

Scripting Turtles

ST 2020



Richard Jansen

1762581



# 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
* Programeer 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 programeer 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

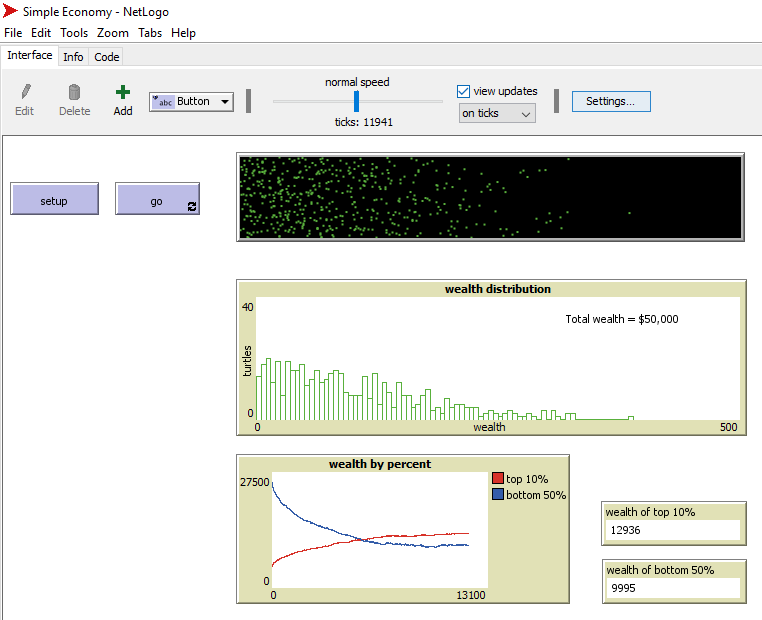
Netlogo is agent-based programming. In Netlego heet agents Turtles. Naast turtles heb je ook patches en links. Agents (Turles) kun je een set instructies geven wat elke turtle aanhoudt. Elke turtle kan bij de globale simulatie variabele, bij de patche variabelen en met degene waar het mee gelinkt is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

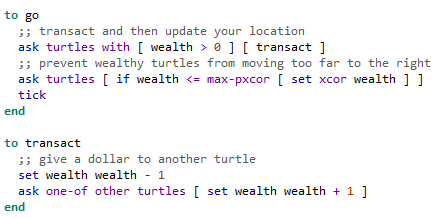
## De Tutorial

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



fig 2

### De Aanpassing

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

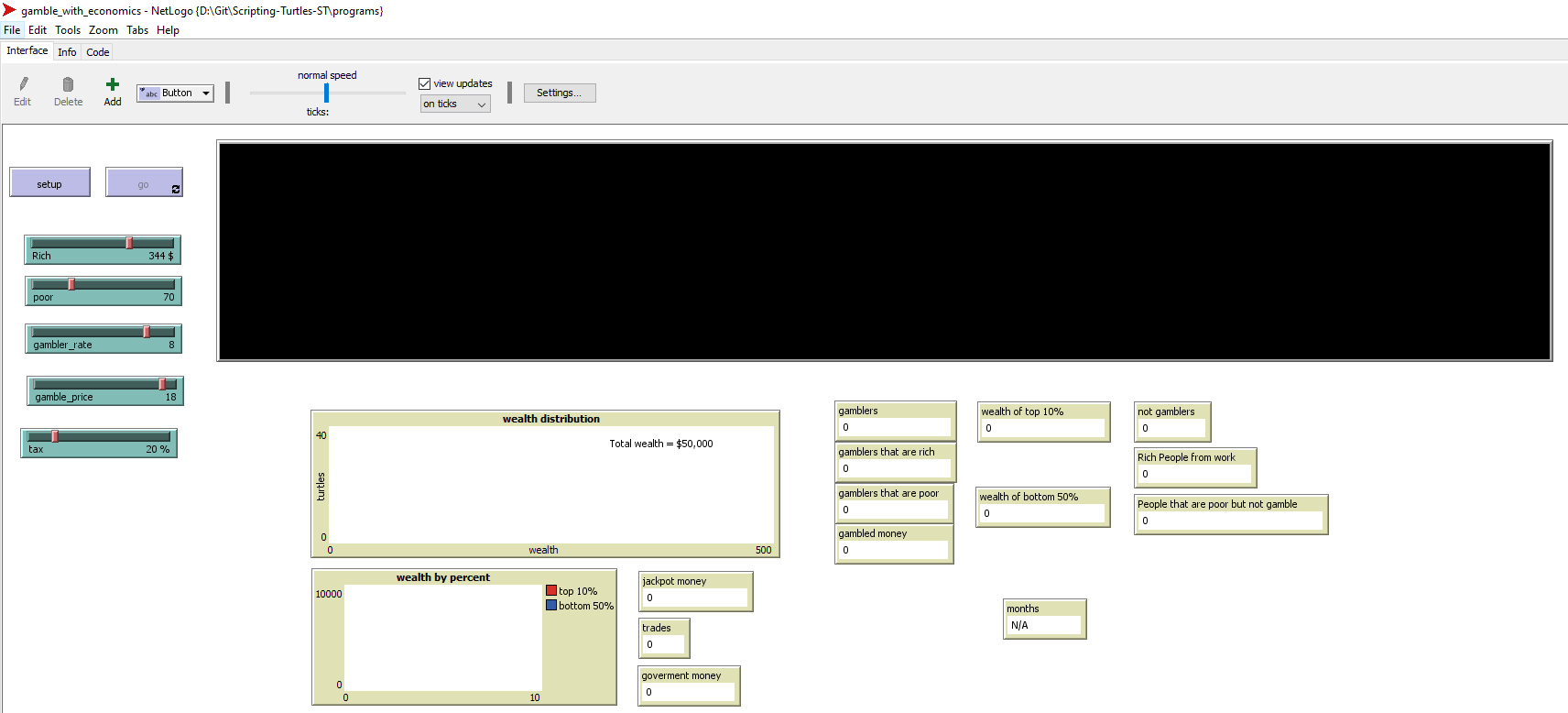
Bij het creëren van een turtle(agent), wordt er gelijk een willekeurige personage en salaris toegewezen. Aan de hand van het personage, wordt er bepaald of deze turtle mee doet aan gokken voor een loterij of niet. Verder krijgt elke turtle na 30 ticks (een maand) salaris. Bij elke trade die er plaats 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\*

### Gui

Zo ziet de gui eruit



# Opdracht 2

## initiele staat

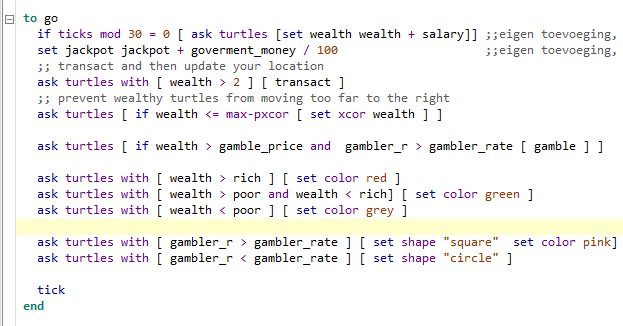
In de initiele staat, bestaat de turtle uit;

* Een wealth van 100
* Een 0 tot 10 random variabelen wat betrekt tot zijn gok persoonlijkhei
* Een 0 tot 30 random variabelen wat betrekt 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



## Functie "Act"

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

Als de turtle een hoog genoegen 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