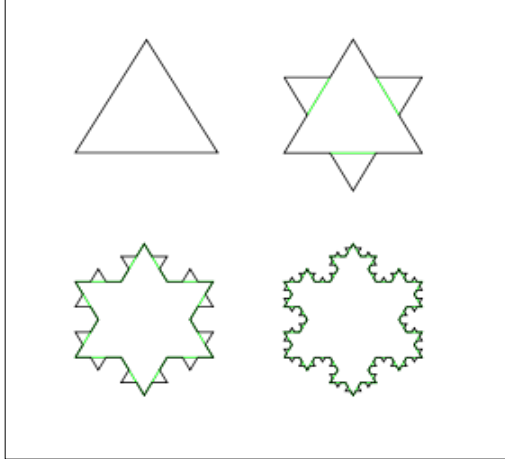


Opdracht JSF31, week 4

We gaan een Java FX applicatie maken die een eenvoudige fractal tekent, de zogenaamde Koch fractal. Het tekenen van deze fractal begint met een driehoek, waarbij elke zijde vervolgens wordt opgedeeld in 4 lijnstukken. Dit proces kan willekeurig vaak herhaald worden, waarbij er steeds fijnere verdelingen gemaakt worden. Dit zijn de verschillende levels van de fractal. Als voorbeeld zijn hieronder de levels 1 t/m 4 weergegeven.



Voor meer info over dit type fractal, zie <http://nl.wikipedia.org/wiki/Koch-kromme>.

Doelstellingen

- ervaring opdoen met het maken van een GUI in Java FX
- toepassen Observer/Observable patroon
- tijd meten
- ervaren dat een langdurige berekening de GUI kan blokkeren

Voor deze applicatie zijn een aantal klassen gegeven:

1. `JSF31KochFractalFX` (Java FX applicatie)
2. `KochFractal` (klasse om de Koch fractal te berekenen)
3. `Edge` (één lijnstuk van de Koch fractal)

Deze klassen hoeven in principe niet bekeken en niet veranderd te worden.

Er is nog een andere klasse (`KochManager`) die je zelf moet schrijven.

Op de laatste 2 pagina's van deze opdracht zijn beschrijvingen van de klassen `KochFractal` en de Java FX applicatie `JSF31KochFractalFX` te vinden.

Opdracht

We gaan achtereenvolgens de klassen `KochFractal`, `Edge` en `JSF31KochFractalFX` gebruiken. Deze klassen zijn te vinden in het `JSF31KochFractalFX` – startup project op SharePoint.

1. Maak een console applicatie in NetBeans en kopieer het package `calculate` vanuit het `KochFractal` – startup project hierin. We gaan nu de coördinaten van alle edges van de fractal van level 1 uitrekenen en in het console tonen. Doe dit door in de `main` methode:

- het `KochFractal` object te maken,
- het level op 1 in te stellen en tenslotte
- de 3 `generate...Edge` methodes één voor één aan te roepen.

Als we het hierbij zouden laten, zouden we geen resultaat zien omdat er geen Observer is voor de `KochFractal` Observable.

Voeg een Observer toe die de edge-coördinaten print. Dit kan in principe op 2 manieren:

- Maak een nieuwe klasse die de interface `Observer` implementeert, implementeer daarin de methode `update`, zodanig dat deze het begin- en eindpunt van een edge afdruckt (zorg voor een duidelijke begeleidende tekst in de output). De betreffende edge is te vinden in de tweede parameter (`arg`) van `update`. Deze kan bijvoorbeeld als volgt gebruikt worden:

```
Edge e = (Edge)arg;
```

Maak in de `main` methode een object van deze klasse en voeg dat als Observer toe aan het `KochFractal` object.

- Of maak van de al bestaande klasse (waar ook de `main` methode in zit) een Observer door deze de interface `Observer` te laten implementeren, en door de methode `update` aan deze klasse toe te voegen.

Maak in de `main` methode een object van deze klasse en voeg dat als Observer toe aan het `KochFractal` object.

2. Als het goed is worden nu de coördinaten van 3 edges afgedrukt omdat we het level hadden ingesteld op 1, hetgeen een driehoek voorstelt. De eind-coördinaat van elke edge is gelijk aan de begin-coördinaat van de volgende edge.
Stel het level vervolgens in op 2. Er moeten dan 12 edges geprint worden, waarbij ook weer de eind-coördinaat van elke edge gelijk is aan de begin-coördinaat van de volgende edge. Merk op dat alle geprinte coördinaten in de range `[0..1]` liggen.

3. We gaan nu verder met de Java FX applicatie in het project `JSF31KochFractalFX`. Creeer eerst een `KochFractal` object en voeg een Observer toe. Maak een nieuwe klasse `KochManager` in het package `calculate`. De constructor van deze klasse wordt aangeroepen in `JSF31KochFractalFX` met een referentie naar de Java FX applicatie zelf. De klasse `KochManager` heeft dus een private attribuut `application` dat een waarde krijgt in de constructor:

```
private JSF31KochFractalFX application;  
public KochManager(JSF31KochFractalFX application) {  
    this.application = application;  
}
```

Vanuit de klasse JSF31KochFractalFX worden de volgende twee methoden aangeroepen:

```
public void changeLevel(int nxt)
public void drawEdges()
```

In de methode `changeLevel()` wordt het level van de Koch fractal aangepast en worden de edges getekend:

```
public void changeLevel(int nxt) {
    koch.setLevel(nxt);
    drawEdges();
}
```

In de methode `drawEdges()` wordt het panel gecleared en worden de edges gegenereerd en getekend:

```
public void drawEdges() {
    application.clearKochPanel();
    koch.generateLeftEdge();
    koch.generateBottomEdge();
    koch.generateRightEdge();
}
```

Net als in de Console-applicatie moet `KochManager` het `Observer` interface implementeren. Het daadwerkelijk tekenen van een edge kan dan gedaan worden door de methode `update()`:

```
public void update (Observable o, Object arg) {
    application.drawEdge((Edge)arg);
}
```

Test nu je applicatie. Als het goed is kun je het level van de Koch fractal verhogen en verlagen. Probeer ook of je de Koch fractal kunt verplaatsen (slepen met linkermuisknop), de fractal kunt vergroten (linker muisknop) en de fractal kunt verkleinen (rechter muisknop).

4. Definieer een `ArrayList` als private attribuut van de klasse `KochManager`. In methode `update` moet nu telkens de edge toegevoegd worden aan deze `ArrayList`. Als je dit goed doet zijn, nadat de 3 `generate...Edge` methodes klaar zijn, alle edges berekend en opgeslagen in de `ArrayList`. Gebruik een `for`-statement om alle edges uit de `ArrayList` te tekenen in methode `drawEdges()`.
5. Gebruik de `TimeStamp` klasse van week 2 om steeds de tijd te bepalen die nodig is om de fractal te tekenen (bepaal dus hoe lang `drawEdges()` duurt). Geef deze tijd weer door een aanroep van `application.setTextCalc()`.
6. Geef ook het aantal edges van de fractal weer door een aanroep van `application.setTextNrEdges()`. Klasse `KochFractal` heeft een methode

om dit aantal op te vragen. Bewaar deze versie van het project.

7. Run het programma voor steeds hogere levels en maak een tabel waarin voor oplopende levels de benodigde tijd en het aantal edges weergegeven wordt.
Bij een bepaald level zul je een heap-overflow melding krijgen omdat de `ArrayList` dan te groot wordt voor het beschikbare geheugen. Het kan ook gebeuren dat de berekeningen extreem lang gaan duren, dus stop als dat meer dan 30 seconden is.
8. Zoals we het nu geïmplementeerd hebben wordt een fractal iedere keer opnieuw berekend als `drawEdges()` wordt aangeroepen, dus als er gezoomd wordt, als de fractal verplaatst wordt, en als het level veranderd wordt. In principe doen we dan te veel, want als een fractal gezoomd of verplaatst wordt, veranderen de edges niet. We hoeven eigenlijk alleen de edges opnieuw te berekenen bij het veranderen van level. Verplaats daarom de aanroepen van de 3 `generate...Edge` methodes naar `changeLevel()`. Zorg dat de fractal opnieuw getekend wordt als de edges berekend zijn. Test weer het zoomen, verplaatsen en veranderen van level.
9. Bepaal nu apart de tijd die nodig is om de fractal te *berekenen* (de aanroep van de 3 `generate...Edge` methodes) en om de fractal te *tekenen* (het `for`-statement). Geef deze 2 tijden weer door een aanroep van `application.setTextCalc()` en een aanroep van `application.setTextDraw()`. Ga na dat, bij zoomen en verplaatsen, het tekenen nu sneller verloopt dan voorheen.
10. Door welke thread worden de berekeningen gedaan?
Run je applicatie nogmaals voor het hoogste level uit opdracht 7, en probeer deze via de GUI (dus niet via NetBeans) te stoppen terwijl er nog gerekend wordt. Lukt dit?
Verklaar dit.

Toon de resultaten van vraag 6 en vraag 9 aan de docent, evenals de tabel van opdracht 7 en het antwoord op de vragen van opdracht 10.

Klasse KochFractal in package calculate

Deze klasse hoeft NIET aangepast te worden.

Dit is de klasse waarin de zijden van de fractal berekend worden.

Deze klasse is een Observable en heeft daarnaast de volgende public methodes:

```
public void setLevel(int lvl)
```

Hiermee wordt het level van de fractal bepaald.

Level 1 is een driehoek; bij de hogere levels worden de zijden steeds verder opgedeeld.

```
public int getLevel()
```

Om het huidige level op te vragen.

```
public int getNrOfEdges()
```

Om het aantal edges (lijnstukken) bij het huidige level op te vragen.

```
public void cancel()
```

Om de huidige berekening van edges (lijnstukken) te stoppen.

```
public void generateLeftEdge()
```

```
public void generateBottomEdge()
```

```
public void generateRightEdge()
```

Hiermee worden, voor het huidige level, alle edges (lijnstukken) van de fractal berekend. Er is voor elke zijde van de begindriehoek een aparte methode:

`generateLeftEdge()` voor de groene zijde links, `generateBottomEdge()` voor de blauwe zijde beneden en `generateRightEdge()` voor de rode zijde rechts. De berekende edges worden één voor één doorgegeven aan de Observer(s).

Voor elke edge wordt dus opnieuw methode `update` van de Observer aangeroepen, met de betreffende edge als parameter. Het is aan de Observer om te bepalen wat er precies met deze informatie gedaan wordt.

Let op: als deze `generate...` methodes door een bepaalde thread worden aangeroepen, dan zal diezelfde thread vervolgens de `update` van de Observer(s) aanroepen.

Klasse JSF31KochFractalFX in package jsf31kochfractalfx

Deze klasse hoeft NIET aangepast te worden.

Dit is een Java FX applicatie met daarin een Canvas (het panel waarin de Koch fractal wordt getekend) en een aantal textvelden en buttons:

Met deze applicatie kunnen de volgende operaties uitgevoerd worden:

1. fractal vergroten (linker muisknop)
2. fractal verkleinen (rechter muisknop)
3. fractal verplaatsen (slepen met linker muisknop)
4. het level van de fractal verhogen (button Increase Level)
5. het level van de fractal verlagen (button Decrease Level)
6. de huidige fractal resizen zodat deze precies in het panel past (button Fit Fractal)

JSF31KochFractalFX heeft de volgende publieke methoden die je nodig hebt voor de uitvoering van de opdracht:

<code>public void clearKochPanel()</code>	Maakt het panel zwart
<code>public void drawEdge(Edge e)</code>	Tekent een edge in het panel
<code>public void setTextCalc(String text)</code>	Zet text achter label 'Calculating:'
<code>public void setTextDraw(String text)</code>	Zet text achter het label 'Drawing:'
<code>public void requestDrawEdges()</code>	Tekent alle edges.

Aanroep van de methode `requestDrawEdges()` zorgt ervoor dat de edges van de Koch fractal getekend worden door de Java FX Application Thread (benodigd voor volgende opdrachten).