение**

Данная глава посвящена изучению основных контейнеров компоновки (Layouts) в .NET MAUI. Компоновки являются фундаментальными элементами для структурирования пользовательского интерфейса, так как они определяют, каким образом дочерние элементы будут располагаться и взаимодействовать друг с другом на странице. Понимание принципов работы различных компоновок, таких как StackLayout, Grid и FlexLayout, критически важно для создания адаптивных и визуально привлекательных интерфейсов для кроссплатформенных настольных и мобильных приложений.

**2. Основная часть**

**2.1. Основные концепции и принципы**

.NET MAUI предоставляет несколько типов контейнеров компоновки, каждый из которых имеет свои особенности и предназначен для решения определенных задач по организации UI.

* **StackLayout и VerticalStackLayout/HorizontalStackLayout:**
  + **StackLayout**: Располагает дочерние элементы в одномерном стеке — либо вертикально (по умолчанию), либо горизонтально (если свойство Orientation установлено в Horizontal).
  + **VerticalStackLayout**: Оптимизированная версия StackLayout для вертикального расположения. Элементы располагаются друг под другом.
  + **HorizontalStackLayout**: Оптимизированная версия StackLayout для горизонтального расположения. Элементы располагаются друг рядом с другом в строке.
  + **Основные свойства:**
    - Spacing: Задает расстояние между дочерними элементами.
    - Padding: Внутренние отступы контейнера.
    - VerticalOptions, HorizontalOptions: Определяют выравнивание самого StackLayout внутри родительского контейнера и выравнивание дочерних элементов внутри StackLayout (если у дочерних элементов не переопределены эти свойства).
* **Grid (Сетка):**
  + Располагает дочерние элементы в строках и столбцах. Является мощным и гибким контейнером для создания сложных структур.
  + **Основные концепции:**
    - **RowDefinitions**: Коллекция объектов RowDefinition, определяющих строки сетки и их высоту (Height). Высота может быть абсолютной, пропорциональной (\*, 2\* и т.д. – "звездочная" высота) или автоматической (Auto).
    - **ColumnDefinitions**: Коллекция объектов ColumnDefinition, определяющих столбцы сетки и их ширину (Width). Ширина задается аналогично высоте.
    - **Присоединенные свойства Grid.Row, Grid.Column**: Используются для указания строки и столбца, в которых должен располагаться дочерний элемент. Нумерация начинается с 0.
    - **Grid.RowSpan, Grid.ColumnSpan**: Позволяют элементу занимать несколько строк или столбцов.
    - RowSpacing, ColumnSpacing: Расстояние между строками и столбцами соответственно.
* **FlexLayout:**
  + Гибкий контейнер компоновки, основанный на концепции Flexbox из веб-разработки. Позволяет располагать дочерние элементы в одном направлении (строка или столбец) и управлять их размерами, выравниванием и порядком.
  + **Основные свойства:**
    - Direction: Направление основного потока элементов (Row (по умолчанию), Column, RowReverse, ColumnReverse).
    - Wrap: Определяет, будут ли элементы переноситься на новую строку/столбец, если не помещаются в текущий (NoWrap (по умолчанию), Wrap, Reverse).
    - JustifyContent: Выравнивание дочерних элементов вдоль основной оси (Start, Center, End, SpaceBetween, SpaceAround, SpaceEvenly).
    - AlignItems: Выравнивание дочерних элементов вдоль поперечной оси (Stretch (по умолчанию), Start, Center, End).
    - AlignContent: Выравнивание строк/столбцов (если Wrap активен) вдоль поперечной оси.
    - **Присоединенные свойства FlexLayout.Basis, FlexLayout.Grow, FlexLayout.Shrink**: Управляют размерами и гибкостью отдельных дочерних элементов.
      * Basis: Начальный размер элемента вдоль основной оси.
      * Grow: Коэффициент увеличения элемента, если есть свободное пространство.
      * Shrink: Коэффициент сжатия элемента, если пространства недостаточно.

**2.2. Примеры кода с пояснениями**

Для демонстрации работы компоновок были созданы отдельные страницы: StackLayoutDemoPage.xaml, GridDemoPage.xaml и FlexLayoutDemoPage.xaml.

* **StackLayoutDemoPage.xaml**:

<?xml version="1.0" encoding="utf-8" ?>

<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="HelloMauiApp.StackLayoutDemoPage"

Title="StackLayout Demo">

<ScrollView>

<VerticalStackLayout Spacing="10" Padding="20">

<Label Text="VerticalStackLayout Example" FontSize="Large" HorizontalOptions="Center"/>

<Frame Padding="10" CornerRadius="5" BorderColor="LightGray">

<VerticalStackLayout Spacing="5">

<Label Text="Item 1" BackgroundColor="LightBlue" Padding="10"/>

<Label Text="Item 2" BackgroundColor="LightCoral" Padding="10"/>

<Label Text="Item 3" BackgroundColor="LightGoldenrodYellow" Padding="10"/>

</VerticalStackLayout>

</Frame>

<Label Text="HorizontalStackLayout Example" FontSize="Large" HorizontalOptions="Center" Margin="0,20,0,0"/>

<Frame Padding="10" CornerRadius="5" BorderColor="LightGray">

<HorizontalStackLayout Spacing="5">

<Label Text="Left" BackgroundColor="LightGreen" Padding="20,10"/>

<Label Text="Center" BackgroundColor="LightPink" Padding="20,10"/>

<Label Text="Right" BackgroundColor="LightSkyBlue" Padding="20,10"/>

</HorizontalStackLayout>

</Frame>

<Label Text="StackLayout (Vertical by default) with Spacing" FontSize="Large" HorizontalOptions="Center" Margin="0,20,0,0"/>

<Frame Padding="10" CornerRadius="5" BorderColor="LightGray">

<StackLayout Spacing="15">

<BoxView Color="Red" HeightRequest="50" WidthRequest="100" HorizontalOptions="Center"/>

<BoxView Color="Green" HeightRequest="50" WidthRequest="100" HorizontalOptions="Center"/>

<BoxView Color="Blue" HeightRequest="50" WidthRequest="100" HorizontalOptions="Center"/>

</StackLayout>

</Frame>

<Label Text="StackLayout (Horizontal) with Alignment" FontSize="Large" HorizontalOptions="Center" Margin="0,20,0,0"/>

<Frame Padding="10" CornerRadius="5" BorderColor="LightGray" HeightRequest="150">

<StackLayout Orientation="Horizontal" Spacing="5" VerticalOptions="Center" HorizontalOptions="Center">

<Label Text="Start" BackgroundColor="AliceBlue" VerticalOptions="Start" Padding="10"/>

<Label Text="Center" BackgroundColor="AntiqueWhite" VerticalOptions="Center" Padding="10"/>

<Label Text="End" BackgroundColor="Aqua" VerticalOptions="End" Padding="10"/>

<Label Text="Fill" BackgroundColor="Aquamarine" VerticalOptions="Fill" Padding="10"/>

</StackLayout>

</Frame>

</VerticalStackLayout>

</ScrollView>

</ContentPage>

*Пояснение:* Демонстрирует использование VerticalStackLayout и HorizontalStackLayout, а также обычного StackLayout с разными ориентациями и отступами.

* **GridDemoPage.xaml**:

<?xml version="1.0" encoding="utf-8" ?>

<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="HelloMauiApp.GridDemoPage"

Title="Grid Demo">

<ScrollView>

<VerticalStackLayout Spacing="20" Padding="10">

<Label Text="Simple 2x2 Grid" FontSize="Large" HorizontalOptions="Center"/>

<Grid RowSpacing="5" ColumnSpacing="5">

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Label Grid.Row="0" Grid.Column="0" Text="(0,0)" BackgroundColor="LightBlue" Padding="20" HorizontalTextAlignment="Center"/>

<Label Grid.Row="0" Grid.Column="1" Text="(0,1)" BackgroundColor="LightCoral" Padding="20" HorizontalTextAlignment="Center"/>

<Label Grid.Row="1" Grid.Column="0" Text="(1,0)" BackgroundColor="LightGoldenrodYellow" Padding="20" HorizontalTextAlignment="Center"/>

<Label Grid.Row="1" Grid.Column="1" Text="(1,1)" BackgroundColor="LightGreen" Padding="20" HorizontalTextAlignment="Center"/>

</Grid>

<Label Text="Grid with ColumnSpan and RowSpan" FontSize="Large" HorizontalOptions="Center" Margin="0,20,0,0"/>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="70" />

<RowDefinition Height="70" />

<RowDefinition Height="70" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<BoxView Grid.Row="0" Grid.Column="0" Color="Red" />

<Label Grid.Row="0" Grid.Column="1" Grid.ColumnSpan="2" Text="Spans 2 Columns" BackgroundColor="LightGray" TextColor="Black" VerticalTextAlignment="Center" HorizontalTextAlignment="Center" />

<Label Grid.Row="1" Grid.Column="0" Grid.RowSpan="2" Text="Spans 2 Rows" BackgroundColor="LightSteelBlue" TextColor="Black" VerticalTextAlignment="Center" HorizontalTextAlignment="Center" />

<BoxView Grid.Row="1" Grid.Column="1" Color="Green" />

<BoxView Grid.Row="1" Grid.Column="2" Color="Blue" />

<BoxView Grid.Row="2" Grid.Column="1" Color="Orange" />

<BoxView Grid.Row="2" Grid.Column="2" Color="Purple" />

</Grid>

<Label Text="Calculator-like UI (simplified)" FontSize="Large" HorizontalOptions="Center" Margin="0,20,0,0"/>

<Grid ColumnSpacing="1" RowSpacing="1" BackgroundColor="Black">

<Grid.RowDefinitions>

<RowDefinition Height="2\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Label Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="4" Text="0" FontSize="48" TextColor="White" HorizontalTextAlignment="End" VerticalTextAlignment="Center" BackgroundColor="DarkSlateGray" Padding="0,0,20,0"/>

<Button Grid.Row="1" Grid.Column="0" Text="C" BackgroundColor="Gray" TextColor="White" CornerRadius="0"/>

<Button Grid.Row="1" Grid.Column="1" Text="+/-" BackgroundColor="Gray" TextColor="White" CornerRadius="0"/>

<Button Grid.Row="1" Grid.Column="2" Text="%" BackgroundColor="Gray" TextColor="White" CornerRadius="0"/>

<Button Grid.Row="1" Grid.Column="3" Text="/" BackgroundColor="Orange" TextColor="White" CornerRadius="0"/>

<Button Grid.Row="2" Grid.Column="0" Text="7" BackgroundColor="DarkGray" TextColor="White" CornerRadius="0"/>

<Button Grid.Row="2" Grid.Column="1" Text="8" BackgroundColor="DarkGray" TextColor="White" CornerRadius="0"/>

<Button Grid.Row="2" Grid.Column="2" Text="9" BackgroundColor="DarkGray" TextColor="White" CornerRadius="0"/>

<Button Grid.Row="2" Grid.Column="3" Text="X" BackgroundColor="Orange" TextColor="White" CornerRadius="0"/>

<Button Grid.Row="3" Grid.Column="0" Text="4" BackgroundColor="DarkGray" TextColor="White" CornerRadius="0"/>

<Button Grid.Row="3" Grid.Column="1" Text="5" BackgroundColor="DarkGray" TextColor="White" CornerRadius="0"/>

<Button Grid.Row="3" Grid.Column="2" Text="6" BackgroundColor="DarkGray" TextColor="White" CornerRadius="0"/>

<Button Grid.Row="3" Grid.Column="3" Text="-" BackgroundColor="Orange" TextColor="White" CornerRadius="0"/>

<Button Grid.Row="4" Grid.Column="0" Text="1" BackgroundColor="DarkGray" TextColor="White" CornerRadius="0"/>

<Button Grid.Row="4" Grid.Column="1" Text="2" BackgroundColor="DarkGray" TextColor="White" CornerRadius="0"/>

<Button Grid.Row="4" Grid.Column="2" Text="3" BackgroundColor="DarkGray" TextColor="White" CornerRadius="0"/>

<Button Grid.Row="4" Grid.Column="3" Text="+" BackgroundColor="Orange" TextColor="White" CornerRadius="0"/>

</Grid>

</VerticalStackLayout>

</ScrollView>

</ContentPage>

*Пояснение:* Демонстрирует определение строк и столбцов, размещение элементов в ячейках, использование Grid.RowSpan и Grid.ColumnSpan, а также пример создания интерфейса, подобного калькулятору.

* **FlexLayoutDemoPage.xaml**:

<?xml version="1.0" encoding="utf-8" ?>

<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="HelloMauiApp.FlexLayoutDemoPage"

Title="FlexLayout Demo">

<ScrollView>

<VerticalStackLayout Spacing="15" Padding="10">

<Label Text="FlexLayout Direction=Row, Wrap=Wrap" FontSize="Medium" HorizontalOptions="Center"/>

<Frame BorderColor="LightGray" Padding="5">

<FlexLayout Direction="Row" Wrap="Wrap" AlignItems="Start" JustifyContent="SpaceAround">

<Label Text="Item 1" BackgroundColor="LightBlue" Padding="10" Margin="5" WidthRequest="100"/>

<Label Text="Item 2" BackgroundColor="LightCoral" Padding="10" Margin="5" WidthRequest="150"/>

<Label Text="Item 3" BackgroundColor="LightGoldenrodYellow" Padding="10" Margin="5" WidthRequest="80"/>

<Label Text="Item 4" BackgroundColor="LightGreen" Padding="10" Margin="5" WidthRequest="120"/>

<Label Text="Item 5" BackgroundColor="LightPink" Padding="10" Margin="5" WidthRequest="90"/>

</FlexLayout>

</Frame>

<Label Text="FlexLayout Direction=Column, AlignItems=Center" FontSize="Medium" HorizontalOptions="Center" Margin="0,15,0,0"/>

<Frame BorderColor="LightGray" Padding="5" HeightRequest="250">

<FlexLayout Direction="Column" AlignItems="Center" JustifyContent="SpaceEvenly">

<Button Text="Button A" WidthRequest="200"/>

<Button Text="Button B" WidthRequest="150"/>

<Button Text="Button C" WidthRequest="250"/>

</FlexLayout>

</Frame>

<Label Text="FlexLayout with FlexBasis, FlexGrow, FlexShrink" FontSize="Medium" HorizontalOptions="Center" Margin="0,15,0,0"/>

<Frame BorderColor="LightGray" Padding="5">

<FlexLayout Direction="Row" HeightRequest="100">

<BoxView Color="CornflowerBlue" FlexLayout.Basis="150" FlexLayout.Grow="1" />

<BoxView Color="DarkOrange" FlexLayout.Basis="100" FlexLayout.Grow="2" />

<BoxView Color="SpringGreen" FlexLayout.Grow="1" />

</FlexLayout>

</Frame>

<Label Text="(CornflowerBlue: Basis=150, Grow=1. DarkOrange: Basis=100, Grow=2. SpringGreen: Basis=auto(0), Grow=1)" FontSize="Small" HorizontalOptions="Center"/>

</VerticalStackLayout>

</ScrollView>

</ContentPage>

*Пояснение:* Демонстрирует различные свойства FlexLayout, такие как Direction, Wrap, AlignItems, JustifyContent, и присоединенные свойства FlexLayout.Basis, FlexLayout.Grow для управления гибкостью дочерних элементов.

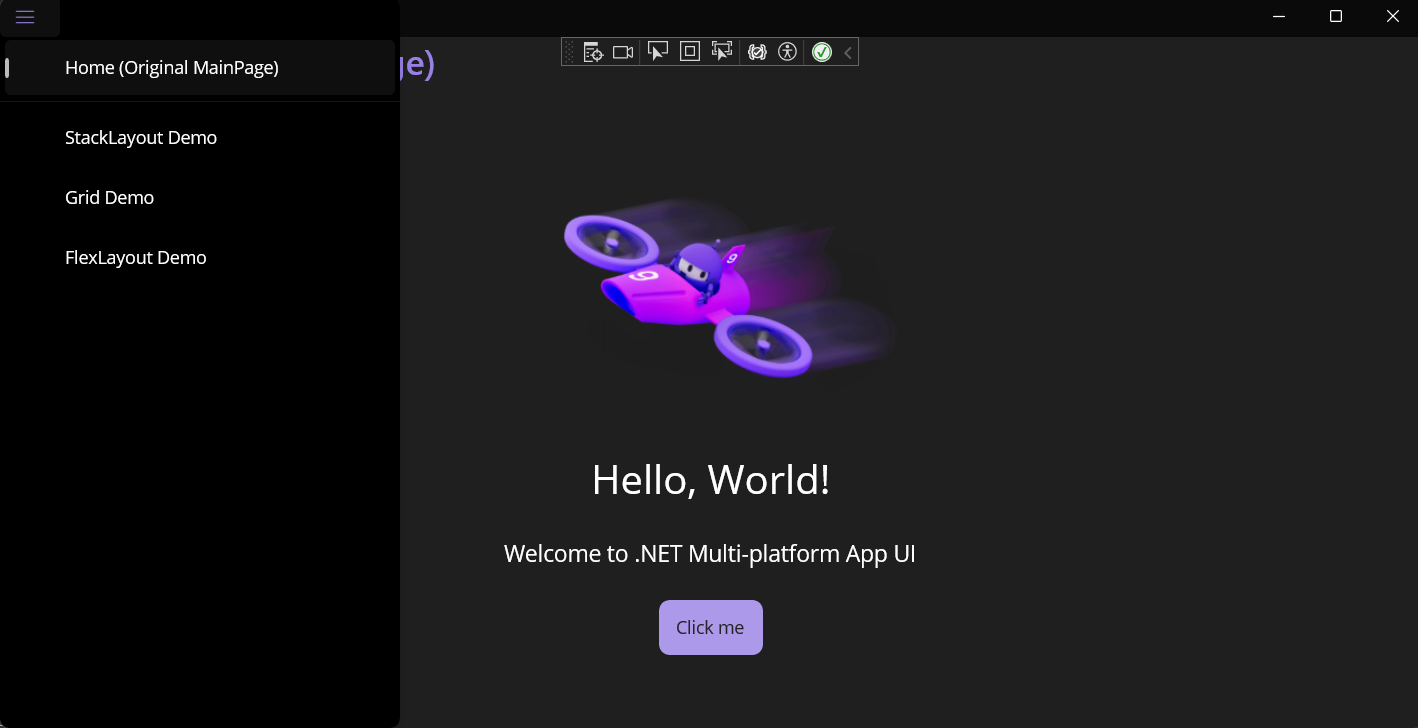
**3. Заключение**

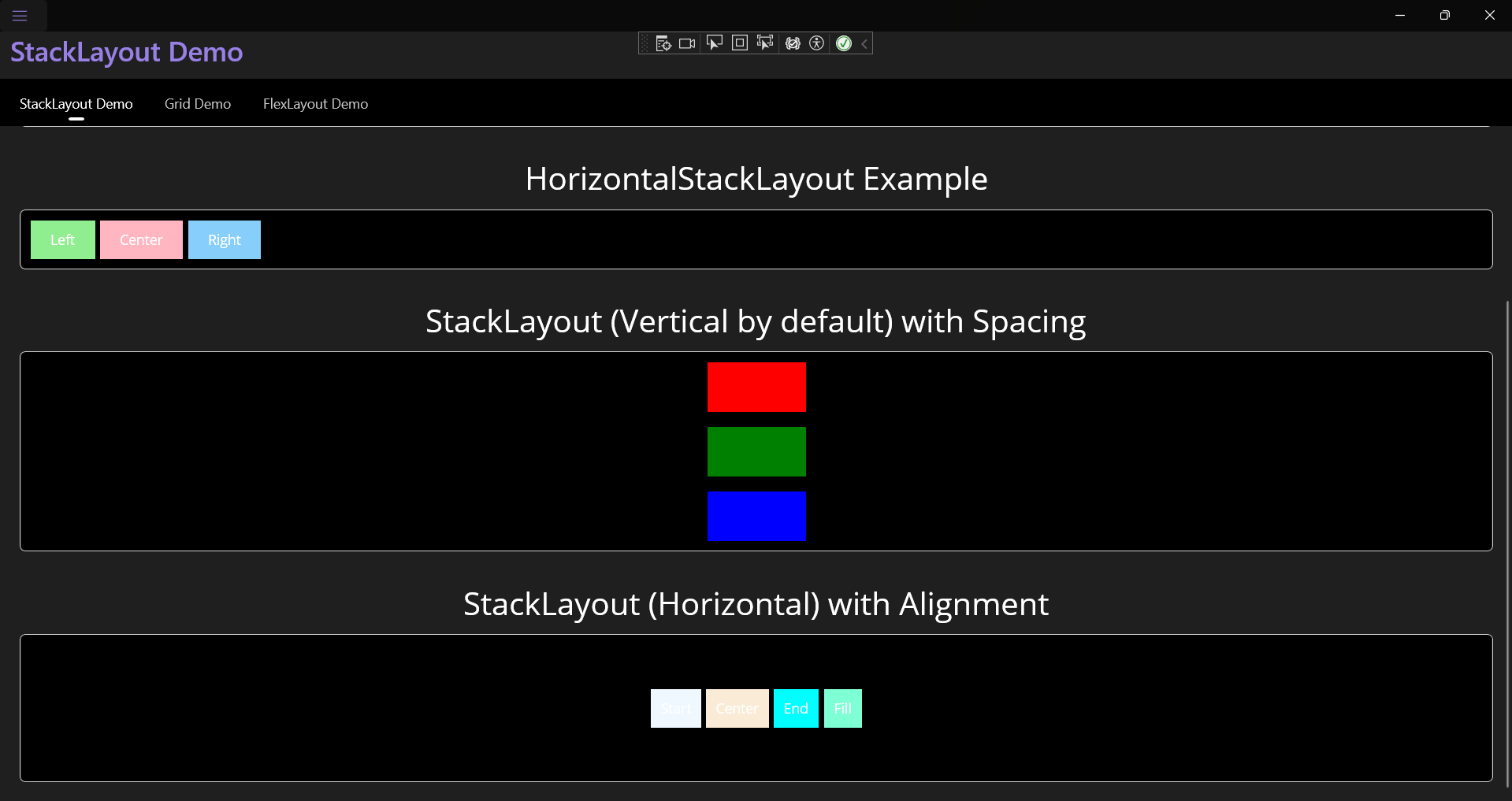
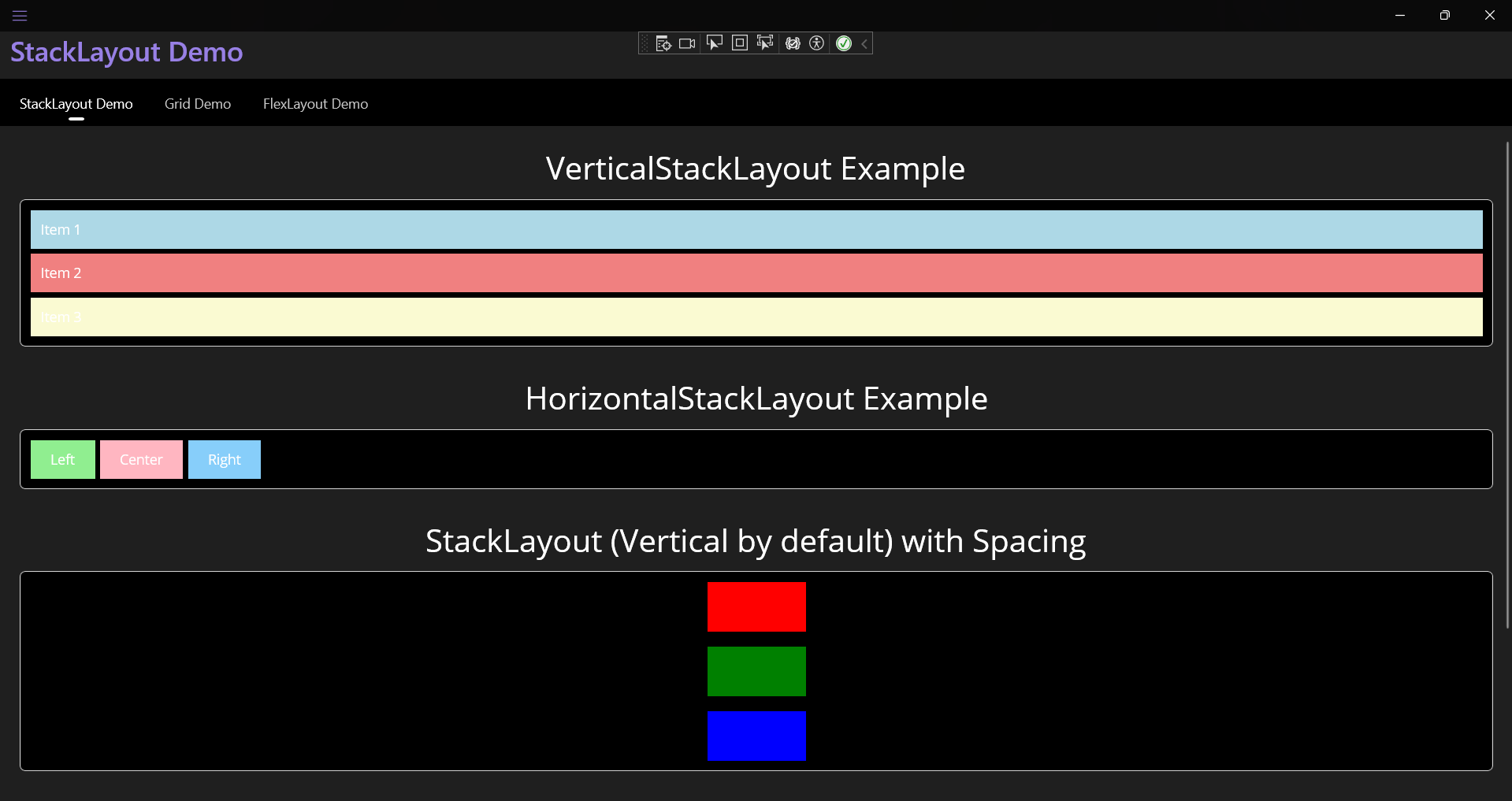
Изучение и практическое применение компоновок StackLayout, Grid и FlexLayout является неотъемлемой частью разработки пользовательских интерфейсов на .NET MAUI. Каждый из этих контейнеров предлагает свой набор инструментов для организации элементов, и их правильный выбор и комбинация позволяют создавать сложные, адаптивные и интуитивно понятные интерфейсы для настольных приложений. StackLayout хорош для простых линейных последовательностей, Grid – для табличных структур и точного позиционирования, а FlexLayout – для создания гибких и адаптирующихся под содержимое макетов.

**4. Скриншоты и примеры**

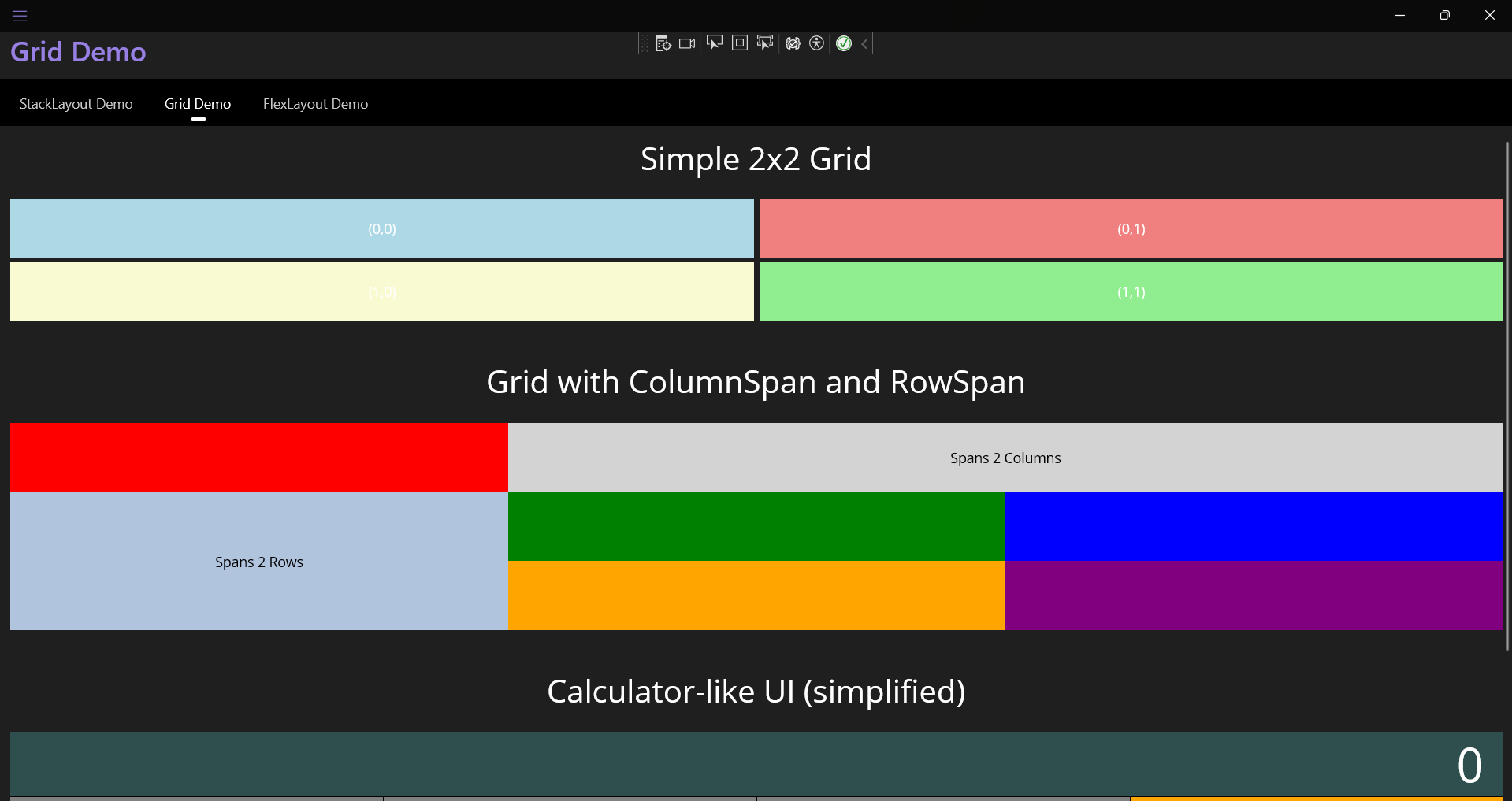
Для демонстрации работы компоновок были созданы и настроены страницы StackLayoutDemoPage, GridDemoPage и FlexLayoutDemoPage. Навигация между ними осуществлялась с помощью бокового всплывающего меню, сконфигурированного в AppShell.xaml. Примеры были запущены на платформе Windows.

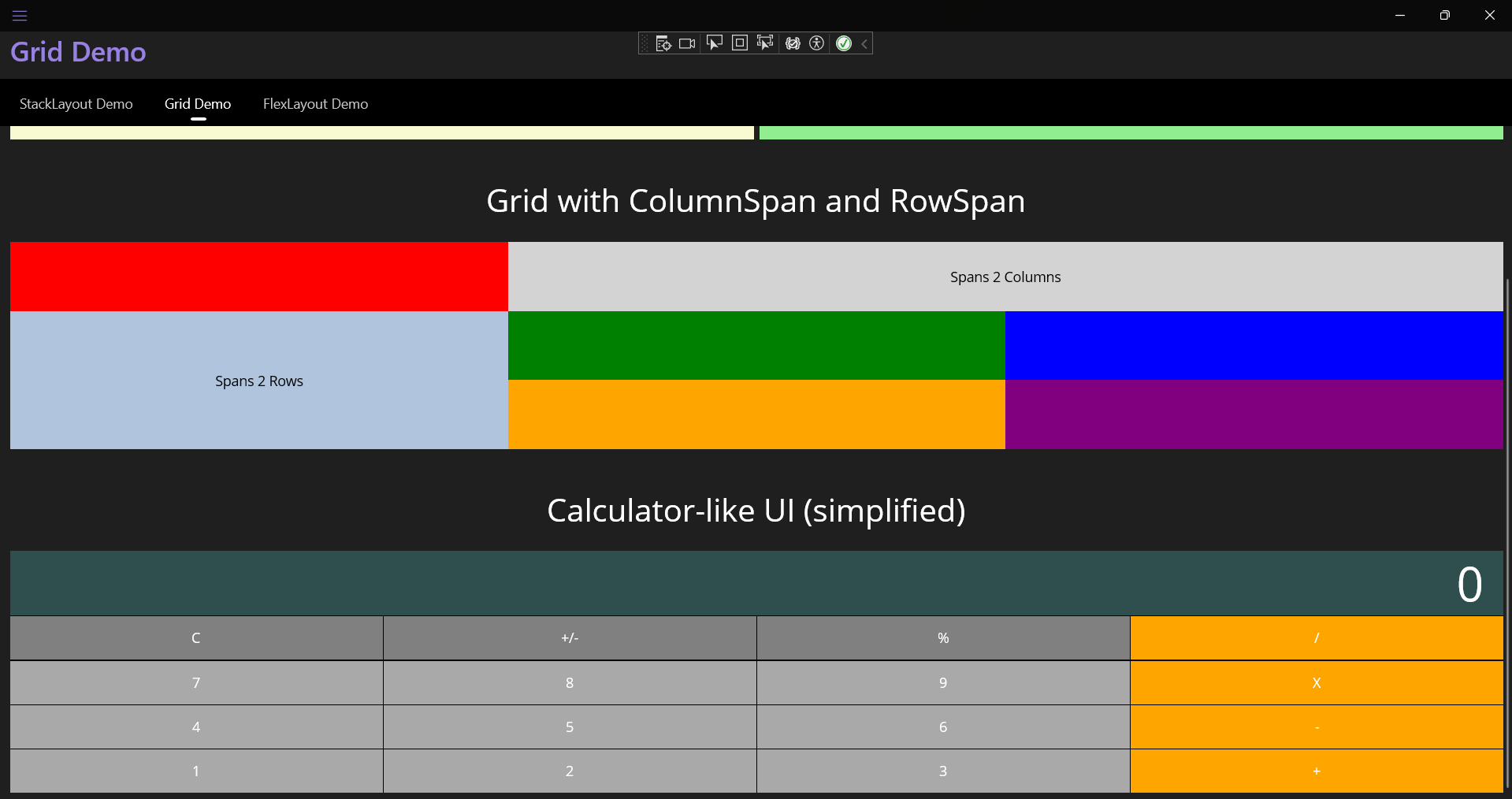
* **Демонстрация StackLayoutDemoPage:**



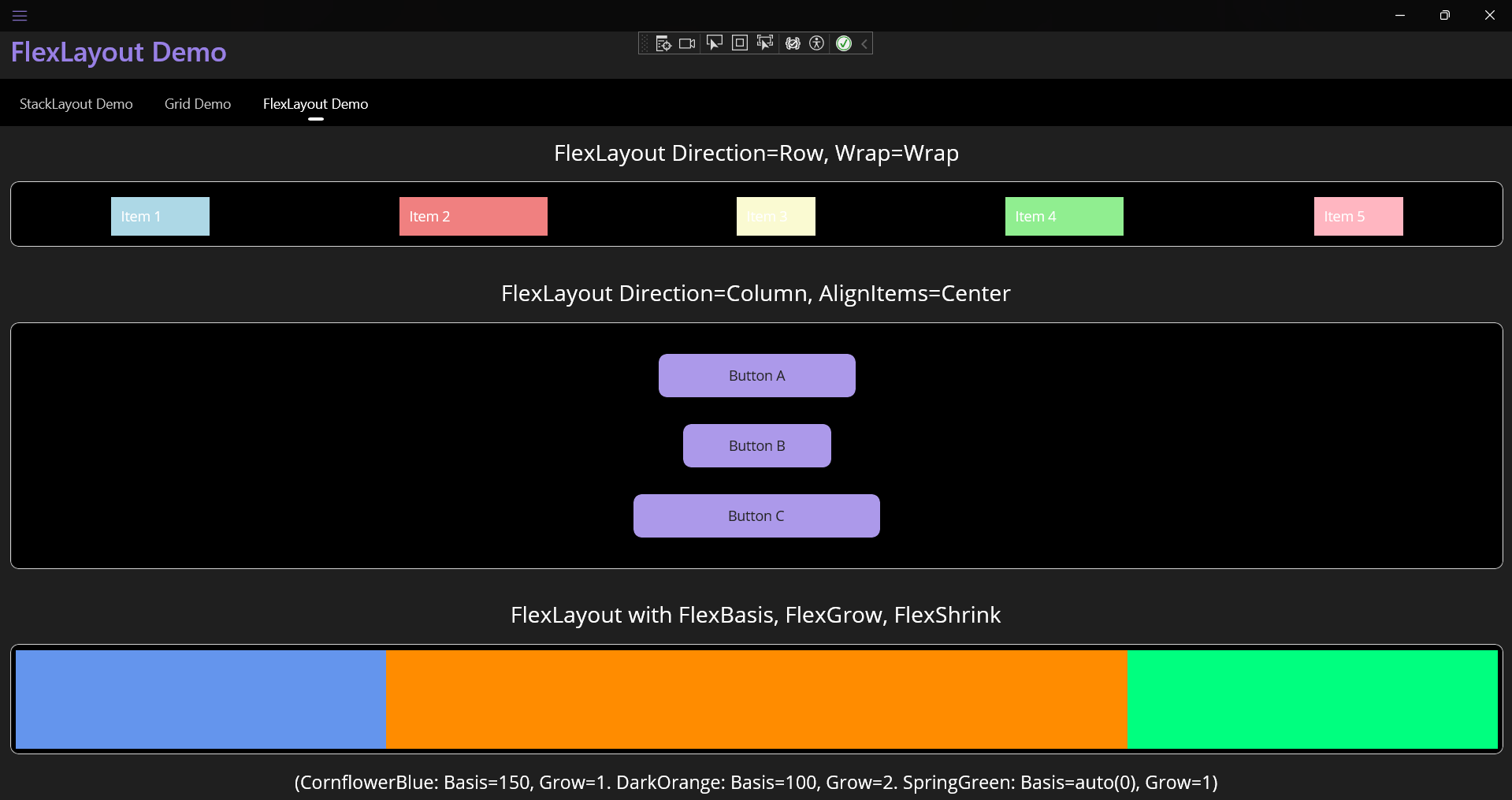
*"Рис. 3.1. Различные примеры использования StackLayout, VerticalStackLayout и HorizontalStackLayout."*

* **Демонстрация GridDemoPage:**



  
*"Рис. 3.2. Примеры использования Grid, включая ColumnSpan, RowSpan и макет калькулятора."*

* **Демонстрация FlexLayoutDemoPage:**

  
*"Рис. 3.3. Примеры использования FlexLayout с различными свойствами направления, переноса и гибкости элементов."*

**Процесс выполнения примеров:**

1. В проекте HelloMauiApp были созданы три новые страницы XAML: StackLayoutDemoPage.xaml, GridDemoPage.xaml, FlexLayoutDemoPage.xaml.
2. Каждая страница была заполнена XAML-кодом, демонстрирующим соответствующий тип компоновки, согласно примерам из [metanit.com/sharp/maui/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fmetanit.com%2Fsharp%2Fmaui%2F) (главы 3.1, 3.2, 3.3).
3. Файл AppShell.xaml был модифицирован для включения навигации к этим трем страницам через боковое всплывающее меню (FlyoutItem).
4. Файл App.xaml.cs был настроен так, чтобы MainPage = new AppShell();.
5. Приложение было скомпилировано и запущено на платформе Windows. Навигация между страницами через боковое меню работала корректно. Все элементы на страницах отображались согласно ожиданиям.

**5. Документация**

* **Ссылка на GitHub с исходным кодом:** <https://github.com/Nikita-Levuskin/C-sharp-practic/tree/main/1>
* **Описание выполненных шагов:**
  1. Изучены материалы по теме "Компоновки: StackLayout, Grid, FlexLayout" (главы 3.1, 3.2, 3.3) с ресурса [metanit.com/sharp/maui/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fmetanit.com%2Fsharp%2Fmaui%2F).
  2. Созданы три отдельные страницы (StackLayoutDemoPage, GridDemoPage, FlexLayoutDemoPage) для демонстрации каждого типа компоновки.
  3. Реализованы примеры XAML-разметки для каждой страницы, иллюстрирующие основные свойства и возможности компоновок.
  4. Настроена навигация между созданными страницами с использованием AppShell.xaml и бокового всплывающего меню.
  5. Проведено тестирование отображения и поведения компоновок на платформе Windows.
  6. Сделаны скриншоты работающих примеров.