

Szkolenie Xamarin 3 / 4

Wprowadzenie do siatkowego układu stron

Grid

Adam Szczerbiak
Witold Goralski
Robert Zimnoch



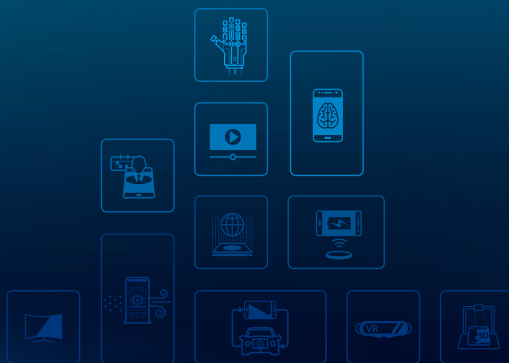
Plan warsztatów

I Wprowadzenie do Xamarin
Pierwsza aplikacja i XAML

II Układy stron i nawigacja

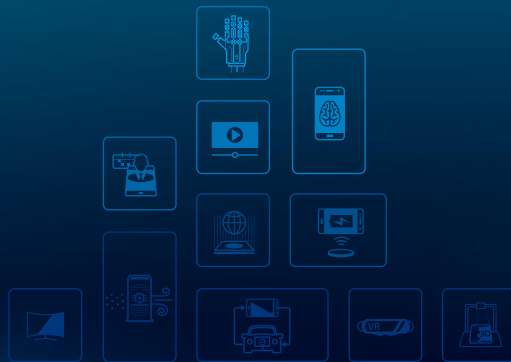
III Układ siatkowy - Grid

IV Zaawansowane układy stron



Poprzednie warsztaty

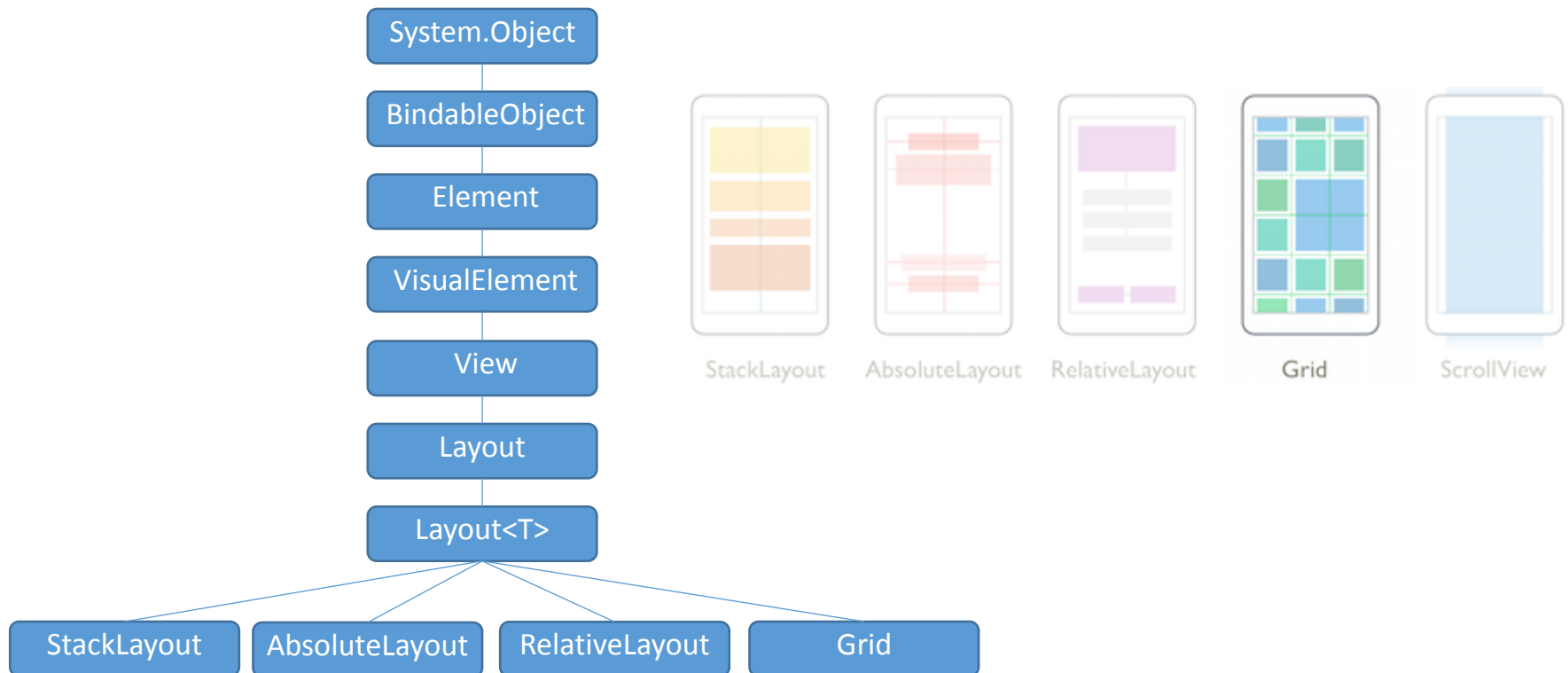
- `AbsoluteLayout`
- `RelativeLayout`
- `NavigationPage`
- `Praca dla chętnych`



Klasa Layout<T>

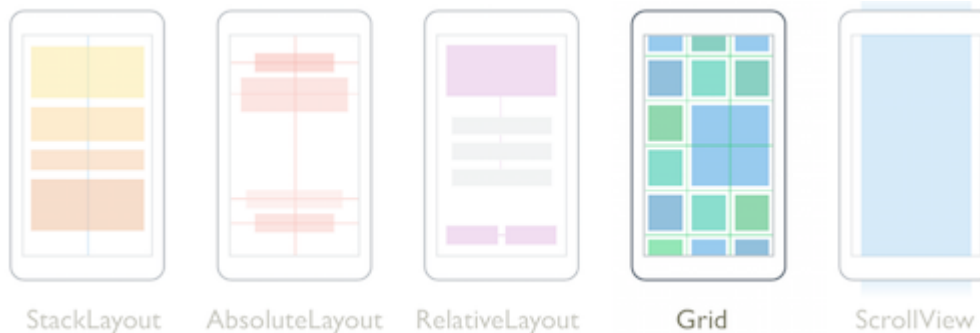


❶ Hierarchia klas StackLayout, AbsoluteLayout, RelativeLayout i Grid



Grid – praktyka użytkowania

- ! Układ przypomina dwuwymiarową tabelę
- ! Rozmiar wierszy i kolumn jest definiowany przez RowDefinitions i ColumnDefinitions
- ! Możliwość zagnieżdżania innych układów w celu dokładnego pozycjonowania konkretnych elementów
- ! Pomocny w projektowaniu interfejsu w rotacji pionowej i poziomej



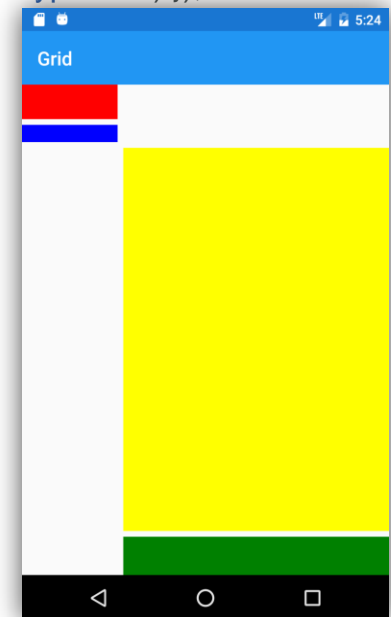
Definiowanie wierszy i kolumn

C#

```
var grid = new Grid();
grid.RowDefinitions.Add (new RowDefinition { Height = new GridLength(2, GridUnitType.Star) });
grid.RowDefinitions.Add (new RowDefinition { Height = new GridLength (1, GridUnitType.Star) });
grid.RowDefinitions.Add (new RowDefinition { Height = new GridLength(400)});
grid.RowDefinitions.Add (new RowDefinition { Height = GridLength.Auto});
grid.ColumnDefinitions.Add (new ColumnDefinition{ Width = new GridLength (100) });
grid.ColumnDefinitions.Add (new ColumnDefinition{ Width = new GridLength (1, GridUnitType.Star) });
```

XAML

```
<Grid>
  <Grid.RowDefinitions>
    <RowDefinition Height="2*" />
    <RowDefinition Height="*" />
    <RowDefinition Height="400" />
    <RowDefinition Height="Auto" />
  </Grid.RowDefinitions>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="100" />
    <ColumnDefinition Width="*" />
  </Grid.ColumnDefinitions>
</Grid>
```



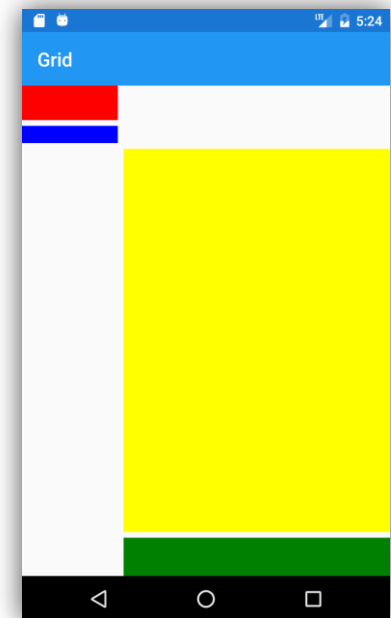
Pozycjonowanie elementów

| C#

```
grid.Children.Add (new BoxView { BackgroundColor = Color.Red }, 0, 0);  
grid.Children.Add (new BoxView { BackgroundColor = Color.Blue }, 0, 1);  
grid.Children.Add (new BoxView { BackgroundColor = Color.Yellow }, 1, 2);  
grid.Children.Add (new BoxView { BackgroundColor = Color.Green }, 1, 3);
```

| XAML

```
<Grid>  
.  
.  
.  
    <BoxView BackgroundColor="Red" Grid.Column="0" Grid.Row="0" />  
    <BoxView BackgroundColor="Blue" Grid.Column="0" Grid.Row="1" />  
    <BoxView BackgroundColor="Yellow" Grid.Column="1" Grid.Row="2" />  
    <BoxView BackgroundColor="Green" Grid.Column="1" Grid.Row="3" />  
</Grid>
```



Grid



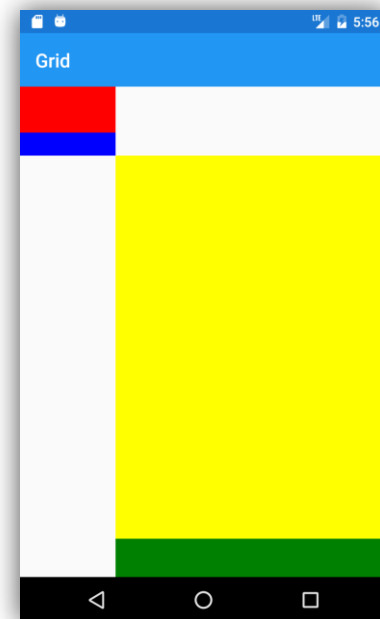
Rozstaw elementów

C#

```
var grid = new Grid { ColumnSpacing = 0, RowSpacing = 0 };
```

XAML

```
<Grid ColumnSpacing="0" RowSpacing="0" >
```



Grid



Rozpiętość komórek

| C#

```
grid.Children.Add(new BoxView { BackgroundColor = Color.Red }, 0, 2, 0, 1);  
grid.Children.Add(new BoxView { BackgroundColor = Color.Blue }, 0, 1);  
grid.Children.Add(new BoxView { BackgroundColor = Color.Yellow }, 1, 2, 1, 3);  
grid.Children.Add(new BoxView { BackgroundColor = Color.Green }, 0, 3);
```

lub

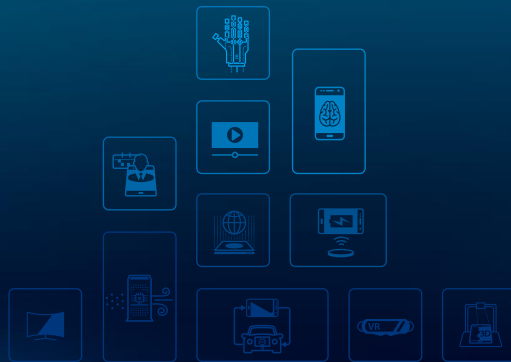
```
Grid.SetColumnSpan(grid.Children[0], 2);  
Grid.SetRowSpan(grid.Children[2], 2);
```

| XAML

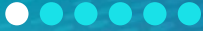
```
<Grid>  
  <BoxView BackgroundColor="Red" Grid.Column="0" Grid.Row="0" Grid.ColumnSpan="2"/>  
  <BoxView BackgroundColor="Blue" Grid.Column="0" Grid.Row="1" />  
  <BoxView BackgroundColor="Yellow" Grid.Column="1" Grid.Row="1" Grid.RowSpan="2" />  
  <BoxView BackgroundColor="Green" Grid.Column="1" Grid.Row="3" />  
</Grid>
```



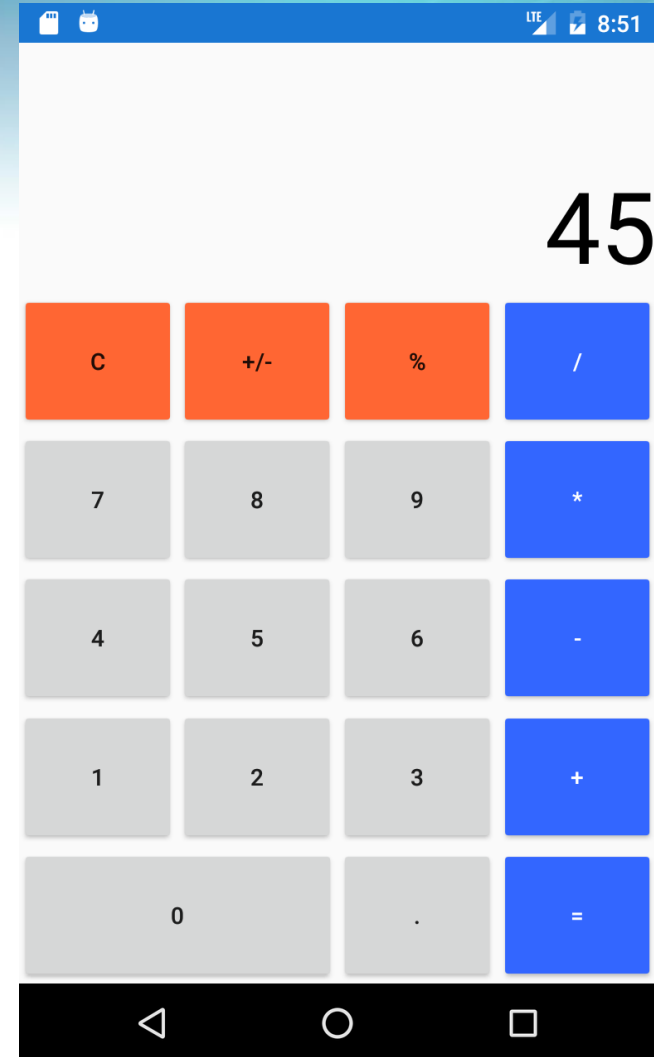
Przykładowa aplikacja Kalkulator



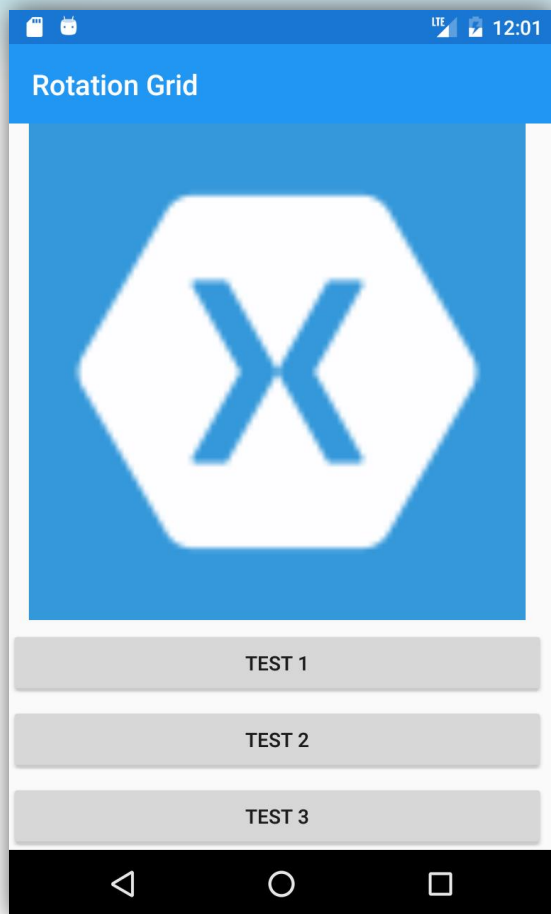
Kalkulator



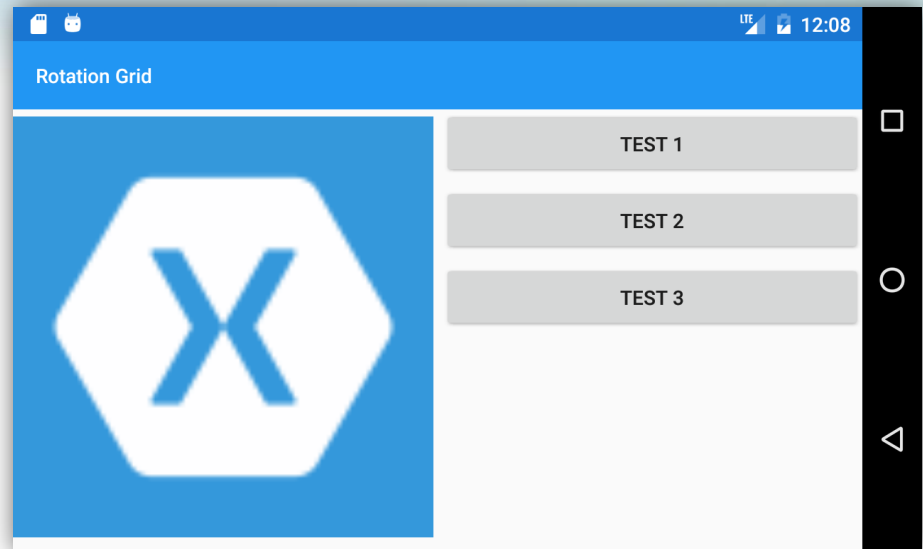
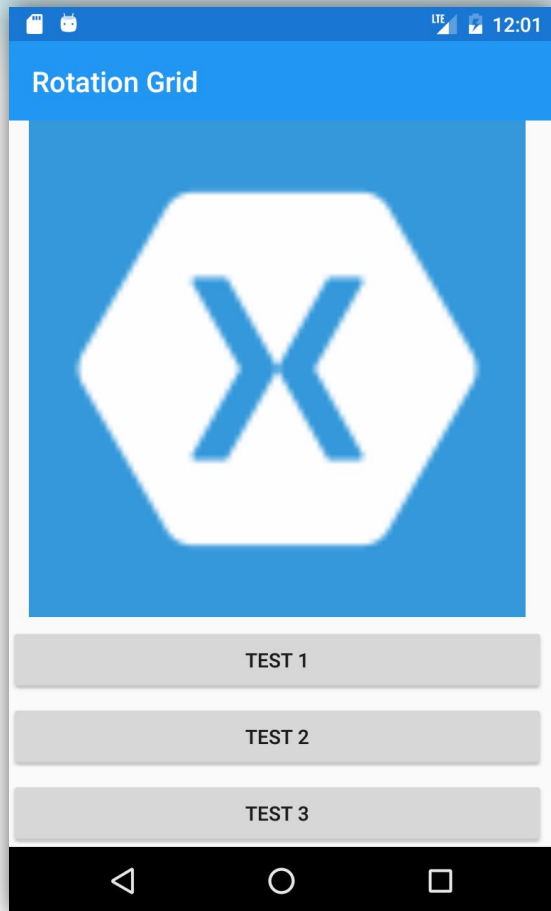
- Układ interfejsu jak na obrazku
- Pełna klawiatura numeryczna
- Przyciski funkcji matematycznych
- Przyciski:
 - czyszczenia
 - dodatnia/ujemna liczba
 - zamiany na procent



Grid



Grid



Rotacja XAML

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms" xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml" x:Class="App1.GridView">
  <ContentPage.Content>
    <Grid x:Name="rotationGrid">
      <Grid.RowDefinitions>
        <RowDefinition Height="*" />
        <RowDefinition Height="Auto" />
      </Grid.RowDefinitions>
      <Grid.ColumnDefinitions>
        <ColumnDefinition Width="*" />
        <ColumnDefinition Width="0" />
      </Grid.ColumnDefinitions>
      <Image Source="icon.png" Grid.Column="0" Grid.Row="0" />
      <StackLayout x:Name="btnStack" Grid.Column="0" Grid.Row="1" >
        <Button Text="Test 1"/>
        <Button Text="Test 2"/>
        <Button Text="Test 3"/>
      </StackLayout>
    </Grid>
  </ContentPage.Content>
</ContentPage>
```

Rotacja C#

```
private double width;
private double height;
protected override void OnSizeAllocated(double width, double height)
{
    base.OnSizeAllocated(width, height);
    if (width != this.width || height != this.height)
    {
        this.width = width;
        this.height = height;
        if (width < height)
        {
            rotationGrid.RowDefinitions[1].Height = GridLength.Auto;
            rotationGrid.ColumnDefinitions[1].Width = new GridLength(0, GridUnitType.Absolute);
            Grid.SetRow(btnStack, 1);
            Grid.SetColumn(btnStack, 0);
        }
        else
        {
            rotationGrid.RowDefinitions[1].Height = new GridLength(0, GridUnitType.Absolute);
            rotationGrid.ColumnDefinitions[1].Width = new GridLength(1, GridUnitType.Star);
            Grid.SetRow(btnStack, 0);
            Grid.SetColumn(btnStack, 1);
        }
    }
}
```

Przykładowa aplikacja Tic Tac Toe



Tic Tac Toe

- Układ interfejsu jak na obrazku
- Licznik wygranych
- Przycisk resetu gry
- Wyświetlanie, kto aktualnie ma ruch
- Interfejs w dwóch orientacjach



Praca dla chętnych



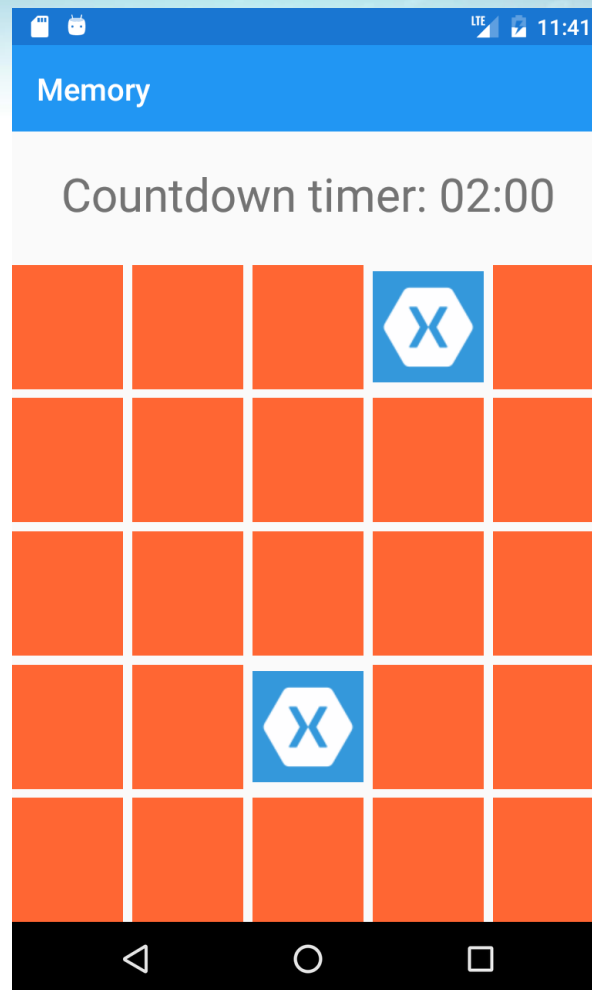
Praca dla chętnych

Tabliczka mnożenia



	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

Gra Memory



Kolejne warsztaty

■ ScrollView

■ ListView



Dziękujemy!