Pie Menu

* Pie Menu.cs
  + DependencyProperties: Werte können hierdurch im XAML-PieMenu-Element verwendet werden
  + DrawMenu(…): Zeichnet den Kreis letztendlich
    - Bestimmt full\_sector
    - Berechnet Innen- und Außenradius, GapAngle, Zentrum
    - **Erstelle PathFigure und füge Segmente hinzu**
    - **Erstelle PathGeometry und füge PathFigure hinzu**
    - Setzte Background + Border Brush
    - DrawGeometry
    - Erstelle Formatted Text und setze dessen Punkt und Drehwinkel (RotateTransform)
    - Setze
* PieMenuItem.cs
  + DependencyProperties: Werte können hierdurch im XAML-PieMenuItem-Element verwendet werden

# ToDo

* Binding noch anders umsetzen? Zwischenzeitige Behandlung der ObservableCollections als List verhindert?

**Solved Problems**

* Problem, dass Variablen des PieMenu nicht duch Binding aktualisiert werden, durch Code-Behind-Werte-Aktualisierung gelöst
* Problem: Binding wird erst nach OnRender des PieMenu aktualisiert. Wie kann OnRender nach erfolgreichem Binding erneut ausgeführt werden? -> pieMenu.InvalidateVisual
* Binding
  + Unterschiedliche Sources haben Binding erschwert. Mit dynamicResource konnten für jeden TagView und jedes PieMenuItem eigene Sources definiert werden
  + Mit <h:BindingProxy x:Key="proxy" Data="{Binding}" /> als Resource und Verwendung durch {Binding Data.BindingTest1, Source={StaticResource proxy}} können die Werte gebindet werden. Allerdings werden diese erst nach Auslösen eines Commands sichtbar. Und auch nur wenn der Header richtig gebindet ist. Falls dieser null ist wird das Binding nicht aktualisiert:
* Initialisierungsverlauf:
  + Start der Anwendung: SV.xaml.cs -> SVM -> STV.xaml.cs -> STVM -> STV.xaml.cs -> STVM ->
  + Tag-Platzierung: STV.xaml.cs -> PieMenuItem SubHeader get (für alle Verwendungen) ->