

Wprowadzenie układów stron oraz nawigacji

Adam Szczerbiak
Witold Goralski

AbsoluteLayout RelativeLayout NavigationPage



Plan warsztatów

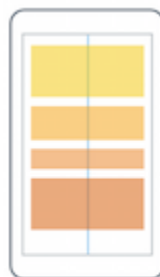
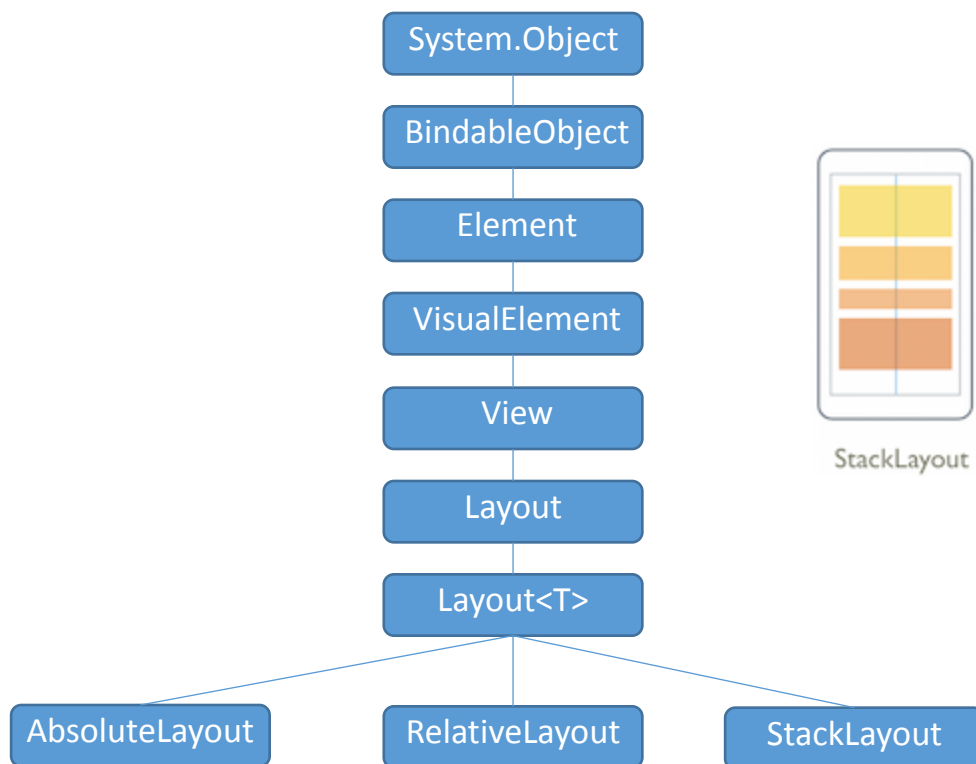
- I Wprowadzenie do Xamarin
Pierwsza aplikacja i XAML
- II Układy stron i nawigacja
- III Układ siatkowy - Grid
- IV Zaawansowane układy stron



Klasa Layout<T>



Położenie klas **AbsoluteLayout**, **RelativeLayout** i **StackLayout** w hierarchii obiektów **Xamarin.Forms**:



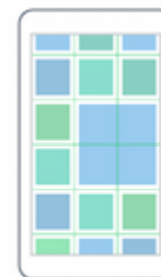
StackLayout



AbsoluteLayout



RelativeLayout



Grid



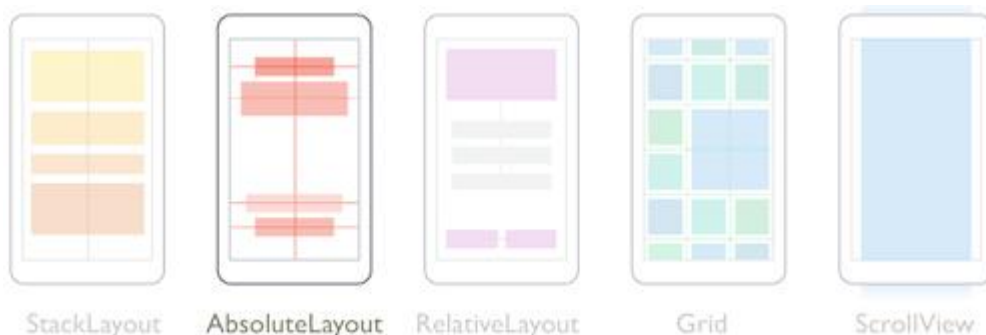
ScrollView

AbsoluteLayout



AbsoluteLayout – praktyka użytkowania

- | Układy obiektów o znanych i niezmiennych wymiarach
- | Układ jest jednoznacznie opisany za pomocą zestawu pozycji
- | Możliwość wykorzystania także wartości proporcjonalnych
- | Możliwość łączenia z innymi układami
w celu dokładnego pozycjonowania konkretnych elementów
- | Wykorzystanie AbsoluteLayoutFlags



StackLayout

AbsoluteLayout

RelativeLayout

Grid

ScrollView

Używanie wartości proporcjonalnych oraz absolutnych

| Wartości proporcjonalne:

```
var label = new Label {Text = "I'm bottom center on every device."};  
AbsoluteLayout.SetLayoutBounds(label, new Rectangle(0.5, 1, 0.5, 0.1));
```

| Wartości absolutne:

```
var label = new Label {Text = "I'm centered on phone but no other device."};  
AbsoluteLayout.SetLayoutBounds(label, new Rectangle(115, 150, 100, 100));
```

AbsoluteLayout



🔄 Wartość proporcjonalna czy absolutna?

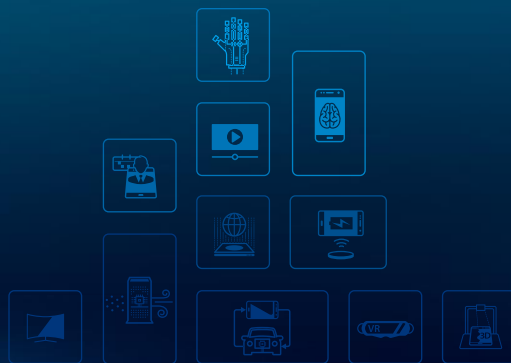
```
AbsoluteLayout.SetLayoutBounds(label, new Rectangle(1, 1, 100, 300));
```

→ Stosowanie **AbsoluteLayoutFlags**:

- ! **None** – wszystkie wartości traktowane są jako absolutne
- ! **All** – wszystkie wartości traktowane są jako proporcjonalne
- ! **WidthProportional** – tylko szerokość traktowana jest jako proporcjonalna
- ! **HeightProportional** – tylko wysokość traktowana jest jako proporcjonalna
- ! **XProportional** – tylko x traktowany jest jako wartość proporcjonalna
- ! **YProportional** – tylko y traktowany jest jako wartość proporcjonalna
- ! **PositionProportional** – tylko x i y traktowane jako proporcjonalne
- ! **SizeProportional** – tylko wysokość i szerokość traktowane jako proporcjonalne

```
var label = new Label {Text = "I'm bottom center on every device."};  
AbsoluteLayout.SetLayoutBounds(label, new Rectangle(0.5, 1, 0.5, 0.1));  
AbsoluteLayout.SetLayoutFlags(label, AbsoluteLayoutFlags.All);
```

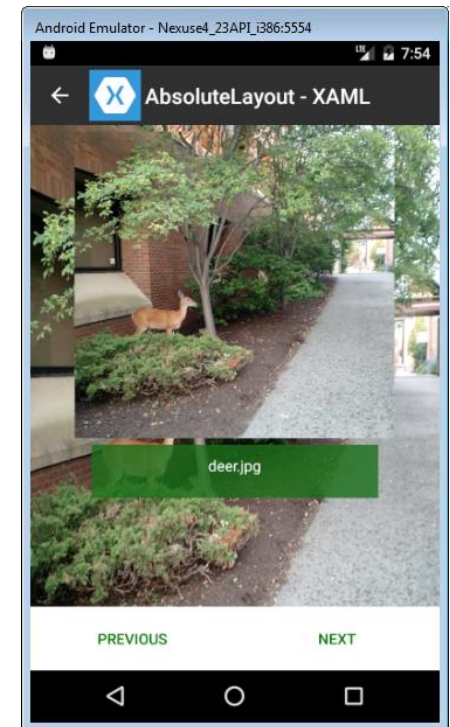
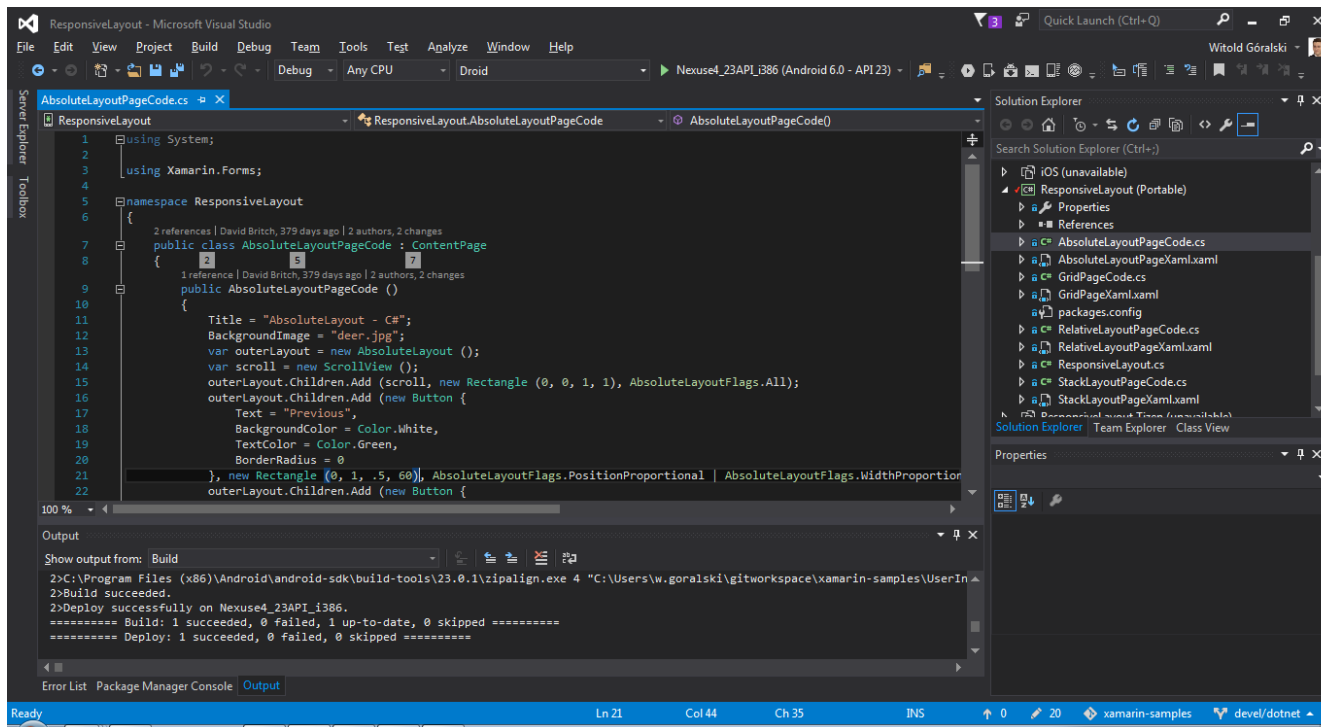
Przykładowa aplikacja ResponsiveLayout



ResponsiveLayout

Analiza aplikacji ResponsiveLayout

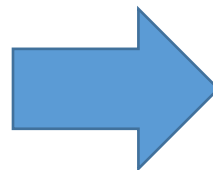
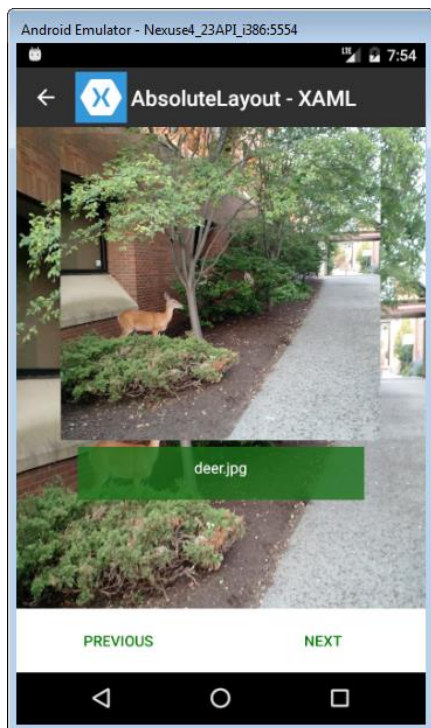
- Wybierz **File > Open > Project/Solution**
- Wejdź do katalogu **UserInterface** w katalogu z przykładami
- Wejdź do katalogu **ResponsiveLayout**
- Wybierz plik **ResponsiveLayout.sln**



Zadanie 3.1

Aplikacja ResponsiveLayout

- Przeanalizuj działanie aplikacji
- Zmień układ aplikacji tak, by odpowiadał on schematowi z obrazka:



XAML

Zadanie 3.1 – rozwiązanie (1/2)



```
<?xml version="1.0" encoding="UTF-8"?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  x:Class="ResponsiveLayout.AltAbsoluteLayoutPageXaml"
  Title="AltAbsoluteLayout - XAML"
  BackgroundImage="ada.jpg">
  <ContentPage.Content>
    <AbsoluteLayout>
      <Button Text="Back" AbsoluteLayout.LayoutBounds="0,0,.5,60"
        AbsoluteLayout.LayoutFlags="PositionProportional,WidthProportional"
        BackgroundColor="White" TextColor="Green"
        BorderRadius="0" />
      <Button Text="Forth" AbsoluteLayout.LayoutBounds="1,0,.5,60"
        AbsoluteLayout.LayoutFlags="PositionProportional,WidthProportional"
        BackgroundColor="White" TextColor="Green"
        BorderRadius="0" />
      <ScrollView AbsoluteLayout.LayoutBounds="0,60,1,1"
        AbsoluteLayout.LayoutFlags="XProportional,SizeProportional">
        <AbsoluteLayout>
          <Image Source="ada.jpg" AbsoluteLayout.LayoutBounds="0.05,0.05,0.45,0.45"
            AbsoluteLayout.LayoutFlags="All" />
          <Image Source="ada.jpg" AbsoluteLayout.LayoutBounds="0.95,0.05,0.45,0.45"
            AbsoluteLayout.LayoutFlags="All" />
          <BoxView Color="#CC1A7019" AbsoluteLayout.LayoutBounds="0.5,0.5,0.7,50"
            AbsoluteLayout.LayoutFlags="PositionProportional,WidthProportional" />
          <Label Text="Ada" AbsoluteLayout.LayoutBounds="0.5,0.5,1,50"
            AbsoluteLayout.LayoutFlags="PositionProportional,WidthProportional"
            HorizontalTextAlignment="Center"
            TextColor="White" />
          <Image Source="ada.jpg" AbsoluteLayout.LayoutBounds="0.05,0.95,0.45,0.45"
            AbsoluteLayout.LayoutFlags="All" />
          <Image Source="ada.jpg" AbsoluteLayout.LayoutBounds="0.95,0.95,0.45,0.45"
            AbsoluteLayout.LayoutFlags="All" />
        </AbsoluteLayout>
      </ScrollView>
    </AbsoluteLayout>
  </ContentPage.Content>
</ContentPage>
```

AbsoluteLayoutPageXaml.xaml

Zadanie 3.1 – rozwiązanie (2/2)



```
<ContentPage.Content>
  <AbsoluteLayout>
    <Button Text="Back" AbsoluteLayout.LayoutBounds="0,0,.5,60"
      AbsoluteLayout.LayoutFlags="PositionProportional,WidthProportional"
      BackgroundColor="White" TextColor="Green"
      BorderRadius="0" />
    <Button Text="Forth" AbsoluteLayout.LayoutBounds="1,0,.5,60"
      AbsoluteLayout.LayoutFlags="PositionProportional,WidthProportional"
      BackgroundColor="White" TextColor="Green"
      BorderRadius="0" />
    <ScrollView AbsoluteLayout.LayoutBounds="0,60,1,1"
      AbsoluteLayout.LayoutFlags="XProportional,SizeProportional">
      <AbsoluteLayout>
        <Image Source="ada.jpg" AbsoluteLayout.LayoutBounds="0.05,0.05,0.45,0.45"
          AbsoluteLayout.LayoutFlags="All" />
        <Image Source="ada.jpg" AbsoluteLayout.LayoutBounds="0.95,0.05,0.45,0.45"
          AbsoluteLayout.LayoutFlags="All" />
        <BoxView Color="#CC1A7019" AbsoluteLayout.LayoutBounds="0.5,0.5,0.7,50"
          AbsoluteLayout.LayoutFlags="PositionProportional,WidthProportional" />
        <Label Text="Ada" AbsoluteLayout.LayoutBounds="0.5,0.5,1,50"
          AbsoluteLayout.LayoutFlags="PositionProportional,WidthProportional"
          HorizontalTextAlignment="Center"
          TextColor="White" />
        <Image Source="ada.jpg" AbsoluteLayout.LayoutBounds="0.05,0.95,0.45,0.45"
          AbsoluteLayout.LayoutFlags="All" />
        <Image Source="ada.jpg" AbsoluteLayout.LayoutBounds="0.95,0.95,0.45,0.45"
          AbsoluteLayout.LayoutFlags="All" />
      </AbsoluteLayout>
    </ScrollView>
  </AbsoluteLayout>
</ContentPage.Content>
</ContentPage>
```

AbsoluteLayoutPageXaml.xaml

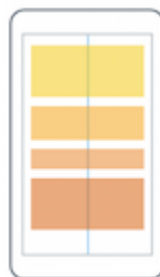
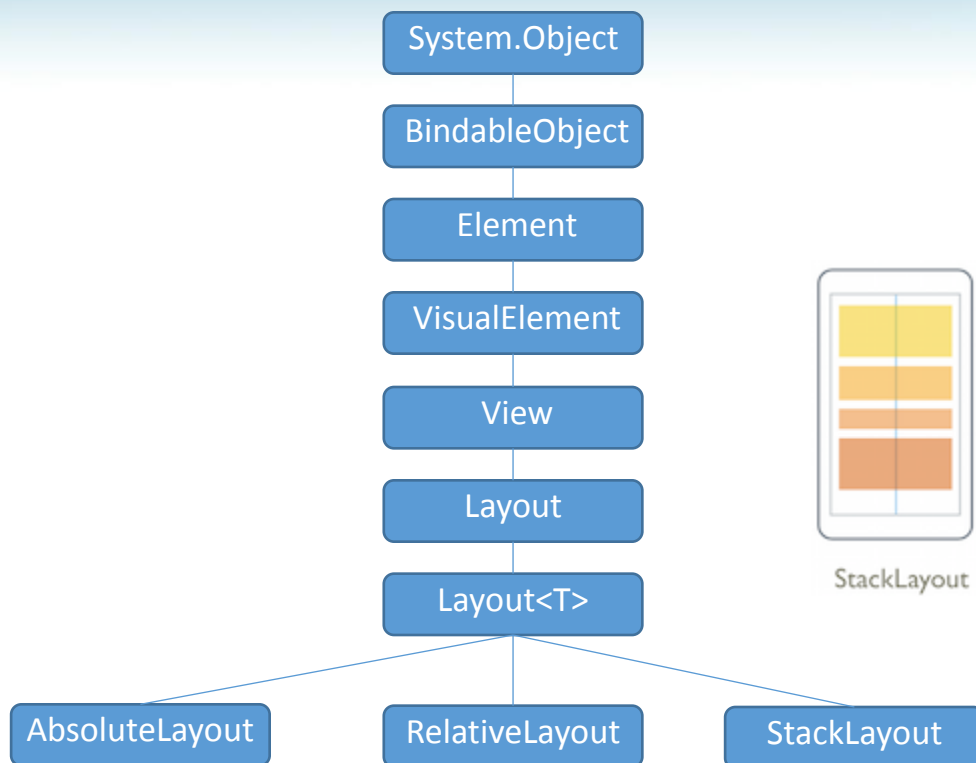
Wprowadzenie układu RelativeLayout Κερίτιλεγαλόντ



Hierarchia Layout<T>



Hierarchia klas **AbsoluteLayout**, **RelativeLayout** i **StackLayout**



StackLayout



AbsoluteLayout



RelativeLayout

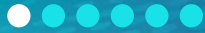


Grid



ScrollView

RelativeLayout

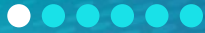


Praktyka stosowania układu RelativeLayout

- Type** – typ relatywny do rodzica lub innego widoku
- Property** – parametr charakterystyczny (jego dotyczyć będzie ograniczenie)
- Factor** – współczynnik skali dla parametru charakterystycznego
- Constant** – wartość przesunięcia (nie zależy od wielkości parametru)
- ElementName** – nazwa widoku do którego odnosi się wiązanie

```
<BoxView Color="Green" WidthRequest="50" HeightRequest="50"  
    RelativeLayout.XConstraint =  
        "{ConstraintExpression  
            Type=RelativeToParent,  
            Property=Width,  
            Factor=0.5,  
            Constant=-100}"  
    RelativeLayout.YConstraint =  
        "{ConstraintExpression  
            Type=RelativeToParent,  
            Property=Height,  
            Factor=0.5,  
            Constant=-100}" />
```

RelativeLayout

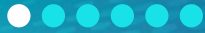


Praktyka stosowania układu RelativeLayout

- Type** – typ relatywny do rodzica lub innego widoku
- Property** – parametr charakterystyczny (jego dotyczyć będzie ograniczenie)
- Factor** – współczynnik skali dla parametru charakterystycznego
- Constant** – wartość przesunięcia (nie zależy od wielkości parametru)
- ElementName** – nazwa widoku do którego odnosi się wiązanie

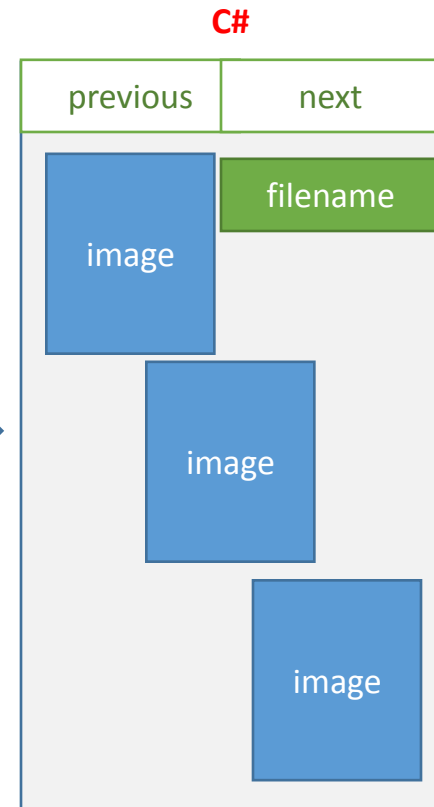
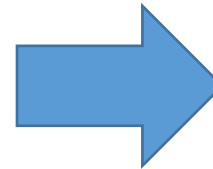
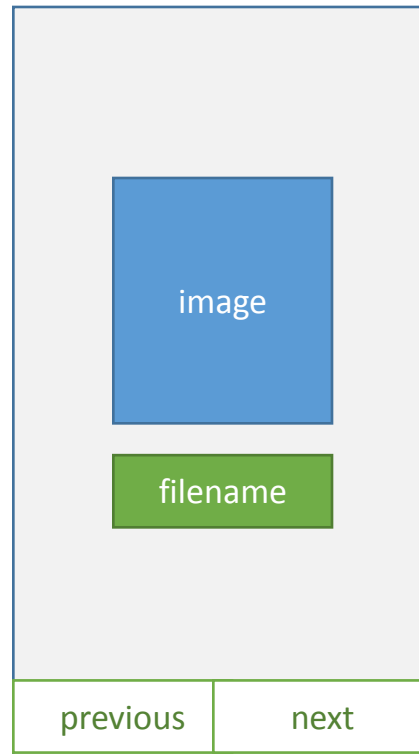
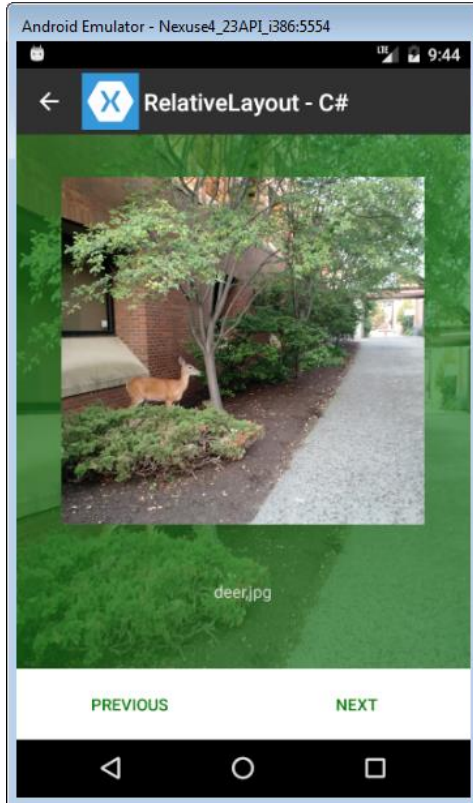
```
layout.Children.Add(box, Constraint.RelativeToParent((parent) =>
{
    return (.5 * parent.Width) - 100;
}),
Constraint.RelativeToParent((parent) =>
{
    return (.5 * parent.Height) - 100;
}),
Constraint.Constant(50), Constraint.Constant(50));
```


ResponsiveLayout



🔄 Aplikacja ResponsiveLayout

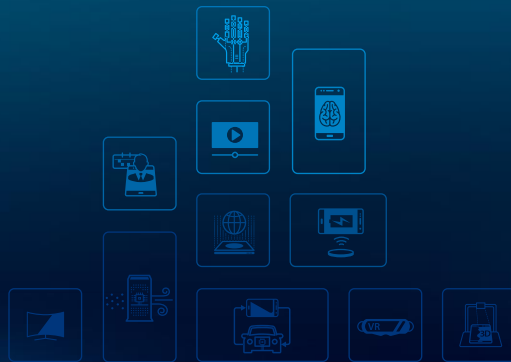
- Przeanalizuj działanie aplikacji
- Zmień układ aplikacji tak, by odpowiadał on schematowi z obrazu poniżej



Wprowadzenie do nawigacji z zastosowaniem komponentu NavigationPage

NavigationPage

zastosowaniem komponentu

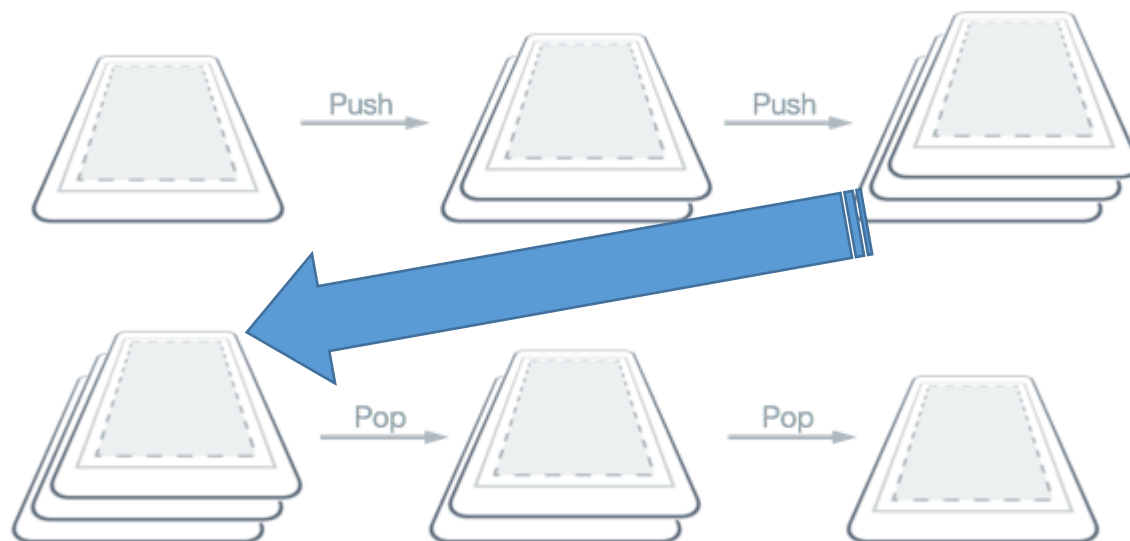


NavigationPage



- ⦿ Działa jak stos widoków (last-in, first-out)

// Page appearance not animated
await **Navigation.PushAsync** (**new** Page2Xaml (), **false**);



// Page appearance not animated
await **Navigation.PopAsync** (**false**);

NavigationPage



Wstawienie widoku pod aktualny widok na stosie

```
Navigation.InsertPageBefore (new MainPage (), this);  
await Navigation.PopAsync ();
```

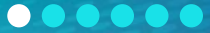


Usunięcie dowolnego widoku ze stosu

```
Navigation.RemovePage (Page removeThisPage);
```

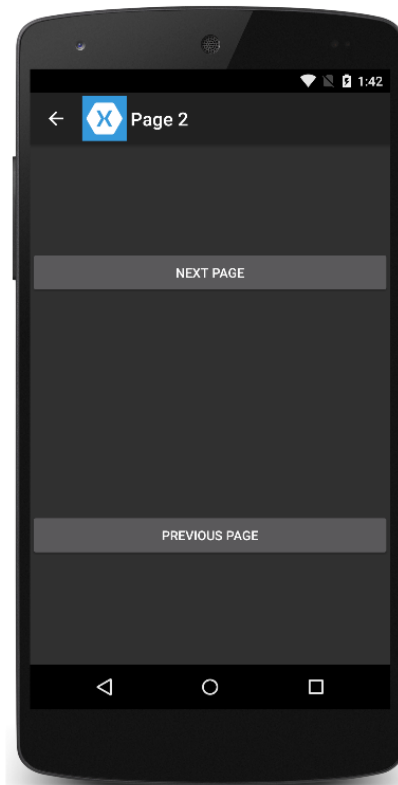


Hierarchical Navigation



Aplikacja Hierarchical

| Przeanalizuj działanie aplikacji



Dziękujemy!