Verslag Mandelbrot

Logic

```
public int[,] MandelbrotFractal(int maxRow, int maxColumn, int iterations, double zoom, double offsetX, double offsetY, CancellationToken tokenSource)
```

Deze functie berekent de iteratiepunten van het Mandelbrot-fractaal eerst worden x en y geschaald naar (-2, 2) waarna dan de formule wordt gebruikt om te zien hoeveel iteraties er nodig zijn om een straal te krijgen die groter is dan 2. Als dit bereikt is stopt de while loop en wordt dat getal toegevoegd in de array. Als eindresultaat krijg je een tweedimensionale array waarbij de plaats van de iteratiewaarde gelijk is aan een coördinaat van de bitmap.

```
public DoublePoint Scaling(int X, int Y, int maxRow, int maxColumn, double zoom, double offsetX, double offsetY)
```

Deze functie schaalt een punt van de bitmap naar een punt op de Mandelbrot-fractaal.

```
public int[,] GreyScale(int maxRow, int maxColumn, int[,] mandelPoints, int iteration, CancellationToken tokenSource)
```

Deze functie zet een array van iteratiepunten om naar de juiste kleur. Hierbij wordt de waarde geschaald naar 255 en wordt er zo de grijswaarde berekend.

```
public int[,] Banding(int maxRow, int maxColumn, int[,] mandelPoints, CancellationToken tokenSource)
```

Deze functie zet een array van iteratiepunten om naar de juiste kleur. Hierbij worden de even waarden op wit gezet en de oneven op zwart.

```
public int[,] UglyBanding(int maxRow, int maxColumn, int[,] mandelPoints, CancellationToken tokenSource)
```

Deze functie zet een array van iteratiepunten om naar de juiste kleur. Dit is hetzelfde principe als banding maar worden er meerdere kleuren toegevoegd en in plaats van oneven of even wordt er gekeken of het deelbaar is door 4, 3, 2 of geen van die.

```
public int[,] Colors(int maxRow, int maxColumn, int[,] mandelPoints, int iteration, CancellationToken tokenSource)
```

Deze functie zet een array van iteratiepunten om naar de juiste kleur. Hier worden de waarden ook geschaald naar 255 maar in plaats van gewoon de grijswaarde te pakken wordt er de modulus gepakt van een bepaald getal en dit wordt dan vermenigvuldigd om als maximumwaarde weer 255 te krijgen. Hierbij krijg je het mooiste effect.

```
public class DoublePoint
```

Deze class wordt gebruikt om doubles te kunnen gebruiken voor een Point.

MainViewModel

```
private void CreateBitmap()
```

Bij deze functie wordt de Bitmap aangemaakt met de juiste grootte.

```
private async void Resize(double[] size)
```

Deze functie wordt opgeroepen als het venster een nieuwe grootte krijgt. Deze krijgt de grootte dan mee en roept CreateBitmap() terug op om een nieuwe Bitmap te maken met de nieuwe grootte.

```
private async Task SetPixels(CancellationTokenSource tokenSource)
```

Deze functie roept een functie op om de iteratiearray om te zetten naar een met de kleurwaarden in en gebruikt dan WritePixels om deze in de bitmap te zetten.

```
private async Task DrawMandel()
```

Deze functie maakt eerst een nieuwe CancellationTokenSource aan om alle vorige Tokens te kunnen cancelen als deze functie opnieuw wordt aangeroepen. Deze functie start een Task die MandelBrotFractal oproept om de iteratiearray te krijgen en timet deze ook. Hierna wordt SetPixels opgergoepen.

```
private async void ZoomInMandel() en private async void ZoomOutMandel()
```

Als er omhoog gescrold wordt gaat ZoomInMandel opgeroepen worden, deze functie gaat de Zoomfactor * 2 doen. Als er omlaag wordt gescrold dan gaat ZoomOutMandel opgeroepen worden en wordt de Zoomfactor / 2 gedaan. DrawMandel wordt op het einde opgeroepen om hem dan opnieuw te tekenen.

```
private void MouseChanged(Point point)
```

Wanneer de muis beweegt wordt deze functie opgeroepen om de positie van de muis en het aantal iteraties eronder weer te geven.

```
private async void ResetMandel()
```

Deze functie Zet de offSets terug op 0 en de Zoom op 1. DrawMandel wordt op het einde opgeroepen om hem opnieuw te tekenen.

```
private async void Panning(Point moved)
```

Met deze functie wordt de offset afgetrokken met de afstand die de muis al slepend heeft afgelegd. Deze wordt dan nog gedeeld door de grootte en de zoomfactor om niet direct aan de andere kant te zitten. DrawMandel wordt op het einde opgeroepen om hem opnieuw te tekenen.

MainWindow

```
private void Scroll(object sender, MouseWheelEventArgs e)
```

Deze functie wordt opgeroepen wanneer er wordt gescrold. Als e.Delta positief is wordt de ZoomInCommand opgevraagd en als er e.Delta negatief is ZoomOutCommand.

```
private void WindowPreviesMouseMove(object sender, MouseEventArgs e)
```

Wanneer de muis beweegt wordt de positie opgeslagen en doorgestuurd in het MouseChangedCommand om de nieuwe coördinaten te krijgen. Als de linkermuisknop ingedrukt wordt en er gesleept wordt dan gaat het PanningCommand ook uitgevoerd. Deze krijgt mee hoeveel pixels de muis bewogen heeft.

```
private void ChangeSize(object sender, SizeChangedEventArgs e)
```

Als het venster vergroot of verkleint wordt gaat het ResizeCommand uitgevoerd worden. Deze krijgt de nieuwe grootte mee.