Siatka (Grid) - jednostki

Dotychczas, w większości przypadków, używaliśmy gwiazdki dla określenia szerokości i wysokość, która mówi, że wiersz lub kolumna powinna zająć pewien procent całej dostępnej przestrzeni. Jednak dostępne są jeszcze dwa inne sposoby na kreślenie szerokości i wysokości kolumn i wierszy: jednostki bezwzględne i automatyczna (Auto) szerokość/wysokość. Spróbujmy zrobić siatkę, gdzie wykorzystamy wszystkie trzy jednostki:

```
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="*"/>
            <ColumnDefinition Width="Auto"/>
            <ColumnDefinition Width="150"/>
        </Grid.ColumnDefinitions>
        <Button Grid.Column="0">Przycisk 1</Button>
        <Button Grid.Column="1">Przycisk 2 może być taki szeroki na ile wprowadzimy
tekst</Button>
        <Button Grid.Column="3">Przycisk 3</Button>
    </Grid>
```



W tym przykładzie pierwszy przycisk ma szerokość gwiazdki, drugi ma szerokość ustawioną na Auto, natomiast trzeci ma sztywną wartość 150 pikseli.

Rezultat widoczny na zrzucie ekranu pokazuje, jak drugi przycisk ma dokładnie taką szerokość jaka jest niezbędna do narysowania jego długiego tekstu, trzeci przycisk jest szeroki dokładnie na 150 pikseli – tak jak zostało to zdefiniowane w kolumnie, natomiast trzeci przycisk, ze zmienną wartością szerokości, bierze całą pozostałą przestrzeń.

Jeśli kilka kolumn (lub wierszy) ma zmienną (gwiazdkową) szerokość to automatycznie dzielą one między sobą pozostałą szerokość (wysokość) – czyli tę, która nie jest wykorzystywana przez kolumny (wiersze) o wartościach bezwzględnych lub tych, które mają automatyczną szerokość (wysokość). Jest to szczególnie widoczne podczas zmiany rozmiaru okna.

Jest to bardzo użyteczna technika do wykorzystania podczas projektowania całego spektrum różnych okien dialogowych. Na przykład, wyobraź sobie prosty formularz kontaktowy, gdzie użytkownik może wprowadzić swoje imię, adres e-mail i komentarz. Pierwsze dwa pola mają zazwyczaj ściśle określoną wysokość, podczas gdy ostatnie pole zajmuje tyle miejsca ile tylko może oferując tym samym więcej miejsca na komentarz.

Grid - scalanie komórek

Domyślnym zachowaniem panelu Grid jest to, że każda kontrolka zajmuje jedną komórkę. Jednak czasem możemy chcieć, aby kontrolka zajmowała obszar większy niż jeden rząd czy kolumna. Na szczęście Grid dostarcza do tego odpowiednie narzędzia, w postaci dołączonych właściwości ColumnSpan oraz RowSpan. Domyślną wartością dla tych właściwości jest 1, jednak można określić większy zakres kolumn oraz wierszy jakie mają zostać zajęte.

Poniższy przykład przedstawia użycie właściwości ColumnSpan:

```
<Grid>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="1*"/>
        <ColumnDefinition Width="1*"/>

    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
        <RowDefinition Height="1*"/>
        <RowDefinition Height="1*"/>
    </Grid.RowDefinitions>
    <Button Grid.Column="0" Grid.Row="0">Przycisk 1</Button>
    <Button Grid.Column="1" Grid.Row="0" Click="Button_Click">Przycisk 2</Button>
    <Button Grid.ColumnSpan="2" Grid.Row="1">Przycisk 3</Button>

</Grid>
```

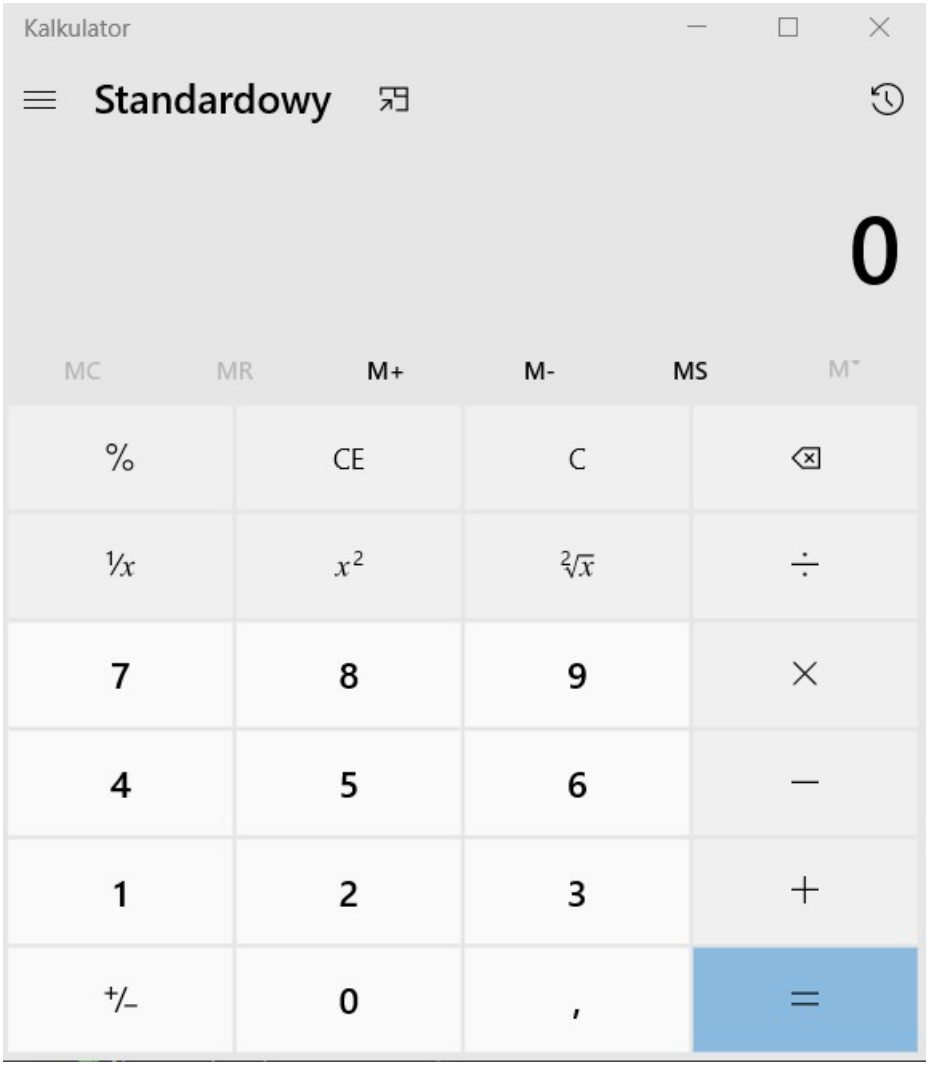


W kodzie zdefiniowaliśmy dwie kolumny oraz dwa wiersze, które zajmują taką samą przestrzeń. Pierwsze dwa przyciski wykorzystują kolumny w normalny sposób. Jednak w przypadku trzeciego przycisku dodanie ColumnSpan sprawiło, że zajmuje on przestrzeń dwóch kolumn.

Przykład jest na tyle prosty, że mogliśmy równie dobrze użyć do tego kombinacji różnych paneli. Jednak w bardziej zaawansowanych przypadkach takie rozwiązanie jest naprawdę poręczne.

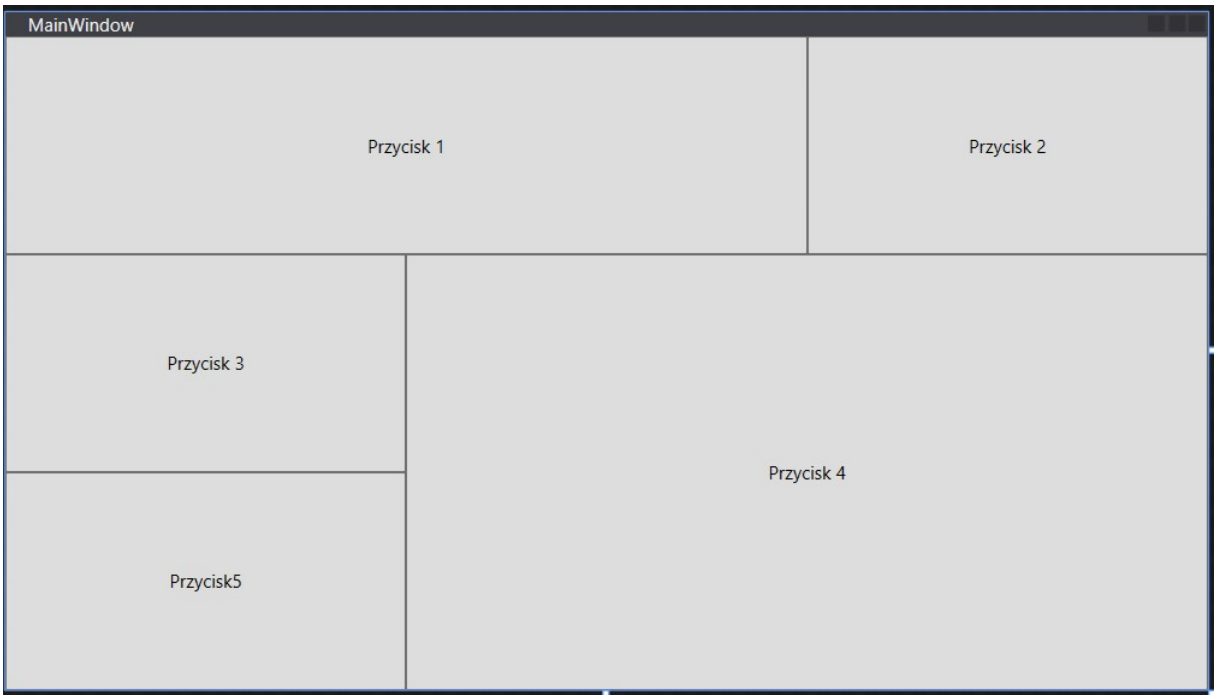
Ćwiczenie:

Korzystając z siatki wykonaj następujący wygląd kalkulatora

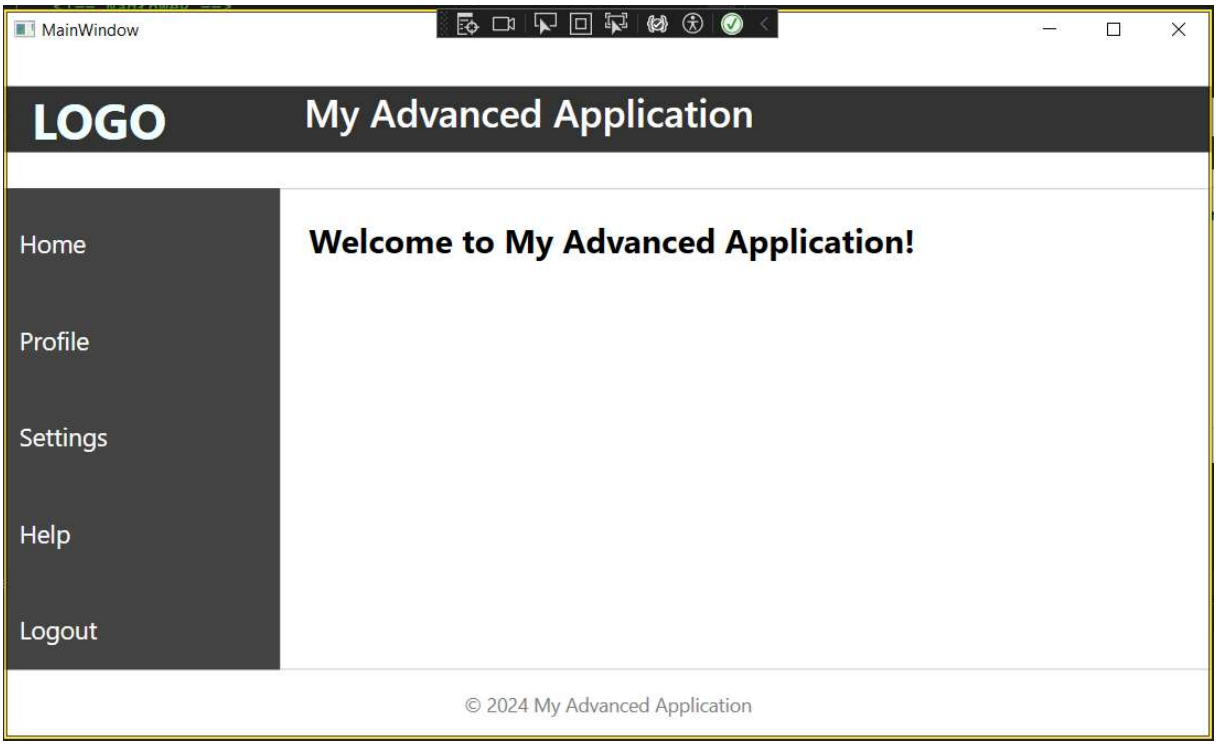


	Rodzaj	Główny bohater	Krótki opis
Sharknado	Horror	Rekin	Wirujące szczęki
Sharktopussy		Zmutowany rekin	Macki, dużo macek
Piranha 3DD		Pirania	Ostre rwanie

Ćw2.Dokonaj następującego podziału siatki:



Wykonaj następujący wygląd aplikacji korzystając z siatki:



Rozwiązanie powyższego zadania:

```
    <Grid>
        <!-- Definicja wierszy i kolumn -->
        <Grid.RowDefinitions>
            <RowDefinition Height="100"/>
            <!-- Nagłówek -->
            <RowDefinition Height="*"/>
            <!-- Sekcja główna (menu + treść) -->
            <RowDefinition Height="50"/>
            <!-- Stopka -->
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="200"/>
            <!-- Menu boczne -->
            <ColumnDefinition Width="*"/>
            <!-- Sekcja treści -->
        </Grid.ColumnDefinitions>

        <!-- Nagłówek -->
        <StackPanel Grid.Row="0" Grid.ColumnSpan="2" Background="#333"
Orientation="Horizontal" VerticalAlignment="Center">
            <TextBlock Text="LOGO" FontSize="36" FontWeight="Bold"
Foreground="Azure" Margin="20,0,100,0"/>
            <TextBlock Text="My Advanced Application" FontSize="28"
FontWeight="SemiBold" Foreground="White" />
        </StackPanel>


        <!-- Menu boczne -->
        <StackPanel Grid.Row="1" Grid.Column="0" Background="#444"
VerticalAlignment="Top">
            <Button Content="Home" FontSize="18" Foreground="White"
Background="#444" Height="60" BorderBrush="Transparent" Margin="0,10,0,0"
                    HorizontalContentAlignment="Left" Padding="10,0"/>
            <Button Content="Profile" FontSize="18" Foreground="White"
Background="#444" Height="60" BorderBrush="Transparent" Margin="0,10,0,0"
                    HorizontalContentAlignment="Left" Padding="10,0"/>
            <Button Content="Settings" FontSize="18" Foreground="White"
Background="#444" Height="60" BorderBrush="Transparent" Margin="0,10,0,0"
                    HorizontalContentAlignment="Left" Padding="10,0"/>
            <Button Content="Help" FontSize="18" Foreground="White"
Background="#444" Height="60" BorderBrush="Transparent" Margin="0,10,0,0"
                    HorizontalContentAlignment="Left" Padding="10,0"/>
            <Button Content="Logout" FontSize="18" Foreground="White"
Background="#444" Height="60" BorderBrush="Transparent" Margin="0,10,0,0"
                    HorizontalContentAlignment="Left" Padding="10,0"/>
        </StackPanel>

        <!-- Sekcja treści -->
        <Border Grid.Row="1" Grid.Column="1" Background="White"
BorderBrush="#CCC" BorderThickness="1" Padding="20">
            <TextBlock Text="Welcome to My Advanced Application!" FontSize="24"
FontWeight="Bold"/>
        </Border>

        <!-- Stopka -->
        <TextBlock Grid.Row="2" Grid.ColumnSpan="2" Text="© 2024 My Advanced
Application"
                   FontSize="14" HorizontalAlignment="Center"
VerticalAlignment="Center" Foreground="Gray"/>
    </Grid>
```

```
 <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="*" />
            <ColumnDefinition Width="*" />
            <ColumnDefinition Width="*" />
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
            <RowDefinition Height="*" />
            <RowDefinition Height="*" />
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>
        <Button Grid.ColumnSpan="2">Button 1</Button>
        <Button Grid.Column="3">Button 2</Button>
        <Button Grid.Row="1">Button 3</Button>
        <Button Grid.Column="1" Grid.Row="1" Grid.RowSpan="2"
Grid.ColumnSpan="2">Button 4</Button>
        <Button Grid.Column="0" Grid.Row="2">Button 5</Button>
    </Grid>
```

W przypadku trzech kolumn i trzech wierszy mielibyśmy dziewięć komórek. Jednak w powyższym przykładzie skorzystaliśmy z rozszerzania kontrolek na pozostałe komórki, tak aby wypełnić całą przestrzeń tylko pięcioma przyciskami.

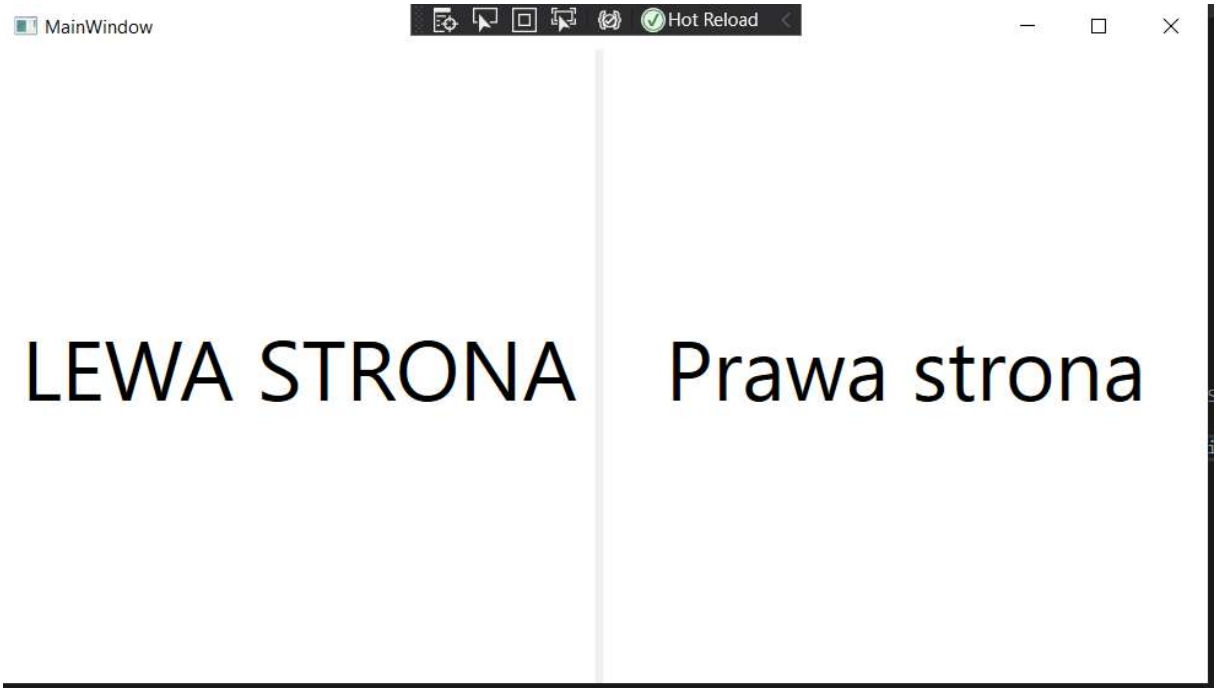
GridSplitter

Stosując panel Grid można bardzo łatwo dzielić dostępną przestrzeń na pojedyncze komórki . Wykorzystując definicję kolumn oraz wierszy, jesteśmy w stanie określić ile przestrzeni mają one zajmować. Jednak co zrobić jeżeli będziemy chcieli pozostawić ten wybór użytownikowi? Z pomocą przychodzi GridSplitter.

W celu zastosowania właściwości GridSplitter należy go dodać do kolumny, lub wiersza z odpowiednią ilością przestrzeni, np. 5 pikseli. Umożliwi to użytkownikowi na przeciąganie go z jednej strony na drugą, lub z góry na dół, tym samym zmieniając rozmiar kolumn lub wierszy, po obu jego stronach.

```
<Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="*"/>
            <ColumnDefinition Width="5"/>
            <ColumnDefinition Width="*"/>
        </Grid.ColumnDefinitions>
        <TextBlock FontSize="55" HorizontalAlignment="Center"
VerticalAlignment="Center" TextWrapping="Wrap">LEWA STRONA</TextBlock>
        <GridSplitter Grid.Column="1" Width="5" HorizontalAlignment="Stretch" />
        <TextBlock Grid.Column="2" FontSize="55" HorizontalAlignment="Center"
VerticalAlignment="Center" TextWrapping="Wrap">Prawa strona</TextBlock>


    </Grid>
```



Stworzyliśmy prostą siatkę wraz z dwoma kolumnami o takiej samej szerokości oraz kolumną o szerokości 5 pikseli pomiędzy nimi. Boczne kolumny zawierają tylko kontrolkę TextBlock aby zilustrować problem. Jak możemy zobaczyć na zrzucie ekranu, GridSplitter jest reprezentowany przez szarą linię pomiędzy dwiema kolumnami wraz z efektem zmiany kursora, kiedy ten jest nad, aby pokazać, że możliwa jest zmiana szerokości.

GridSplitter w poziomie

GridSplitter jest bardzo prosty w zastosowaniu i można go stosować także do dzielenia interfejsu w poziomie. W gruncie rzeczy zmiany jakich należy dokonać w kodzie, aby podział interfejsu był horyzontalny są minimalne. Po prostu zamienimy kolumny na wiersze oraz określimy wysokość GridSplitter'a zamiast szerokości.

```
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="*" />
            <RowDefinition Height="5" />
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>
        <TextBlock FontSize="55" HorizontalAlignment="Center"
VerticalAlignment="Center" TextWrapping="Wrap">Góra</TextBlock>
        <GridSplitter Grid.Row="1" Height="5" HorizontalAlignment="Stretch" />
        <TextBlock Grid.Row="2" FontSize="55" HorizontalAlignment="Center"
VerticalAlignment="Center" TextWrapping="Wrap">Dół</TextBlock>

    </Grid>
```



Mój przykład kalkulatora:



```
 <Grid>
        <!-- Definicja wierszy i kolumn -->
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*"/>
            <RowDefinition Height="*"/>
            <RowDefinition Height="*"/>
            <RowDefinition Height="*"/>
            <RowDefinition Height="*"/>
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="*"/>
            <ColumnDefinition Width="*"/>
            <ColumnDefinition Width="*"/>
            <ColumnDefinition Width="*"/>
        </Grid.ColumnDefinitions>

        <!-- Ekran kalkulatora -->
        <TextBox Grid.Row="0" Grid.ColumnSpan="4" FontSize="32"
TextAlignment="Right" VerticalAlignment="Center" Padding="10" Text="0"
IsReadOnly="True"/>

        <!-- Przyciski -->
        <!-- Pierwszy rząd -->
        <Button Grid.Row="1" Grid.Column="0" Content="7" FontSize="24" />
        <Button Grid.Row="1" Grid.Column="1" Content="8" FontSize="24" />
        <Button Grid.Row="1" Grid.Column="2" Content="9" FontSize="24" />
        <Button Grid.Row="1" Grid.Column="3" Content="/" FontSize="24"
Background="LightGray"/>

        <!-- Drugi rząd -->
        <Button Grid.Row="2" Grid.Column="0" Content="4" FontSize="24" />
        <Button Grid.Row="2" Grid.Column="1" Content="5" FontSize="24" />
        <Button Grid.Row="2" Grid.Column="2" Content="6" FontSize="24" />
        <Button Grid.Row="2" Grid.Column="3" Content="*" FontSize="24"
Background="LightGray"/>

        <!-- Trzeci rząd -->
        <Button Grid.Row="3" Grid.Column="0" Content="1" FontSize="24" />
        <Button Grid.Row="3" Grid.Column="1" Content="2" FontSize="24" />
        <Button Grid.Row="3" Grid.Column="2" Content="3" FontSize="24" />
        <Button Grid.Row="3" Grid.Column="3" Content="-" FontSize="24"
Background="LightGray"/>

        <!-- Czwarty rząd -->
        <Button Grid.Row="4" Grid.Column="0" Content="0" FontSize="24" />
        <Button Grid.Row="4" Grid.Column="1" Content="." FontSize="24" />
        <Button Grid.Row="4" Grid.Column="2" Content="=" FontSize="24"
Background="LightGreen"/>
        <Button Grid.Row="4" Grid.Column="3" Content="+" FontSize="24"
Background="LightGray"/>

        <!-- Dodatkowe przyciski -->
        <Button Grid.Row="5" Grid.Column="0" Content="C" FontSize="24"
Background="LightCoral"/>
        <Button Grid.Row="5" Grid.Column="1" Content="+/-" FontSize="24" />
        <Button Grid.Row="5" Grid.Column="2" Content="%" FontSize="24" />
        <Button Grid.Row="5" Grid.Column="3" Content="√" FontSize="24"
Background="LightGray"/>

    </Grid>
```

Wyjaśnienie:

- **Grid.RowDefinitions** i **Grid.ColumnDefinitions** definiują strukturę siatki, gdzie mamy 6 wierszy (pierwszy na ekran kalkulatora, pozostałe na przyciski) oraz 4 kolumny.
- Przyciski kalkulatora (cyfry i operatory) są rozmieszczone w odpowiednich wierszach i kolumnach siatki.
- Każdy przycisk ma ustawiony rozmiar czcionki za pomocą `FontSize`, a przyciski operacyjne (jak "=", "+", "-", itp.) mają lekko wyróżniające się tło.
- **TextBox** służy jako ekran wyświetlający aktualnie wprowadzane liczby i wyniki działań.

Ten layout odzwierciedla podstawowy wygląd kalkulatora w systemie Windows. Możesz teraz uruchomić tę aplikację, a przyciski będą reagowały na interakcje!