

# Informatica

## Modulehandleiding ipose

### Praktische Opdracht Software Engineering

### A-mazing Challenge 20



Challengeweek: Software Engineering  
Collegejaar: 2016-2017  
Auteur: Thomas Boose, Michiel Boere, Roland Westveer  
Hogeschool Leiden: Opleiding informatica

Hogeschool Leiden,  
Opleiding Informatica

April 2017  
Versie 3.0

---

# Inhoud

Wat zit er in de enveloppe? .....	4
Synopsis.....	4
Wat heb je nodig? .....	5
Leerdoelen .....	5
Planning .....	7
Fase 1 aanvang (maandag 10 april 8:30 uur - dinsdag 11 april 12:00 uur) .....	7
Fase 2 detaillering (dinsdag 11 april 12:00 uur - woensdag 12 april 17:00 uur) .....	7
Fase 3 bouw (woensdag 12 april 12:00 uur - donderdag 13 april 12:00 uur): .....	7
Fase 4 verander de Maze (dinsdag 11 april 8:30 uur – donderdag 13 april 8:30) .....	8
Fase 5 oplevering (donderdag 13 april 12:00 uur - donderdag 13 april 16:00 uur) .....	8
Voorbeelddefinitie glade.....	9
Beschrijving van de taal 20 .....	9
Lijst van hardware-, software- en verbruikskosten.....	10
De tegels van the glade .....	11
Bom: [B0, B1 .. B8].....	11
Kleur: [C0..C8] .....	11
Doel: [D1 .. D9].....	11
Bonus: [E1 .. E9] .....	11
Obstakel: [O0 .. O3].....	11
Draai: [R0 .. R3] .....	11
Start: [S0 .. S3].....	12
Uitwerking van een glade .....	12
Voorbeeldprogramma .....	13
Bijlage 1: Beoordelingsmodel A-mazing Challenge 20 .....	14
Bijlage 2: Syntax specificatie 20 .....	16

# Versiebeheer

<b>Versie</b>	<b>Datum</b>	<b>Status</b>	<b>Wijzigingen</b>
0.1	01-03-2015	Pre final	Resultaat van 6 maanden collaboratie
0.2	14-03-2015	Pre final	Verificatie verwerk en afspraken met OM
1.0	21-4-2015	Final	
1.1	12-02-2016	Pre Final	Nieuwe ronde nieuwe kansen
2.1	3-04-2017	Pre Final	Aanpassingen gedaan voor nieuwe challenge

# Wat zit er in de enveloppe?

- Deze beschrijving en zo niet, dan weet je niet wat je mist :-)
- Vijf tickets voor "The Maze Runner" in bioscoop Lido maandag 10 april om 16:00 uur.
- Vijf schriften met lijntjes en vijf pennen om aantekeningen te kunnen maken tijdens de colleges.
- Een groot schrift met centimeter ruitjes om glades te schetsen.
- Een setje kleurpotloden om de glades te schetsen.
- Inlogcode voor de "zandbak-website" waar jullie kunnen oefenen met het oplossen van glades.
- Een HS Leiden CardBoard.

## Synopsis

Deze challenge draait om the glade. The glade is een vierkant veld, omringd door een doolhof van hoge muren. Het was altijd relatief rustig in the glade totdat er een jongen, genaamd Thomas verscheen. Hij doodt een griever en daardoor is het doolhof van slag. Grieviers zijn robots en deze moeten geprogrammeerd worden.

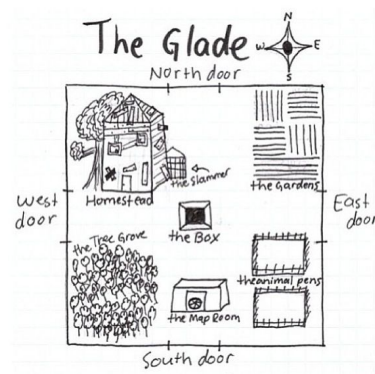
Voorheen gingen 's nachts alle openingen van de Maze dicht waardoor er een relatieve rust heerste in the glade. Maar sinds Thomas een griever doodde gaan de deuren niet meer dicht.

De bewoners van the glade besluiten om zich te beschermen tegen de grieviers door obstakels op te werpen in the glade. Aangezien grieviers robots zijn, en geprogrammeerd worden, hopen de bewoners op deze manier de belangrijkste voorzieningen, ofwel Doelen, te kunnen beschermen tegen deze "relatief" domme apparaten.

Doelen mogen natuurlijk niet afgesloten worden. De runners moeten nog wel terug kunnen komen. Alle doelen moeten bereikbaar blijven maar moeilijk bereikbaar zodat de domme grieviers op afstand gehouden worden.

Dit is de context van deze challenge. Jullie spelen in deze challenge in groepen van vijf. Eerst als bewoner van de glade en daarna de programmeur van een griever. Als bewoner is het belangrijk om the glade zo te bouwen dat het niet makkelijk voor een griever is om de doelen te bereiken. Er moet echter een beloopbaar pad zijn van start tot einde voor de runners.

Als programmeur van een griever is het belangrijk om een programma te schrijven dat de griever efficiënt en effectief naar de bron(-nen) of doel(-en) van the glade leidt.



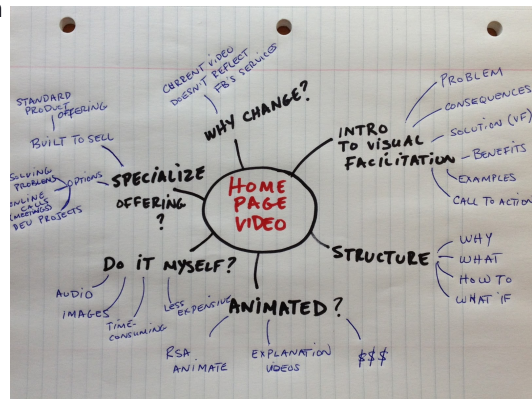
# Wat heb je nodig?

Om deze challenge aan te kunnen gaan, zullen jullie deze instructie moeten bestuderen. Ook is het belangrijk om de film "The Maze Runner" te hebben gezien zodat jullie een gedeelde vocabulaire opbouwen. Daarom vinden jullie in de enveloppe ook vijf tickets voor deze film.

Daarnaast bieden wij jullie in fase 1 voorbeeldopgaven aan in een zandbak omgeving zodat jullie een beeld kunnen krijgen van de mogelijkheden die je als bewoner van de glade hebt en de mogelijkheden die een programmeur van een griever heeft. Naast inhoudelijke workshop komen er weblectures beschikbaar over relevante onderwerpen. De balie is de hele week geopend om jullie vragen te beantwoorden.

Het is belangrijk dat de student die een workshop volgt aantekeningen maakt zodat hij of zij de kennis die aangedragen is, kan overdragen aan de rest van de groep.

Om alle groepsleden de mogelijkheid te bieden om zijn of haar ideeën kenbaar te maken aan de groep hebben we voor genoeg ruitjespapier en kleurpotloden gezorgd.



Wij raden jullie aan om de ideeën eerst op papier uit te werken, te overleggen in de groep en daarna één persoon verantwoordelijk te maken voor het daadwerkelijk uitwerken van the glade of de programma's op de website.

# Leerdoelen

De challenge doen we niet alleen omdat het heel erg leuk is. We willen ook graag dat jullie er iets van leren. Naast de leerdoelen die voor alle challenges gelden, hebben we leerdoelen specifiek voor deze challenge beschreven.

- Rekenen: Er zal van jullie verwacht worden dat je veel berekeningen maakt. Met name bij het bepalen van de meest efficiënte strategie bij het programmeren van een griever wint de groep die het best zijn sommetjes uitwerkt. Het gaat hierbij niet om zeer complexe berekeningen maar eenvoudigweg; optellen, aftrekken, vermenigvuldigen en delen; maar dan wel zeer precies.
- Uitdagen: Bij het bouwen van the glade zullen jullie als bewoner de programmeurs van grieviers moeten uitdagen. Het is niet goed voor je eigen score als niemand jullie glade kan oplossen. Het aantal verschillende goede oplossingen dat andere groepen inleveren voor jullie glade bepaalt namelijk een groot deel van je cijfer. m.a.w:
  - Stel dat niemand jullie glade oplost, dan verdien je 0 van de 2 punten.
  - Zijn alle oplossingen van de groepen die jullie glade oplossen goed, maar verschillend in efficiëntie (verbruik), dan verdien je 2 van de 2 punten.
  - Is je eigen oplossing ondanks dat, het meest efficiënt, dan heb je nog eens 1 punt te pakken.
- Oplossen en Programmeren: Een deel van je cijfer hangt ervan af hoe efficiënt en effectief jullie zijn in het bepalen van een strategie om de uitdagingen van de andere groepen op te lossen. Natuurlijk is het oplossen niet genoeg, je moet de oplossing ook nog programmeren. Maar grieviers zijn geen standaard apparaten. De taal waarin ze geprogrammeerd moeten worden is niet bepaald standaard.
- Code interpreter maken: Breidt de Maze uit door de Python interpreter aan te passen. Bedenk en realiseer je een leuke nieuwe mogelijkheid in de Maze dan verdien je 2 punten.

- Visualiseren en vormgeven: Het laatste leerdoel betreft het visualiseren en vormgeven van jullie uitdaging / glade. Daarbij is het belangrijk dat je zo realistisch mogelijk, in tijd en vorm de oplossing van je eigen uitdaging / glade weergeeft. Elke groep krijgt een 3D bril van ons om met behulp van Unity een 3D visualisatie van je Glade te maken. Programmeer in Unity hoe de Griever door jullie Maze loopt volgens jullie optimale oplossing. Maak een filmpje van de 3D visualisatie.

## Eindeisen

De volgende eindeisen zijn opgesteld voor deze challenge:

Deel A (het maken van een uitdagende opdracht)

Groep maakt een uitdagende opdracht, Glade, die een discriminerende werking heeft.

Groep voorziet deze opdracht, Glade, van een werkende voorbeeldoplossing.

Groep maakt geen vergissingen of rekenfout waardoor de eigen oplossing slechter scoort dan veel / het merendeel van de oplossingen van de collega groepen.

Groep scoort zelf relatief hoog ten opzichte van de topscore van alle groepen.

Deel B (het oplossen van de uitdagingen van andere groepen)

Het aantal overwonnen uitdagingen is groot t.o.v. het totaal aantal aangeboden opdrachten.

Het overgebleven deel van het beschikbare budget is hoog t.o.v. dat van de best presterende groep.

Deel C (realiseren van een software oplossing)

Breidt de Glade omgeving uit door nieuwe mogelijkheden in de Glade te scheppen binnen de daarvoor gestelde kaders. De Glade moet wel blijven werken bijvoorbeeld.

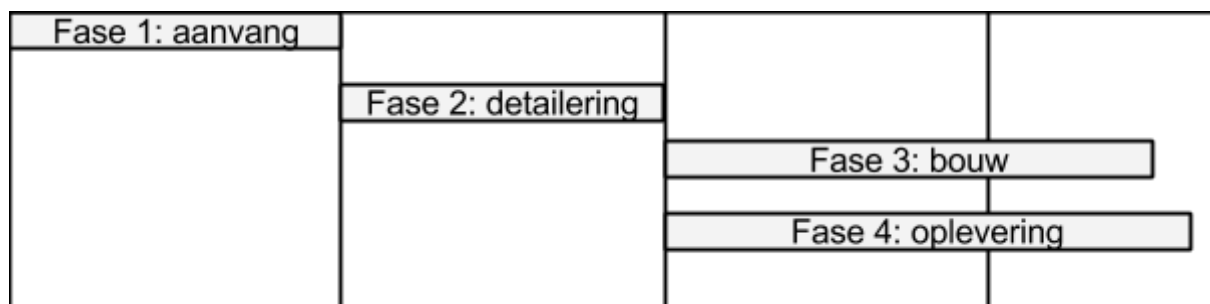
Deel D (visualisatie en vormgeving)

Groep visualiseert de eigen opdracht / uitdaging en de eigen oplossing.

Groep geeft vorm aan de eigen opdracht.

# Planning

Na de kickoff start fase 1.



## Fase 1 aanvang (maandag 10 april 8:30 uur - dinsdag 11 april 12:00 uur)

In deze fase worden er een workshop over syntaxdiagrammen aangeboden en een workshop over 3D visualisatie in Unity. Hier kunnen jullie per groep één student naartoe afvaardigen. Zorg ervoor dat je pen en papier bij je hebt om aantekeningen te maken. Daarnaast zijn er weblectures beschikbaar ter ondersteuning van de andere fases zoals, kostenkaart en de Python interpreter aanpassen.

Aan de hand van deze workshop en weblectures, deze beschrijving en jullie eigen gezonde verstand kunnen jullie in de zandbak-omgeving een aantal voorbeelduitdagingen proberen op te lossen door je griever te voorzien van een programma.

## Fase 2 detaillering (dinsdag 11 april 12:00 uur - dinsdag 11 april 17:00 uur)

Als fase 2 start plaatsen wij een bericht op ELO. Jullie kunnen dan de inlogcodes van de productieomgeving ophalen bij de balie. Vanaf dat moment kunnen jullie bij de balie de inlogcodes van de productieomgeving halen.

Vervolgens ontwerpen jullie een eigen versie van the glade. Hiervoor is een editor beschikbaar, maar het is verstandig om eerst een ontwerp op papier te maken. Behalve dat jullie de obstakels, doelen en vallen plaatsen, bepalen jullie ook de kosten voor de verschillende hardware onderdelen, de kosten voor het programmeren en de kosten voor het verbruik van de griever of runner door middel van het invullen van een kostenkaart.

Als proof of concept, bewijs dat jullie uitdaging opgelost kan worden, schrijven jullie een programma dat een runner kan volgen om de uitdaging op te lossen. Dit programma mag maximaal 20 regels code bevatten en dient binnen budget alle doelen te bereiken.

De meeste werkzaamheden, met name het denkwerk, van fase 2 kunnen jullie op papier al gedaan hebben voordat jullie de inloggegevens van de productieomgeving ontvangen. Als jullie dit goed gedaan hebben, dan is het invoeren een kleinigheid.

Zodra jullie tevreden zijn met jullie glade, de kostenkaart en jullie programma, publiceren jullie deze en starten direct de fases 3 en 4. Let op: Jullie mogen jullie glade slechts drie keer proberen te publiceren. Na drie pogingen gaat jullie account op slot en kan je dit onderdeel niet meer behalen. Het is wel mogelijk om jullie glade lokaal te testen. Het publiceren hiervan moet uiterlijk op dinsdag 11 april om 17:00 uur gedaan zijn. Zo niet, dan eindigt de challenge en hebben jullie een onvoldoende.

## Fase 3 bouw (dinsdag 11 april 12:00 uur - donderdag 13 april 13:00 uur):

Tijdens fase 3 programmeren jullie voor iedere uitdaging een griever. Voor alle glades kun je maximaal drie opleveringen doen. Wees dus zuinig op je pogingen en probeer bij de derde oplevering geen nieuwe trucjes. Een deel van jullie cijfer wordt bepaald door het budget dat jullie over hebben en het aantal uitdagingen dat jullie weten op te lossen.



#### **Fase 4 verander de Maze (dinsdag 11 april 8:30 uur – donderdag 13 april 13:00)**

Direct na aanvang van de challengeweek kan je met je groep aan de slag om de Maze zelf te wijzigen. Wij stellen hiervoor de broncode beschikbaar. Waar kan je aan denken: Jullie kunnen bijvoorbeeld de Griever meer mogelijkheden geven om te lopen of de Maze meer mogelijkheden geven, zoals geheime deuren of speciale bommen. Denk eraan om wel de kostenkaart aan te passen. Jullie behalen deze fase door de aangepast broncode in te leveren bij de balie en met de hele groep een kort assessment te doen bij de balie. Zorg ervoor dat de broncode uiterlijk donderdagochtend 13 april voor 8:30 in ons bezit is.

#### **Fase 5 oplevering (donderdag 13 april 08:30 uur - donderdag 13 april 13:00 uur)**

Fase vijf start direct na fase 2. In deze fase maken jullie een visualisatie van jullie eigen glade en het programma dat jullie zelf hebben gemaakt voor de runners.

Hierbij is het belangrijk dat er sprake is van een vloeiende lijn, dat de tijd voor het lopen van één stap en het draaien gelijk is aan één seconde zoals deze opdracht voorschrijft en dat jullie glade waarheidsgetrouw en aantrekkelijk gevisualiseerd is.

Er is een bonus te verdienen als jullie naast de visualisatie ook de stappen door het programma laten zien, zoals "Step into" in de debugger van Eclipse. Ook kan het wisselen van camerastandpunt (runner view, bird view, target view) op extra waardering rekenen.

Jullie uploaden de film in de vorm van een video-bestand naar elo voor de uiterlijke deadline: donderdag 13 april 13:00 uur. Het videobestand moet afgespeeld kunnen worden in VLC en mag niet groter zijn dan 100 MB.

Tussen 13:00 en 15:00 presenteren jullie je eigen Glade en oplossing aan elkaar en de docenten. Om 16:00 maken wij de beoordelingen bekend en is de centrale afsluiting.



# Voorbeelddefinitie glade

Om een beeld te krijgen van de werking van het spel volgt hier een voorbeeld van een glade.

Een glade is een matrix van 20 bij 20 cellen die allen van een bepaald type zijn. Iedere cel is dus een obstakel, een gekleurde tegel waarover een griever of runner kan lopen, een bom, een draaiende plaat, een doel, een startpunt of een bonus veld waar extra geld te verdienen is.

Bij een glade hoort een lijst met kosten voor de aanschaf van hardware, het programmeren van programmacode (software) en het verbruik van resources, tijd en ruimte. Het startkapitaal van een runner of griever is altijd 2020 euro.

Daarnaast dient een glade voorzien te worden van een voorbeeldprogramma waarmee het minimaal binnen budget opgelost wordt. Bedenk wel dat jullie veel tijd kunnen nemen voor het maken van jullie glade en het programma voor de runners terwijl een programmeur van een griever slechts beperkt de tijd heeft om hem op te lossen.

## Beschrijving van de taal 20

Een voorbeeldprogramma waarmee een runner alle doelen binnen een glade kan bereiken, is onderdeel van de oplevering van een glade. Hier volgt een beknopte beschrijving van de taal 20. Als bijlage vind je een formele specificatie in de vorm van syntaxdiagrammen.

De verschillende vergelijkingsoperatoren die je mag gebruiken zijn:  $==$ ,  $<$ ,  $>$ ,  $!=$ .  
Deze operatoren kun je inzetten bij een "Zolang" of een "Als" regel.

De rekenkundige operatoren die je kunt gebruiken zijn  $+$ ,  $-$ ,  $*$ ,  $\%$  en  $/$  er bestaan enkel integer variabelen, hele getallen,  $\%$  geeft de restwaarde van een breuk en  $/$  het hele deel:

$5 / 3 = 1$ $5 \% 3 = 2$
-----------------------------

*In tegenstelling tot wat je op school hebt geleerd worden alle operaties op volgorde verwerkt. De expressie:  $2 + 3 * 7 \% 4$  wordt verwerkt alsof er staat:  $((2 + 3) * 7) \% 4$*

In bijlage 2 vind je een formele definitie van de syntax van de taal 20

## Lijst van hardware-, software- en verbruikskosten

Dit is een voorbeeld van een kostenoverzicht bij een glade. De items staan vast, de kosten worden door de bewoners van the glade vastgesteld:

Startkapitaal: 2020

Hardware (aanschaf)		Verbruik		Software (per geschreven statement)	
kompas	10	stapVooruit	2	zolang (lus)	4
zwOog (alles is wit behalve zwart)	3	stapAchteruit	3	als (keuze)	4
kleurOog	15	draaiLinks	5	opdracht	2
variabele (a .. z)	1	draaiRechts	5	toekenning	2
		zwOog	1		
		kleurOog	1		
		kompas	1		
		duw obstakel (schade)	65K		
		toewijzing (a = 1) (het uitvoeren van een toekenning)	1		
		operatie (+, - , *, %, /)	2		
		vergelijking (=, <, >, !=)	1		

## De tegels van the glade

Een glade is een matrix van 20 bij 20 van tegels van een bepaald type. Dit type wordt aangeduid met 2 tekens. Deze wereld kan worden gedefinieerd d.m.v. 20 regels met ieder 40 tekens:

### Bom: [B0, B1 .. B8]

Een bom is een tegel die explodeert. Als de griever of een runner zich op de tegel bevindt tijdens het exploderen dan is de griever of runner kapot. Een B0 explodeert direct bij betreden. Een [B1 .. B8] explodeert 1, 2 .. 8 seconden na het betreden door een griever of runner. Zowel het draaien van een griever of runner als het zetten van een stap kost 1 seconde. Op een B1 kun je dus niet draaien maar je kunt er wel overheen stappen. Na de explosie wordt een Bx veld een 00 veld (puin). De kleur van de bom is zwart(0)

### Kleur: [C0..C8]

Een kleur-tegel is een tegel waarop een griever of runner kan staan. Met een "oog" kan de kleur van de vloer afgelezen worden. Een zwart/wit oog ziet alles wit behalve de zwarte tegels.

	Zwart wit oog	Kleuren oog
Wit	1	8
Grijs	1	7
Rood	1	6
Oranje	1	5
Geel	1	4
Groen	1	3
Blauw	1	2
Paars	1	1
Zwart	0	0

### Doel: [D1 .. D9]

Een doel moet bereikt worden. Meerdere doelen moeten op volgorde worden bereikt. Je mag over een doel heen lopen en er later terugkeren als het aan de beurt is. Er zijn per glade maximaal 9 doelen: D1 .. D9. Ieder doel komt hooguit 1 keer voor en er is minimaal 1 doel gedefinieerd. Als je op het laatste doel stapt, nadat alle doelen op volgorde zijn bereikt, ben je klaar. Het programma hoeft niet zelf te "stoppen" bij het laatste doel. De kleur van een doel is geel (4)

### Bonus: [E1 .. E9]

Bij een bonusveld kan een griever of runner extra geld verdienen. Een "Ex" veld levert  $2^x$  extra euro op. Bijvoorbeeld: een E2 veld levert 4 euro op en een E8 veld 256 euro. De kleur van een bonusveld is geel (4). Nadat de bonus is opgepikt blijft er een lege gele tegel achter. Per glade kan iedere bonus [E1 .. E9] maar 1 keer voorkomen.

### Obstakel: [O0 .. O3]

Tegen een obstakel kun je aanlopen. De griever of runner blijft dan staan maar duwen kost wel geld. Er zijn verschillende soorten obstakels: O0=Puin, O1=Heg, O2=Steen, O3=Hout

### Draai: [R0 .. R3]

Als een griever of runner op een draaivlak staat dan draait hij. Bij R1 draait de griever of runner 90 graden met de klok mee, R2 180 graden met de klok mee, R3 270 graden per seconde met de klok mee. Als een griever of runner op een R0 veld komt draait hij een willekeurig aantal graden (0,90,180 of 270). Je heb een kompas nodig om na een willekeurige draai te weten in welke richting je staat.

kompas	
Noord	0
Oost	1
Zuid	2
West	3

Als je op een R1 veld stapt en er direct weer afstapt ben je 90 graden gedraaid maar als je zelf een draaibeweging maakt op een R1 veld dan draait de schijf een 2e keer voordat je er weer af bent. Een draaibeweging van een griever of runner kost immers ook 1 seconde net als een stap. De kleur van een draaischijf is blauw (2).

### Start: [S0 .. S3]

Er is in iedere glade 1 veld waarop de griever of runner start. De griever of runner kan in vier richtingen worden opgesteld: Noord = 0, Oost = 1, Zuid = 2 en West = 3. De kleur van het startveld is zwart (0).

### Uitwerking van een glade

O1	O1	O1	O1	O1	O1	O1	O1	O1	O1	O1	O1	O1	O1	O1	O1	O1	O1	O1	O1
O1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	D1	C1	C1	C1	C1	B0
O1	O3	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	B0
O1	B0	O3	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	B0
O1	O3	O3	O3	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	R2	C1	C1	C1	C1	B0
O1	C1	C1	B0	O3	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	B0
O1	C1	C1	C1	O3	O3	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	B0
O1	C1	C1	C1	C1	O3	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	B0
O1	C1	C1	C3	C1	C4	C1	C1	C1	C1	C1		C1	C1		C1	C1	C1	C1	B0
O1	C1	C1	C1	C1	C1	C1	B0	O3	B0	O3	B0	O3	B0	O3	B0	O3	B0	O3	B0
O1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	B0
O1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	B0
O1	C1	C1	E4	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	B0
O1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	B0
O1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	B0
O1	C1	C1	C1	C1	C1	O2	O2	O2	O2	O2	O2	O2	O2	O2	O2	O2	O2	O2	B0
O1	C1	C1	S0	C1	C2	O2	O2	O2	O2	O2	O2	O2	O2	O2	O2	O2	O2	O2	B0
O1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	C1	E6	C1	C1	C1	C1	B0
O1	B0	B0	B0	B0	B0	B0	B0	B0	B0	B0	B0	B0	B0	B0	B0	B0	B0	B0	B0

## Voorbeeldprogramma

Bij een ontwerp van een glade moet ook een programma / oplossing worden geleverd die werkt. Deze oplossing moet een runner alle doelen laten bereiken zonder dat het budget op raakt. Het voorbeeldprogramma mag niet meer dan 20 regels code omvatten.

<b>Programma</b>	<b>Softw</b>	<b>Hardw</b>	<b>Verbruik</b>
gebruik kleurOog		-15	
gebruik a		-1	
zolang kleurOog != 3 {	-4		9 * -2 (verg.&kleur)
stapVooruit	-2		8 * -2
}			
draaiRechts	-2		1 * -5
zolang kleurOog != 0 {	-4		9 * -2
stapVooruit	-2		8 * -2
}			
stapVooruit	-2		1 * -2
zolang kleurOog != 0 {	-4		3 * -2
stapVooruit	-2		2 * -2
}			
draaiLinks	-2		1 * -5
a = 7	-2		1 * -1
zolang a > 0 {	-4		7 * -1
stapVooruit	-2		7 * -2
a = a - 1	-2		6 * -3 (toek.&oper)
}			
=====			
	-34	-16	-130 = <b>-180</b>

# Bijlage 1: Beoordelingsmodel A-mazing Challenge 20

Dit beoordelingsmodel bestaat uit 4 delen.

Deel A en B, beide ter waarde van 3 punten, zullen automatisch gescoord worden door het ipose systeem dat jullie gedurende deze week gebruiken om je Glade te publiceren en de oplossingen in te leveren. De stand is continue te volgen op de ipose site. Publicaties en oplossingen van andere groepen kunnen invloed hebben op deze score. Streef dus naar een ruime marge om teleurstellingen aan het einde van de challengeweek te voorkomen.

Deel C, ter waarde van 2 punten, kan je behalen door deze opdracht in te leveren bij de balie en vervolgens met de hele groep uitgenodigd worden door de examinerator om een kort assessment te ondergaan.

Deel D, ter waarde van 2 punten, zal op de laatste dag van de challenge door de examinerator beoordeeld worden.

Onderdeel	Score toelichting	Score
Deel A: Eigen glade		
Maken werkende glade, kostenkaart en voorbeeld oplossing	Go / No go	0
Discriminerende werking	Aantal verschillende scores / totaal aantal scores	1
Toernooi prestatie	Ranking van de eigen oplossing / totaal aantal scores	2
Relatieve prestatie	Eigen score / top score eigen glade	3
Eindscore	als 0 == "no go" dan 0 anders $\text{score}[1] + \text{score}[2] + \text{score}[3]$	A

Onderdeel	Score toelichting	Score
Deel B: glades oplossen		
Oplossen	Aantal werkende oplossingen / totaal aantal glades	1
Overgebleven budget	Som (overgebleven budget) / topscore	2
Eindscore	$(\text{score}[1] * 2) + \text{score}[2]$	B

Onderdeel	Score toelichting	Score
Deel C: Maze veranderen		
Implementatie in Python	Examinator Grade [0,1]	1
Aanpassen kostenkaart	Examinator Grade [0,1]	2
Eindscore	score[1] + score[2]	C

Onderdeel	Score toelichting	Score
Deel D: Visualisatie		
Vormgeving eigen glade	Examinator Grade [0,1]	1
Visualisatie eigen glade	Examinator Grade [0,1]	2
Eindscore	score[1] + score[2]	D

Onderdeel	Score toelichting	Score
Totaal eigen glade	Verplicht > 1 geen compensatie mogelijk	1
Totaal wedstrijd	Verplicht > 1 geen compensatie mogelijk	2
Totaal maze	Compensatie van een 1 is mogelijk	3
Totaal visualisatie	Compensatie van een 1 is mogelijk	4
Eindscore	som(A,B,C,D)	

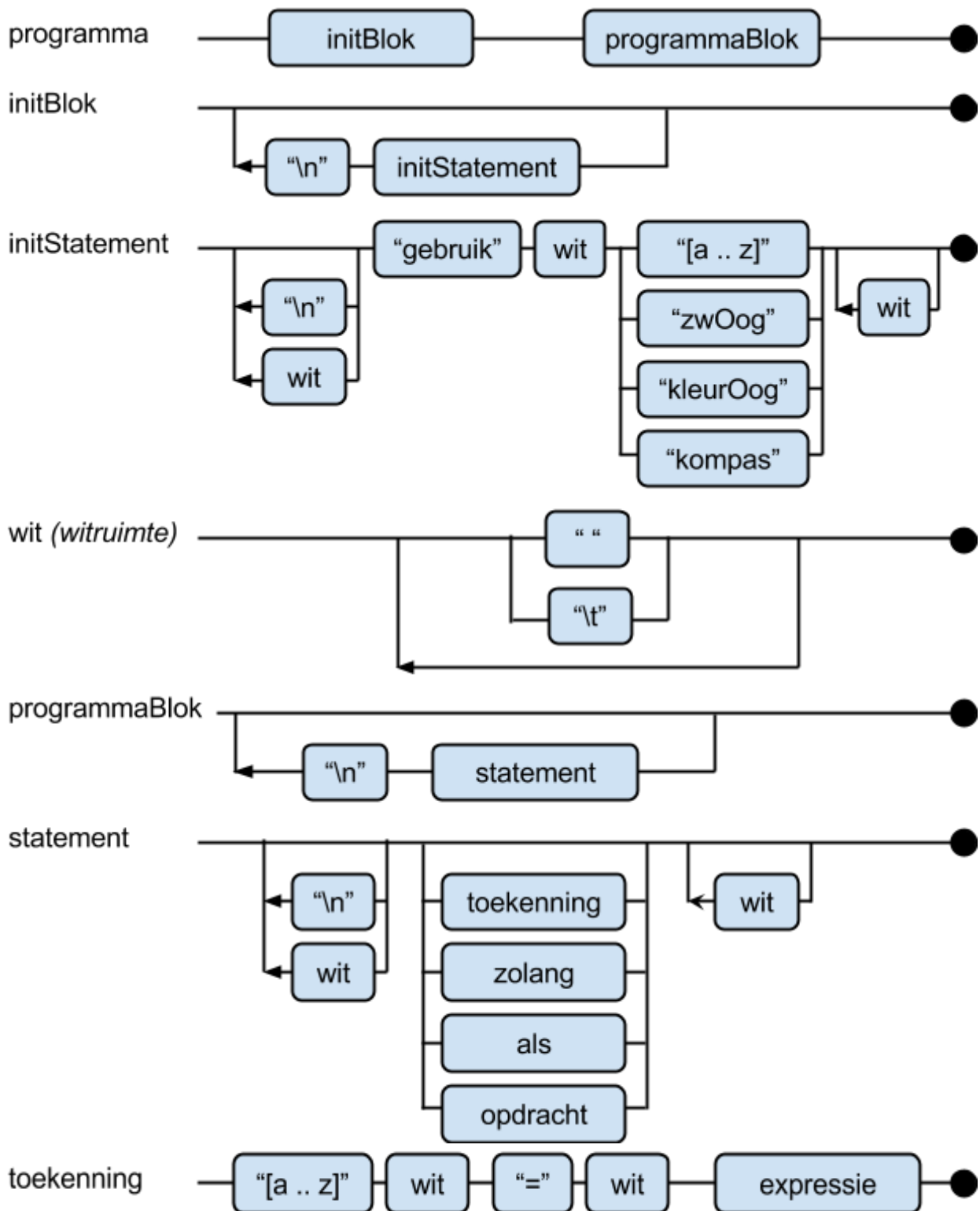
Nadat de eindscore door de examiner bekend gemaakt is kunnen jullie vragen om een motivatie. De examiner zal dan kort uitleggen hoe het cijfer tot stand is gekomen. Als jullie vinden dat de argumentatie niet klopt of als jullie een beoordeling per individuele student wensen dan kunnen jullie dit direct kenbaar maken door te vragen om een herbeoordeling.

Deze herbeoordelingen zullen plaatsvinden in de week direct volgend op de challengeweek in de vorm van een assessment. De examiner zal jullie individueel of per groep vragen om op gesprek te komen. Tijdens dit gesprek zal de examiner a.d.h.v. de antwoorden die je / jullie geven op de vragen bepalen of de individuele beoordelingen verschillend moeten zijn of dat jullie misschien wel aan de eisen voldoen maar geen voldoende hebben gekregen omdat de andere groepen excelleerden.

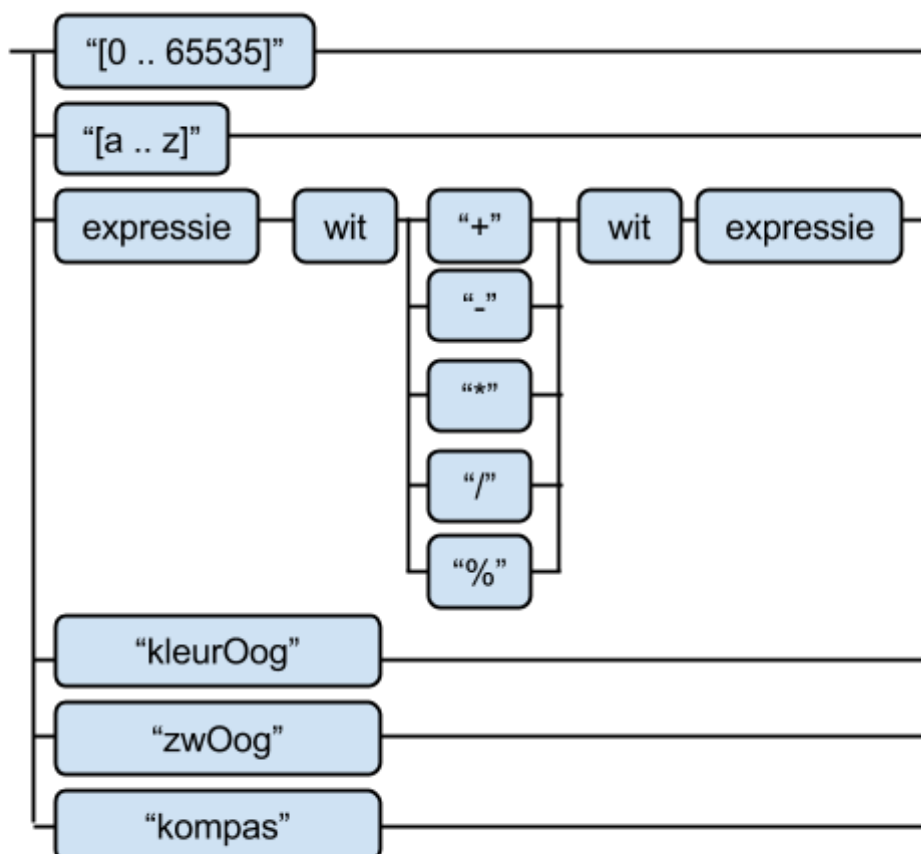
Hou er rekening mee dat het resultaat van het assessment ook kan zijn dat je score naar beneden moet worden bijgesteld. Het is dus niet geheel risicoloos. De herbeoordeling d.m.v. een assessment staat los van jullie recht op een herkansing in de week daaropvolgend.



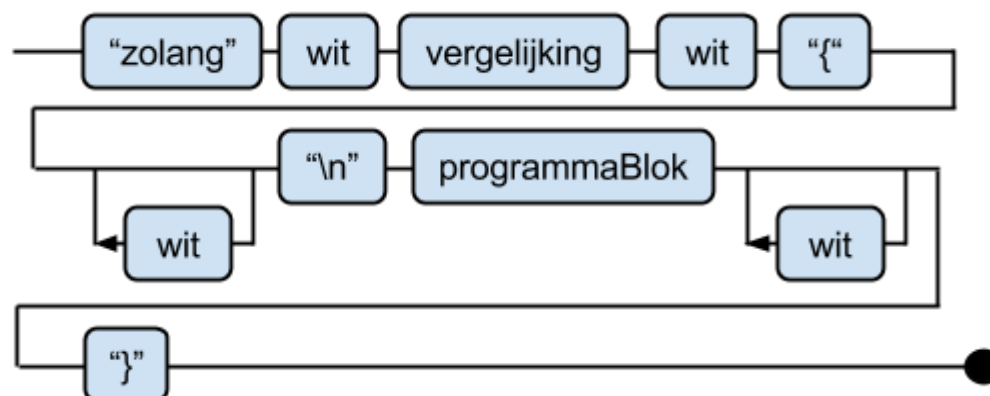
## Bijlage 2: Syntax specificatie 20



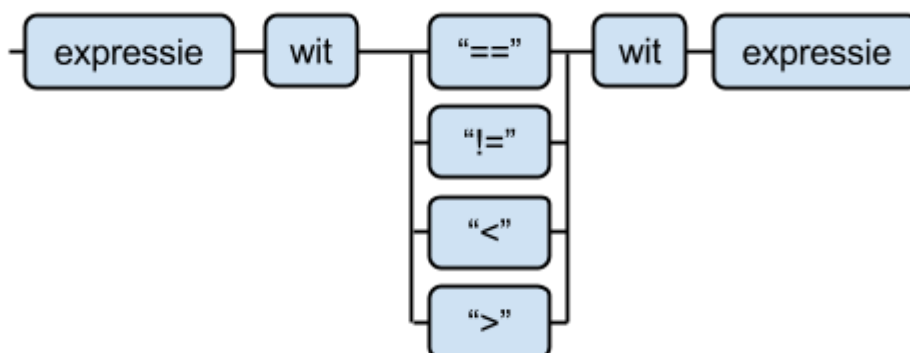
expressie



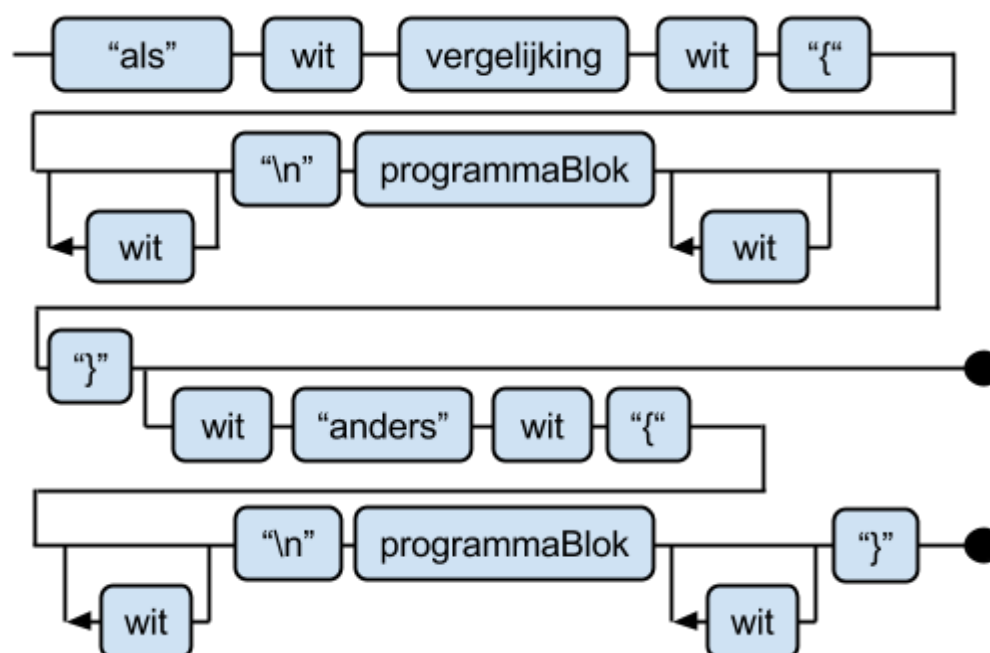
zolang



vergelijking



als



opdracht

