

Hand-out

Programmieren

Periode 2

01. Formulieren

1. Inleiding

In deze handout gaan we beginnen met formulieren. Tot nu toe hebben we alleen gewerkt in de console application en dit was om er voor te zorgen dat we ons alleen maar hoefden te focussen op het schrijven van de code.

Bij formulieren is het namelijk zo dat we niet alleen te maken hebben met programma code, maar ook met een visuele presentatie van een formulier. Deze visuele presentatie noemen we ook wel de GUI en staat voor Graphical User Interface. Deze GUI wordt opgemaakt aan de hand van markup language (ML), in dit geval XAML.

Binnen het Microsoft .NET framework kennen we drie manieren om formulieren te maken.

1. Winforms
2. WPF (Windows Presentation Foundation)
3. UWP (Universal Windows Platform)

Wij gaan werken met WPF omdat dit momenteel de industrie standard is. Winforms is enigszins deprecated geworden en UWP is nog te nieuw en ontworpen om op windows 10 te draaien.

In deze hand-out ga leren wat WPF inhoudt en hoe de techniek in elkaar zit.

2. XAML

XAML staat voor Extensible Application Markup Language en word uitgesproken als "zammel". XAML is een opmaaktaal en geen programmeer taal. Het word gebruikt om het formulier op te maken in een mens-leesbaar document en kan door de programmeur aangepast en uitgebreid worden.



```
<?xml version="1.0" encoding="utf-8" ?>
<Window x:Class="WPF_Test.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:WPF_Test"
        mc:Ignorable="d"
        Title="MainWindow" Height="350" Width="525">
    <Grid>
        <Button Content="Button" HorizontalAlignment="Left" Margin="241,163,0,0" VerticalAlignment="Top" Width="75"/>
    </Grid>
</Window>
```

Zoals je in het voorbeeld kunt zien begint het XAML document met een Window tag. Tags zijn woorden die tussen het < en > teken staan en eigenschappen hebben. HTML XML en XAML zijn markup talen die gebruik maken van tags.

Vervolgens zien we binnen de window tag een Grid tag en in de grid tag hebben we een Button tag. Je zou dus ook kunnen zeggen Window.Grid.Button.

In de volgende les gaan we dieper in op dit onderwerp.

3. Code-behind

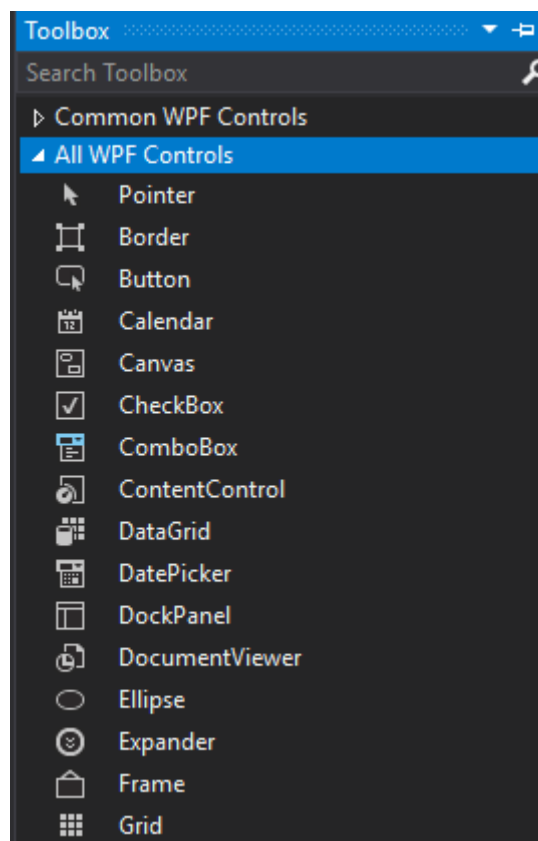
Code-behind betekent, de code achter de XAML. Code kennen we inmiddels want daar hebben we de hele eerste periode mee gewerkt, XAML is nieuw en het mooiste is dat we de XAML kunnen aansturen met code, vandaar code-behind. Code-behind is niets meer dan een .cs bestand zoals we die ook al kennen uit de eerste periode.

Ieder XAML document kent zijn eigen .cs bestand, dit bestand is voor dat XAML bestand de code-behind. Als we een nieuw project openen krijgen we een XAML bestand met een .cs bestand er bij. Net als in een console applicatie heeft deze structuur een basis, kent het een constructor en weet de interpreter waar hij het programma moet uitvoeren.

Als we bijvoorbeeld de button tag uit het vorige onderwerp voorzien van een "name" eigenschap dan kunnen we in de code-behind de waarde die we hier in zetten gebruiken om de button programmatisch te behandelen en het programma dus interactief te maken.

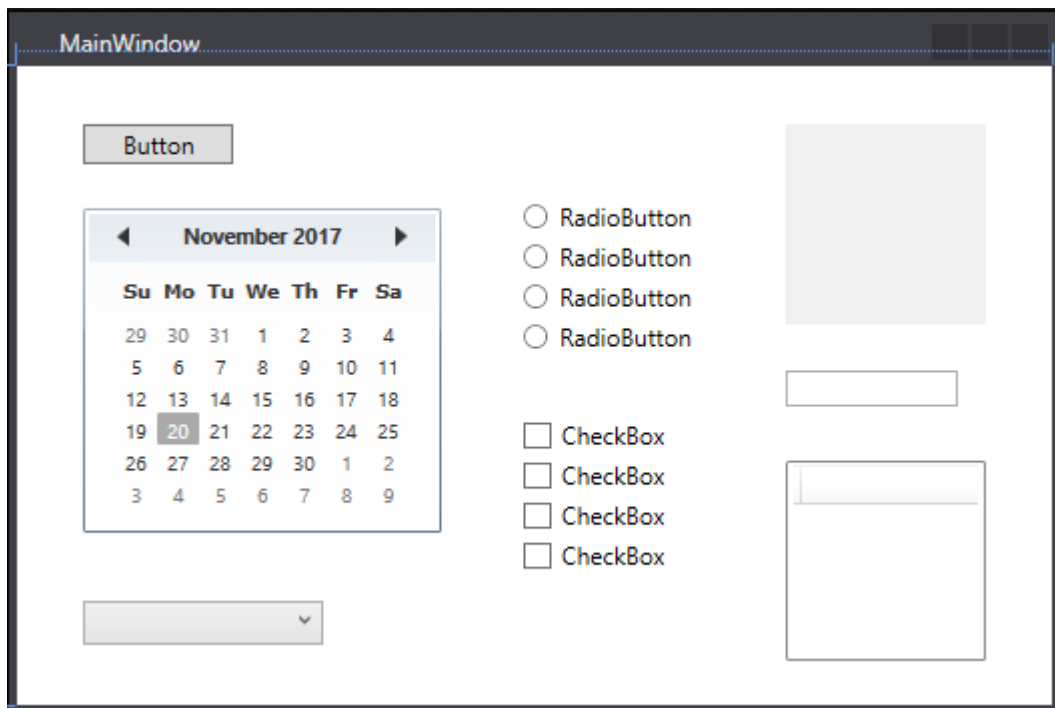
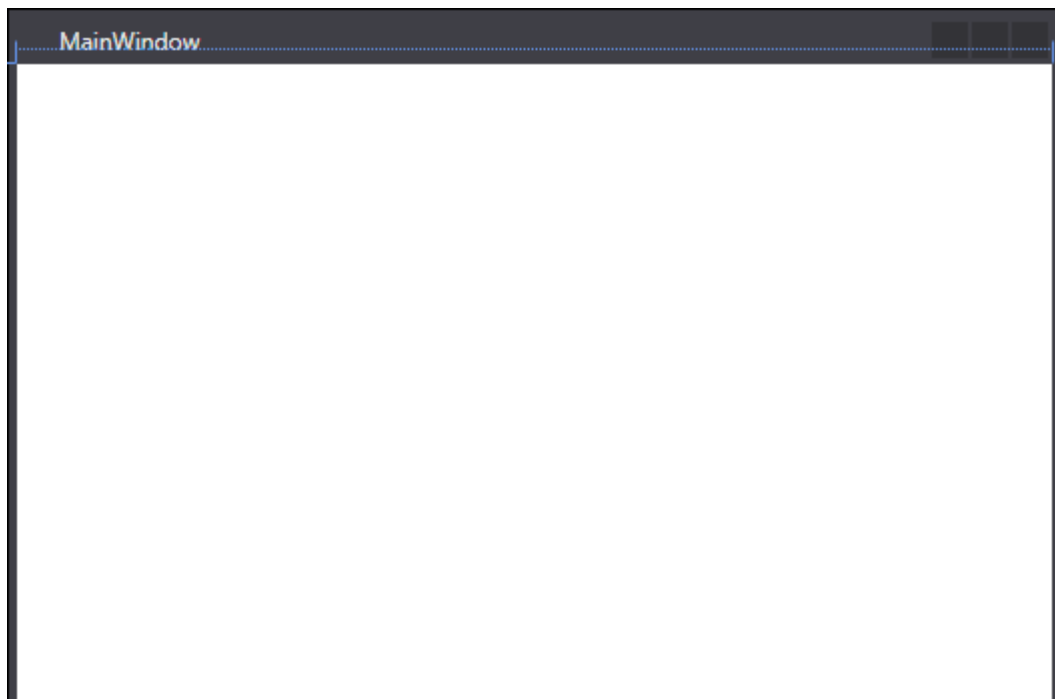
4. Toolbox

De Toolbox kun je vinden aan de linkerkant van Visual Studio en bevat een lijst met alle Tools die je kunt gebruiken wanneer je gebruik maakt van WPF. Daaronder vallen ook de Controls. Controls zijn onderdelen waarmee je een formulier of programma opmaakt, aanstuurt of bediend.



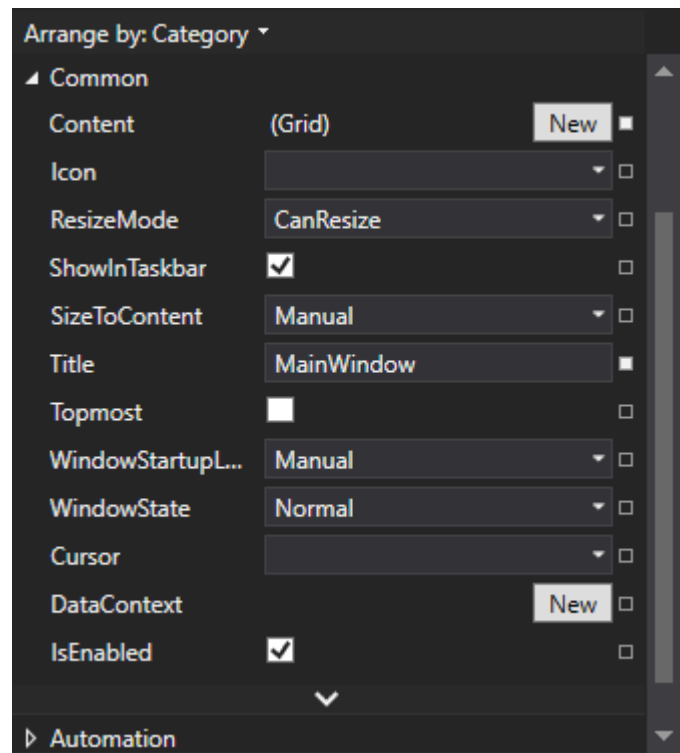
5. Designer

De Designer is een visuele weergave van de XAML en de code-behind samen. De Designer is een WYSIWYG editor wat staat voor What You See Is What You Get. Dit betekent dat ik vanuit de Toolbox tools of controls kan slepen naar de Designer, dit noemen we Drag & Drop. Vervolgens kunnen we de controls met de muis naar de juiste plek op het scherm slepen. De Designer is ook wel een live weergave van de GUI.



6. Properties

De properties, of ook wel “eigenschappen” zijn binnen visual studio rechts onder in beschikbaar, onder de solution explorer. In deze ruimte kun je waarden invoeren per GUI element, maar later zal je er al snel achter komen dat dit veel makkelijker is om direct in de XAML of in de code-behind te doen. In deze les gaan we nog niet veel doen met de properties en dus gaan we hier ook niet te diep op in, makkelijk is te weten dat ze bestaan en waar je ze kunt vinden.



7. Demo

In dit stuk gaan we klassikaal een demo maken. Het is de bedoeling dat je mee kijkt en zo stapsgewijs houvast krijgt op het onderwerp, maak daar waar nodig is aantekeningen in je schrift of in je kladblok. Als je klaar bent ga je zelf aan de slag dus let goed op!

8. Opdracht

In deze opdracht ga je een nieuw project beginnen, dit moet een WPF project zijn en binnen dit project ga je een formulier maken. Het maakt hier niet uit wat voor formulier, het doel is om te kijken hoe ver je kunt komen door zelf op onderzoek uit te gaan. Als je vragen hebt of er niet uit komt laat dit dan even weten aan de docent en vraag om hulp.