Agent-based modelling of complex systems

Janusz Szwabiński

Lecture #1: Introduction

Contact data

- janusz.szwabinski@pwr.edu.pl
- office hours (C-11 building, room 5.16):
    - Monday, 9.00-11.00
    - Wednesday, 9.00-11.00
    - preferably **make an appointment via email**, providing details of your problem
- http://prac.im.pwr.wroc.pl/~szwabin/index

Course overview

1. Introduction to agent-based modelling
2. Creating simple models
3. Exploring and extending models
4. Components of agent-based models
5. Analyzing agent-based simulations
6. Verification and validation of models
7. Computational roots of agent-based modelling
8. Examples of interesting models

Reading

1. U. Wilensky, W. Rand, „An Introduction to Agent-Based Modeling”
2. S. F. Railsback, V. Grimm, „Agent-Based and Individual-Based Modeling: A Practical Introduction”
3. R. Siegfried, „Modeling and Simulation of Complex Systems: A Framework for Efficient Agent-Based Modeling and Simulation”

Assessment

- each assignment in the lab will be graded on a 100 point basis
  - submissions that do not run will receive at most 20% of the points
  - in case of not meeting the deadline for the assignment, the score will by reduced by 25% for each day of the delay

Average score	Grade
$X < 65$	2.0
$65 \leq X < 70$	3.0
$70 \leq X < 75$	3.5
$75 \leq X < 85$	4.0
$85 \leq X < 95$	4.5
$95 \leq X \leq 100$	5.0

Assessment

- two quizzes in the lectures: midterm and final
  - closed-book
  - one DIN A4 page of handwritten notes allowed
- final score according to the formula:

$$Score = 0.4 * L + 0.2 * Q_m + 0.4 * Q_f$$

Outlook

- Definition of complex systems
- Notion of emergence
- Agent-based models
- Why model?
- Useful tools

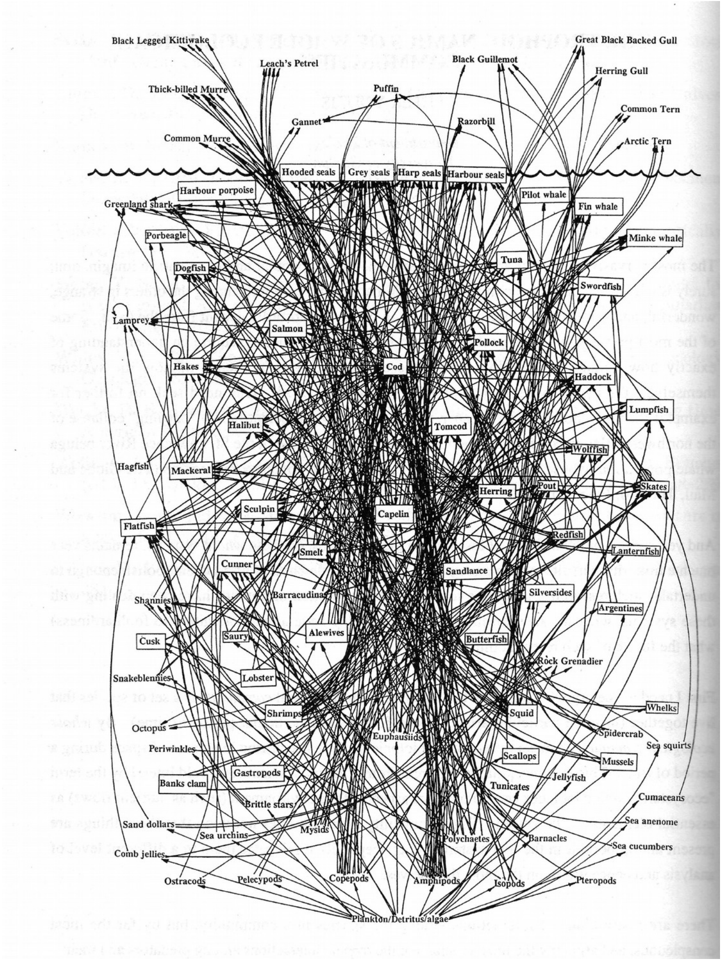
Complex systems

- a system composed of many components which may interact with each other:
  - Earth’s global climate
  - human brain
  - ecosystems
  - universe
- their behavior intrinsically difficult to model due to dependencies, relationships or interactions between their parts
- typical properties: nonlinearity, emergence, self-organization, adaptation, feedback loops
- usually whole is more than a sum of parts

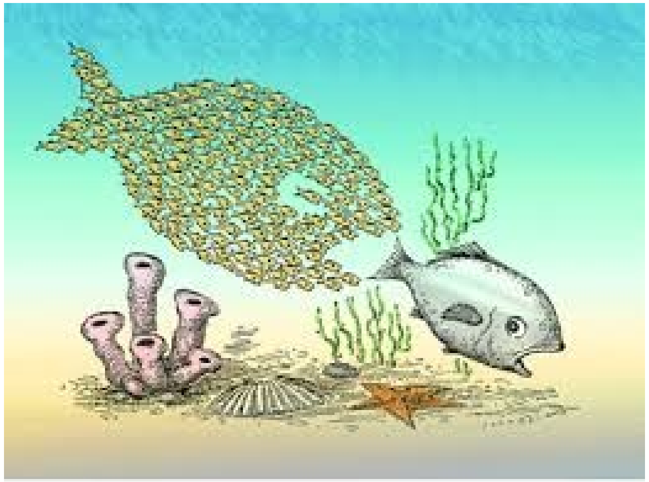
Complex systems

Complex systems

Complex systems



Emergence



- a macro-level phenomenon
  as a result of local-level interactions
- examples:
  - life
  - path formation (desire paths)
  - traffic jams
  - flocking behavior
  - mexican waves, standing ovation

Emergence

Emergence

Emergence

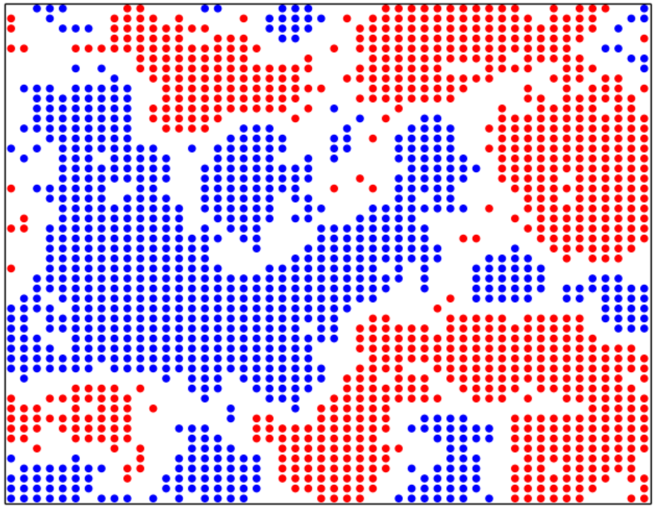
Emergence

Emergence

Emergence

Emergence

Models



- simplifications of reality
- many kinds: **formal mathematical** ones, stories, cartoons etc
- focus on the main characteristics of a problem
- different modelling approaches
- each approach involves its own set of theories, concepts, mathematical techniques and accepted procedures for constructing and testing models

Models: reductionistic vs integrated models

- reductionistic Newtonian approach:
  - deterministic
  - a system (like the solar system) can be described as a machine
  - we can predict its development if we understand all the mechanisms
  - its relative simplicity had a great appeal to scientists
- integrated models:
  - no predictability of the full trajectory of the system
  - the system adjusts constantly to changing context (like a flock of birds)
  - related to changes in our perception of the world (thermodynamics, quantum mechanics, evolution)
  - very often irreversible, stochastic and out-of-equilibrium

Models: deductive vs inductive reasoning

- inductive reasoning – a process where the starting point is the observation of a phenomenon

  Observation → pattern → tentative hypothesis → theory

  - many social and life scientists use this approach
  - useful if one does not know what to look for

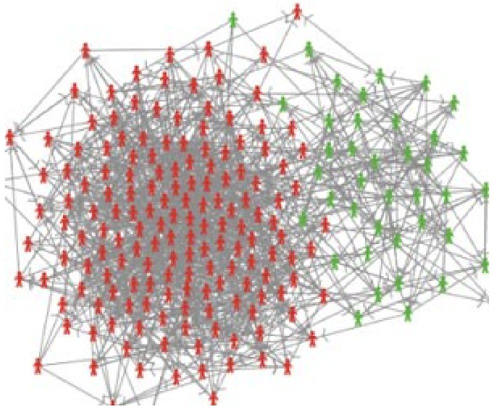
- deductive reasoning – an approach that follows the opposite direction

  Theory → hypothesis → observation → confirmation

Models: deductive vs inductive reasoning

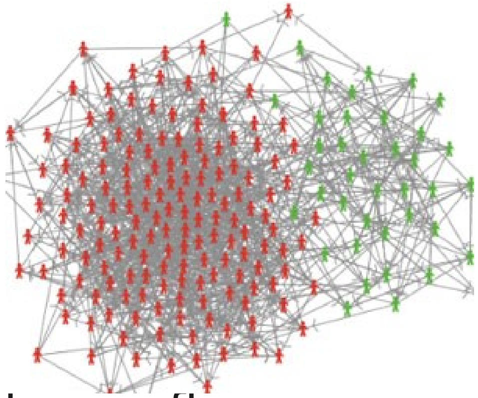
- equally valid methods of deriving knowledge
- one often needs both methods to understand complex processes
- inductive approach from the modelling perspective:
  - data set from a case study as starting point
  - statistics from the data set
  - model that may fit the data → first step towards a theory
- deductive approach from the modelling perspective:
  - start with an abstract model using rules based on existing knowledge
  - what kind of patterns will evolve within the model evolution?
  - test or adjust existing theories based on the results

Agent-based models



- a class of computational models for simulating the actions and interactions of autonomous agents (both individual or collective entities)
- combine elements of game theory, complex systems, emergence, computational sociology, multi-agent systems, and evolutionary programming
- Monte Carlo methods are used to introduce randomness
- often used on non-computing related scientific domains including biology, ecology and social science
- used to search for explanatory insight into the collective behavior of agents obeying simple rules
- a kind of microscale model
- attempt to re-create and predict the appearance of complex phenomena
- K.I.S.S. ("Keep it simple, stupid") principle

Agent-based models



- typical elements:
  - numerous agents specified at various scales
  - decision-making heuristics
  - learning rules or adaptive processes
  - an interaction topology
  - an environment
- usually implemented as computer simulations, either as custom software, or via ABM toolkits
- these simulations are used to test how changes in individual behaviors will affect the system's emerging overall behavior

Why model?

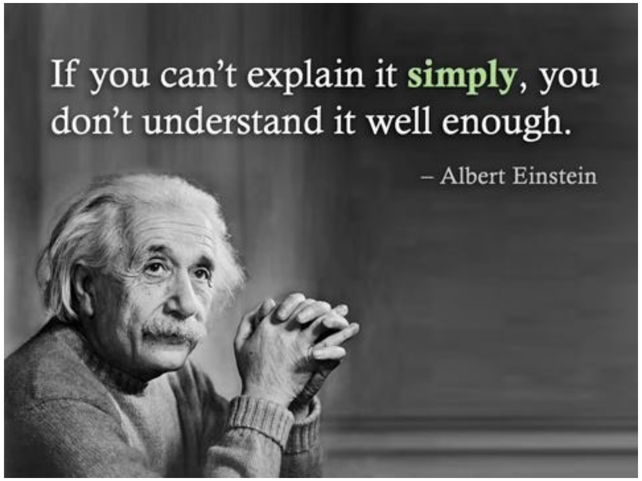
- Predict
- Explain
- Guide data collection
- Illuminate core dynamics
- Suggest dynamical analogies
- Discover new questions
- Promote a scientific habit of mind
- Bound outcomes to plausible ranges
- Illuminate core uncertainties
- Reveal the apparently simple to be complex and vice versa
- Offer crisis options in near-real time
- Demonstrate tradeoffs/suggest efficiencies
- Challenge the robustness of prevailing theory through perturbations
- Expose prevailing wisdom as incompatible with available data
- Train practitioners
- Discipline the policy dialogue
- Educate the general public

Further reading: J. M. Epstein, „Why Model?", JASSS vol. 11, no. 4 12, http://jasss.soc.surrey.ac.uk/11/4/12.html

Predict

- prediction often presumed to be the goal of modelling
- it might be a goal, particularly if one admits statistical prediction in which stationary distributions (of epidemic sizes for instance) are the regularities of interest

Explanation

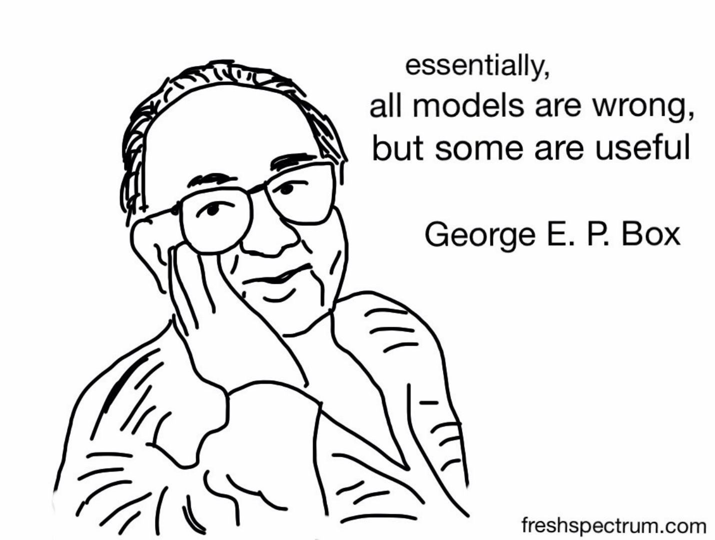


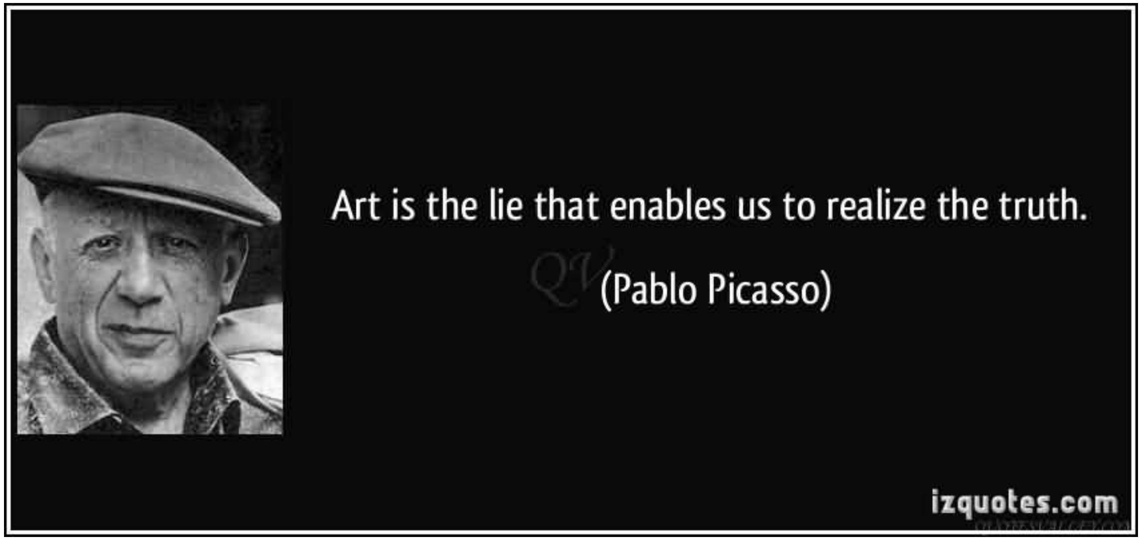
- it does not imply prediction!:
  - plate tectonics
  - electrostatics
  - evolution
- plausible behavioral rules → macroscopic explananda (large scale regularities)

Guided data collection

- collecting a lot of data and then running regressions on it is not the only approach to science
- theory often precedes data collection (e.g. Maxwell's electromagnetic theory, general relativity)
- without models it is not always clear what data to collect

Core dynamics





- even the best models are wrong in an engineering sense
- some of them are fruitfully wrong
  - they illuminate abstractions (Lotka-Volterra ecosystem model, SIR model, Hooke's Law)
  - they capture qualitative behaviors of overarching interest (e.g. predator-prey cycles, epidemic threshold)

Analogies

- a huge variety of seemingly unrelated processes have formally identical models (e.g. Coulomb’s Law and gravitational attraction)
- applying a powerful pre-existing theory (model) to an unexplored field may lead to rapid advances

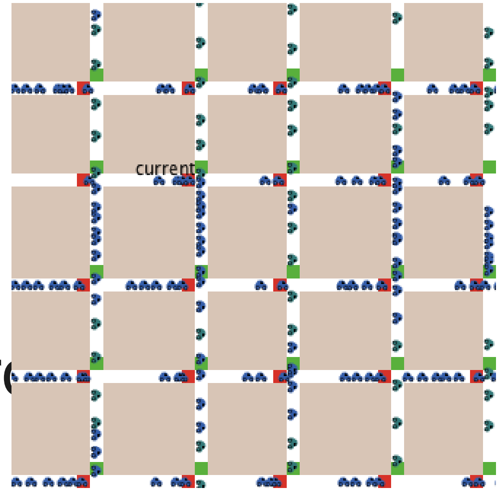
New questions

- new questions produce huge advances
- models can help us to discover them

Tools

- Python (with Numpy/Scipy, Networkx and many others modules) as the course programming language
- Matlab and GNU Octave are reasonable high-level alternatives
- Netlogo, Repast, MASON
- Mesa
- PyCX
- for more advanced models you may resort to compiled programming languages as C/C++ or Fortran

Netlogo

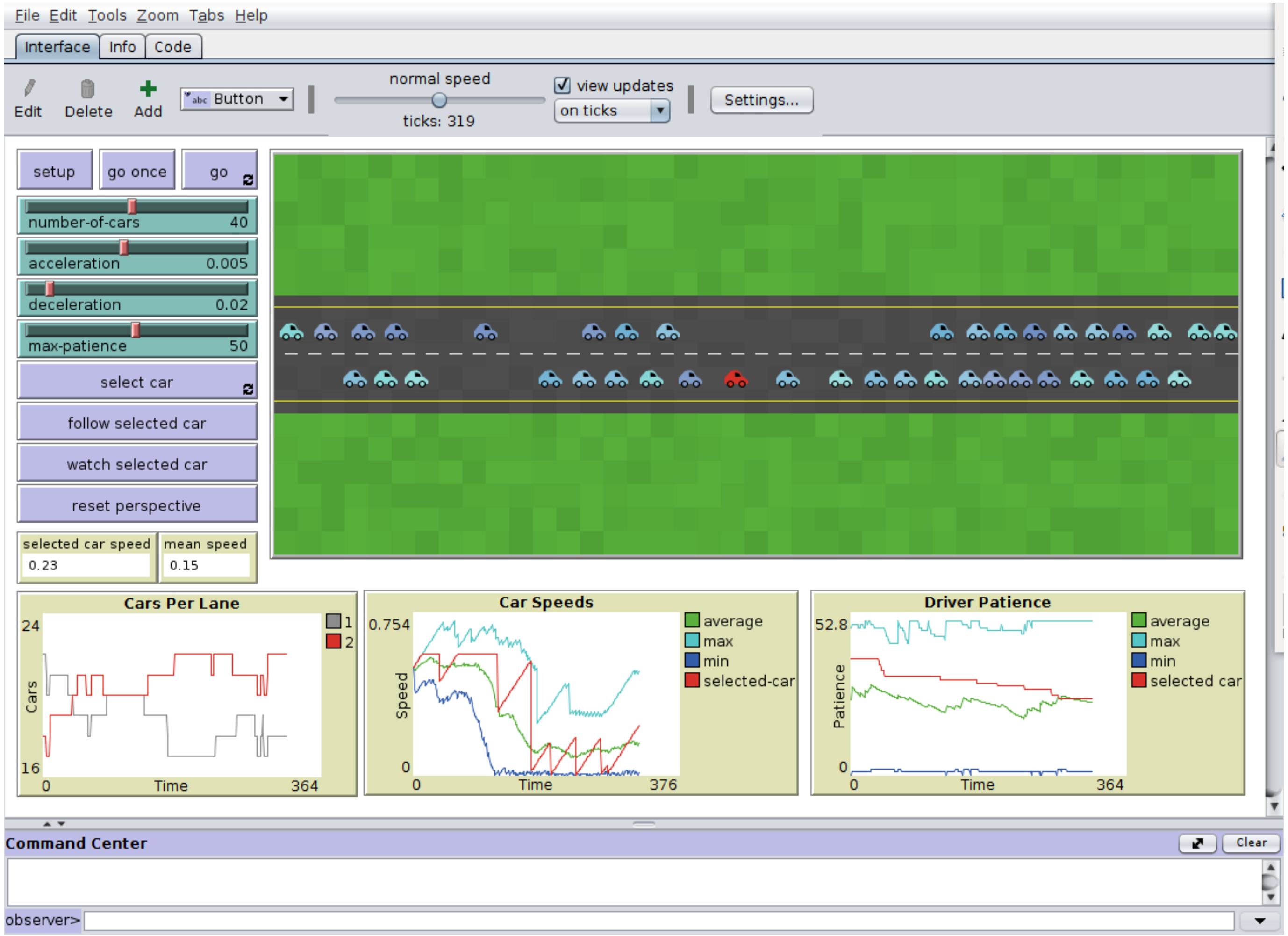


- https://ccl.northwestern.edu/netlogo/
- a multi-agent programmable modeling envir
- open source and freely available
- authored by Uri Wilensky and developed at Northwestern's Center for Connected Learning and Computer-Based Modeling
- designed in the spirit of the Logo programming language, to be "low threshold and no ceiling" → easy to pick up
- designed for multiple audiences in mind, in particular:
  - teaching children in the education community
  - for domain experts without a programming background

Netlogo

- its environment enables exploration of emergent phenomena (supported by the GUI)
- an extensive models library (a variety of domains, such as economics, biology, physics, chemistry, psychology, system dynamics)
- it allows authoring of new models and modification of existing ones
- used in a wide variety of educational contexts: from elementary school to graduate one
- used by researchers worldwide
- an online version available at https://www.netlogoweb.org/

Netlogo

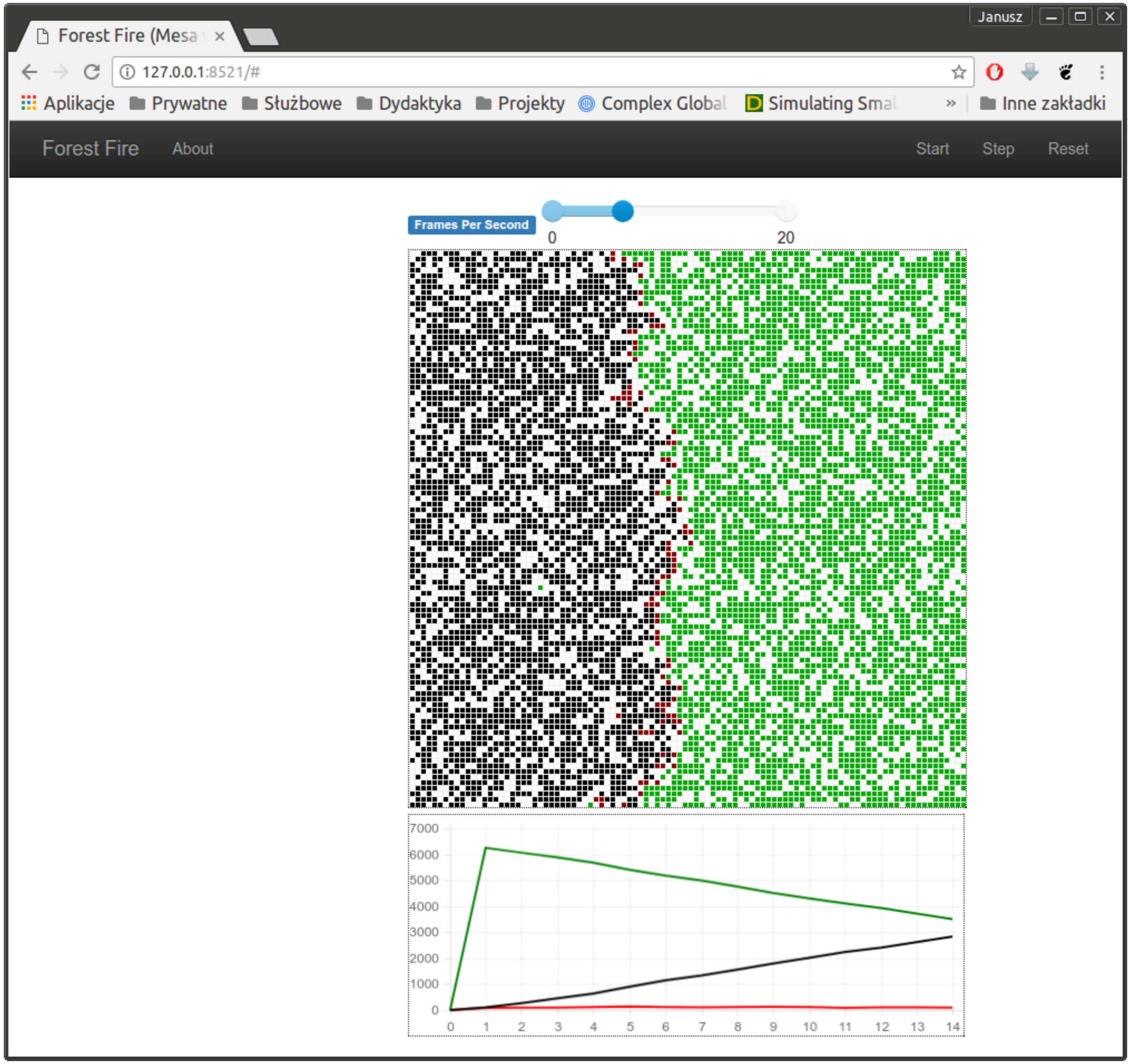


File Edit Tools Zoom Tabs Help
Interface Info Code
Edit Delete Add Button
normal speed
ticks: 319
view updates
on ticks
Settings...
setup
go once
go
number-of-cars 40
acceleration 0.005
deceleration 0.02
max-patience 50
select car
follow selected car
watch selected car
reset perspective
selected car speed 0.23
mean speed 0.15
Cars Per Lane
Car Speeds
Driver Patience
average
max
min
selected-car
Time
Cars
Speed
Patience
Command Center
Clear
observer>

Mesa

- http://mesa.readthedocs.io/en/latest/tutorials/intro_tutorial.html
- agent-based modeling framework in Python
- it allows:
  - to quickly create agent-based models using built-in core components (such as spatial grids and agent schedulers) or customized implementations
  - to visualize models using a browser-based interface
  - to analyze the results using Python’s data analysis tools
- Python 3-based counterpart to NetLogo, Repast, or MASON

Mesa



Forest Fire (Mesa
Janusz
127.0.0.1:8521/#
Aplikacje
Prywatne
Służbowe
Dydaktyka
Projekty
Complex Global
Simulating Sma
Inne zakładki
Forest Fire
About
Start
Step
Reset
Frames Per Second
0
20
7000
6000
5000
4000
3000
2000
1000
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

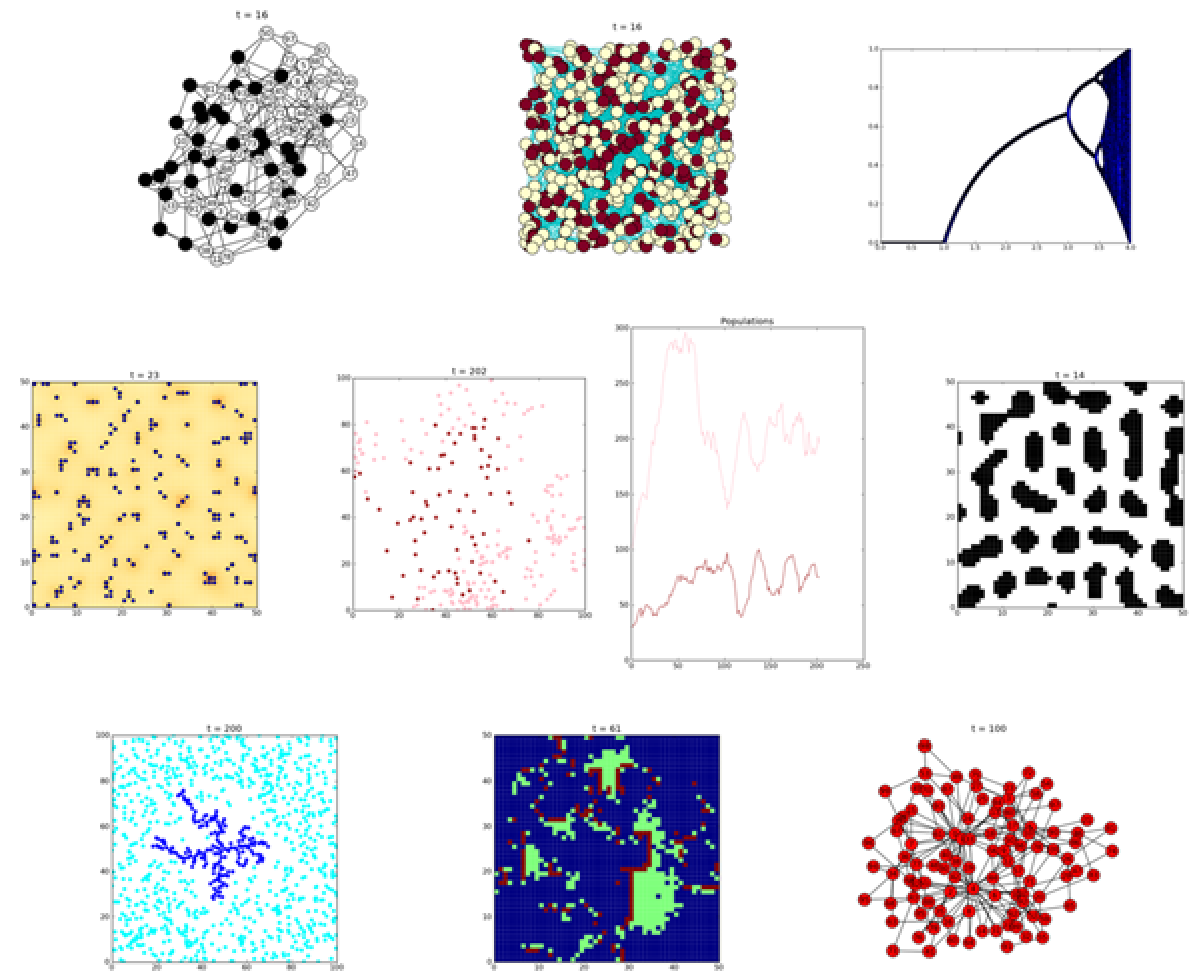
PyCX

- http://pycx.sourceforge.net/
- an online repository of simple Python sample codes for dynamic complex systems simulations:
  - iterative maps
  - cellular automata
  - dynamical networks
  - agent-based models
- runs on plain Python
- philosophy:
  - simplicity
  - ☟readability
  - ☟generalizability
  - ☟pedagogical values

PyCX

- coding style:
  - one simulation model, one .py file
  - 📢 same three-part structure for all dynamic simulations
    → initialization, visualization, updating
  - 📢no object-oriented programming
  - 📢frequent use of global variables
- first steps (provided Python is already installed):
  - download a PyCX sample code of your interest
  - run it
  - read it
  - change it as you like!!!

PyCX



Case study: forest fire model

Case study: forest fire model

- early versions go back to Henley (1989) and Drossel and Schwabl (1992)
- defined as a cellular automaton on a grid with L^d cells (L – sidelength of the grid, d – dimension of the grid)
- a cell can be:
  - empty
  - occupied by a tree
  - burning
- in the simplest version 2 rules executed simultaneously:
  1. A burning cell turns into an empty cell
  2. A tree will burn if at least one neighbor is burning

Case study: forest fire model

Demos