

Mandelbrotgenerator

Ing. Thomas Herzog

Version 1.0, April 29 2016

Folgendes Dokument stellt die Dokumentation für die dritte Übung dar. Diese Übung ist in drei Teile unterteilt:

- [Sync Generator](#)
- [Async Generator Version 1](#)
- [Async Generator Version 2](#)
- [Parallel Generator](#)

Sync Generator

Dieser Abschnitt beschäftigt sich mit der synchronen Implementierung. In dieser Version wird das *Image* synchron erstellt. Damit diese Implementierung wiederverwendet werden kann, wurde die synchrone Implementierung mit folgender Methodenspezifikation implementiert.

```
Bitmap GenerateImage(int startIdx, int endIndex, Area area, Func<bool> cancel);
```

Methodenparameter:

- **startIdx** (Der Startindex ab welcher Spalte das Bild generiert werden soll)
- **endIndex** (Der Endindex bis zu welcher Spalte das Bild generiert werden soll)
- **area** (Die *Area* Instanz mit den anderen Parametern)
- **cancel** (Die *Cancel*-Funktion, über die abgebrochen werden kann)

Daher ist die Implementierung einerseits synchron und andererseits auch asynchron zu verwendbar, da durch die Angabe des `startIdx` und `endIdx`, die Berechnung der Pixels auf einen Teil des Bildes beschränkt werden kann.

Siehe [SyncImageGenerator.cs](#)

Async Generator Version 1

Dieser Abschnitt beschäftigt sich mit der asynchronen Implementierung in der Version 1. In dieser Version wurde ein *Thread* verwendet, der die Generation des Bildes asynchron abarbeitet.

Siehe [AsyncGeneratorV1.cs](#)

Async Generator Version 2

Dieser Abschnitt beschäftigt sich mit der asynchronen Implementierung in der Version 2. In dieser Version wurde ein *BackgroundWorker* verwendet, der die Generation des Bildes asynchron abarbeitet.

Siehe [AsyncGeneratorV2.cs](#)

Parallel Generator

Dieser Abschnitt beschäftigt sich mit der parallelen Implementierung. In dieser Version werden *n-BackgroundWorker* verwendet, welche die Teile des Bildes parallel und asynchron berechnen. Hierbei wird die Arbeit der Bildgeneration auf die *BackgroundWorker* aufgeteilt und abschließend zu einem Gesamtbild zusammengefügt. Dabei wird die *area.Height* auf die *BackgroundWorker* aufgeteilt. Nachdem jeder *Worker* fertig ist, wird sein Teil des generierten Bildes in das Gesamtbild aufgenommen.

Siehe [ParallelGenerator.cs](#)

Tests

Folgender Abschnitt beschäftigt sich mit den Tests des Mandelbrotgenerators.

Folgende Tests wurden mit einem *Build* der Anwendung durchgeführt, welcher mit einer *Release*-Konfiguration und für eine x64 Architektur erstellt wurde.

Table 1. Synchron/Parallel Generator

| Durchlauf | Synchron | Parallel |
|-----------|----------|----------|
| 1 | 2.49 | 1.13 |
| 2 | 2.54 | 1.08 |
| 3 | 2.60 | 1.17 |
| 4 | 2.60 | 1.35 |
| 5 | 2.55 | 1.10 |
| 6 | 2.63 | 1.16 |
| 7 | 2.55 | 1.02 |
| 8 | 2.54 | 1.03 |
| 9 | 2.56 | 1.10 |
| 10 | 2.55 | 1.02 |

Synchroner Generator:

Mean time: 2.561 sec

Std deviation: 0.00156556 sec

Paralleler Generator:

Mean time: 1.116 sec

Std deviation: 1.25169956 sec