

SCRIPTING TURTLES

ST 2020



RICHARD JANSEN
1762581

Contents

Opdracht 1	2
Algemeen	2
Pros	2
Cons	2
De Tutorial.....	2
Economie model.....	3
De Aanpassing	4
Gui.....	5
Opdracht 2	6
initiele staat.....	6
Functie "See" of "Perceive"	6
Functie "Act"	6
Functie "Update"	6
Opdracht 3	7
Inaccessible	7
Non-Deterministic (Stochastic)	7
Episodic.....	7
Static	7
Continuous	7
Opdracht 4	8
accessible	8
Non-Deterministic (Stochastic)	8
Non-Episodic	8
dynamic.....	8
Continuous	8

Opdracht 1

Algemeen

In het kort is Netlogo een eenvoudig programma waarvoor je niet veel programmeer kennis voor nodig hebt ten opzichte van unity of de Mesa extensie voor python. Verder is Netlogo matig in 3D en focust het meer op een model in 2D.

Pros

- Snel en makkelijk te leren
- Programmeer taal is heel simpel opgezet
- Resultaten zijn makkelijk te delen met anderen mensen
- Exporteren van data is makkelijk te doen via csv bestanden
- Makkelijk gui's maken

Cons

- Niet heel uitgebreid qua programmeer taal waar door je mogelijk complexe dingen minder snel/goed kan doen
- Heel snel wordt de code omslachtig maar kan ook niet kleiner
- Niet een hele grote community die actief is op stack overflow voor vragen
- Eigen Sprites maken kost veel tijd, je kan geen eigen icon's overzetten van anderen programma's zoals de Adobe Software

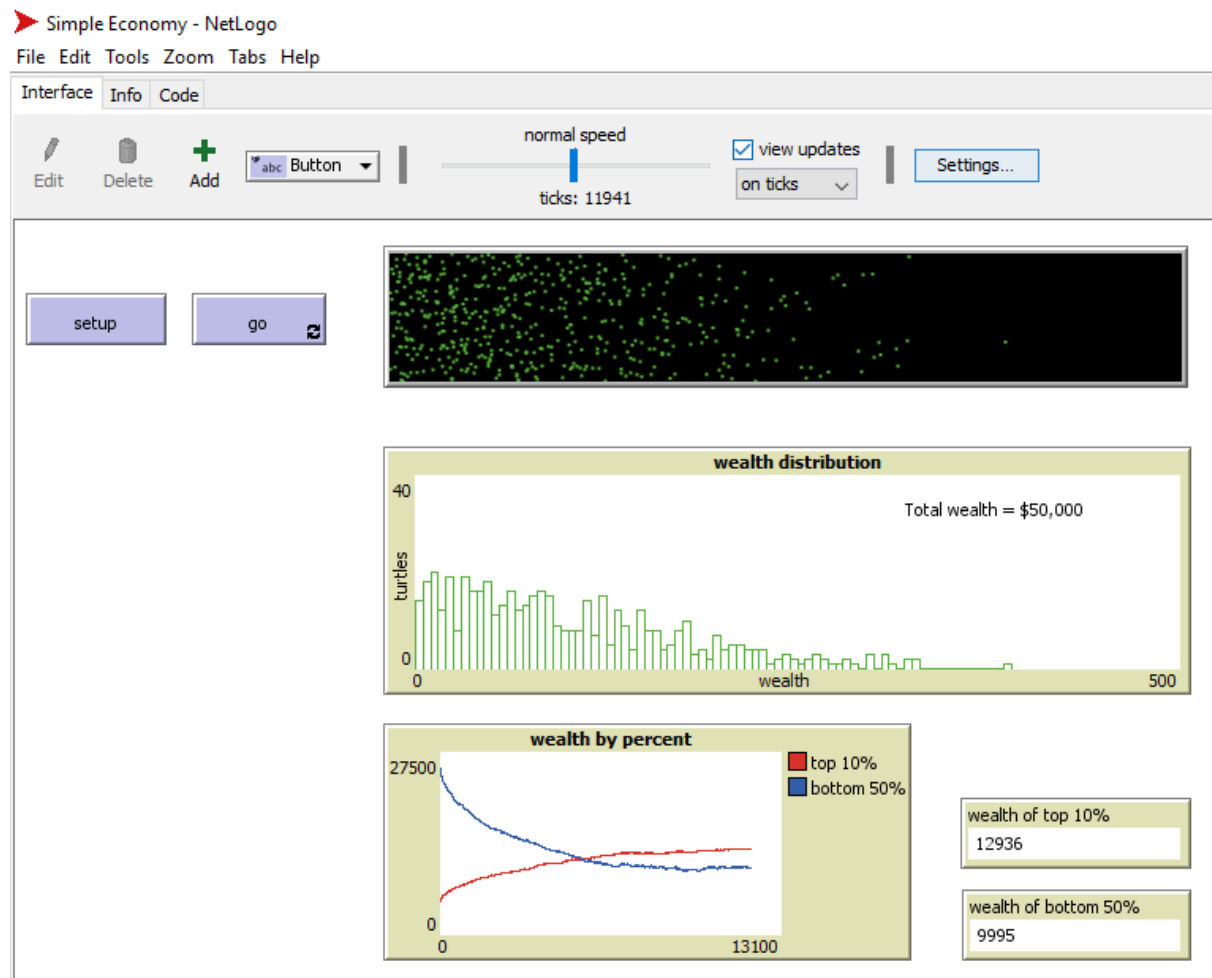
Netlogo is agent-based programming. In Netlogo heet agents Turtles. Naast turtles heb je ook patches en links. Agents (Turtles) kun je een set instructies geven wat elke turtle aanhoudt. Elke turtle kan bij de globale simulatie variabele, bij de patches variabelen en met degene waar het mee gelinkt is (als je die link hebt gemaakt)

De Tutorial

De tutorial van Netlogo die wij hebben gekregen van school was zeer duidelijk en goed te volgen. Alle belangrijke onderwerpen over agent-based programming wordt hier goed in behandeld en geeft een goed totaal plaatje. Er wordt goed inhoudelijk verteld over elk model dat als voorbeeld wordt gegeven. Na het volgen van de tutorial is het makkelijk genoeg om de meeste basismodellen goed te begrijpen en zelf ook dingen te maken met behulp van de handleiding van Netlogo zelf.

Economie model

In dit model wordt er een simulatie gedraaid van wat er gebeurt als iedere agent 100 “euro” krijgt en elke tick een willekeurige anderen agent 1 euro geeft. Zolang de agent meer geld heeft dan 0 euro, gaat het iemand anders een euro geven. De code (*fig 2) laat zien hoe kort en krachtig zo simulatie kan zijn



```
to go
  ;; transact and then update your location
  ask turtles with [ wealth > 0 ] [ transact ]
  ;; prevent wealthy turtles from moving too far to the right
  ask turtles [ if wealth <= max-pxcor [ set xcor wealth ] ]
  tick
end

to transact
  ;; give a dollar to another turtle
  set wealth wealth - 1
  ask one-of other turtles [ set wealth wealth + 1 ]
end
```

fig 2

De Aanpassing

Ik heb in de simulatie een aantal componenten toegevoegd die mij interessant leken. Ik heb 2 sliders toegevoegd voor de visualisatie om een selectie te kunnen maken tussen rijke en arme agent. Dit is puur visueel namelijk grijs voor arm en rood voor rijk en wat ertussen zit groen.

Bij het creëren van een turtle(agent), wordt er gelijk een willekeurige personage en salaris toegewezen. Aan de hand van het personage, wordt er bepaald of deze turtle mee doet aan gokken voor een loterij of niet. Verder krijgt elke turtle na 30 ticks (een maand) salaris. Bij elke trade die er plaats vindt, wordt daarboven op ook belasting op uitgevoerd. Een deel van deze belasting gaat naar de jackpot toe.

Verder zijn er nog enkele monitors toegevoegd om informatie duidelijk te maken dat je niet moet gokken om rijk te worden 😊

Volgende bladzijde staat het programma*

setup go

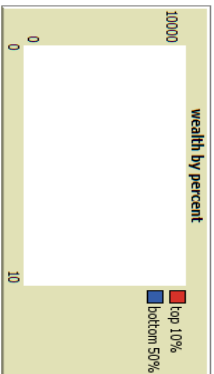
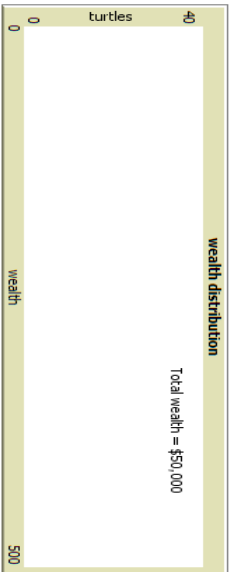
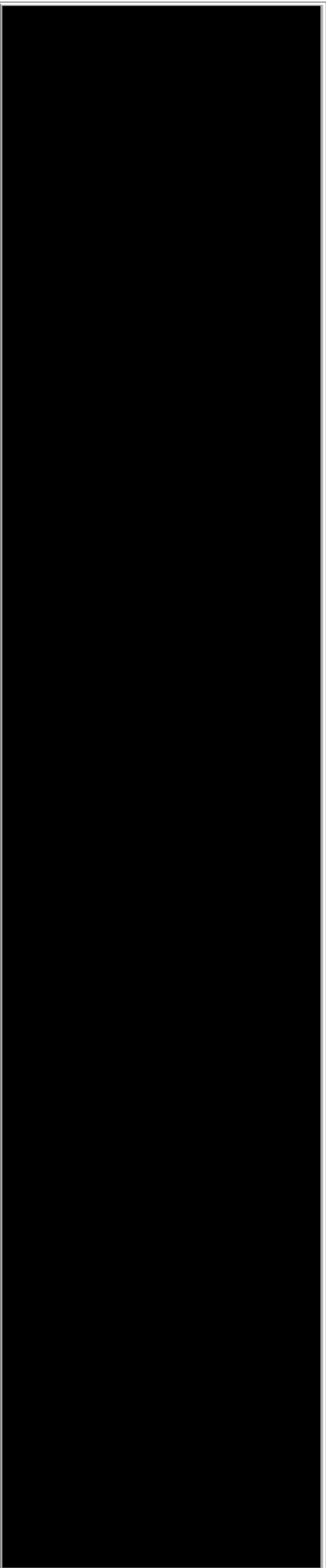
Rich 344 \$

poor 70

gambler_rate 8

gambler_price 18

tax 20 %



gamblers 0

gamblers that are rich 0

gamblers that are poor 0

gamblers money 0

wealth of top 10% 0

wealth of bottom 50% 0

not gamblers 0

Rich People from work 0

People that are poor but not gamble 0

jackpot money 0

trades 0

government money 0

months N/A

Gui

Zo ziet de gui eruit

Opdracht 2

initiele staat

In de initiele staat, bestaat de turtle uit;

- Een wealth van 100
- Een 0 tot 10 random variabelen wat betreft tot zijn gok persoonlijkheid
- Een 0 tot 30 random variabelen wat betreft tot zijn salaris per maandag

Functie "See" of "Perceive"

In deze functie wordt eerst gekeken of het tijd is voor salaris. Als dit zo is, dan krijgt de turtle salaris boven op zijn huidige geld.

Elke tick is een update voor de turtle

```
to go
  if ticks mod 30 = 0 [ ask turtles [set wealth wealth + salary]] ;;eigen toevoeging,
  set jackpot jackpot + government_money / 100 ;;eigen toevoeging,
  ;; transact and then update your location
  ask turtles with [ wealth > 2 ] [ transact ]
  ;; prevent wealthy turtles from moving too far to the right
  ask turtles [ if wealth <= max-pxcor [ set xcor wealth ] ]

  ask turtles [ if wealth > gamble_price and gambler_r > gambler_rate [ gamble ] ]

  ask turtles with [ wealth > rich ] [ set color red ]
  ask turtles with [ wealth > poor and wealth < rich ] [ set color green ]
  ask turtles with [ wealth < poor ] [ set color grey ]

  ask turtles with [ gambler_r > gambler_rate ] [ set shape "square" set color pink ]
  ask turtles with [ gambler_r < gambler_rate ] [ set shape "circle" ]

  tick
end
```

Functie "Act"

Als de turtle meer dan 2 euro heeft, dan gaat het handelen.

Als de turtle een hoog genoeg gok persoonlijkheid heeft en genoeg geld heeft om te gokken, dan gaat het een poging doen met gokken.

Functie "Update"

In het laatste gedeelte van de tick, wordt er gekeken naar het geld en de gok persoonlijkheid om de turtle de juiste kleur en shape te geven.

Opdracht 3

Inaccessible

In deze simulatie kan een turtle alleen bij zijn eigen variabelen om een beslissing te maken. Het kijkt niet naar de omgeving BEHALVE of het genoeg geld heeft om het inleggeld te betalen voor

Non-Deterministic (Stochastic)

In dit model zijn er veel random factors zoals salaris en gokken. Hierdoor is elke simulatie anders dan een ander.

Episodic

Elke actie die er gedaan wordt, heeft (bijna) geen effect op de volgende cyclus. De agent gaat niet anderen dingen proberen ten opzichte van de voorgaande acties.

Static

Het environment in de simulatie blijft hetzelfde. Er is alleen wel een inflatieprobleem in mijn programma maar dat is in het echt ook 😊

Continuous

In principe kan deze simulatie oneindig lang doorgedraaid worden.

Opdracht 4

Een simulatie over stock beurzen en trading daarin. Een agent kan bij de gegevens van een aandeel en daarop beleggen. Elke dag verandert de prijs van een aandeel en dus ook de waarde van de agent.

accessible

In deze simulatie kan een agent bij de gegevens van aandeel. Een aandeel kan een agent zijn in abstracte context

Non-Deterministic (Stochastic)

In dit model zijn er random variabelen of dingen die niet voorspelbaar zijn zoals de koers van een aandeel

Non-Episodic

Als meerdere agents meerdere aandelen kopen van 1 bedrijf, dan heeft dat invloed op de beurs prijs

dynamic

Het environment verandert aan de hand van hoe duur de aandelen zijn. Dit komt door het voorgaande gedrag.

Continuous

In principe kan deze simulatie oneindig lang doorgedraaid worden.