# Praktische opdracht Computational Science

In deze praktische opdracht ga je de wetenschappelijke methode toepassen op het modelleren van een fenomeen, zoals beschreven in hoofdstuk 3 van het lesmateriaal.

Volg de aanwijzingen voor het uitwerken van deze praktische opdracht en lever regelmatig en op tijd de vereiste documenten in.

## Groepjes vormen en casus kiezen

Vorm groepjes. De docent geeft aan met hoeveel mensen je een groepje vormt. Kies met elkaar een casus uit de lijst. Je kunt zowel een bestaand model uitbreiden, als een heel nieuw model maken.

In overleg met je docent kun je zelf een casus bedenken.

## Casus en onderzoeksvraag

Beschrijf wat je gaat modelleren en met welke doel aan de hand van de volgende vragen:

1. Wat weet je van dit fenomeen? Voer eventueel het benodigde (literatuur-) onderzoek uit.
2. Welke (deel van het) fenomeen wil je modelleren?
3. Waar hoop je achter te komen met behulp van jouw model?

## Model ontwerpen

Ontwerp je model aan de hand van de onderstaande vragen. Geef steeds aan welke overwegingen en keuzes je hebt gemaakt. Voorbeeld:
De schapen kunnen zich voortplanten. Als twee schapen elkaar tegenkomen, dan is er een kans van 20% dat er een nieuw schaap ontstaat. We hebben besloten om het geslacht van de schapen niet te modelleren omdat dat in dit geval niet relevant is.

De eerste twee vragen zijn alleen relevant als je een nieuw model ontwerpt.

1. Wat zijn de voornaamste soorten agents die betrokken zijn bij dit fenomeen?
2. In wat voor omgeving functioneren deze agents? Zijn er ook omgevingsagents aanwezig?
3. Welke eigenschappen hebben de agents? (Beschrijf per type agent.)
4. Welk gedrag vertonen de agents? (Beschrijf per type agent.)
5. Welke interacties hebben de agents onderling? En met de omgeving?
6. Als je dit fenomeen in discrete tijdseenheden zou beschrijven, wat gebeurt er dan in elke tijdseenheid (d.w.z. bij elke tik van de klok) en in welke volgorde?

## Model implementeren

Beschrijf in detail (eventueel met pseudocode) de eigenschappen, het gedrag en de interacties van alle (omgevings-) agents. Beschrijf hoe de tijd verloopt in het model en wat de volgorde van gebeurtenissen is. Met andere woorden: geef aan hoe de *setup*- en *go*-procedures eruit zullen zien.

Implementeer je model in NetLogo. Doe niet te veel tegelijk: schrijf de code stukje voor stukje en blijf testen (verificatie)!

## Model valideren

Valideer je model aan de hand van de volgende vragen:

1. Microvalidatie: in hoeverre komt het gedrag van de agents overeen met het gedrag dat in werkelijkheid wordt geobserveerd? Als het niet (helemaal) overeenkomt, zijn de verschillen dan relevant voor je onderzoeksvraag?
2. Macrovalidatie: in hoeverre komt het gedrag van de het systeem als geheel in jouw model overeen met het gedrag dat in werkelijkheid wordt geobserveerd. Als het niet (helemaal) overeenkomt, zijn de verschillen dan relevant voor je onderzoeksvraag?

## Experiment, analyse en conclusie

Gebruik je model om het antwoord op jouw onderzoeksvraag te vinden aan de hand van de volgende punten:

1. Beschrijf gedetailleerd hoe het experiment eruit ziet. Als je gebruik maakt van *BehaviorSpace*, beschrijf dan hoeveel experimenten er zijn uitgevoerd en met welke parameters.
2. Vermeld je uitkomsten op een geschikte manier (beschrijvend, m.b.v. een tabel, een grafiek enz.)
3. Analyseer de uitkomsten.
4. Beantwoord jouw onderzoeksvraag.

## Reflectie

Reflecteer op het modeleerproces aan de hand van de volgende vragen:

1. Wat ging goed? Wat kon beter?
2. Heb je aannames gedaan, die je de volgende keer toch anders zou doen? Welke aspecten van het model wil je veranderen?
3. Heb je in het begin van bepaalde zaken (agents of gedrag) afgezien, terwijl je nu denkt dat je ze toch had moeten meenemen in het model? Maak een *wishlist* van de zaken die je in een volgende versie van het model zou willen toevoegen, verwijderen en / of wijzigen.

## Model en documentatie inleveren:

* Een document met uitwerkingen van de hoofdvragen 1 t/m 6 uit de opdracht
* Het NetLogo-bestand van je model

# Casussen

Kies een casus uit deze lijst. In overleg met de docent kun je ook zelf een casus aandragen.

# Model uitbreiden

## Hoofdstuk 2 Computational Science, opdracht 28

## Hoofdstuk 2 Computationa Science, opdracht 29

## Simple economy

Dit model vind je in de NetLogo Models Library/IABM textbook/chapter 2

Maak de volgende wijzigingen:

1. Maak een monitor die het totale bedrag aan geld in het systeem weergeeft.
2. Belastingen invoeren: de turtles met meer dan 200 euro betalen 5% belasting. Implementeer deze wijziging. Wordt er in jouw model bij elke tik bij elke agent gekeken of hij belasting moet betalen?
3. Arme turtles krijgen een bijstandsuitkering. Bepaal zelf de bedragen en momenten van het betalen en implementeer de wijzigingen.
4. De wijzigingen uit de laatste twee opgaven zijn waarschijnlijk *hard coded*: je hebt in de code een getal ingevoerd voor belasting en bijstand. Voor een model is het handiger als dat parameters zijn. Voeg de nodige parameters aan je model toe en zorg dat deze aan te passen zijn in de interface.
5. In het huidige model is er geen verband tussen de geïnde belastingen en uitgekeerde bijstand. We introduceren een overheidskas waaruit niet meer uitgegeven kan worden aan bijstand dan dat er binnenkomt aan belastingen. Implementeer deze wijziging. Maak ook een monitor die aangeeft hoeveel geld er in de overheidskas is.
6. In de echte wereld hebben mensen te maken met tal van andere zaken waardoor geld „ontstaat” of „verdwijnt”. Voorbeelden zijn inflatie, economische groei, diefstal, het verkrijgen van een erfenis etc. Zoek een interessante economische gebeurtenis uit en voeg die toe aan je model.
7. Desgewenst maak je ook nog andere wijzigingen.

## Disease Solo

Dit model vind je in models library/biology

Maak de volgende wijzigingen:

1. Mensen kunnen genezen
2. Mensen kunnen ingeënt worden
3. Zieke mensen kunnen doodgaan
4. Mensen kunnen zich voortplanten. Bedenk zelf of kinderen in dat geval ziekte of immuniteit kunnen erven
5. Voeg zelf nog minstens twee aanpassingen toe

## Wolf Sheep predation

Dit model vind je in models library/biology

Maak de volgende wijzigingen:

1. Als een wolf een schaap tegenkomt, hoeft hij hem niet op te eten. Er is dus een kans dat het schaap overleeft.
2. Wolven leren steeds beter op schapen te jagen: hoe meer schapen een wolf heeft opgegeten, hoe groter de kans dat hij een schaap opeet als hij er één tegenkomt.
3. Schapen leren wolven te ontlopen: hoe meer wolven een schaap tegengekomen is (uiteraard zonder dat het opgegeten is), des de groter de kans dat het de ontmoeting overleeft.
4. Wolven gaan actief achter schapen aan.
5. Schapen gaan de wolven actief ontwijken.
6. Voeg zelf nog minstens twee aanpassingen toe.

# Een nieuw model maken

## Verkeersplein

Bij een kruispunt met stoplichten ontstaan er regelmatig files tijden het spitsuur. De gemeente overweegt om het kruispunt te vervangen door een verkeersplein.

Zou dat de doorstroming van het verkeer ten goede komen?

## Leven op Mars

Kunnen mensen echt op Mars leven?

NASA[[1]](#footnote-1) en SpaceX[[2]](#footnote-2) denken van wel. Het idee is dat mensen op den duur zelfvoorzienend op Mars zouden kunnen leven, zonder afhankelijk te zijn van bevoorrading vanaf de Aarde.

Stel dat er geschikt onderdak geregeld is. Wat is er dan nodig om voldoende water, lucht en voedsel te produceren om te overleven? Is het mogelijk om zelfvoorzienend te worden?

## Wachtrij

Een bank wil een nieuwe filiaal openen. Met het oog op de klantenvriendelijkheid wil men de service in het filiaal zo organiseren dat de wachttijd voor de klanten zo klein mogelijk is.
Bij een balie kan men verschillende soorten zaken doen. Sommige daarvan zijn vrij snel klaar (bijvoorbeeld saldo opvragen, geld storten) en duren maximaal 5 minuten. Andere zaken duren wat langer (bijvoorbeeld een rekening openen of opheffen) en dat duurt tussen de 3 en 10 minuten. Ten slotte zijn er zaken die lang duren en waarvoor men twee keer moet komen (bijvoorbeeld het regelen van een lening of hypotheek). In dat geval duurt elke bezoek tussen de 5 en 20 minuten.

Hoe kunnen de diensten het beste over de balies worden verdeeld? En wat betekent dit voor de wachtrijen van de klanten?

## Kaaspakhuis

Kaas wordt na de productie opgeslagen in kaaspakhuizen om te rijpen en in afwachting van verkoop. Hoe ouder de kaas, des te hoger de opbrengt, maar opslag in een pakhuis brengt ook kosten met zich mee.

Een kleine kaasproducent wil zijn verkoopstrategie optimaliseren, ofwel zijn winst maximaliseren. Op zijn boerderij worden elke maand 25 boerenkazen geproduceerd en in zijn pakhuis heeft hij ruimte voor 400 kazen. Als het pakhuis vol is, kan hij kiezen: hij zet de productie stop, of hij geeft een aantal kazen aan een goed doel. Hij kan kiezen uit een aantal verkoopkanalen. Hij kan leveren aan een biologische winkel waar hij langlopende contracten mee kan sluiten, of aan een streekmarkt met maandelijkse contracten waar hij hogere prijzen kan rekenen.

Hoe zit een optimale verkoopstrategie eruit?

## Aardappels

Een biologische aardappelteler wil zijn winst maximaliseren. Daartoe moet hij rekening houden met kosten die hij maakt voor pootgoed, mest, het verwijderen van aangetaste planten etc., maar ook met plagen en ziektes die zich makkelijk verspreiden als aardappels te dicht bij elkaar zijn geplant. Hij kan alleen aardappels van goede kwaliteit verkopen. De rest gooit hij weg.

Wat is de optimale strategie voor het planten en telen van aardappels?

## Brand op school

Bij brand in een school of een ander gebouw waar zich veel mensen bevinden, is het belangrijk dat mensen tijdig gealarmeerd kunnen worden om snel het gebouw te kunnen verlaten. Mensen die niet op tijd naar buiten kunnen komen lopen gevaar om dit niet te overleven.

De leiding van jouw school heeft advies nodig over het aantal alarmen en de positie van alarmen en nooduitgangen om de begane grond goed te kunnen evacueren. Idealiter wordt er ook rekening gehouden met het evacueren van mensen in een rolstoel.

Wat is op basis van jouw model het advies?

## Weerstand

Maak een model van elektrische weerstand. Controleer of de volgende natuurkundige wetten kloppen:

1. Wet van Ohm
2. Voor serieschakeling: 
3. Voor parallelschakeling: 
1. https://www.nasa.gov/topics/journeytomars/index.html [↑](#footnote-ref-1)
2. http://www.spacex.com/mars [↑](#footnote-ref-2)