|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
|  |  |  | |  |
| Experiment Dynamische kaartlaag  Stein Jonker  02-11-2022 | | |
| Minor DMP | | Mario de Vries |

|  |  |
| --- | --- |
|  |  |
| Inhoudsopgave  [1 Inleiding 3](#_Toc118711106)  [2 Hypothese 4](#_Toc118711107)  [3 Resultaten 5](#_Toc118711108)  [3.1 Interactie 5](#_Toc118711109)  [3.2 Bewerken 5](#_Toc118711110)  [4 Conclusies 6](#_Toc118711111)  [5 Bronnen 7](#_Toc118711112)  [6 Bijlagen 8](#_Toc118711113)  [6.1 Bewerken onbewerkbare lagen 8](#_Toc118711114) | |

|  |  |
| --- | --- |
|  |  |

# Inleiding

In dit document beschrijf ik het proces en de resultaten van mijn tweede experiment. In dit experiment heb ik onderzocht of het mogelijk is om een kaartlaag aan te passen op basis van acties die een gebruiker uitvoert. Denk bijvoorbeeld aan een kaart die de afstand naar een boom visualiseert. Wanneer de gebruiker een boom plaatst, moet de kaart worden bijgewerkt.

De aanleiding tot dit experiment komt voort uit het idee voor een groen-planning tool en het 3-30-300 beleid. Dit beleid is in het Arnhemse coalitieakkoord 2022-2026 genoemd.

Dit beleid streeft naar:

* Ieder woning heeft uitzicht op minimaal drie bomen
* Een wijk heeft minimaal 30 procent kroondekking
* De afstand naar een park vanaf een woning is maximaal 300 meter.

De huidige 3-30-300 situatie is al in kaart gebracht. Wanneer de gebruiker een boom plaatst in de applicatie, dan moet de kaart worden bijgewerkt met de nieuw geplaatste bomen. Of deze aanpassing mogelijk is, wordt onderzocht in dit experiment.

In het experiment is gebruik gemaakt van de Arcgis Javascript API om een kaart te bouwen.

|  |  |
| --- | --- |
|  |  |

# Hypothese

Het verwachtte resultaat van het experiment is dat het wel mogelijk gaat zijn om een kaartlaag aan te passen op basis van gebruikers acties (bijv. het plaatsen van een boom).

Arcgis biedt al een mogelijkheid om attributen aan te passen met de [Editor widget](https://developers.arcgis.com/javascript/latest/sample-code/widgets-editor-3d-scenelayer/). De gebruikers interactie vindt hier plaats op een andere manier, door te klikken op een onderdeel van het gebouw.

Het aanpassen van attributen zal vervangen moeten worden door het plaatsen van een boom. Vervolgens moet het effect van de boom op twee kaartlagen worden verwerkt: de bomenlaag en de laag die beïnvloed wordt, in dit geval de 3-30-300 kaart.

|  |  |
| --- | --- |
|  |  |

# Resultaten

## Interactie

Om de gebruiker bomen te laten plaatsen kan gebruik gemaakt worden van de [editor widget](https://developers.arcgis.com/javascript/latest/sample-code/widgets-editor-3d-scenelayer/) van Arcgis.

Deze widget laadt automatisch de kaartlagen in die als “editable” zijn gemarkeerd. Dit wordt door de uitgever van de kaartlaag bepaald. Wanneer een laag editable is, maar je deze niet wil bewerken kun je dit configuren. Daarom heb ik geprogrammeerd dat elke laag niet-editable is (zelfs als ze dat wel zijn) behalve als het id van de laag begint met “editable”

De kaartlaag waarin de bomen staan is niet gemarkeerd als editable. Als dit wel zo zou zijn, dan zouden de bomen die de gebruiker plaatst ook echt worden opgeslagen. De bijgewerkte kaartlaag is dan voor iedereen te zien.

Omdat de gebruiker bomen neerzet die in de toekomst er komen te staan, ontstaat er datavervuiling. Er is dan geen onderscheid meer tussen bomen die er echt staan en die geplanned zijn.

Dit probleem kan op twee manieren worden opgelost:

* Een nieuwe, lege kaartlaag in Arcgis Online maken voor bomen die geplanned zijn. Dat betekent een extra kaartlaag, die allebei tegelijkertijd gevisualiseerd kunnen worden.

Deze laag kan voor iedere gebruiker worden aangemaakt, of één waar iedere gebruiker in werkt.

* Een kopie maken van de bestaande kaartlaag. Deze heeft geen verbinding met de originele laag en kan worden bijgewerkt, maar niet worden opgeslagen.

Ik heb gekozen om een lege kaartlaag aan te maken in Arcgis Online waarin alle gebruikers samenwerken. De onderbouwing voor deze keuze is te vinden in bijlage 6.1.

Deze kaartlaag is simpel aan te maken: op de website van Arcgis Online kan een nieuwe, lege, objectenlaag worden aangemaakt.

Voor deze laag kun je vervolgens velden configureren. Hiervoor heb ik de velden KROON\_DIAMETER en BOOMSOORT ingesteld. De grootte van de boom kan ik afleiden aan de kroon diameter (de diameter van het bladerdek) en de soort kan ik gebruiken om verschillende 3D modellen neer te zetten. In Figuur 1 op de volgende pagina is een screenshot te zien van het menu waarin bomen aangemaakt kunnen worden. De drie zichtbare bomen hebben allen een kroondiameter van 15 meter. De soorten van de bomen verschillen, deze zijn van links naar rechts: populier, berk, eik.



Figuur 1 Plaatsen van bomen

## Bewerken kaartlagen

Zoals in de hypothese genoemd, zal de bewerking in twee kaartlagen moeten worden verwerkt:

* De bomenplanning kaartlaag
* De 3-30-300 kaartlaag

Het bewerken van de bomenplanning kaartlaag doet de editor widget van Arcgis. Hier hoeft niets aan veranderd te worden.

Het bewerken van de 3-30-300 kaartlaag moet worden gedaan wanneer de bomenplanning laag is veranderd. Daarom voeren we een functie uit wanneer de ‘edits’ event voorkomt. Deze functie wordt geregistreerd in Figuur 2

Figuur 2 Registreren edits event

|  |  |
| --- | --- |
|  |  |

De onLayerEdited functie controleert eerst of wel objecten zijn aangemaakt of gewijzigd: zo niet, dan stopt het uitvoeren van de functie.

Vervolgens controleert de functie of de laag die gewijzigd is andere lagen beïnvloed. Dit is niet-standaard functionaliteit, maar kan worden geconfigureerd in het bestaande configuratiesysteem.

De configuratie van de bomenplanning kaart is te zien in Figuur 3.

Figuur 3 Configuratie bomenplanning laag

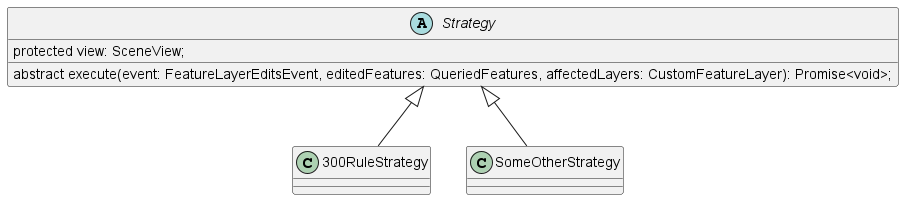
Met het attribuut “affects” kan een lijst worden opgegeven. Elk item in de lijst bevat een id van een andere kaartlaag en de naam van een strategie die moet worden toegepast om deze kaartlaag aan te passen. Elke laag bevat andere attributen, die met een ander algoritme aangepast moeten worden. De 3-30-300 kaart werkt bijvoorbeeld met scores, die een bepaalde berekening vereisen. Een andere laag heeft wellicht een algoritme nodig om een nieuwe afstand te berekenen.

Het systeem moet dus open staan voor nieuwe strategieën en kunnen wisselen per kaartlaag. De volgende vraag staat in het boek “Applying UML Patterns” van Craig Larman:

“How to design for varying, but related, algorithms or policies? How to design for  
the ability to change these algorithms or policies?” (Larman, 2004)

Deze vraag kan worden beantwoord met het strategy pattern; “Define each algorithm/policy/strategy in a separate class, with a common interface.” (Larman, 2004)

Een schematische weergave is in Figuur 4 te zien.



Figuur 4 Schematisch Strategy Pattern

|  |  |
| --- | --- |
|  |  |

De abstracte klasse Strategy is de common interface, waarin een execute functie zit. De verschillende implementaties van deze functie bevinden zich in elke subklasse. Door een nieuw object van de subklasses te maken kun je makkelijk wisselen van implementatie.

Voor elk beïnvloedde laag wordt dus de juiste strategie opgezocht en uitgevoerd. De implementatie voor de 3-30-300 kaart moet eigenlijk een berekening bevatten om scores opnieuw uit te rekenen. Echter is deze berekening onbekend en essentieel voor het experiment. Daarom is het algoritme tijdelijk een algoritme die alle scores naar 10 zet.

Het algoritme haalt eerst alle polygonen in de buurt van de geplaatste boom op. Dit is voor nu een radius van 20 meter. Deze polygonen krijgen ieder een score van 10.

Echter komt hier een nu bekend probleem voor: als de originele laag wordt aangepast ontstaat er datavervuiling.

Ik heb er nu voor gekozen om een lege, client-side laag aan te maken, die later alleen de bijgewerkte polygonen bevat. Zo heb je twee lagen, waarvan één alleen de aangepaste polygonen bevat.

Deze oplossing vermijdt de complexiteit om kopieën te maken op de server.

Wel is er een nadeel: de effecten van geplaatste bomen worden niet opgeslagen. Daarom zijn de effecten van geplaatste bomen weg na het herladen van de pagina. De bomen worden wel opgeslagen.

|  |  |
| --- | --- |
|  |  |

# Conclusies

Een goede uitbereiding is om de strategie uit te voeren op het moment dat de bomenplanning kaart laadt. Het effect op de omgeving wordt dan elke keer opnieuw berekend voor al geplaatste bomen in de bomenplanning laag. Op die manier zijn de effecten toch zichtbaar.

|  |  |
| --- | --- |
|  |  |

# Bronnen

Larman, C. (2004). *Applying UML patterns.* Pearson.

|  |  |
| --- | --- |
|  |  |

# Bijlagen

## Bewerken onbewerkbare lagen

|  |  |
| --- | --- |
| Probleem | Nieuwe objecten plaatsen op een kaartlaag die niet gemarkeerd is op |
| Oplossing | Er wordt een lege kaartlaag gemaakt in Arcgis Online om voor iedereen in samen te werken.  Dit is een technisch simpele oplossing want het werkt met de huidige architectuur. Omdat iedereen in één kaartlaag werkt kan het ook als communicatiemiddel werken, want je ziet elkaars ambities. |
| Alternatieven | 1. Een lege kaartlaag voor iedereen. Dit is technisch lastiger omdat idealiter deze kaartlagen automatisch aangemaakt worden, maar dit wordt door de versie van Arcgis JS die ik gebruik niet ondersteund. Dit vereist een andere versie waardoor de complexiteit van de applicatie toeneemt. Deze andere versie vereist architectuur veranderingen. 2. Een kopie van de bomenlaag creëren in de kaart, zonder verbinding naar een server. Het nadeel van deze oplossing is dat het veel performance kost want alle bomen moeten eerst worden gedownload uit de originele laag.   Daarnaast kan de laag niet worden opgeslagen, want er is geen verbinding naar een server. De resultaten kunnen dus niet worden gedeeld of later worden aangepast. |
| Voordelen keuze | * Technisch makkelijker dan alternatieven * Sneller dan alternatieven; kaart bestaat al * Doordat de kaart gedeeld is kan van iedereen de ambities worden ingezien |
| Nadelen keuze | * Handmatig eens een kaart aanmaken |