

# Project “Objectgericht Programmeren”: Deel 2

Prof. Eric Steegmans  
Raoul Strackx

Academiejaar 2010-2011

Deze tekst beschrijft het tweede deel van de opgave voor het project van de cursus ‘*Objectgericht Programmeren*’. Het project wordt bij voorkeur uitgewerkt in dezelfde groepjes van twee personen, maar je mag de samenstelling van je team nu ook wijzigen. Studenten die het eerste deel alleen hebben uitgewerkt, worden dan ook aangemoedigd om alsnog een groepje te vormen. De deadline voor het insturen van je nieuwe groepssamenstelling naar *ogp-project@cs.kuleuven.be* is 28 maart, 12u. Wanneer je groepje niet verandert, hoef je geen mail te sturen. Wanneer dit echter wel het geval is, stuurt één student een mail, met de studenten waarvoor dit gevolgen heeft in CC. Studenten die een groepje opbreken zijn zelf verantwoordelijk voor het verwittigen van hun (ex-)partner!

Na de deadline aanvaarden we geen nieuwe groepjes meer, indien er echter in bepaalde groepjes problemen ontstaan, dan mogen die splitsen. In dat geval moet er individueel worden verdergewerkt en moet dit zo snel mogelijk worden gemeld aan prof. Eric Steegmans.

Indien het project individueel wordt gemaakt, moeten sommige delen van de opgave niet worden uitgewerkt, dit is aangegeven op die plaatsen.

Een aantal zaken van deel 1 zullen in deel 2 veranderen, dit moet weer-spiegeld zijn in je code. Ook met de gekregen feedback na deel 1 dien je rekening te houden bij het maken van deel 2.

## Opgave

In het tweede deel van de opgave worden een aantal zaken omtrent vakjes gewijzigd en komen er een aantal toevoegingen, zoals muren met deuren erin tussen vakjes. Ook wordt nu de klasse *Dungeon* uitgewerkt. Alles van deel 1 waarover niets gezegd wordt, blijft ongewijzigd, al kan het wel onrechtsreeks beïnvloed worden.

## 1 Muren en Deuren [*Walls and Doors*]

In tegenstelling tot deel 1 van de opgave, worden grenzen van vakjes nu niet langer met getallen aangeduid, maar volgens de meer intuïtieve windrichtingen **north**, **east**, **south** en **west**. De boven- en onderkant worden respectievelijk aangegeven met **ceiling** en **floor**. De mogelijke richtingen voor een vakje zullen nooit veranderen.

Vakjes kunnen vanaf nu naast elkaar liggen (zie verder). Tussen twee naast elkaar liggende vakjes kunnen er zich echter een aantal verschillende dingen bevinden, zoals de muren uit deel 1 en nu ook deuren. Elk vakje moet minstens in één richting afgebakend zijn door een muur of een deur. Deuren kunnen verder nooit geplaatst worden in de vloer van een vakje. Tenslotte kan een vakje nooit meer dan 3 deuren bevatten.

### 1.1 Niets

Wanneer er niets tussen aangrenzende vakjes ligt, zijn ze samengevoegd, en hun temperatuur en vochtigheidsgraad zijn gelijk. Hierdoor ontstaan ruimtes, die verderop in deze opgave in detail besproken worden.

### 1.2 Muur [*Wall*]

Een muur tussen twee vakjes wordt door die beide vakjes gedeeld (het zijn dus geen twee ‘deelmuren’ die naast elkaar liggen, en wanneer de muur van een vakje wordt verwijderd, wordt die dus ook voor het ernaast liggende vakje verwijderd). Een muur zorgt ervoor dat de beide vakjes perfect geïsoleerd zijn, en men kan er niet doorheen bewegen.

Een muur heeft de eigenschap dat hij al dan niet uit zichzelf glad kan zijn. Dit wordt voor elke muur voor eens en voor altijd beslist wanneer de muur wordt gebouwd. Een eventuele gladheid geldt voor beide kanten van de muur.

### 1.3 Deur [*Door*]

Met een deur bedoelen we eigenlijk een muur met een deur erin (een deur ter grote van een muur zou er nogal vreemd uitzien). Deze muren (en de deur die erin zit) zijn echter nooit uit zichzelf glad.

Een deur kan zich in open of gesloten toestand bevinden. Wanneer een deur open is, zijn de ernaast liggende vakjes samengevoegd, anders zijn ze geïsoleerd. Men kan de toestand van een deur uiteraard veranderen; dit gebeurt *totaal*.

## 1.4 Constructie

Bij constructie van een vakje moet het mogelijk zijn om muren te voorzien in bepaalde richtingen. De constructor mag zelf de specifieke karakteristieken van die muren bepalen (gladheid, deur). Verder moet het mogelijk zijn om voor een vakje in een gegeven richting (boven, onder, enz.) een nieuwe muur of deur te bouwen, respectievelijk om een bestaande muur of deur af te breken. Muren of deuren die toegevoegd worden mogen niet reeds grenzen aan bestaande vakjes.

Zorg ervoor dat er een bidirectionele associatie bestaat tussen borders en vakjes.

## 2 Kerkers [*Dungeons*]

Kerkeren zijn verzamelingen van vakjes, waarin deze coördinaten krijgen.

### 2.1 Dimensies [*Dimensions*]

Elke Dungeon heeft een uniek driedimensionaal assenstelsel, waarbij coördinaten in x (links/rechts), y (voor/achter) en z (boven/onder) worden gebruikt. Hierbij is de {0,0,0}-coördinaat het vakje aan de voorkant onderaan links en hebben alle andere vakjes positieve coördinaten die boven en/of achter en/of rechts hiervan liggen. Achter komt overeen met het noorden, en rechts met het oosten.

Een dungeon heeft voor al zijn dimensies een maximum, dat voor elke dimensie en elke dungeon kan verschillen, maar voor geen enkel geval ooit groter zal zijn dan `long.MAX_VALUE`. Dit maximum kan tijdens de levensduur van het object enkel nog vergroten.

Alles in verband met de dimensies van dungeons moet *defensief* worden uitgewerkt.

### 2.2 Vakjes in Kerkeren

Kerkeren zijn opgebouwd uit vakjes. Elk vakje bevindt zich op een welbepaalde positie in een kerker. Daarbij is het onmogelijk dat een vakje zich op een coördinaat  $(x, x, x)$  bevindt, i.e. posities die in alle dimensies even ver van de oorsprong liggen. Verder is het zo dat niet meer dan 20% van de vakjes binnen een kerker glad kunnen zijn.

**Update:** Hierbij bedoelen we vakjes die intrinsiek glad zijn (en niet om vakjes die door ijsvorming of door waterafzetting glad worden).

Wanneer een vakje volgens het assenstelsel onmiddellijk naast een ander vakje ligt, moeten beide vakjes zich gedragen als burens. Als er een obstakel geplaatst is op hun grens, moeten ze dus dat obstakel delen. In het andere geval behoren ze tot dezelfde ruimte.

Vakjes op zich zullen nooit weten in welke kerker ze zich precies bevinden, laat staan op welke positie ze zich zouden bevinden. Daarmee is het onmogelijk om in alle mogelijke omstandigheden te verhinderen dat een vakje in meerdere kerkers zou voorkomen. Waar het enigszins kan, moeten dergelijke inconsistenties worden uitgesloten.

Er moet een methode voorzien worden om een vakje op een bepaalde coördinaat toe te voegen aan een kerker. Hierbij is het uiteraard mogelijk dat het toe te voegen vakje zal gaan grenzen aan vakjes die reeds deel uitmaken van de betrokken kerker. In een dergelijk geval wordt het meest gesloten obstakel aan de gemeenschappelijke grens behouden. Muren hebben daarbij dus voorkeur boven deuren, en deuren op hun plaats boven niets. Als er ten gevolge van de samenvoeging vrij kan bewogen worden van het ene vakje naar het andere, moeten temperatuur en vochtigheidsgraad aangepast worden. Dergelijke vakjes creëren ruimtes, die we verderop bespreken.

Naast het toevoegen van vakjes, moet ook het verwijderen ervan worden ondersteund. Bij het verwijderen van een vakje blijven eventuele obstakels met aangrenzende vakjes behouden.

Verder moet een methode worden aangeboden die in (haast) constante tijd kan aangeven of er op een welbepaalde positie al dan niet een vakje staat.

### **3 Ruimtes**

Ruimtes ontstaan doordat vakjes aan mekaar grenzen en niet gescheiden zijn door een obstakel. Een open deur wordt in dit geval niet beschouwd als een obstakel, een gesloten deur uiteraard wel. Wellicht is het niet nodig om ruimtes expliciet als objecten te modelleren. De volgende vereisten moeten wel voldaan zijn:

#### **3.1 Temperatuur en vochtigheid**

##### **Enkel voor groepjes van 2!**

Alle vakjes binnen dezelfde ruimte moeten dezelfde temperatuur en vochtigheid hebben. Wanneer er onevenwichten dreigen te ontstaan (bv. door het toevoegen van een vakje, door het veranderen van de temperatuur, ...), moeten temperatuur en vochtigheid van de ganse ruimte

onmiddellijk worden teruggebracht tot het gewogen gemiddelde van de waarden van alle vakjes. Beschouw als voorbeeld een ruimte van 3 vakjes met een temperatuur van 60 graden. Als je de temperatuur van een van die vakjes met 30 graden verhoogt, zal de temperatuur van de 3 vakjes in de ruimte opgehoogd zijn met 10 graden.

Vakjes aan de randen van een kerker zijn niet noodzakelijk afgebakend door muren. We nemen aan dat kerkers zich bevinden in een soort vacuum, waarbij geen overdracht van temperatuur, vochtigheid, ... plaats vindt.

*Merk op dat dit verschillend is van de **mergeWith** methode uit deel 1 van het project!*

*Voor studenten die alleen werken, mogen er temperatuur- en vochtigheidsverschillen aanwezig zijn binnen een ruimte.*

### 3.2 Bereikbaarheid

Er moet een methode worden voorzien waarmee kan nagegaan worden of een vakje behoort tot dezelfde ruimte als een ander vakje. Van deze methode moet de tijdscomplexiteit in de documentatie worden weergegeven. Verder moet er ook een methode zijn die de ruimte teruggeeft waartoe een vakje behoort in de vorm van een collectie van alle vakjes die tot die ruimte behoren [**Deze laatste methode is enkel voor groepjes van 2**].

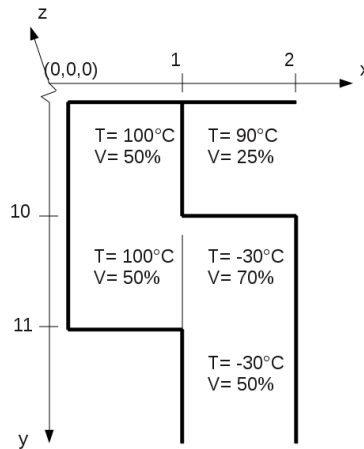
## 4 Hoofdprogramma

De werking van de ontwikkelde klasse moet je kunnen demonstreren. Schrijf hiervoor een hoofdprogramma dat de vakjes in figuur 1 aanmaakt. Print hierbij voor het vakje op coördinaten  $(0, 10, 0)$  de temperatuur, vochtigheidsgraad en de grootte van zijn ruimte af. Open vervolgens de deur die vakjes  $(0, 11, 0)$  en  $(1, 11, 0)$  met elkaar verbinden en print dezelfde informatie opnieuw uit.

**Update:** De coördinaten van de vakjes is gewijzigd om excepties in het hoofdprogramma te vermijden.

## Praktische Richtlijnen

We verwachten dat dit deel van het project volledig kan worden afgewerkt in 30 uur. Hierbij veronderstellen we dat de materie die aan bod komt in de cursus, vooraf volledig verwerkt is. Daarenboven kan het project enkel binnen die tijdsspanne worden uitgewerkt, indien je op een beredeneerde en gedisciplineerde manier te werk gaat. We raden sterk aan om de principes



Figuur 1: De plattegrond van de dungeon uit het hoofdprogramma.

van *Extreme Programming* zoveel mogelijk toe te passen bij de uitwerking van het project.

## 5 Verdere richtlijnen rond de oplossing

1. Enkel de expliciet opgelegde vereisten moeten voorzien worden. Je zal geen indruk maken door bijkomende spelelementen te voorzien in je oplossing.
2. Hoewel de opgave vrij uitgebreid is, hebben we niet gepoogd alle mogelijke details in te vullen. Alles wat niet uitdrukkelijk vermeld is in de opgave mag je dan ook naar eigen inspiratie invullen. Kies daarbij steeds voor de gemakkelijkste oplossing.
3. Dit project moet uitgewerkt worden in Java 5 of hoger.
4. Als opgelegd wordt dat methodes rond een bepaalde karakteristiek nominaal, totaal of defensief moeten uitgewerkt worden, geldt dit ook voor het manipuleren van die karakteristieken in meer complexe methodes (vb. constructoren).
5. Voor de klasse van de vakjes moet alle aspecten van de documentatie formeel en informeel worden uitgewerkt. Voor de klasse van de kerkers volstaat een formele specificatie. Voor alle andere klassen in je oplossing mag je zelf kiezen tussen een formele of een informele specificatie.

## 6 Inleveren

1. Het project moet ingediend worden vóór vrijdag 29 april om 9u. Afwijkingen op dit tijdstip zullen slechts in zeer uitzonderlijke gevallen worden toegestaan.
2. Het indienen gebeurt via Toledo. Dien je oplossing als **JAR-bestand** in. Je kan met de Eclipse omgeving eenvoudig het JAR-bestand genereren met de export functie die je o.a. in het file menu vindt. Let erop dat je al je ontwikkelde klassen selecteert (Select the resources to export), en neem zowel de .java source files als de gegenereerde .class files op! Vermeld je naam en studierichting in commentaar bij het indienen. Bij het indienen moet je bevestigen tot je oplossing effectief werd ingeleverd!

Mocht deze opgave op bepaalde punten onduidelijk zijn, zal verdere toelichting gegeven worden via Toledo. Om tegenstrijdigheden in antwoorden te vermijden, kunnen vragen (enkel i.v.m. de opgave) uitsluitend gestuurd worden naar: [ogp-project@cs.kuleuven.ac.be](mailto:ogp-project@cs.kuleuven.ac.be)

**Veel succes!**