



Peter König, Noud van Kruysbergen

Netradar

Visualisaties van een website programmeren met Processing

Grafische weergaven kunnen verborgen relaties tussen complexe gegevens onthullen. Als je zo'n representatie zelf programmeert, kun je de visualisatie aan je eigen behoefte en smaak aanpassen. De compacte programmeeromgeving Processing zorgt ervoor dat dit overzichtelijk blijft.

De nieuwswebsite nu.nl heeft van een aantal nieuws categorieën telkens één item groot uitgelicht en de laatste zeven nieuwtjes van die rubriek met hun titel onder elkaar gezet. Hierbij staat het laatste nieuws altijd bovenaan. Dit leidt tot een overzichtelijke startpagina, waarbij je in één oogopslag weer helemaal bij de tijd bent.

Het nadeel van deze weergave is dat je veel detailinformatie over het hoofd ziet, bijvoorbeeld de tijd van de publicatie en de lengte van de tekst. Bovendien kan een nieuwsitem een vervolg zijn op eerdere berichten, waar je

normaal pas achterkomt als je een bericht in je browser opent.

Als je geïnteresseerd bent in de onderlinge samenhang tussen nieuwsberichten, kan een visualisatie van de items helpen om onderlinge verbanden inzichtelijk te maken. Als voorbeeld hebben we met de eenvoudige, gratis programmeeromgeving Processing een alternatieve weergave voor de nieuwsberichten op nu.nl gemaakt. In het vorige artikel hebben we de basisbegrippen als variabelen, loops en functies behandeld. Hier breiden we dat uit met externe bibliotheken, gegevens ophalen

van internet en wat objectgeoriënteerd programmeren.

Alles in één oogopslag

De nieuwsberichten verschijnen dan als kleine cirkels op een soort nieuwsradarscherm. Hierbij geeft de kleur van de cirkels aan bij welke rubriek het nieuws hoort, terwijl de grootte van de cirkel een indicatie is voor de lengte van het nieuwsitem. De lijnen tussen de cirkels symboliseren welke nieuwsberichten met elkaar te maken hebben. Als je met de muis over een cirkel beweegt, krijg je de titel en de datum van het bericht te zien. Door met de rechter muisknop op een nieuwsitem te klikken, wordt het betreffende bericht in je browser geopend.

Via de link aan het einde van het artikel vind je de uitvoerbare Java-programma's voor Windows, Mac OS X en Linux, evenals de broncode. De programma's staan in een ZIP-bestand. Nadat je dat hebt uitgepakt, kopieer je de map Netradar in zijn geheel in de schetsboekmap van Processing.

Onze code kan als inspiratiebron dienen voor eigen programmeerprojecten. Je kunt hem aanpassen, ombouwen of stukken hergebruiken. Een aantal principes die we hier gebruiken, kun je immers ook toepassen op andere visualisatievormen van gegevens op websites.

Orde door spanning

Als je de berichten op een willekeurige plek op het scherm zou zetten, zou het door de links tussen de berichten een onoverzichtelijke puinhoop worden waar geen systeem in valt te ontdekken. Om dat te voorkomen, gebruiken we dan ook een handig lay-outalgoritme.

Bij programmeurs is een zogeheten force-directed lay-out populair om netwerkgrafen te visualiseren. Letterlijk vertaald: een ordening op grond van krachten. Het idee is dat je objecten een bepaalde massa meegeeft en tussen objecten een afstotingskracht definieert, vergelijkbaar met de opbouw van elektronen. De objecten die gerelateerd zijn, zijn verbonden met een springveer met een bepaalde veerconstante en een gegeven rustlengte. Dit alles gooi je in een eenvoudige fysische simulatie en dan maar kijken wat eruit komt. Door de afstoting verdelen de objecten zich zo gelijkmatig mogelijk in de ruimte, waarbij de veren ervoor zorgen dat alles wat met elkaar te maken heeft zoveel mogelijk bij elkaar blijft.

Dergelijke krachtgestuurde grafen leiden vaak tot mooie plaatjes, zelfs als je de beginposities willekeurig kiest. Het leuke is dat wat er op het scherm gebeurt zich net zo gedraagt als een fysische constructie met ballen en springveren. Er zijn wel een paar nadelen. Een echt perfecte orde krijg je alleen bij eenvoudige structuren of bij toeval. Hoe meer verbindingen er in het spel zijn, des te waarschijnlijker dat er kruisende lijnen of hardnekkig aan elkaar klevende objecten overblijven. Bovendien komen grote grafen zelden tot rust en blijven ze wat chaotisch heen en weer schommelen als je niet voor genoeg demping en wrijving zorgt. Daarbij vraagt het algoritme aardig wat rekenkracht en zorgt het implementeren voor de nodige hoofdbreken.

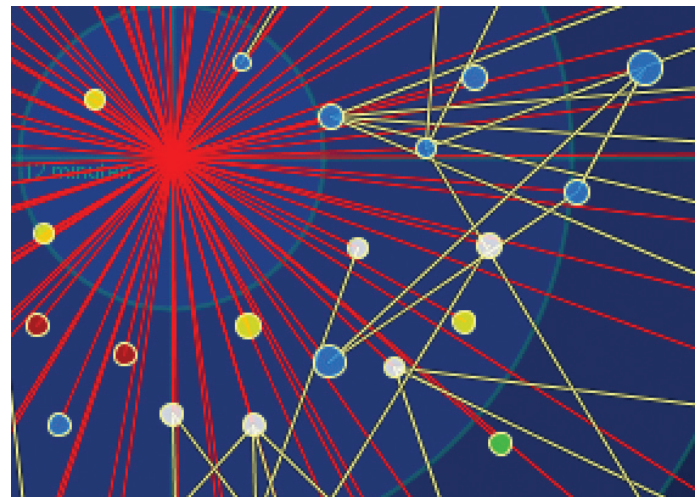
Het laatstgenoemde probleem is gelukkig makkelijk op te lossen. Jeffrey Traer Bernstein heeft speciaal voor Processing een bibliotheek met de naam `traer.physics` gemaakt, waarmee je met een paar regels code objecten (Particle genoemd) kunt maken. Die kun je aan veren koppelen en er aantrekkings- dan wel afstotingskrachten aan meegeven. Een grafische weergave zit er echter niet bij. Maar dat is ook niet zo erg, omdat je met Processing zonder veel moeite mooie, grafische effecten tevoorschijn kunt toveren.

In een omloopbaan

In Newsradar worden alle nieuwsberichten een partikel in een Traer-partikelsysteem. Alle partikels hebben hierbij een identieke massa en stoten elkaar met een kleine kracht af, zodat twee nieuwsberichten niet dezelfde plek innemen. In de bibliotheek zit echter geen routine om botsingen te detecteren. De elastische springveren trekken de gerelateerde nieuwsberichten naar elkaar toe.

De berichten komen op een rond radarscherm, waarbij het actuele nieuws in het centrum verschijnt en oudere berichten verder naar de rand komen te staan. Dit wordt gedaan met behulp van een vast, onzichtbaar

Stugge springveren dienen als afstandshouders die de oudere berichten verder van het centrum van het radarscherm duwen. Hier zijn ze rood getekend, maar in het uiteindelijke programma zie je ze niet.



partikel met de naam `center` in het midden van het radarscherm, dat als ankerpunt voor de weergave dient. Tussen dit punt en ieder nieuwspartikel dient een zeer stijve veer als afstandshouder (`spacer`). De rustlengte daarvan komt overeen met de leeftijd van het nieuwsbericht in minuten. De schaal van de lengte van de afstandshouder is logaritmisch, waardoor je nieuwere berichten makkelijk uit elkaar kunt houden en oudere berichten wat dichter bij elkaar komen te staan. Dit voorkomt meteen dat een nieuwsitem van een paar jaar oud buiten het venster terecht zou komen, wat bij een lineaire schaal wel het geval zou zijn.

Newsradar kijkt in eerste instantie echter alleen naar de berichten op de startpagina van `nu.nl`. Deze zijn in de regel niet ouder dan een paar uur. De parameter `DEPTH` in het programma bepaalt echter tot welk recursieniveau er naar andere berichten moet worden gezocht. Bij 0 wordt er niet verder gekeken naar gerelateerde berichten; bij 1 worden ook de gerelateerde berichten in de nieuwsitems meegegenomen; bij niveau 2 wordt ook gekeken naar de gerelateerde berichten van de gerelateerde berichten, enzovoort. Sommige berichten, bijvoorbeeld over de Mexicaanse griep, kunnen al snel een stortvloed aan gerelateerde berichten oproepen. Een recursieniveau van meer dan 2 kan dan ook tot een onoverzichtelijke weergave leiden. De kant-en-klare programaversies van Newsradar werken dan ook standaard met `DEPTH=1`.

Code-indeling

Als je de broncode van Newsradar in Processing opent, kun je alle parameters en constanten van het programma aanpassen en bijvoorbeeld het recursieniveau verhogen. Je kunt ook dieper in de code duiken en proberen of je hetzelfde principe op een andere nieuwssite kunt toepassen. Voor het eerste heb je niet echt programmeerervaring nodig, maar voor het laatste is enige bekendheid met Java en Processing wel een voordeel. Aan de andere kant zijn de syntaxis en opbouw van de code vrij overzichtelijk, zodat je redelijk snel inzicht krijgt in wat er waar gebeurt.

Na het starten van de code-editor klik je op het menu-item 'File / Sketchbook / Newsradar'. Er openen vier aparte programmabestanden met elk een eigen tabblad. Newsradar bevat het hoofdprogramma, `News` definieert de gelijknamige klasse, in `drawHelpers` staan een aantal hulpfuncties voor de grafische weergave en in `processSource` staan functies om de website en links te analyseren.

Je maakt een nieuw bestand door op het pijltje rechtsboven te klikken voor een nieuw tabblad. Het opdelen van een programma in dergelijke deelbestanden zorgt voor overzicht, al maakt Processing er intern gewoon één groot codebestand van voordat de compiler er op los wordt gelaten. Het voordeel is dat globaal gedefinieerde variabelen in een bestand zonder meer ook vanuit andere code-tabbladen toegankelijk zijn.

Nieuwtjes

Een klasse is een soort sjabloon waar een programma een groot aantal objecten van hetzelfde type van kan maken, maar die afzonderlijke waarden hebben. De variabelen binnen de klasse `News` beschikken over alle informatie over een nieuwsbericht die bij de verdere programma-afhandeling van belang is. Daar hoort bijvoorbeeld het berichtnummer (`id`) bij, dat zich in de url van een nieuws-item bevindt, evenals de titel van het bericht, hoe oud het nieuws is en bij welke rubriek het hoort. In `News` wordt geen positie opgeslagen, maar een verwijzing naar een `Particle` met de naam `pos`, waarvan je de `x`- en `y`-positie kunt opvragen.

De `News`-objecten beheren hun links in een zogeheten `HashMap`. In de documentatie van Processing zul je dit object niet terug kunnen vinden, aangezien het van het standaard Java-pakket `java.util` geleend is. Je kunt onbeperkt onderdelen uit de Java-API gebruiken, maar je moet jezelf hierbij wel tot Java 1.4 beperken, aangezien Processing op deze oudere versie is gebaseerd. De regel `import java.util.*;` aan het begin van het hoofdprogramma zorgt ervoor dat je de `HashMap` en andere nuttige onderdelen uit het pakket kunt gebruiken.

Een `HashMap` kan objecten (values) van verschillende typen opslaan. Die kun je dan

Als een nieuwsbericht zich onder de muis bevindt, verschijnen de rubriek, de datum en de titel. Als je er met de rechter muis-knop op klikt, wordt het nieuwsbericht in de browser geopend.

via een eenduidige sleutel (key) bereiken, die eveneens van een willekeurig type kan zijn. De afzonderlijke News-items gebruiken de bericht-id van het gerelateerde nieuwsitem als sleutel en de springveer uit het partikelsysteem die de verbinding voorstelt als waarde.

Constructief

Het grootste deel van de code van de klasse wordt in beslag genomen door de constructor, de functie die nieuwe News-objecten initialiseert. Het binnenhalen van de broncode van een webpagina in een Processing-programma is wel héél erg eenvoudig: `loadString(String url)` levert een array van strings terug, waarin de afzonderlijke HTML-regels staan. De constructor moet daar dan wel alle relevante informatie moeizaam uit zien te peuteren. Een voorbeeld:

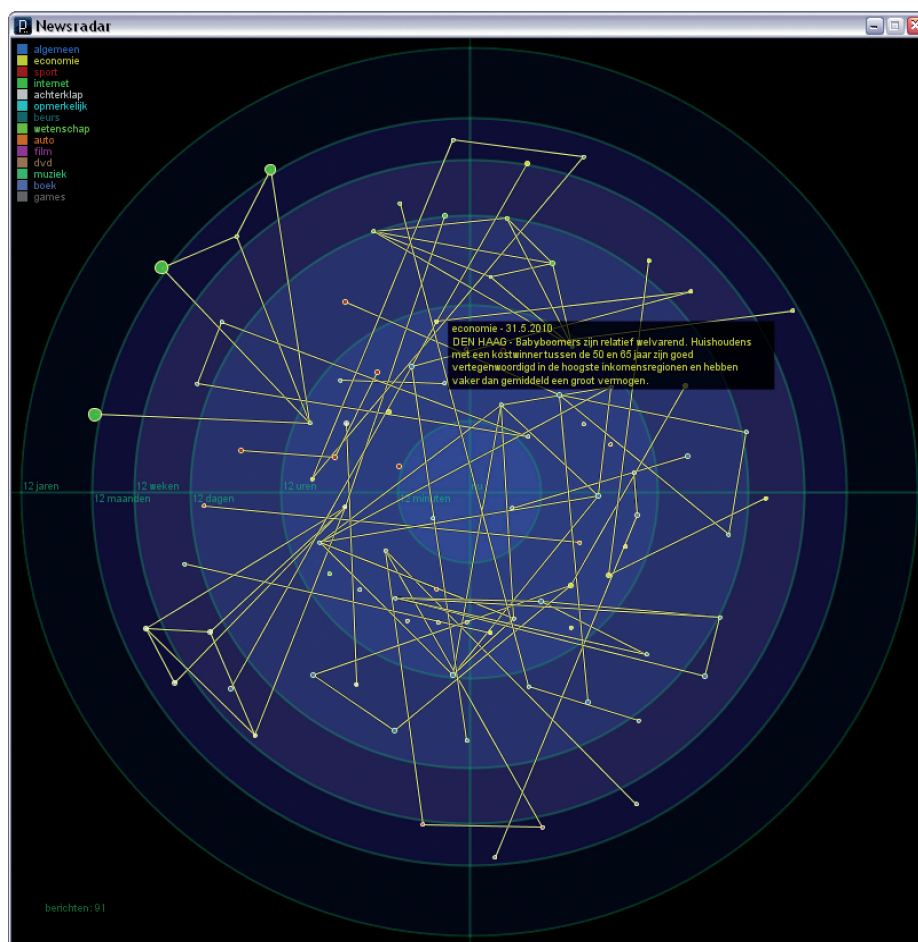
```
if (lines[i].indexOf("<meta name=\"description\"") != -1){
    title = getContent(lines[i]);
}
```

De eerste regel test of de i-de regel de meta-tag 'description' bevat door naar de positie van deze tag in de string te vragen. Als die niet voorkomt, krijg je -1 als resultaat. Als die wel voorkomt, wordt de functie `getContent()` opgeroepen, die van een HTML-regel zoals

```
<meta name="description" content="AMSTERDAM - Zes
van de tien vrouwen vindt dat mannen niet begrijpen
waarover vriendinnen onderling zo'n lol kunnen hebben.
43 procent van de mannen beaamt dit." />
```

alleen de content teruglevert. In dit geval gaat het om de korte inhoud van het nieuwsbericht.

Deze constructie is specifiek voor een nieuwsbericht op `nu.nl`, net als de andere functies die we op maat hebben gemaakt



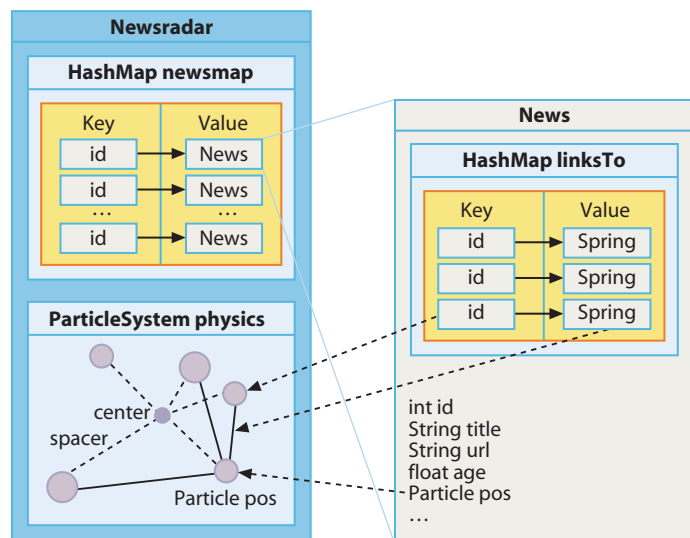
voor deze nieuwssite. Daaronder vallen bijvoorbeeld het achterhalen van de rubriek, het vaststellen van de publicatiedatum en de bericht-id. Als je Newsradar wilt aanpassen voor een andere nieuwssite, zul je zelf moeten gaan sleutelen. Je moet dan de HTML-code van de nieuwspagina's kunnen analyseren en achterhalen waar de benodigde informatie zich bevindt. Ook loop je het risico dat het programma niet meer werkt als `nu.nl` de pagina-opbouw verandert. Dat is nu eenmaal de prijs die je betaalt als je de informatie op zo'n laag niveau uit de HTML-code gaat peuteren.

Het bovengenoemde voorbeeld van de titel is dan een positieve uitzondering. Informatie in de meta-tags van een pagina is niet afhankelijk van de opbouw en inhoud van die pagina en is dus makkelijk te abstraheren. Het achterhalen van de rubriek van het nieuws is echter een geheel andere zaak. Die staat wel in de url van het nieuwsbericht, maar eigenlijk willen we de rubriek uit de pagina zelf kunnen halen. In de HTML vind je die in de regel

```
<a href="/algemeen/"></a>
```

dus daar moet je eerst naar zoeken. Dat doen we op basis van het CSS-commando 'class="cattap"', omdat die alleen daar voorkomt. Het achterhalen van de publicatiedatum verloopt op vergelijkbare wijze; die staat in de regel volgend op 'Uitgegeven:'. Het zou mooier en beter (en meer gestandaardiseerd) zijn als deze datum en de rubriek in een meta-tag zouden staan. Hoe beter een website georganiseerd en gestructureerd is, hoe makkelijker het wordt om dergelijke informatie te achterhalen.

Nadat we de broncode van de pagina op deze wijze globaal hebben onderzocht en in de string `longtxt` hebben gezet, gaat de functie `refineText()` aan de slag voor het fijnere werk. Daar wordt eerst de body van het eigenlijke bericht zo goed mogelijk geabstraheerd. In de HTML-code staat die tussen het begin van de samenvatting (class="summary") tot aan de eerstvol-



Newsradar maakt van de nieuwsberichten van `nu.nl` objecten van de klasse `News`. Het partikelsysteem bepaalt waar de berichten op het scherm moeten komen, waarbij een fysisch systeem van veren, massa's en afstoting wordt gesimuleerd.

gende <div>. De lengte van dit tekstdeel wordt later gebruikt als indicatie voor de omvang van het nieuwsbericht en daarmee voor de diameter van de cirkel waarin het wordt getoond.

Het achterhalen van de gerelateerde nieuwsitems verloopt nog minder gestructureerd. Die staan in de HTML na het deel waar 'ZIE OOK:' op de pagina staat, dus ook dat deel moet je 'handmatig' opzoeken. Vervolgens wordt die substring doorlopen op zoek naar url's, die wel moeten voldoen aan de definitie van een url voor een nieuwsitem. Als een url voldoet, wordt die in linksTo opgeslagen, maar nog wel met de waarde null omdat er nog geen verbinding is. Vervolgens test de functie of het maximale recursieniveau al bereikt is en of het nieuwsitem niet al in Newsradar zit. Als beide niet het geval zijn, initialiseert de constructor zelf een nieuw News-object van het gerelateerde bericht.

Hoofdzaak

Het hoofdprogramma op het tabblad Newsradar bevat zo'n beetje alle denkbare parameters die je voor een visualisatie zou kunnen instellen. De elasticiteit van de springveren wordt vastgelegd met STRENGTH, het aantal minuten voordat er weer naar nieuwe nieuwsberichten moet worden gezocht met REFRESH en de sterkte van de initiële afstoting met REPULSE. In het commentaar in de code staat omschreven wat de diverse parameters voorstellen. Sommige parameters zijn afhankelijk van andere. Zo zijn de logaritmische waarden voor de leeftijd van de nieuwsitems afhankelijk van de schaal SCALE van de venstergrootte SIZE, net als de rustlengte van de springveren LINKRL.

De constante TICK bepaalt of het partikelsysteem realtime (bij een waarde van 1.0) of vertraagd (ongeveer 0.5) moet werken. Als er veel nieuwsberichten zijn, gaan ook veel krachten een rol spelen. Ook van een moderne computer vergt het veel rekenkracht om het hele partikelsysteem voor iedere momentopname door te rekenen. Hierdoor ontstaat er wat meer tijd tussen de momentopnamen, waardoor een aantal objecten een vluchtendens kunnen ontwikkelen. De graaf lijkt dan te exploderen en er komt een aantal berichten buiten het venster terecht. Hoe meer berichten er zijn, des te groter de kans dat dit gebeurt. Het is daarom veel handiger de weergave niet realtime te laten plaatsvinden, maar met een zekere vertraging, zodat er genoeg tijd is om alle berichten en links door te rekenen. We hebben de vertraging-factor daarom omgekeerd evenredig aan het recursieniveau DEPTH gemaakt. Bij DEPTH=0 wordt de vertraging 0,5; bij DEPTH=1 daarentegen 0,33, enzovoort.

Na de declaratie van de diverse constanten en globale variabelen volgen de twee functies die een Processing-programma laten werken: setup() wordt na het starten één keer uitgevoerd, maar draw() telkens opnieuw totdat je het programma afsluit.

Bij het starten van het programma maakt setup() een HashMap aan, die dit keer de naam newsmap meekrijgt. Ook nu dienen de bericht-id's als sleutel, terwijl de bijbehorende News-objecten hier als waarden worden opgeslagen.

Vervolgens maakt het programma een ParticleSystem met de naam physics. Ook die zit in de bibliotheek van Bernstein en werkt als simulator waarin je met functies als makeParticle() en makeSpring() objecten en verbindingen mon-

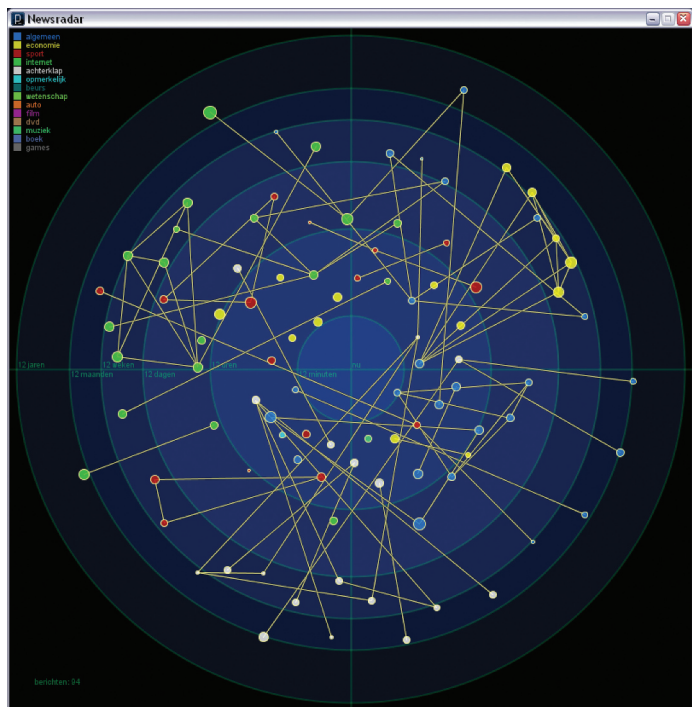
teert. Het eerste partikel is center, dat met makeFixed() op een vaste plek wordt gedefinieerd.

Nieuw nieuws

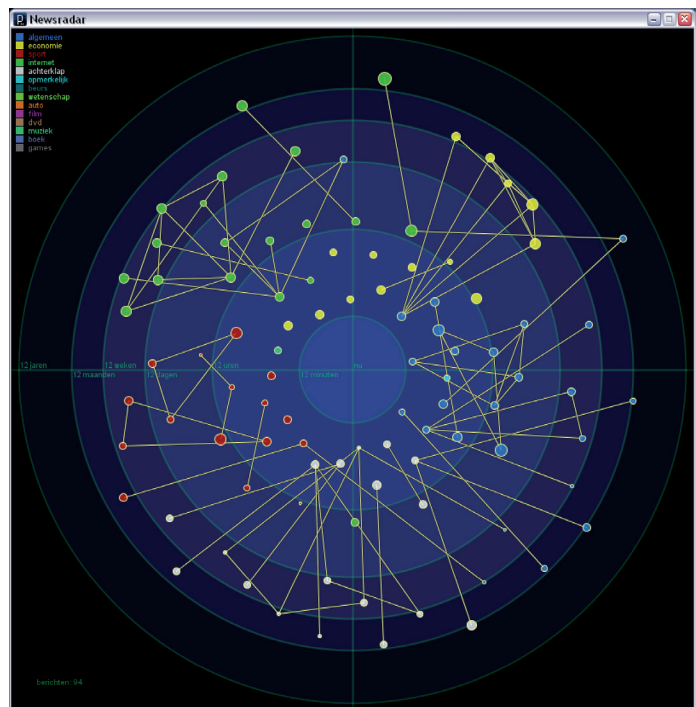
Tijdens het uitvoeren van setup() wordt de systeemtijd in de variabele minuteNextRefresh gezet. Daardoor worden de functies checkWebsite() en checkLinks() aangeroepen als draw() voor de eerste keer wordt uitgevoerd. Deze twee functies zijn voor het overzicht op het tabblad processSource gezet.

De eerste functie speurt de startpagina van nu.nl af naar links naar nieuwsberichten. Daarbij worden de hulpfuncties isolateURLfromMainPage() en getIdFromURL() gebruikt, die eigenlijk precies doen wat hun naam zegt. Hoe ze dat doen wordt in de code (en het commentaar) wel duidelijk. Als de functies een url en een id terugleveren, dan controleert het programma of newsmap de betreffende id al als sleutel heeft. Is dat niet het geval, dan wordt er een nieuw News-object aangemaakt en toegevoegd. Als de id al wel bestaat, dan wordt de leeftijd van dat bericht opnieuw berekend.

Pas wanneer alle nieuwsberichten beschikbaar zijn en hun leeftijd is bijgewerkt, kijkt checkLinks() welke verbindingen er getekend kunnen worden. De hele lijst van aanwezige News-items wordt doorlopen en van elk item wordt in hun HashMap met links gekeken naar verbindingen met berichten die al in het systeem zitten, maar waar nog geen verbinding naar bestaat. Dat is het geval als er in newsmap wel een entry is met de betreffende id, maar er bij linksTo met die id een false terugkomt. In dat geval wordt er in het partikelsysteem een nieuwe veer gemaakt en aan linksTo toegevoegd.



De virtuele veeraandrijving vindt niet altijd de optimale oplossing. In dergelijke gevallen kun je met de muis ingrijpen...



...door nieuwsberichten naar een plek te verslepen waar de met elkaar gerelateerde items zich overzichtelijker kunnen groeperen.

Het voordeel van deze tweetraps werkwijze is dat er in een aantal uitzonderingsgevallen toch een verbinding wordt gelegd als het eigenlijke recursieniveau DEPTH wordt overschreden. Voorbeeld: op de startpagina verschijnt bericht A met een link naar de oudere berichten B en C. Het recursieniveau is 1, dus de links B en C worden niet verder gevolgd. Als die twee echter onderling gelinkt zijn, verschijnt die verbinding toch in de nieuwsradar. De werkwijze heeft ook voordelen als nieuwe nieuwsberichten naar al bestaande nieuwsitems linken.

Tekenuurtje

Processing is van huis uit een programmeer-tool die is ontwikkeld voor grafici en kunst-

```
void draw(){
  if (minute() == minuteNextRefresh){
    checkWebsite();
    checkLinks();
    minuteNextRefresh = (minute() + REFRESH) % 60;
  }
  physics.tick(TICK);
  if (EMERGENCY_BRAKE != 0){
    doEmergencyBraking();
  }
  drawBackground();
  drawGraph();
}

void checkWebsite(){
  String [] lines = loadStrings(URL);
  for (int i=0; i<lines.length; i++){
    if (lines[i].indexOf("<.html">") != -1){
      String curr = isolateURLfromMainPage(lines[i]);
      int currId = getIdFromURL(curr);
      if (!Newsmap.containsKey(currId)){
        News newNews = new News(curr, currId, DEPTH,
          random(SIZE), random(SIZE));
        newsmap.put(currId, newNews);
      } else {
        ((News)newsmap.get(currId)).ageNews();
      }
    }
  }
}

void checkLinks(){
  Set newskeys = newsmap.keySet();
  Iterator it = newskeys.iterator();
  while (it.hasNext()){
    int currNewsId = (Integer)it.next();
    News currNews = (News)newsmap.get(currNewsId);
    Map links = currNews.linksTo;
    Set linkskeys = links.keySet();
    Iterator it2 = linkskeys.iterator();
    while (it2.hasNext()){
      currLinkId = (Integer)it2.next();
      if (newsmap.containsKey(currLinkId)){
        News currLinkedNews = (News)newsmap.get(currLinkId);
        if (links.get(currLinkId) == null){
          Spring linkspring = physics.makeSpring(currNews.pos,
            currLinkedNews.pos, STRENGTH, DAMP,
            max(LINKRL, abs(currNews.age-currLinkedNews.age)));
          links.put(currLinkId, linkspring);
          [...]
        } else {
          ((Spring)links.get(currLinkId)).setRestLength
            (max(LINKRL, abs(currNews.age-currLinkedNews.age)));
        }
      }
    }
  }
}
```

naars. Het is dan ook geen wonder dat het niet zo'n probleem is om dingen op het scherm te toveren. De functie drawGraph() (op het tabblad drawHelpers) vraagt om informatie rechtstreeks van het partikelsysteem, krijgt de verbindingen aangereikt en tekent die als lijnen. Als het eindpunt tenminste niet het partikel center is – de afstandhouders moeten immers onzichtbaar blijven. De functie drawNews hoort bij de klasse News, omdat die informatie van de klasse achterhaalt, zoals de kleur en grootte waarin de cirkel moet worden weergegeven.

Experimenteren

Door met de parameters in de code te spelen, kun je de grenzen van het visualiseren van het partikelsysteem en de virtuele veren experimenteel verkennen. Als je het recursieniveau bijvoorbeeld hoger zet, duurt het starten van het programma steeds langer. Zelfs een 2,3GHz-dualcore-pc met 4 GB werkgeheugen komt bij een grootschalig partikelsysteem niet verder dan een frame per seconde. Dan loop je bovendien het gevaar dat je ontworpen systeem door een te kleine damping van de veren of een te grote onderlinge afstoting van de partikels alle kanten uitschiet. Dan is het handig om met de parameter EMERGENCY_BRAKE voor ieder partikel een maximale snelheid in x- en y-richting in te stellen. Een waarde tussen de 100 en 200 remt de normale bewegingen van de berichten niet al te veel af, maar houdt een dreigende chaos wel in toom. Bij de waarde 0 is de noodrem uitgeschakeld.

Als je het recursieniveau flink opschroeft, zal de grafische weergave meer op een chaotische sterrenhemel gaan lijken. Dat ziet er wellicht mooi uit, maar door het grote aantal nieuwsberichten en links is daar nauwelijks nog structuur uit te destilleren. In de klasse News hebben we toch enigszins geprobeerd wat meer orde in de weergave aan te brengen door de nieuwsberichten op de startpagina afhankelijk van hun rubriek in bepaalde kwadranten onder

te brengen. Zo zullen alle nieuws-items uit de rubrieken algemeen, economie en beurs in het eerste kwadrant (rechtsboven) worden getoond. Je kunt deze voorselectie uitzetten of uitbreiden naar alle niveaus.

Een nieuwsradar als deze heeft eigenlijk alleen maar nut als er hooguit een honderdtal berichten wordt getoond. Het continu bijwerken van het partikelsysteem vergt nogal wat rekenkracht en is daarmee een prima bezigheidstherapie voor een onderpresterende kantoorcomputer om op de achtergrond door de tweede kern te worden uitgevoerd. Op een singlecore-pc wordt het wat lastiger om op die manier te multitasken.

De veren hebben zelf geen massa en zullen daarom niet afstotend werken. Daarom komt een bericht in de praktijk soms per ongeluk op een vreemde link terecht. Bovendien verschijnen clusters van sterk gerelateerde berichtgroepen soms enigszins in de war vanwege de willekeurige positie die ze in het begin krijgen. Die komen dan uit eigen kracht niet uit de war, want het algoritme kan immers niet toveren. Alleen als gebruiker kun je hier en daar ingrijpen. Als je de muisknop ingedrukt houdt, kun je de berichten naar een plek verschuiven waar ze beter tot hun recht komen. Als je op de x-toets drukt, wordt de graaf wat door elkaar geschud, wat tot een beter resultaat kan leiden. Lukt het dan nog niet, probeer dan eens de hoofdletter X.

Uitbouwen

Met kleine veranderingen aan de code kun je het uiterlijk van Newsradar aanpassen. Zo kun je meerdere kenmerken van de nieuwsberichten op het scherm laten weergeven, zoals de datum van eventuele wijzigingen en of er een video bij zit. Bij een andere nieuwssite zou je bijvoorbeeld kunnen aangeven hoeveel reacties er op een forum zijn. Het is dan wel zaak om regelmatig te updaten. Als je bepaalde parameters van het programma of het partikelsysteem vaker wilt aanpassen, kun je daar een schuifregelaar voor maken. Processing heeft zelf geen kant-en-klare GUI-elementen, maar op de projectwebsite kun je daar wel materiaal voor vinden, zoals de controlP5-bibliotheek van Andreas Schlegel.

Het partikelsysteem probeert de lay-out van de nieuwsradar continu te optimaliseren – zelfs als er geen berichten zijn bijgekomen. Als je cpu-tijd daar te kostbaar voor is, kun je bijvoorbeeld de functie doEmergencyBraking() zodanig ombouwen dat alle partikels tot aan de volgende REFRESH met makeFixed() worden vastgezet als hun snelheid in de x- en y-richting beneden de 1.0 komt.

Je kunt Newsradar wellicht omzetten naar een webapplicatie door deze in Processing als Java-applet te exporteren en op je blog te integreren, de force-directed lay-out door een zelfbedacht algoritme te vervangen, of de visualisatie geheel anders aan te pakken. De mogelijkheden zijn wat dat betreft redelijk onbeperkt. (nkr)