ABM in BCM

Diving in with Poseidon

Nicolas Payette

April 18, 2021

How to draw an owl

1.

1. Draw some circles

How to draw an owl

1.

2.

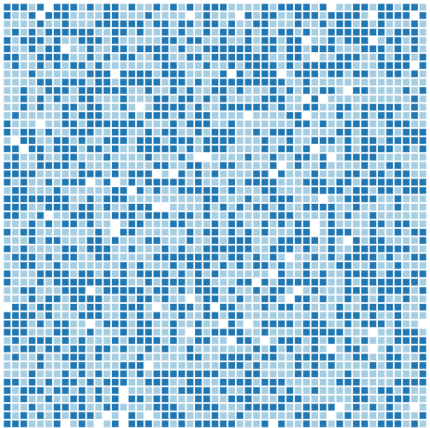
1. Draw some circles

2. Draw the rest of the [illegible] owl

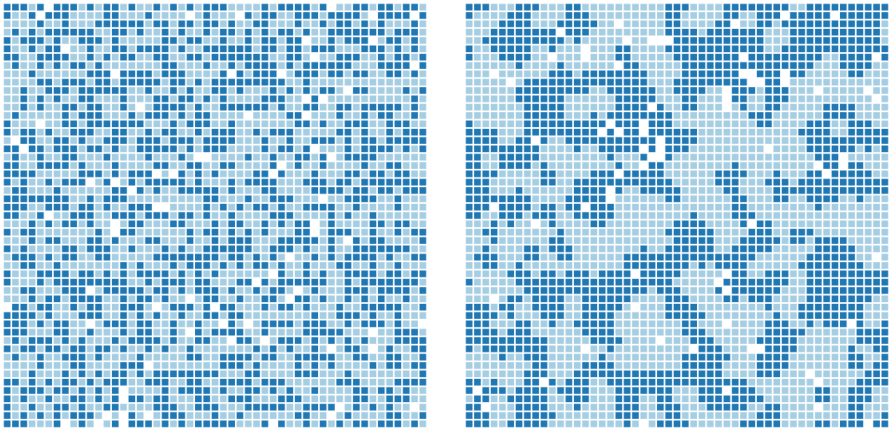
What's the plan?

- A little bit about **agent-based models** (ABM)
- A little bit about the **Poseidon** fisheries ABM
- A little bit about the **NetLogo** modelling platform
- A whole lot about building **Poseidon in NetLogo**!

Early ABM: Schelling's models of segregation

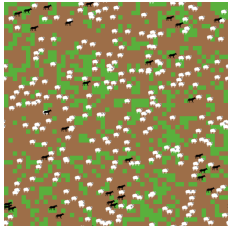


Early ABM: Schelling's models of segregation

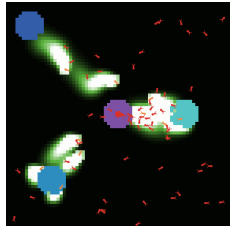


Proportion of similar neighbours wanted: 30%.

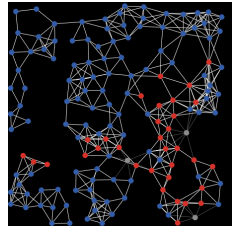
Examples from the NetLogo model library



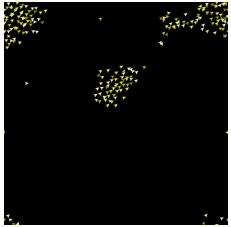
Wolf Sheep Predation



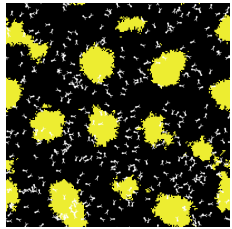
Ants



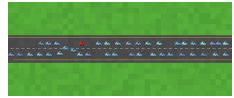
Virus on a Network



Flocking



Termites



Traffic 2 Lanes

Some typical characteristics of ABM

Heterogeneity: agents have their own individual properties and can differ from one another.

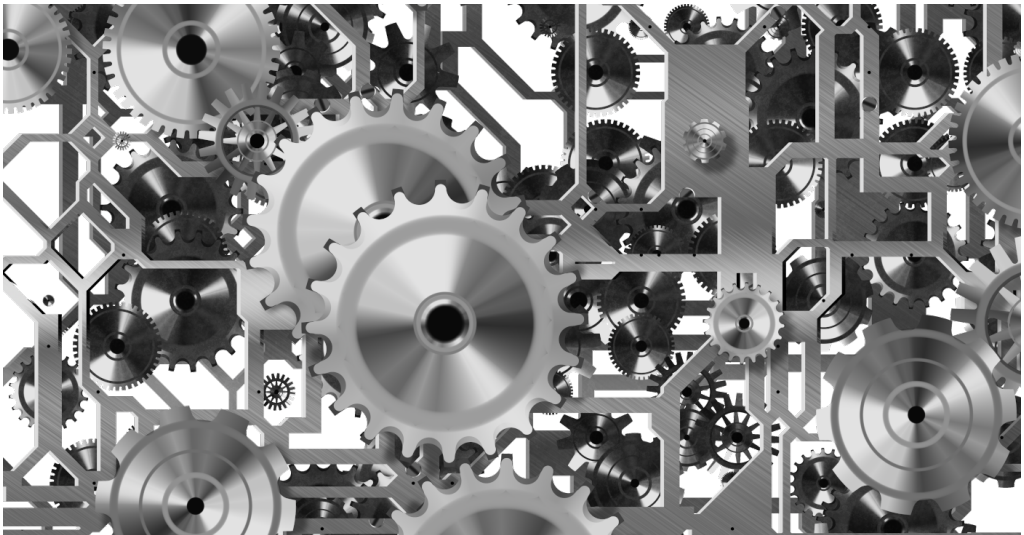
Autonomy: agents makes their own decisions; there is no central control.

Explicit space: agents are located in a well-defined environment (e.g., a network or a grid).

Local interactions: agents interact with their neighours and their local environment.

Bounded rationality: agents are not omnicient and often use behavioural heuristics.

See: Epstein, Joshua M. 2006. Generative Social Science: Studies in Agent-Based Computational Modeling. Princeton University Press.



Why model?

Epstein, J. M. 2008. Journal of Artificial Societies and Social Simulation 11(4):12. http://jasss.soc.surrey.ac.uk/11/4/12.html

1. Explain (very distinct from predict)
2. Guide data collection
3. Illuminate core dynamics
4. Suggest dynamical analogies
5. Discover new questions
6. Promote a scientific habit of mind
7. Bound (bracket) outcomes to plausible ranges
8. Illuminate core uncertainties.
9. Offer crisis options in near-real time
10. Demonstrate tradeoffs / suggest efficiencies
11. Challenge the robustness of prevailing theory through perturbations
12. Expose prevailing wisdom as incompatible with available data
13. Train practitioners
14. Discipline the policy dialogue
15. Educate the general public
16. Reveal the apparently simple (complex) to be complex (simple)

Why model fisheries?

- It's big!
  - 96.4 million tonnes of fish caught in 2018
  - employing 59.51 million people
  - 17% of total animal protein consumed globally
- It's in trouble!
  - 34.2% of stocks are overfished

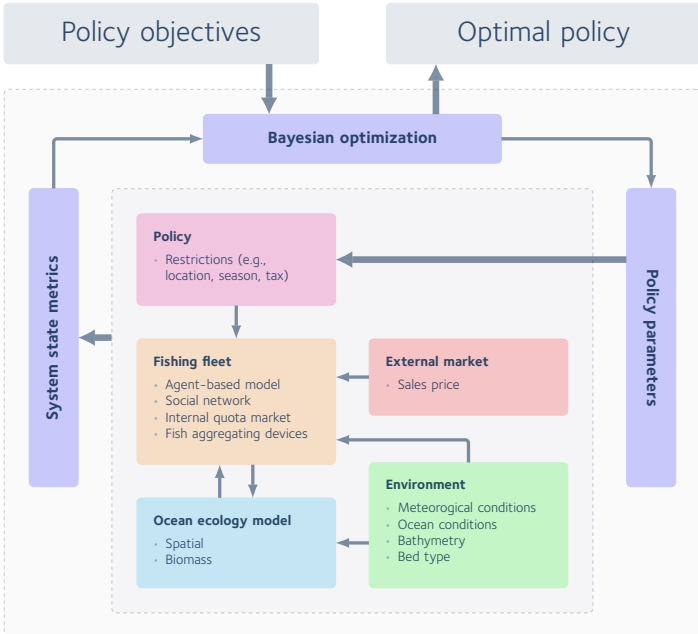
Why use ABM to do it?

- They're inherently spatial;
- they involve complex interactions,
- between smart, adaptive agents.

Source: `http://www.fao.org/3/ca9231en/CA9231EN.pdf`

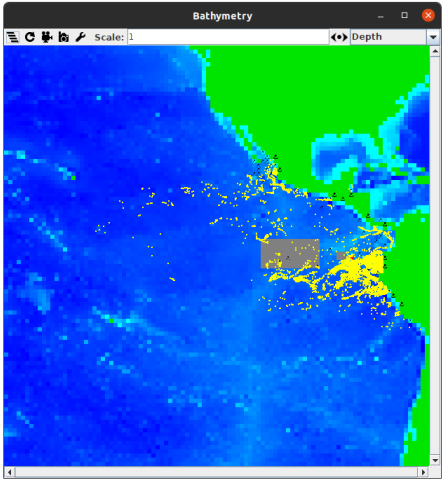
Poseidon

Source: Ricardo André Frantz, `https://w.wiki/wzx`



Poseidon applications

- Conceptual models
- US West Coast Groundfish
- Indonesian Deep water snapper grouper fishery
- Eastern Pacific Tuna Management



What can we put in a minimal fisheries model?

Agents:

- A port,
- fishers,
- and fish!

Some policy to test:

- The size of a marine protected area (MPA)

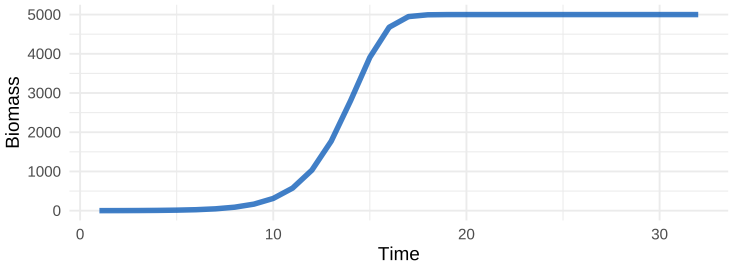
Fisher agents in Poseidon use the **“Explore, exploit, imitate”** behaviour algorithm:

- Should I go exploring?
  - If yes, pick a random spot not too far from my favourite spot (**EXPLORE**)
  - If not, pick a friend and ask: is my favourite spot better than my friend’s favourite spot?
    - If it is, go to my favourite spot (**EXPLOIT**)
    - If it isn’t, go to my friend’s favourite spot (**IMITATE**)
- If the spot I went to was better than my favourite spot, it becomes my new favourite!

A bit of biology

We do not simulate individual fish. We keep track of the total biomass and apply yearly **logistic growth**.

$$\frac{dP}{dt} = rP\left(1 - \frac{P}{K}\right)$$



Pierre-François Verhulst (1804–1849).
Source: `https://w.wiki/x2o`.

ABM frameworks

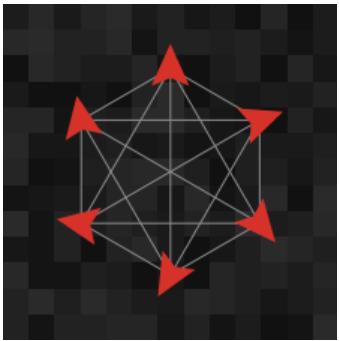
- **NetLogo** (Logo)
  `https://ccl.northwestern.edu/netlogo/`
- **MASON** (Java; MASON is what we use for Poseidon)
  `https://cs.gmu.edu/~eclab/projects/mason/`
- **MESA** (Python)
  `https://github.com/projectmesa/mesa/`
- **Repast** (Groovy or Java)
  `https://repast.github.io/`
- **GAMA** ("GAMA agent modelling language")
  `https://gama-platform.github.io/`
- **HASH** (Python or JavaScript; free but **not** open-source)
  `https://hash.ai/`

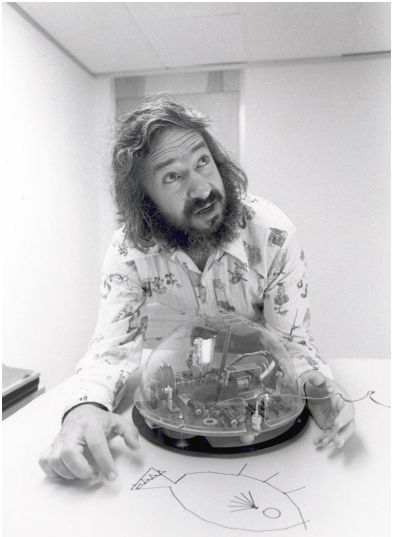
Why NetLogo?

- *Lingua franca* of the ABM community
- All-in-one IDE for GUI building / model coding / experiment running
- Logo language optimized for ABM
- “Low threshold, high ceiling” (i.e., more powerful than you think)

Agents in NetLogo

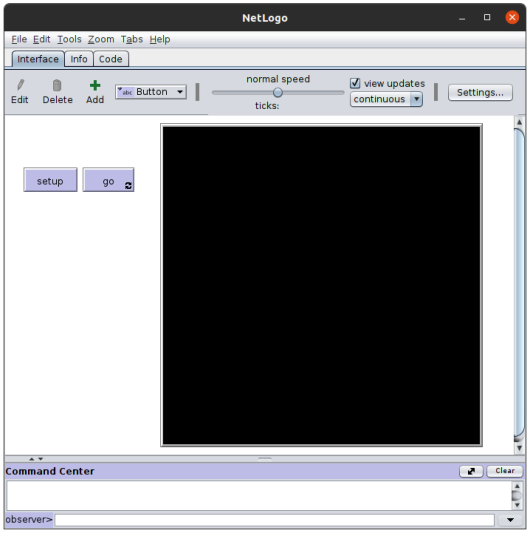
Turtles, patches and links





Seymour Papert, 1928–2016
(Credit: Cynthia Solomon/MIT Media Lab)

Basic skeleton of a NetLogo model




```
to setup
  clear-all
  reset-ticks
end

to go
  tick
end
```

Creating a port

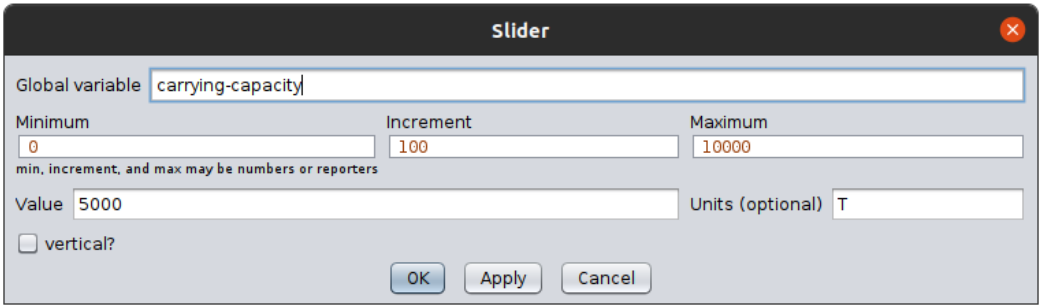
```
breed [ ports port ]

to setup
  clear-all
  create-ports 1 [
    set xcor max-pxcor
    set color lime
    set shape "square"
  ]
  reset-ticks
end
```

Source: https://www.tourismeilesdelamadeleine.com/fr/decouvrir-les-iles/les-iles/ile-de-grande-entree

Adding some fish biomass

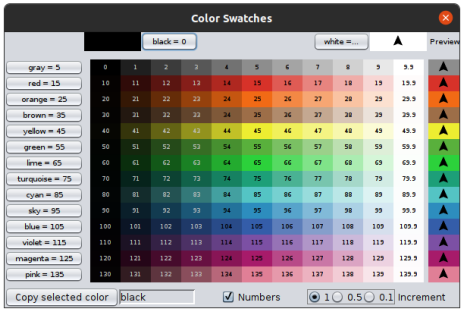
```
breed [ ports port ]
patches-own [ biomass ]

to setup
  clear-all
  create-ports 1 [
    set xcor max-pxcor
    set color lime
    set shape "square"
  ]
  ask patches [ set biomass carrying-capacity ]
  reset-ticks
end
```

Adding a `carrying-capacity` slider



Recoloring patches

```
to recolor-patches
  ask patches [
    set pcolor scale-color blue (biomass / 2) carrying-capacity 0
  ]
end
```



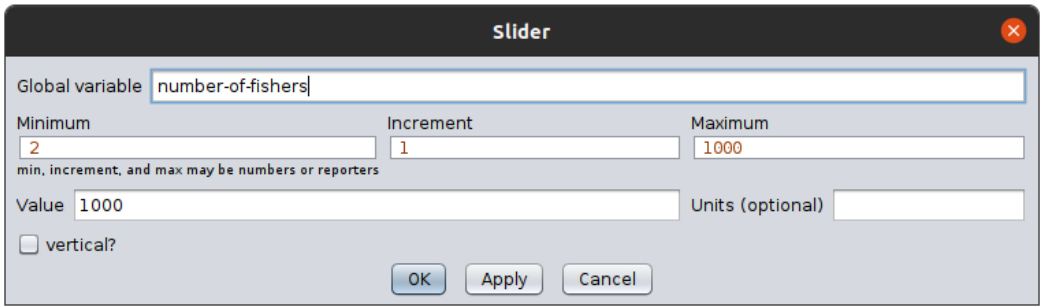
Calling `recolor-patches` from `setup`

```
to setup
  clear-all
  create-ports 1 [
    set xcor max-pxcor
    set color lime
    set shape "square"
  ]
  ask patches [ set biomass carrying-capacity ]
  recolor-patches
  reset-ticks
end
```

Creating some fishers

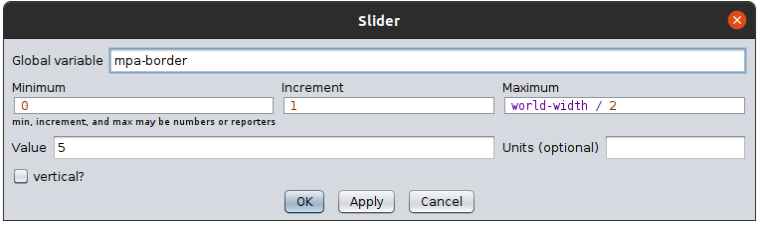
```
breed [ ports port ]
breed [ fishers fisher ]
patches-own [ biomass ]

to setup
  clear-all
  create-ports 1 [
    set xcor max-pxcor
    set color lime
    set shape "square"
  ]
  ask patches [ set biomass carrying-capacity ]
  recolor-patches
  create-fishers number-of-fishers [
    set color yellow
    move-to one-of ports
  ]
  reset-ticks
end
```

Adding a `number-of-fishers` slider



Creating a marine protected area

```
globals [ fishable-patches ] ; put this at the very top of the code tab
breed [ xs x ] ; put this before the other breed declarations

to setup-mpa
  set-default-shape xs "x"
  let mpa patches with [ abs pxcor < mpa-border and abs pycor < mpa-border ]
  ask mpa [ sprout-xs 1 [ set color cyan ] ]
  set fishable-patches patches with [ not member? self mpa ]
end
```

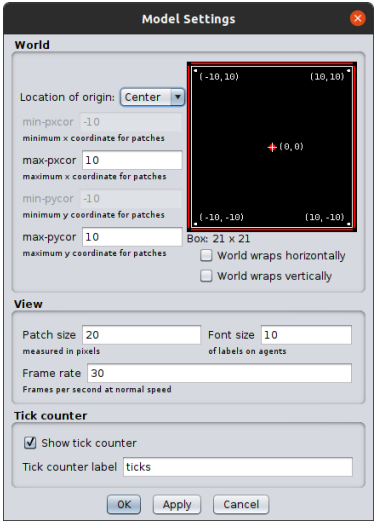


Calling `setup-mpa` from the `setup` procedure

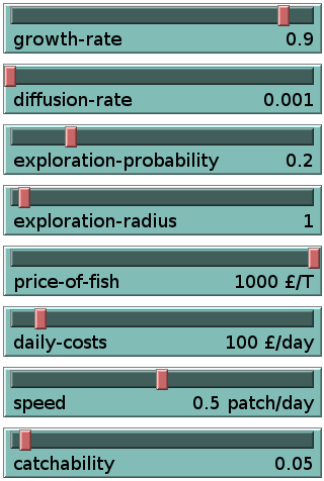
```
breed [ ports port ]
breed [ fishers fisher ]
patches-own [ biomass ]

to setup
  clear-all
  create-ports 1 [
    set xcor max-pxcor
    set color lime
    set shape "square"
  ]
  ask patches [ set biomass carrying-capacity ]
  recolor-patches
  setup-mpa
  create-fishers number-of-fishers [
    set color yellow
    move-to one-of ports
  ]
  reset-ticks
end
```

Now, a bit of house keeping...

Adjusting the model settings



Creating sliders and fisher variables




```
; this goes at the top, as always
fishers-own [
  favourite-destination
  profits-at-favourite-destination
  current-destination
  trip-destination
  trip-costs
  bank-balance
  biomass-in-hold
]
```

Let’s go!

```
to go
  ask fishers [
    set trip-costs trip-costs + daily-costs
    ifelse patch-here = current-destination [
      ifelse any? ports-here [ dock ] [ fish ]
    ] [
      face current-destination
      forward speed
    ]
  ]
  tick
end
```

Fishing!

```
to fish ; fisher procedure
  let biomass-caught biomass * catchability
  set biomass biomass - biomass-caught
  set biomass-in-hold biomass-in-hold + biomass-caught
  set current-destination [ patch-here ] of one-of ports
  set pcolor red
end
```

Docking at the port

```
to dock ; fisher procedure
  let revenues biomass-in-hold * price-of-fish
  set biomass-in-hold 0
  let profits revenues - trip-costs
  set trip-costs 0
  set bank-balance bank-balance + profits

  ifelse trip-destination = favourite-destination [
    set profits-at-favourite-destination profits
  ] [
    if profits > profits-at-favourite-destination [
      set favourite-destination trip-destination
      set profits-at-favourite-destination profits
    ]
  ]
  pick-destination
end
```

Exploring, exploiting, imitating

```
to pick-destination ; fisher procedure
  ifelse random-float 1 < exploration-probability [
    ; explore:
    let r random-poisson exploration-radius
    set trip-destination [ one-of fishable-patches in-radius r ] of favourite-destination
  ] [
    let other-fisher one-of other fishers
    let other-profits [ profits-at-favourite-destination ] of other-fisher
    ifelse profits-at-favourite-destination >= other-profits [
      ; exploit:
      set trip-destination favourite-destination
    ] [
      ; imitate:
      set trip-destination [ favourite-destination ] of other-fisher
    ]
  ]
  set current-destination trip-destination
end
```

Calling `pick-destination` from `setup`

```
to setup
  clear-all
  create-ports 1 [
    set xcor max-pxcor
    set color lime
    set shape "square"
  ]
  ask patches [ set biomass carrying-capacity ]
  setup-mpa
  recolor-patches
  create-fishers number-of-fishers [
    set color yellow
    move-to one-of ports
    set favourite-destination patch-here
    pick-destination
  ]
  reset-ticks
end
```

Updating the biology

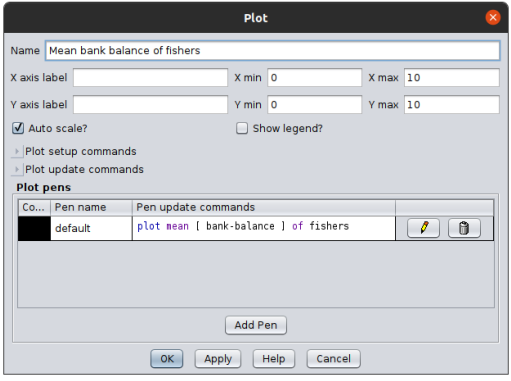
$$\frac{dP}{dt} = rP\left(1 - \frac{P}{K}\right)$$

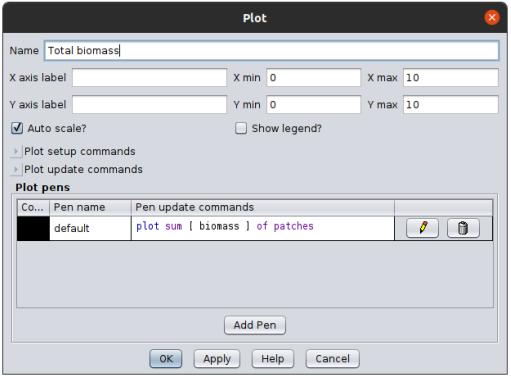
```
to update-biology
  diffuse biomass diffusion-rate
  recolor-patches
  if ticks mod 365 = 0 [
    ask patches [
      set biomass biomass + (
        growth-rate * biomass * (1 - (biomass / carrying-capacity))
      )
    ]
  ]
end
```

Calling update-biology from go

```
to go
  ask fishers [
    set trip-costs trip-costs + daily-costs
    ifelse patch-here = current-destination [
      ifelse any? ports-here [ dock ] [ fish ]
    ] [
      face current-destination
      forward speed
    ]
  ]
  update-biology
  tick
end
```

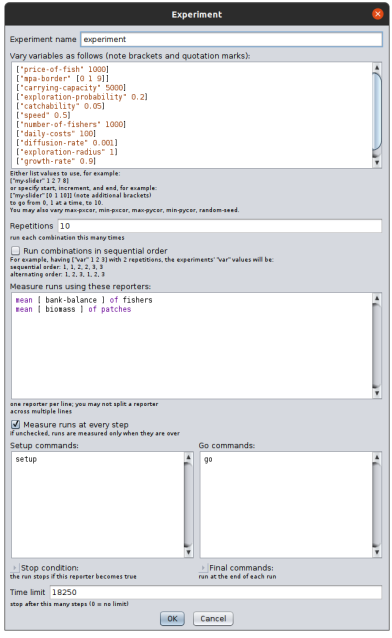
Adding a couple of plots

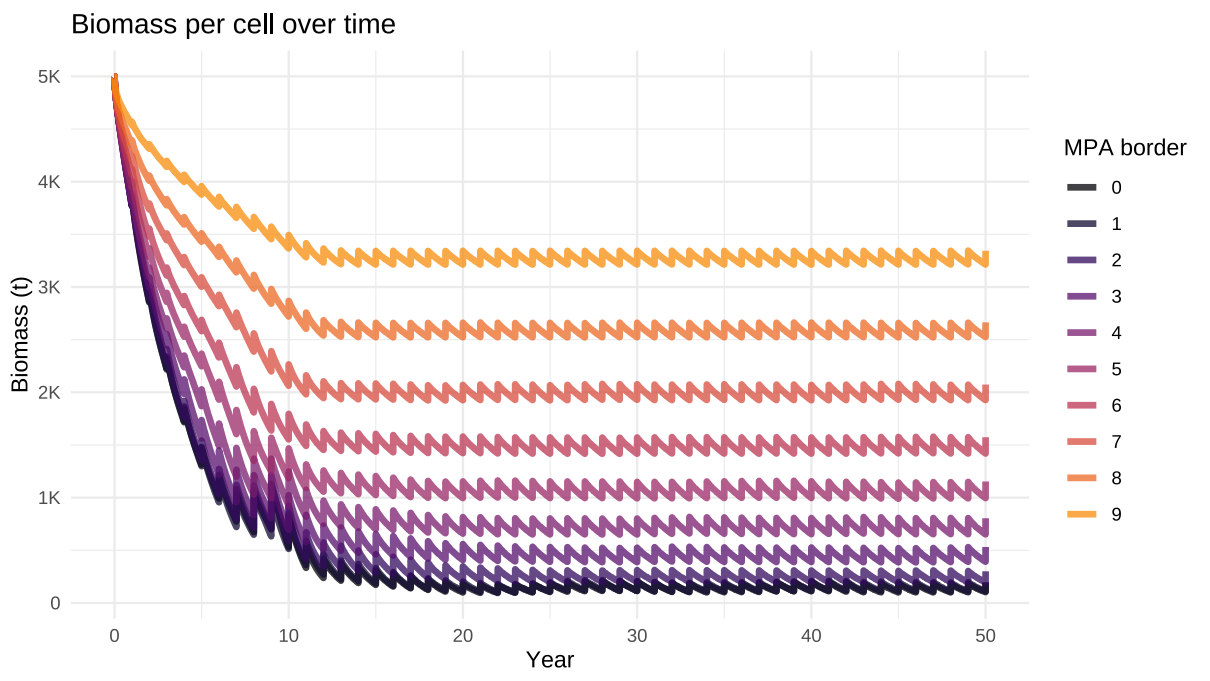


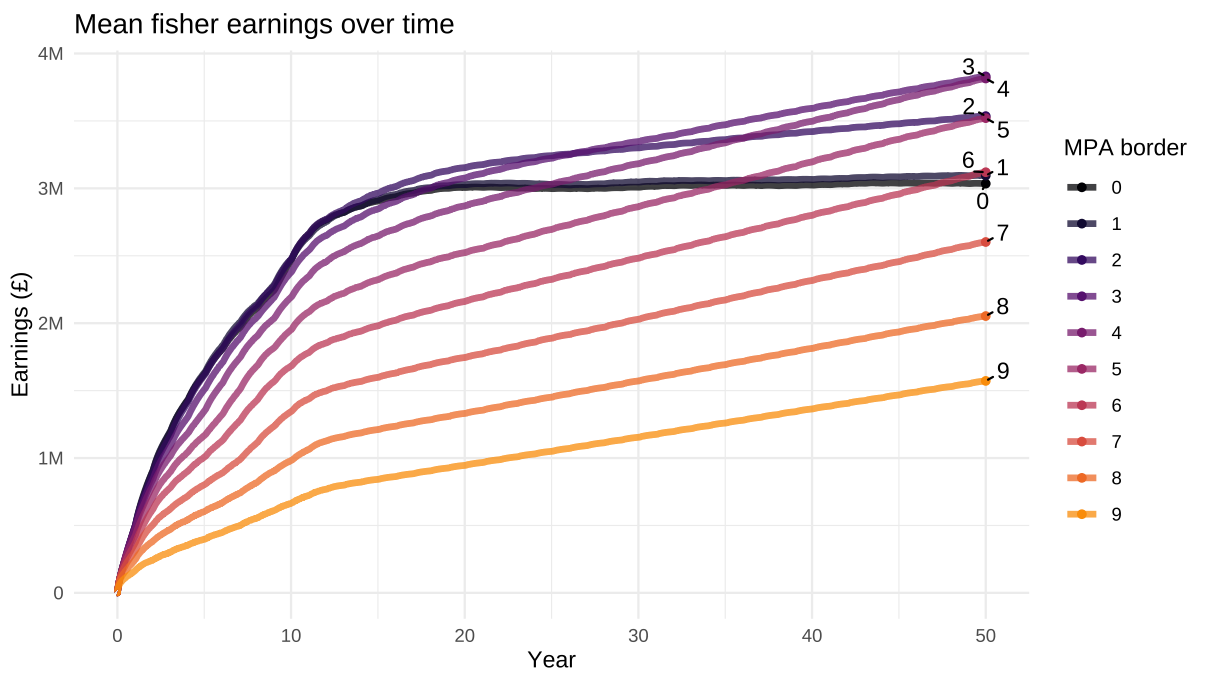


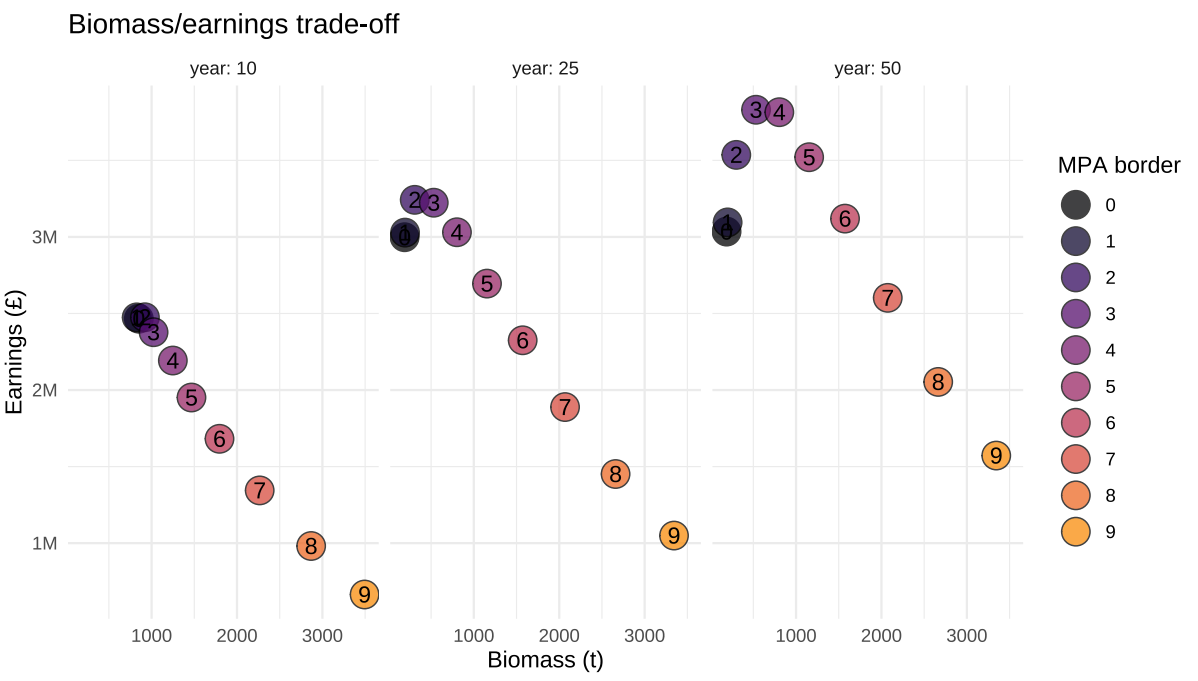
Running an automated experiment through **BehaviorSpace**

```
["price-of-fish" 1000]
["mpa-border" [0 1 9]]
["carrying-capacity" 5000]
["exploration-probability" 0.2]
["catchability" 0.05]
["speed" 0.5]
["number-of-fishers" 1000]
["daily-costs" 100]
["diffusion-rate" 0.001]
["exploration-radius" 1]
["growth-rate" 0.9]
```




Biomass per cell over time
5K
4K
3K
2K
1K
0
Biomass (t)
0
10
20
30
40
50
Year
MPA border
0
1
2
3
4
5
6
7
8
9


Mean fisher earnings over time
4M
3M
2M
1M
0
Earnings (£)
0
10
20
30
40
50
Year
3
4
2
5
6
1
0
7
8
9
MPA border
0
1
2
3
4
5
6
7
8
9


Biomass/earnings trade-off
year: 10
year: 25
year: 50
Earnings (£)
3M
2M
1M
1000
2000
3000
Biomass (t)
MPA border
0
1
2
3
4
5
6
7
8
9

“Homework”

How robust are our results?

- Which parameters are they sensitive to?
- What is the range for which they hold?

Can you add exit/entry into the fishery?

- Hint: agents can `die` or `hatch` new agents.
- How does that affect the fishery’s sustainability?

Readings about ABM

Classics

- Schelling, Thomas C. 1978. Micromotives and Macrobehavior. Norton.
- Resnick, Mitchel. 1997. Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds. MIT Press.
- Epstein, Joshua M., and Robert Axtell. 1996. Growing Artificial Societies: Social Science from the Bottom Up. Brookings Institution Press.

Textbooks

- Railsback, Steven F., and Volker Grimm. 2012. Agent-Based and Individual-Based Modeling: A Practical Introduction. Princeton University Press.
- Wilensky, Uri, and William Rand. 2015. An Introduction to Agent-Based Modeling: Modeling Natural, Social and Engineered Complex Systems with NetLogo. Cambridge, MA: MIT Press.

Current research

- Journal of Artifical Societies and Social Simulation
  `http://jasss.soc.surrey.ac.uk`

Other resources

- `https://ccl.northwestern.edu/netlogo/resources.shtml`
- `https://www.jiscmail.ac.uk/cgi-bin/webadmin?A0=simsoc`