# Complex Systems 530: Computer Modeling of Complex Systems

Lecture 1: Introduction and overview

# Welcome!

- Today
  - Class logistics & introductions
  - Why model? How to choose what kind of model to use?
  - Agent-based models (ABMs)
  - ABM exploration

# Course Info

- Instructors: Marisa Eisenberg & Conrad Kosowsky
- Office Hours - TBD
- Course website & syllabus
  **https://epimath.github.io/cscs-530-materials/**
  (Go through this together & talk about the plan for the class)
- Course Structure
  - Homework/Labs & Project
- Course Goals & Philosophy

# Course overview

- Focus of course: how to design, build, simulate, visualize, analyze, document, and compare agent-based & other complex systems models
- Along the way, we will also explore:
  - A bit of philosophy of modeling/complex systems
  - Intro/intermediate programming
  - Some probability, statistics
  - Cellular automata
  - Some network (graph) theory & data analysis
  - A bit of collective behavior, pattern formation, & game theory
  - Models in social, biological, and physical systems

# Course overview

- We will not focus as much on equation-based models (e.g. ODE’s, PDE’s)—these are also super useful! But there are a lot of existing courses covering these topics (e.g. CMPLXSYS 511, 541)
    - (That said, if math is your jam, we will have some of that too!)
    - If you are curious whether your question is better addressed with ABMs vs. other kinds of models, we will discuss some & can talk more as we go

# Goals for the course

- Learn about models and model-building—how and why to build models for a given problem
- Become a sophisticated consumer of articles involving ABM and gain a solid grounding in some of the key models in the field
- Develop familiarity and competencies in Python, and some familiarity with NetLogo
- Leave this class with an ABM you have designed, built, and analyzed that will (hopefully) be useful to you in your future research!

# Languages for Coding

- Example code and HW files will be mostly in NetLogo & Python 3
    - See website for more information on how to install
    - Google Colab or you can run on your computer (e.g. Anaconda)
- However, for much of the class you can code in whatever language is comfy for you if you prefer (e.g. R, MATLAB) (within reason…)

# Prior Background

- A background in complex systems is not required, but hopefully you have a strong interest!
- Some basic programming and basic math/stats background is **strongly** encouraged.

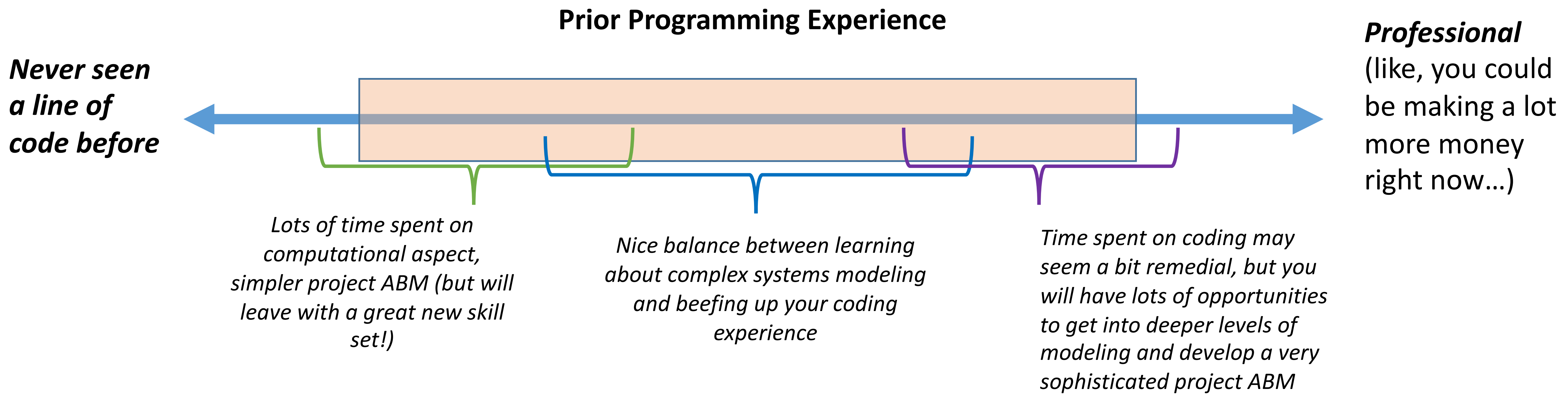
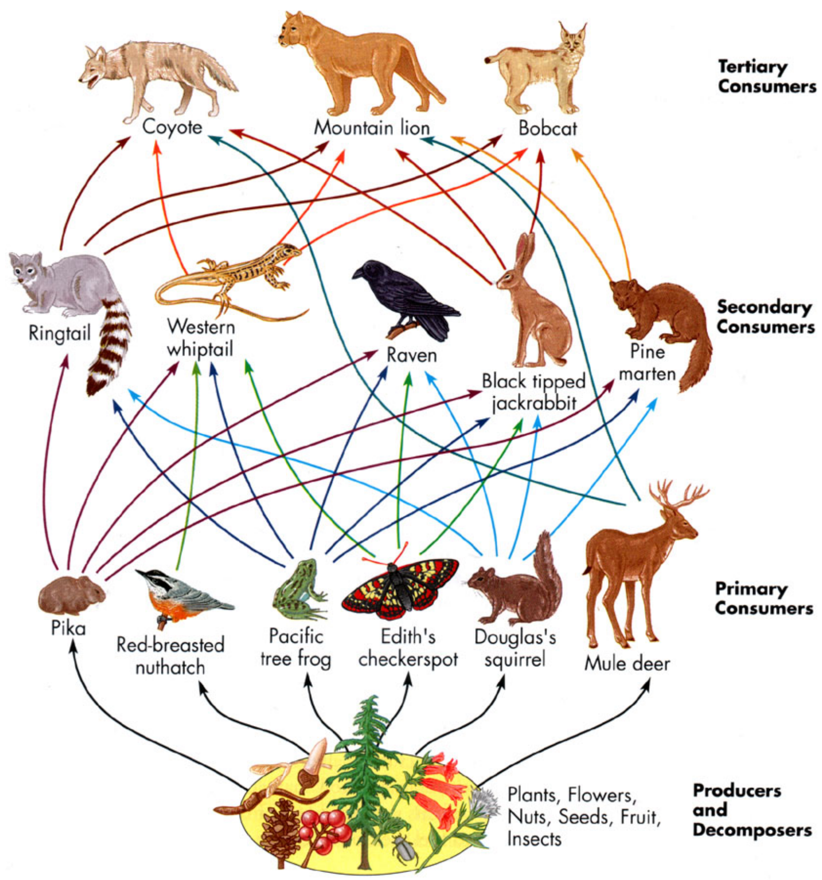


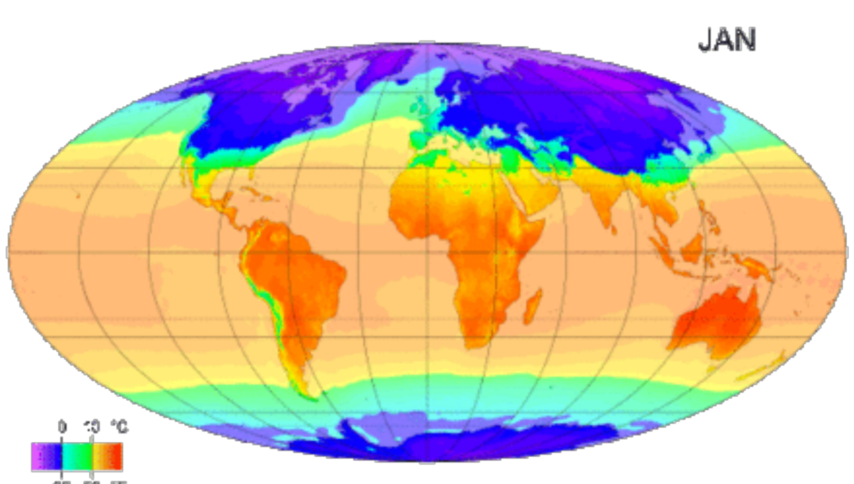

diagram borrowed from Lynette Shaw!

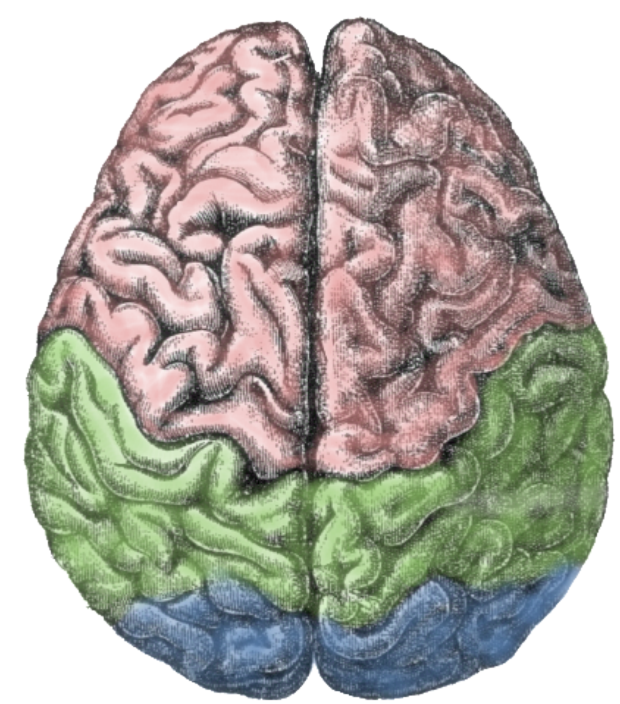
# Introduction

# What is a complex system?




© Daniel Biber, Germany, Shortlist, Professional, Natural World & Wildlife, 2018 Sony World Photography Awards



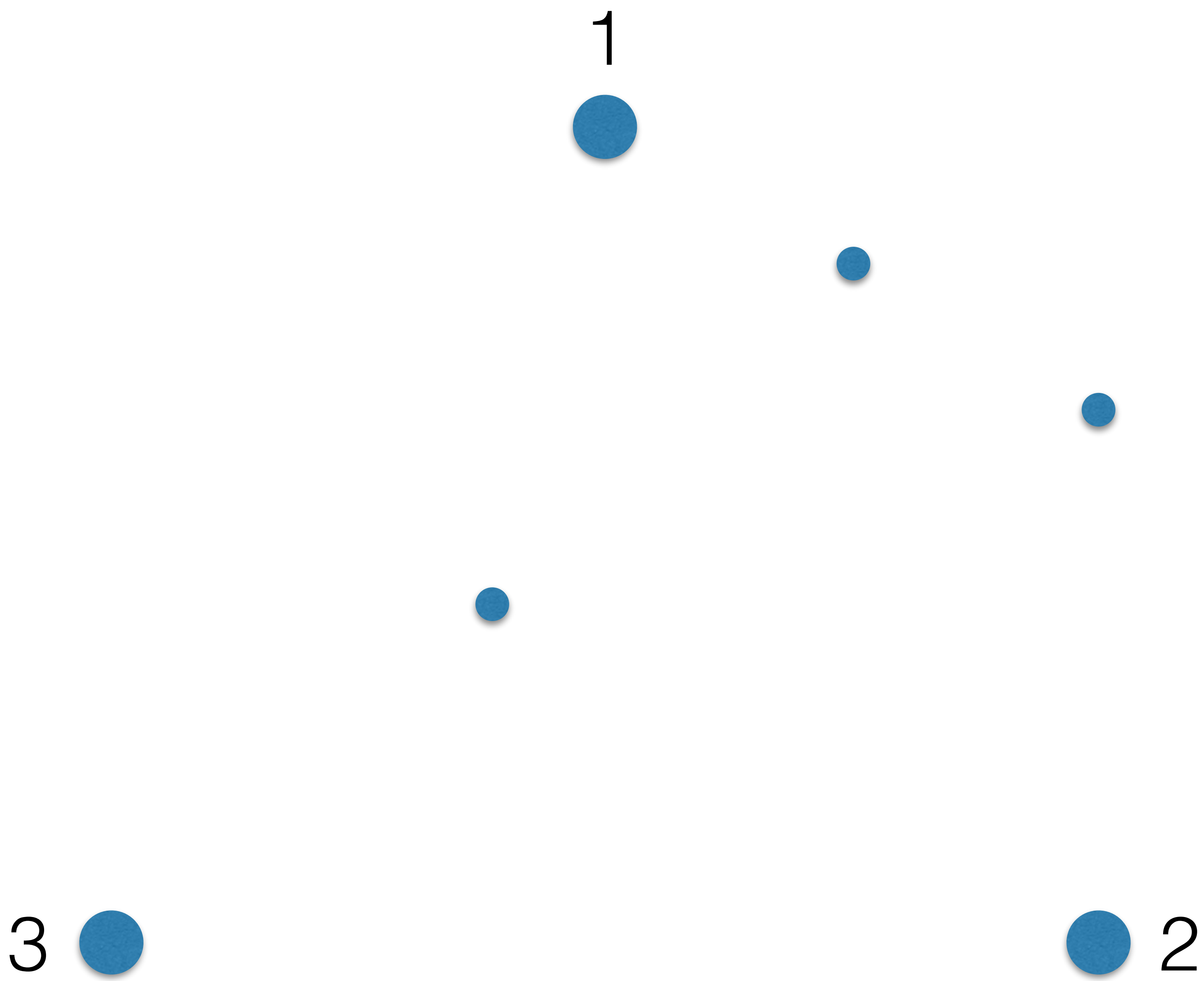



# What is a complex system?

- **A collection of components that interact, often in nonlinear ways**

- Sayama: “Complex systems are networks made of a number of components that interact with each other, typically in a nonlinear fashion. Complex systems may arise and evolve through self-organization, such that they are neither completely regular nor completely random, permitting the development of **emergent behavior at macroscopic scales**.”

# Interactive Example!



# Interactive example

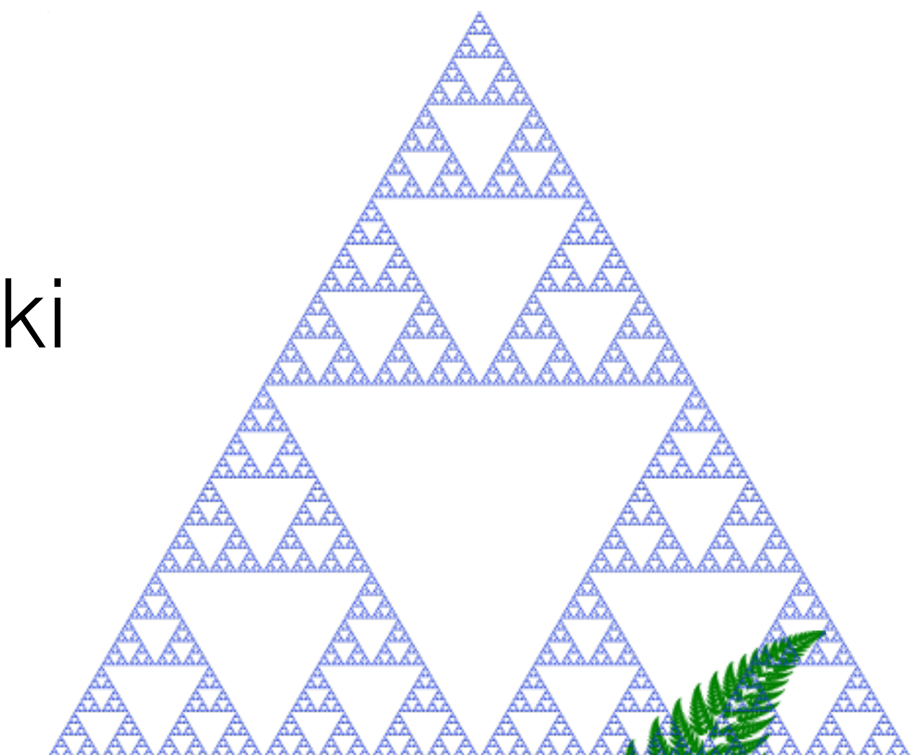
- 1) Mark a point at random within the triangle (can really be anywhere but this is simpler)
- 2) Randomly pick a vertex
- 3) Mark your next point: the point halfway between your current point and the vertex you chose
- Repeat steps 2) and 3)

What do you think will happen? (and why?)

Let’s try it out!

# The chaos game!

- Generates a fractal! This is called the **Sierpiński gasket** or Sierpiński triangle
- Fractals are objects that exhibit self-similarity (ferns, trees, snowflakes, rivers, & more)
- Simple rules can generate surprising emergent patterns!



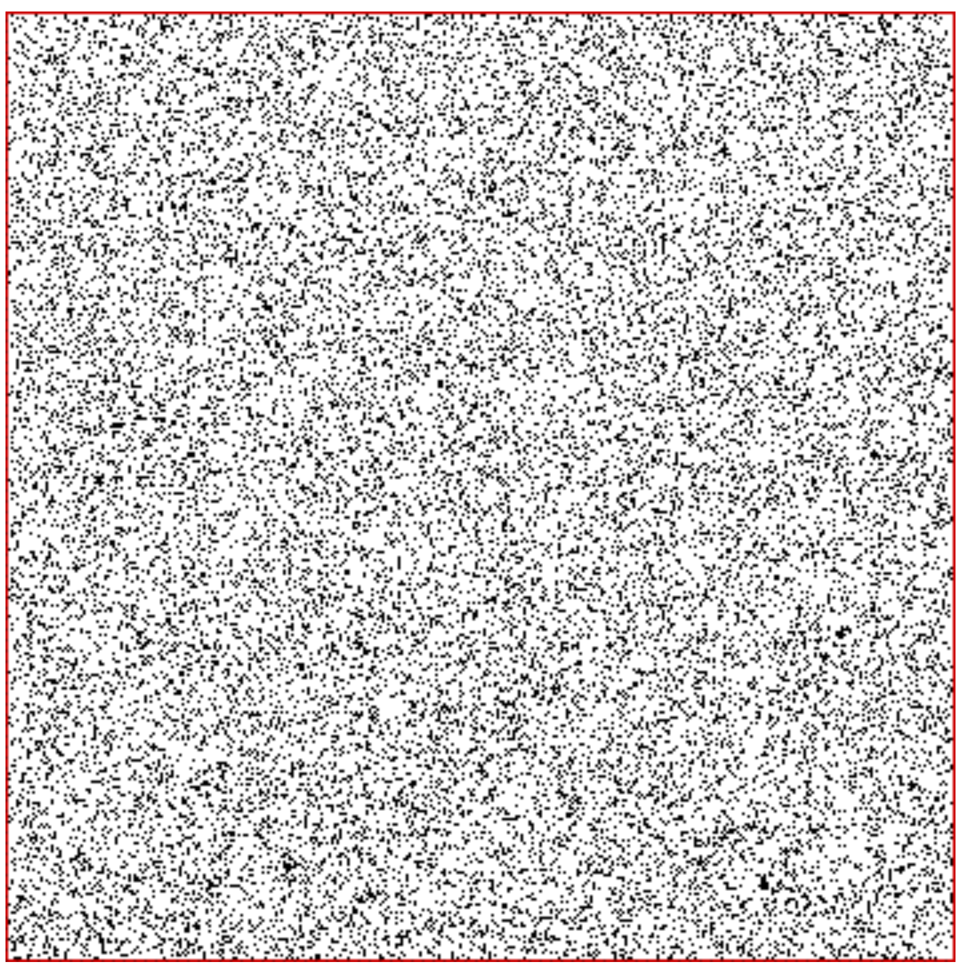
Interactive game is from: http://math.bu.edu/DYSYS/chaos-game/node4.html

# The chaos game

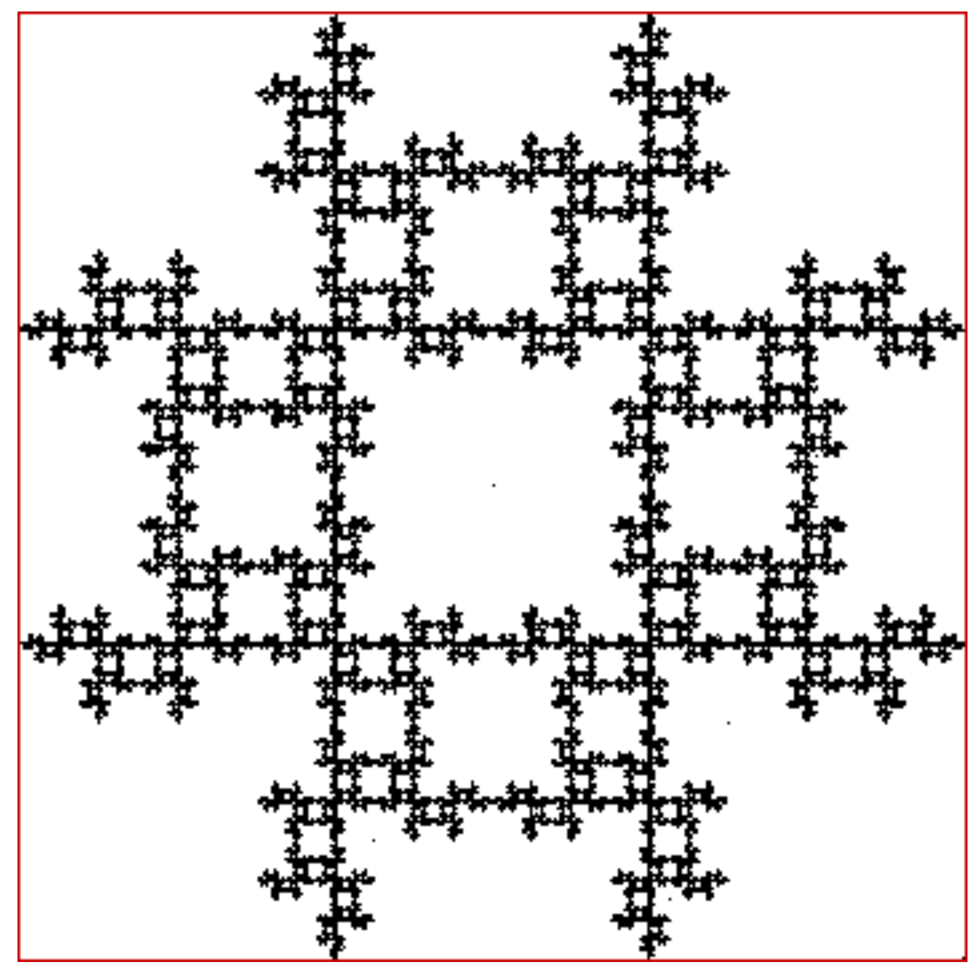
- The Sierpinski gasket is an attractor for this process
- Why does this work? A little intuition—think about transformations—discuss
- We will do more with this later
- Does it work with other shapes?

# The chaos game

Try it with a square?



Use a square but don't choose the same vertex twice



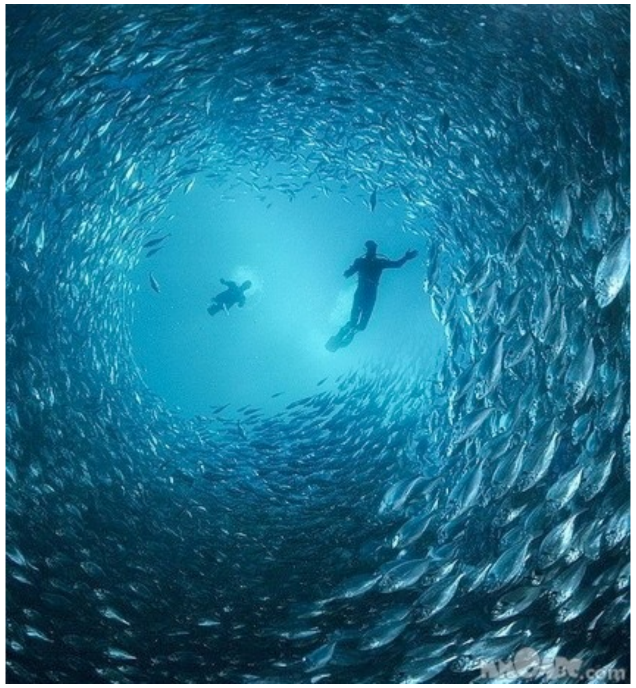
Many other variations!

# Complex Systems

- Emergent behavior
- Self-organization
- Adaptive interactions
- “Fat-Tail” Behavior
- Chaos
- Nonlinearity, tipping points, etc.

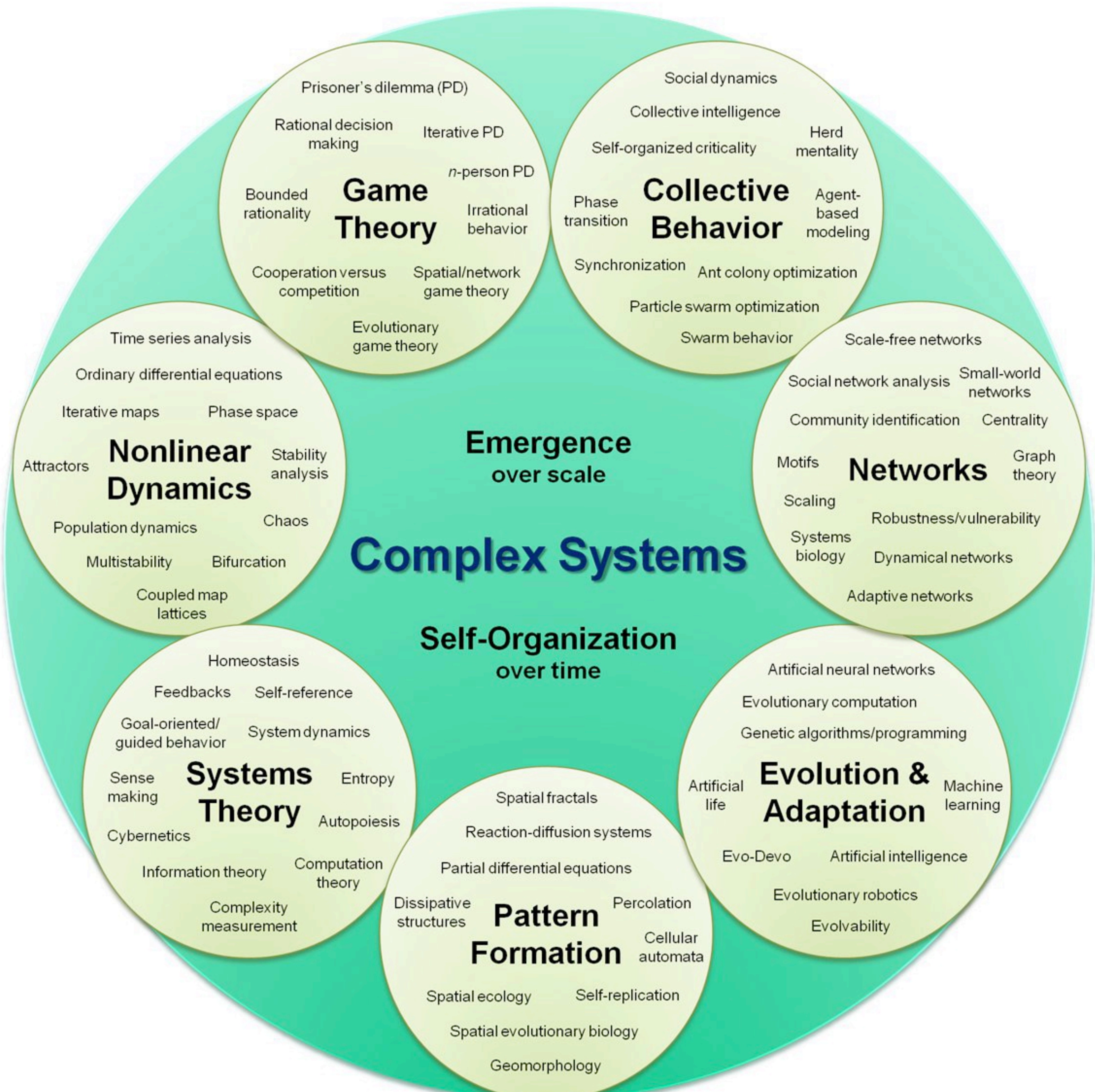
# Example: flocking!

- Flock of geese/school of fish
  - Forms a large, organized pattern
  - But no 'group mind' or leader
  - Birds follow local rules
  - Result is emergent, organized behavior



https://www.youtube.com/watch?v=e8Prw9AZ9jw

# Complex Systems Science




Introduction to the Modeling and Analysis of Complex Systems by Hiroki Sayama

# What is a model?

- In a few words, what does a model mean to you?

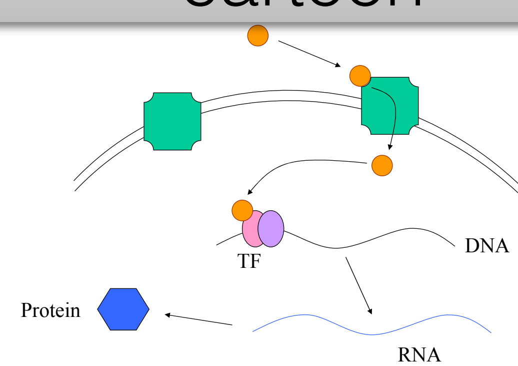
# What is a model?

Reality

Abstraction of Reality

Physical
e.g. miniature, animal model

Diagram
e.g. schematic, map, cartoon




Math/computational
e.g. equations, code

$$\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

```
for i in range(1,10):
    print(i)
```

# Models are abstractions

- They are **simplifications**—and so they are **wrong**!
- But they are also often **useful**
- Abstraction is both their strength & weakness
- The trick is, how much to simplify? What details to keep, which ones to remove?
- Even “simple” systems can lead to remarkably complex behavior

# Why model?

- Discuss

# Why model?

- You already model all the time! All of us build mental models, pictures, etc. of the world and the systems we're interested in.

- Really, we are asking: why build explicit models, where we lay out the mechanisms, assumptions, parameters, etc. as equations/code?

    - If we make our models explicit, we can test, interrogate, explore them—can make sure they are self-consistent, explore more complex scenarios, etc.

# Why model?

- Understand mechanisms, causality
- Help to guide & test control, treatment, intervention strategies,
- Test alternative scenarios (counterfactuals)
- Forecast/predict

# Why model?

From Epstein 2008:

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

# Why model?

| | |
|---|---|
| Known knowns | Known unknowns |
| Unknown knowns | Unknown unknowns |

# Why model?

| | |
|---|---|
| Known knowns | Known unknowns |
| Unknown knowns | Unknown unknowns |

# Two fun examples of unknown knowns

- Not quite the same type of unknown known as for our modeling work but I thought these were fun:
  - Adjectival order: opinion, size, age, shape, colour, origin, material, purpose (e.g. an old purple cat, or brown leather walking shoes)
  - Ablaut reduplication: chit-chat, singsong, flipflop, hip-hop, tit-for-tat, etc. all follow the vowel order i, a, o (bing bang boom!)

https://www.theguardian.com/commentisfree/2016/sep/13/sentence-order-adjectives-rule-elements-of-eloquence-dictionary

# Words of caution

- How to know if you have the right mechanism?
- How much do our assumptions and simplifications affect our outcome?
- May not be possible to, for example, predict certain things from the data/understanding that we have! (e.g. early epidemic curve)
- *Importance of understanding/testing assumptions, uncertainty quantification, model comparison, etc.*

# How to choose a modeling framework?

- Differential equations, stochastic population-based models, agent-based models, many others!
- Deterministic vs. stochastic?
- Discrete vs. continuous?
- Population-based or individual-based?
- Spatial/non-spatial?
- Different frameworks will have different analytical and computational tractability, interpretability, and assumptions

# How to choose a modeling framework?

- Often can model the same process with many different frameworks (ABMs, ODEs, Markov models, etc.)
- Can also sometimes implement the same or equivalent model in different frameworks
- Discuss for:
  - Epidemics
  - Swarming/flocking (e.g. murmuration, fish schooling)
  - Population growth (e.g. birth/death processes)

# How to choose a modeling framework?

- Depends on the problem/question of interest!

- ABMs often particularly advantageous for questions where individual heterogeneity is key (e.g. spatial position of individuals, individuals with varying properties, etc.)

- May also just be more illustrative/clear/interpretable in one framework or another even if equivalent

# How to choose a modeling framework?

- What if the modeling framework you choose affects your results?
- More generally, how to decide how realistic/simplified to make your model?
- Model comparison & inference robustness assessment—more on this later

Koopman JS. Infection transmission science and models. Japanese journal of infectious diseases. 2005 Dec 1;58(6):S.
Pollock KH. Inference robustness vs. criterion robustness: an example. The American Statistician. 1978 Nov 1;32(4):133-6.

# Agent-based models (ABMs)

# Agent-based models!

- Agents - independent “agents” move, interact, explore environment, etc.
- Environment - agents exist in a non-agent environment (can be static or dynamic)
- Rules/interactions - to govern agent behavior, how they interact with the environment, etc.

# Motivating example: video games!

# Agent-based models

- Advantages
  - Can handle situations where population cannot be viewed as aggregates
    - Heterogeneity
  - Often a natural description of system - think in terms of individuals & their decisions/actions
    - Builds from micro (individual process) to macro (overall emergent behavior)
  - Flexible, can account for more complexity/detail

# Agent-based models

- Disadvantages
  - Often harder to develop, document, and validate
  - Less of an existing analytical framework for understanding dynamics, doing parameter estimation, etc.
  - Flexibility towards complexity/complicatedness is both a strength & weakness
  - Can be tempting to make highly complicated & realistic—can make it difficult to know what's going on. Be careful to keep it 'as simple as possible (but not simpler)'

# We will explore

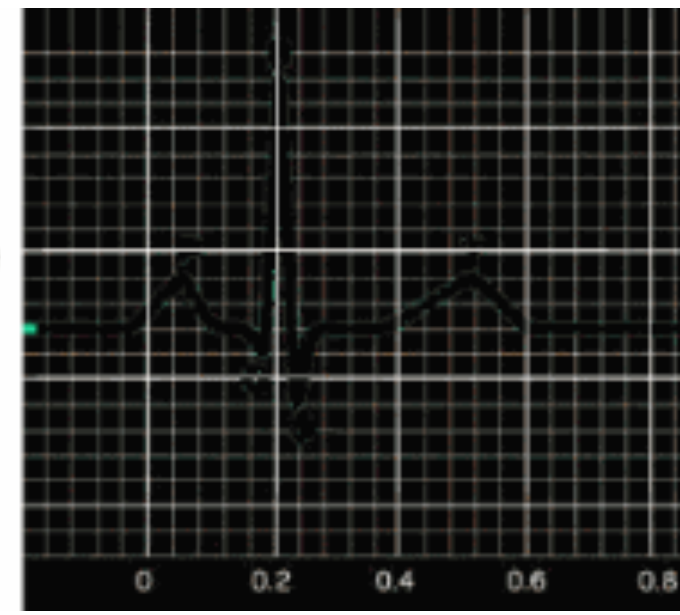
- Designing, building, and simulating ABMS
- Parameter exploration & sampling - exploring the model behavior
- Interpreting results
- Documenting ABMs!
- Alternative models & inference robustness - how do the inferences/predictions/outcomes/explanatory power of our model change as we add realism?

# Let’s try out an ABM!

- Forest fire model: https://ncase.me/simulating/model/
- What do you notice?
- Any interesting behaviors as you adjust things? You can also add other agent types into the mix!
- What happens if we increase tree growth to 5% and start with all trees?
- Modeling by analogy: what else could this model (or similar) be used for?

# Cardiac dynamics & heart muscle tissue

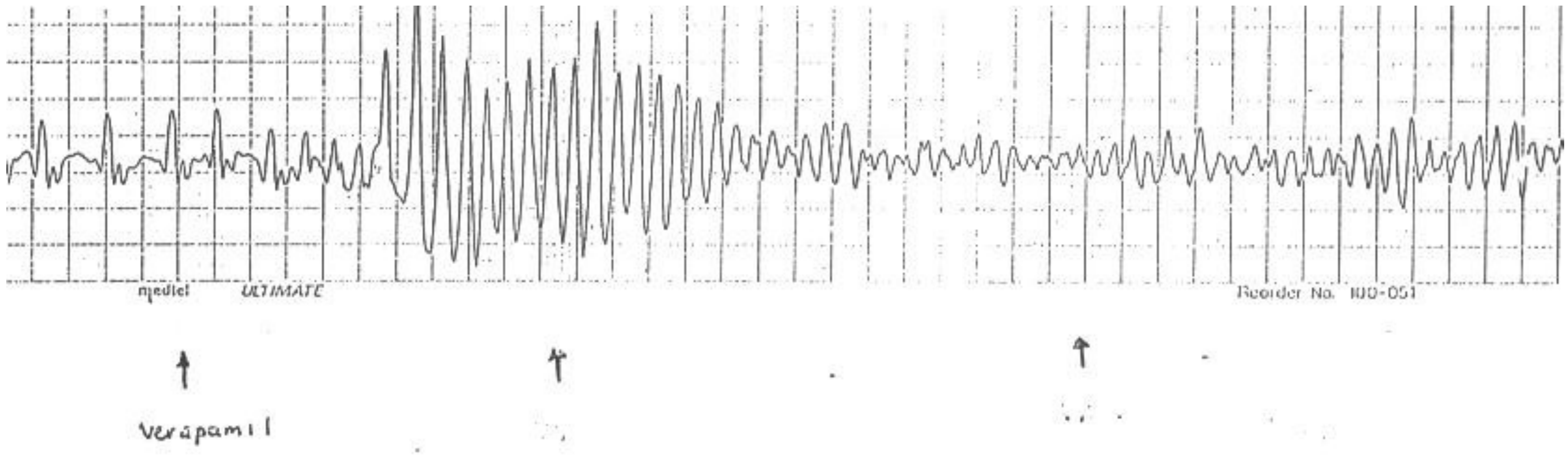
- Normal rhythm: depolarization wave contracts atrium and ventricle in a regular rhythm, originating at sinoatrial node & traveling to atrioventricular node




http://www.scholarpedia.org/article/Cardiac_arrhythmia

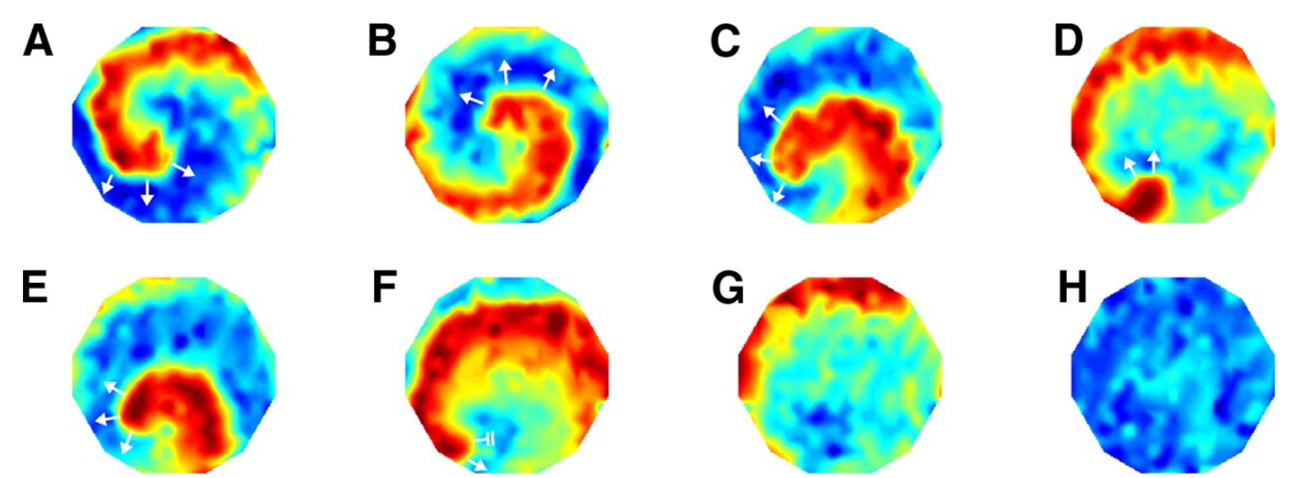
- However, heart can undergo bifurcation to other behaviors—arrhythmias

# Cardiac arrhythmias



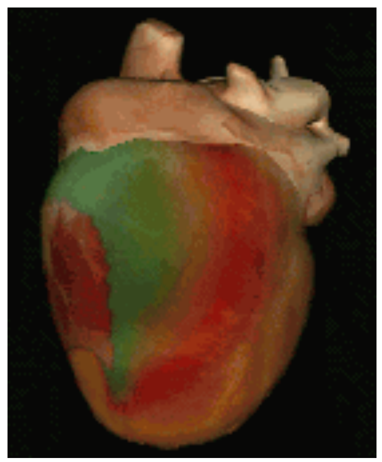

- Tachycardia—>ventricular flutter—>ventricular fibrillation

- These bifurcations can often be understood in terms of spiral waves, period doubling bifurcations, and other spatiotemporal dynamics

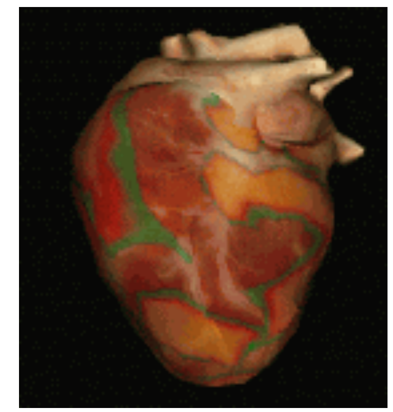



http://ajpheart.physiology.org/content/294/1/H58.figures-only

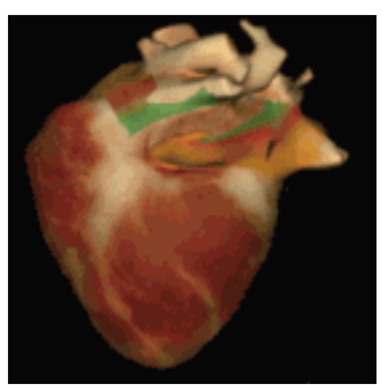
# Spiral/scroll waves in heart arrhythmias



Ventricular tachycardia



Ventricular fibrillation



Atrial flutter

Atrial fibrillation

http://www.scholarpedia.org/article/Cardiac_arrhythmia

# Now let's make one from scratch

- https://ncase.me/sim/?s=blank
- Simple voting model!
- Everyone starts with some random initial preference
- They tally the planned votes of their neighbors and if more than half of neighbors are voting the other way, they switch
- What happens?

Wilensky, U. (1998). NetLogo Voting model. http://ccl.northwestern.edu/netlogo/models/Voting. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

# Voting model

- Things to try:
  - Award ties differently (e.g. switch if exactly half of neighbors vote the other way)
  - Different initial percentages of each party
- Try with 3 parties
  - What happens? Same patterns?

# More emoji ABMs

- https://ncase.me/sim/
- Explore!

# For next time…

- Read
  - “Why Model?”, Epstein 2008
  - Sayama, Chp. 1-2
  - “More is Different,” Anderson 1987
  - Wilensky, Chp. 0-1
- Do
  - Download, install, and try out NetLogo and Python (google colab or install)