Python Mesa

Dr. Thomas Robey
Felina Rivera Calzadillas
Supercomputing Challenge Kickoff
September 30, 2023

Most Popular Agent Based Modeling Software

- Netlogo
- Repast (Java / C++)
- MASON (Java)
- Mesa (Python)
- Agents.jl (Julia)

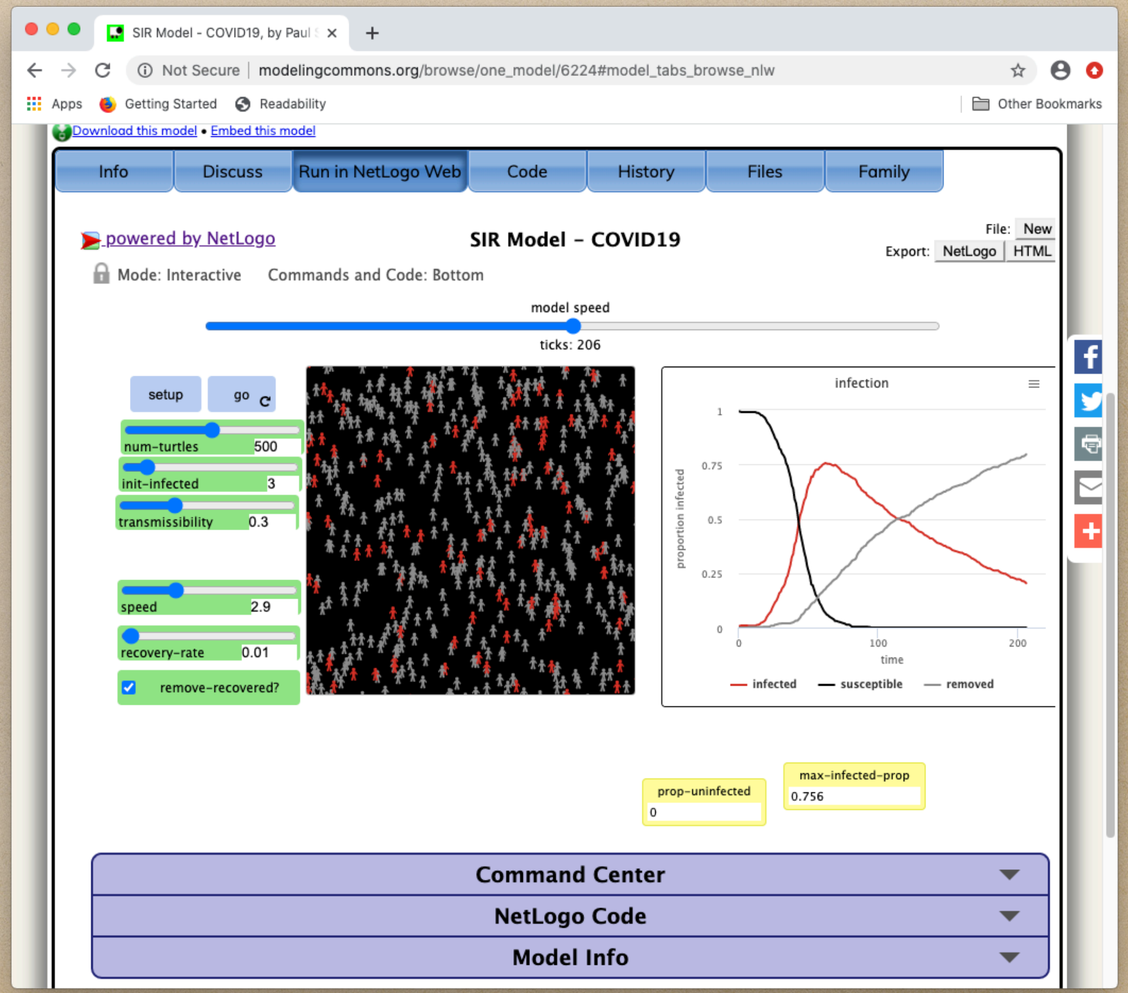
Additional Software

https://www.comses.net/resources/modeling-frameworks/

Netlogo

- Most widely used agent-based modeling environment in both education and research
- Large library of examples
- Self-contained programming environment
- Easy to learn
- Single purpose (only for agent-based modeling)
- 2 D and 3D grid
- ~24 years of development

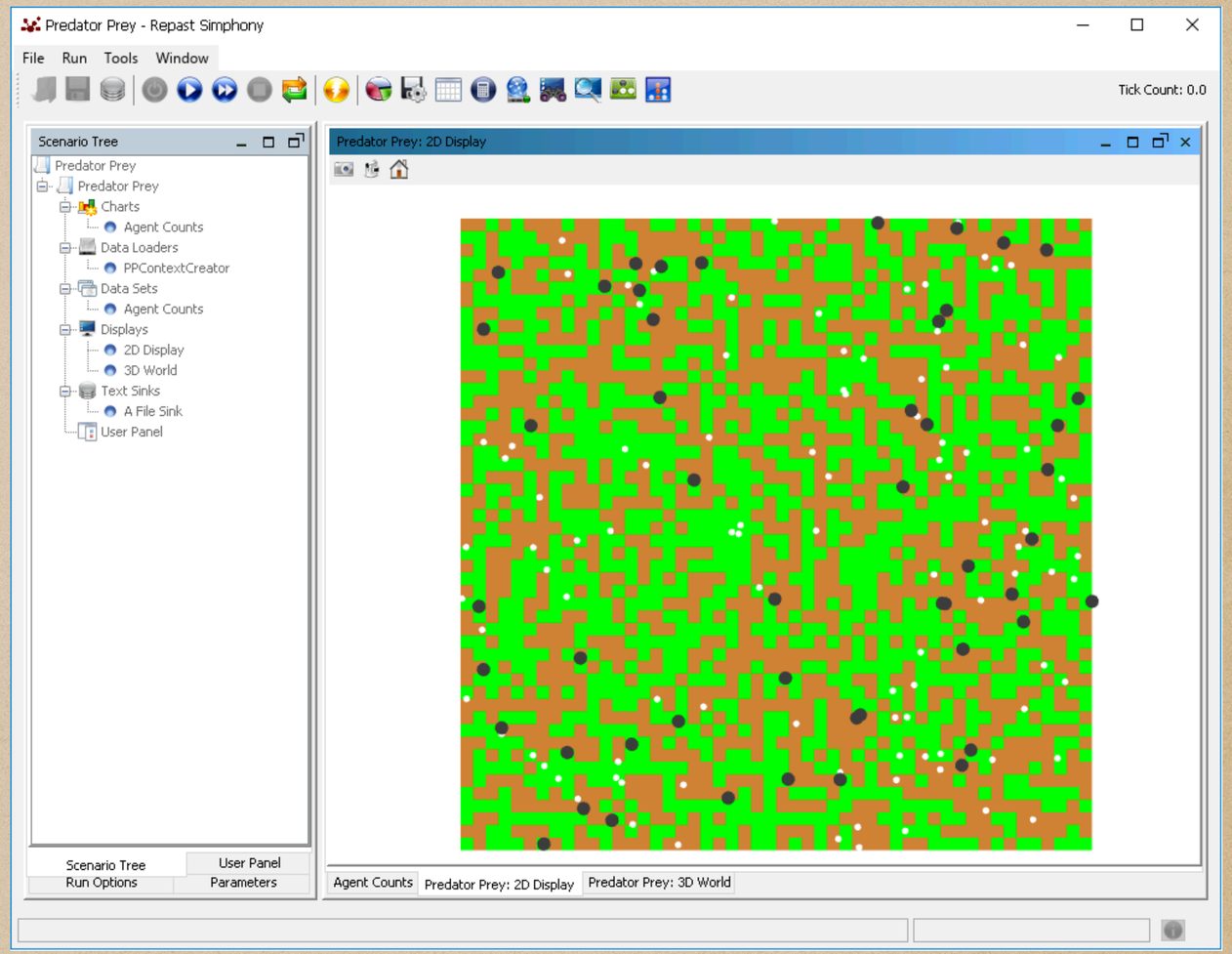
http://ccl.northwestern.edu/netlogo/



Repast

- Repast Simphony (Java)
  - Easy to learn
- Repast for High Performance Computing (C++)
  - Lean and targeted at advanced users
  - Millions of agents and cells
- 2 D and 3 D grids
- Repast statecharts - easy to see agent state
- Argonne National Labs
- ~21 years of development

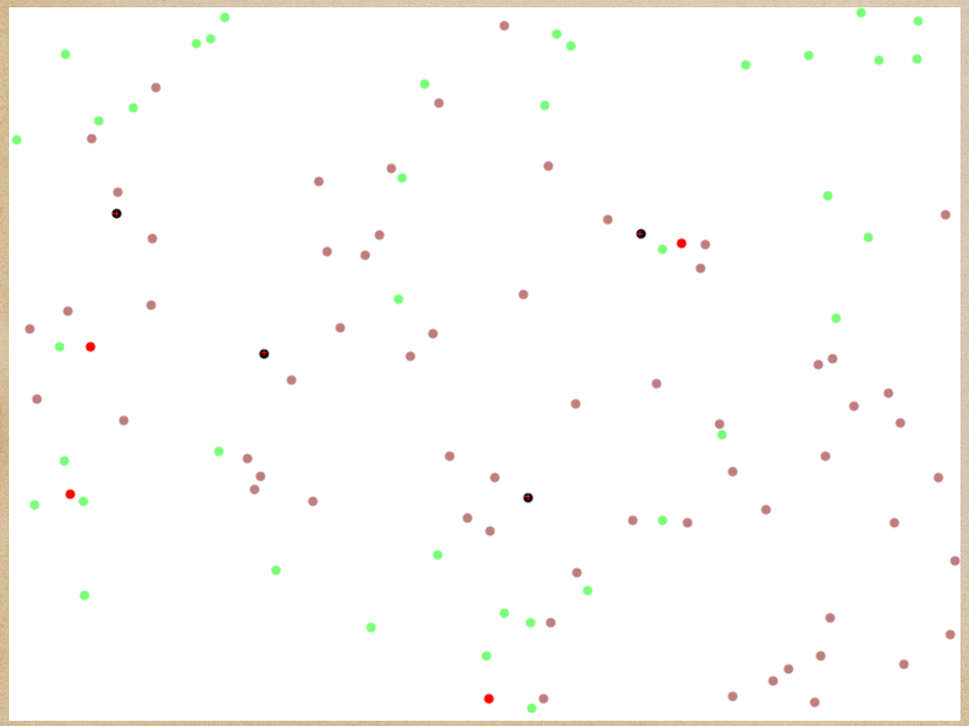
https://repast.github.io/

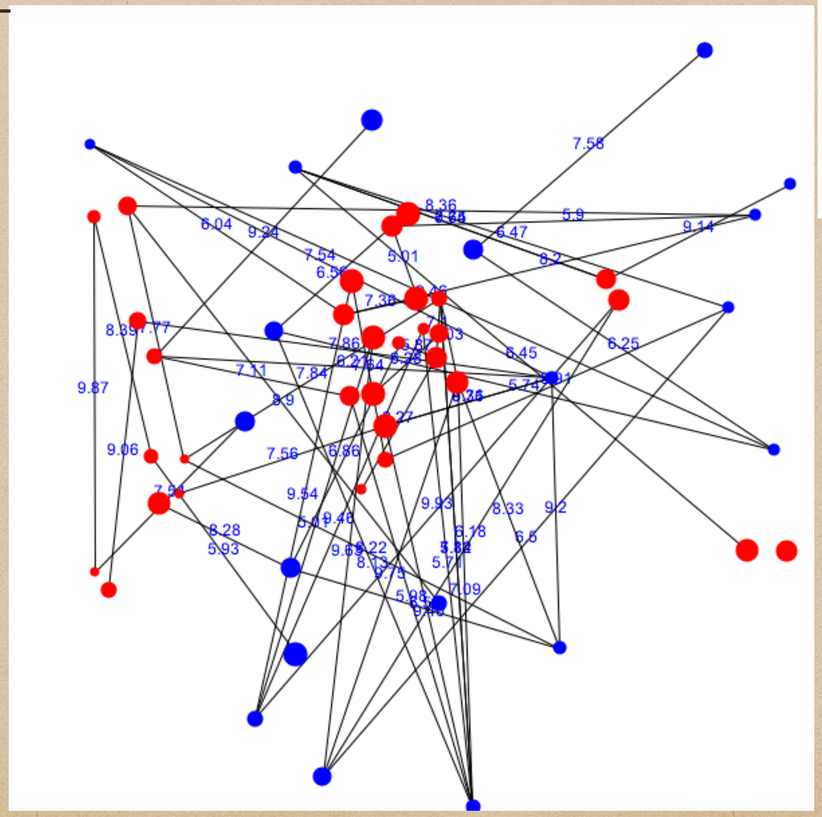


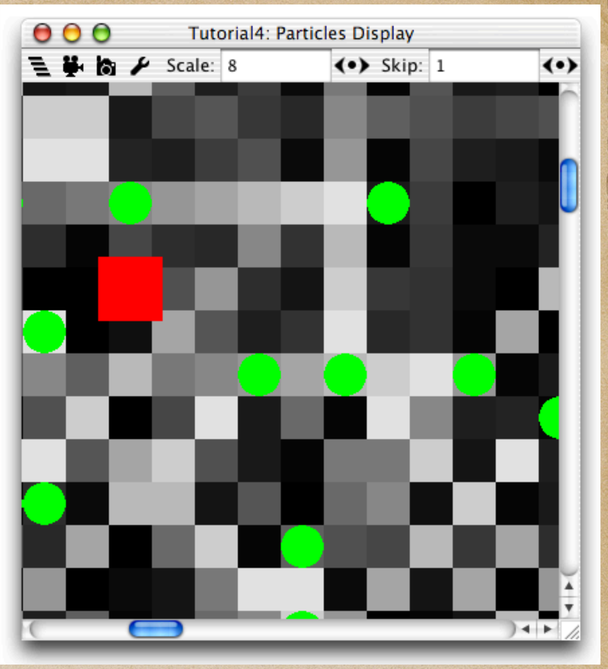
MASON

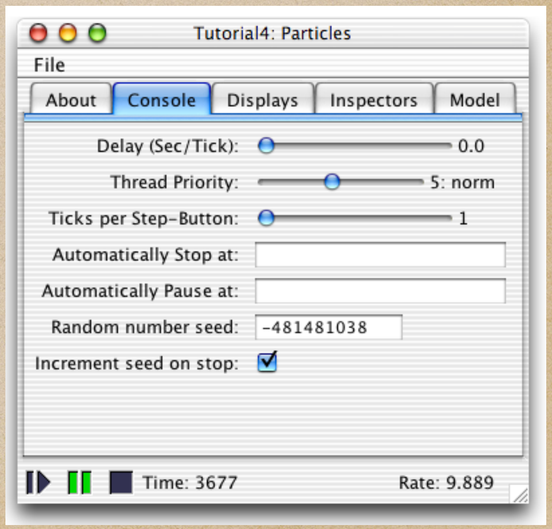
- Fast, portable and fairly small
- Checkpointing
- Continuous, grid (hexagonal, 2 D and 3D square), and network spaces
- George Mason University

https://cs.gmu.edu/~eclab/projects/mason/



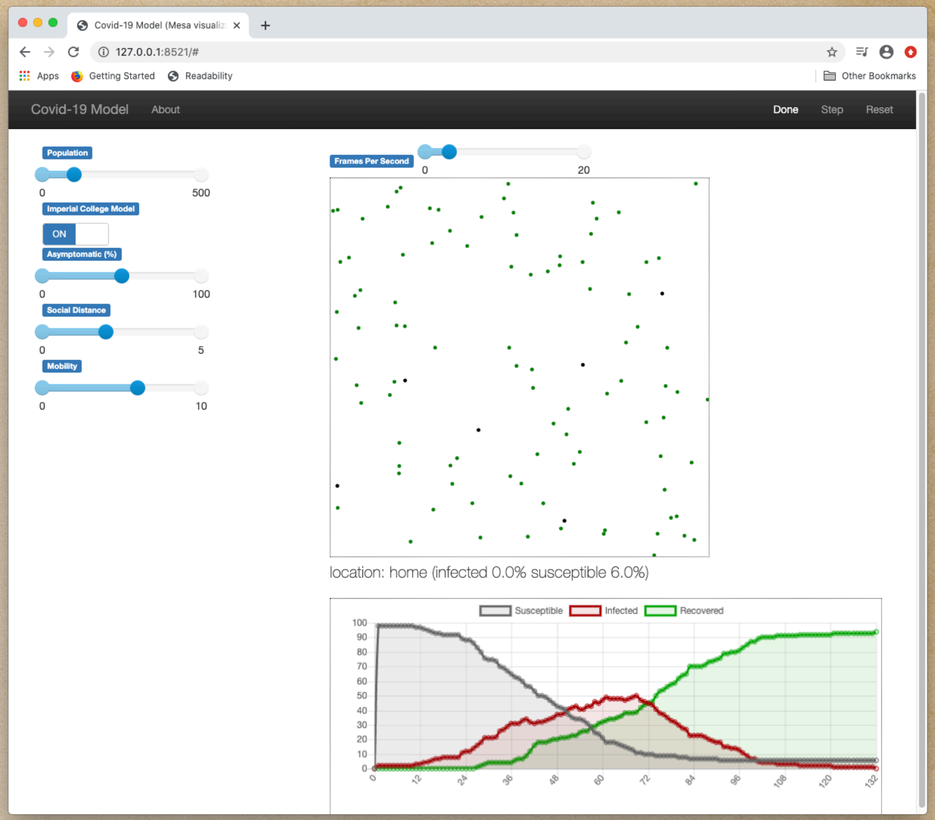


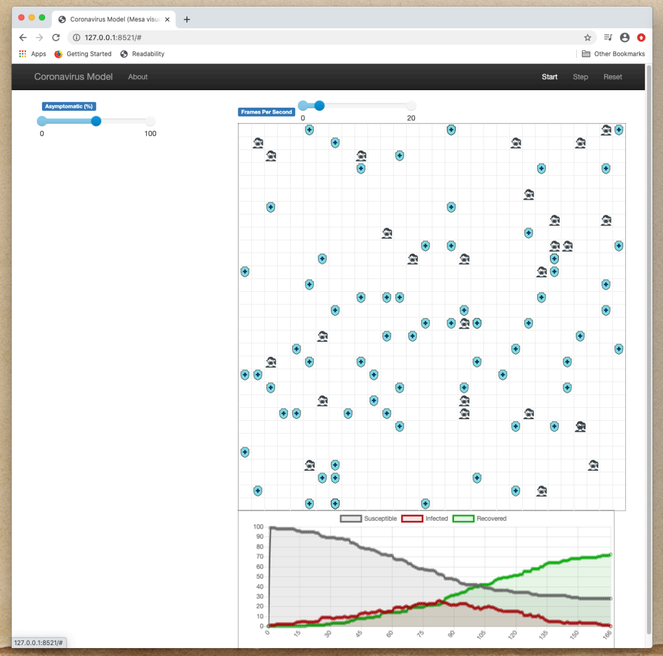


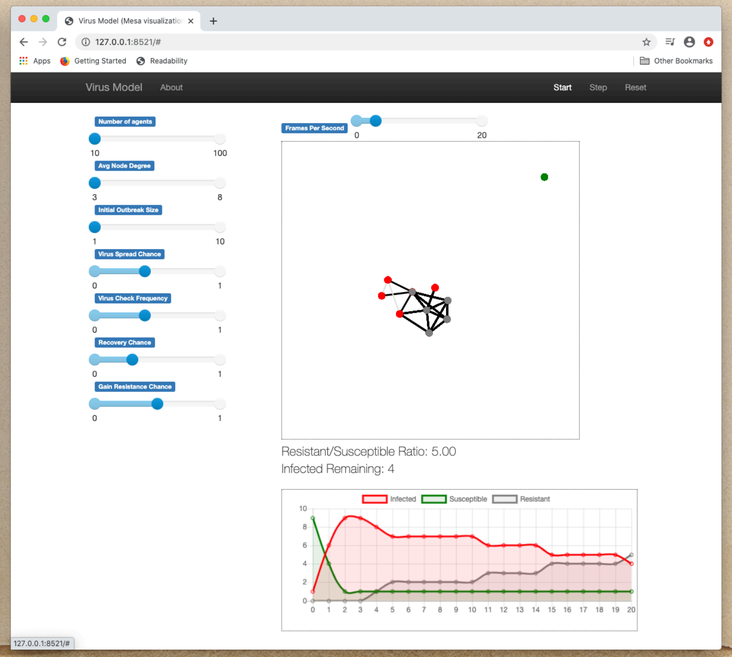


Python Mesa

- Based on Python 3
- General purpose mainstream computer language
- Integration of agent-based models with other software (eg. AI)
- Grid, continuous space and network spaces
- Browser-based visualization
- Newer software framework
- Modular (modeling, analysis and visualization)





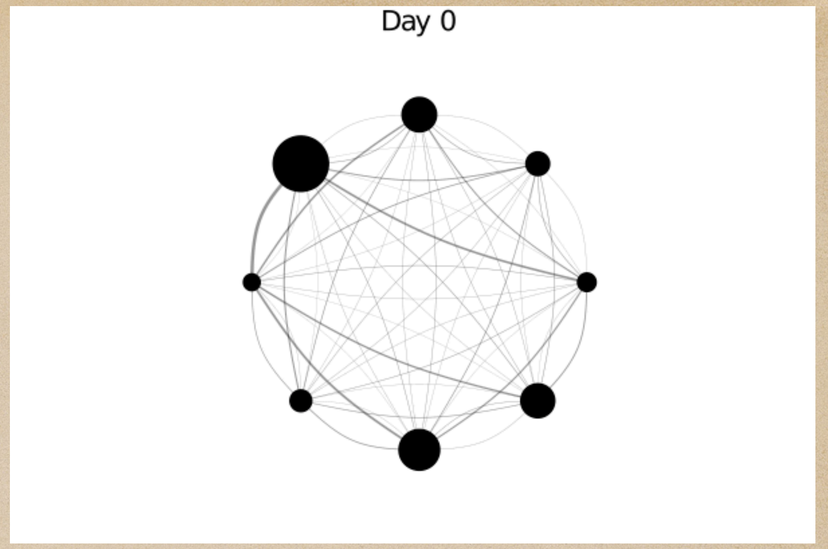


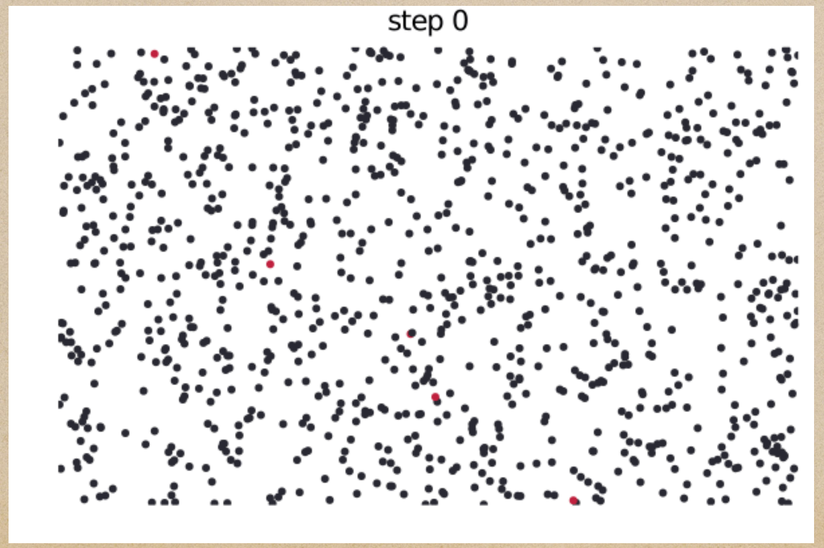
https://mesa.readthedocs.io/en/master/

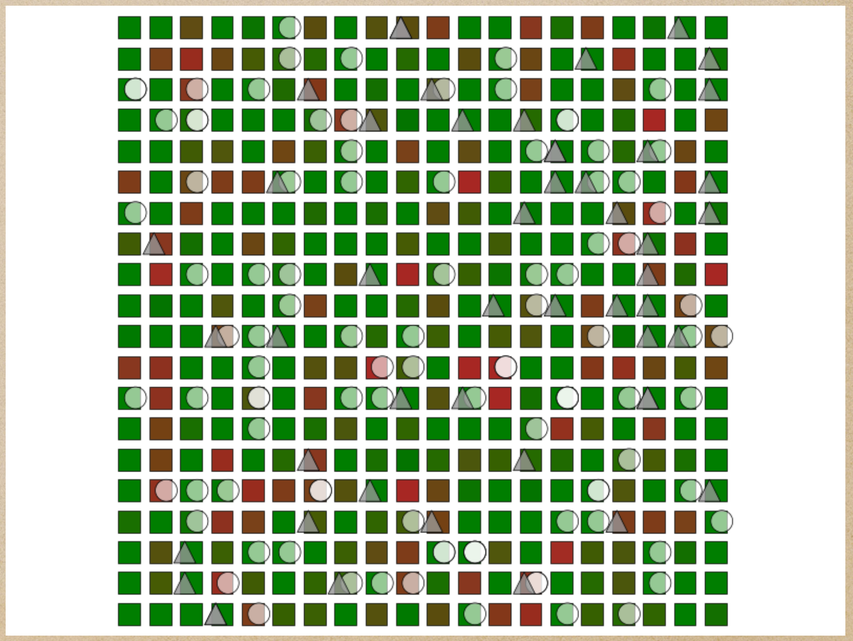
Agents.jl

- New (Julia - 2012, Agents.jl - 2019?)
- General purpose programming language (Julia)
- Easier to use than Java based ABMs (Repast and MASON)
- Claims to be 10 times faster than Mesa
- Grid, continuous and network spaces

https://juliadynamics.github.io/Agents.jl/stable/
https://julialang.org/





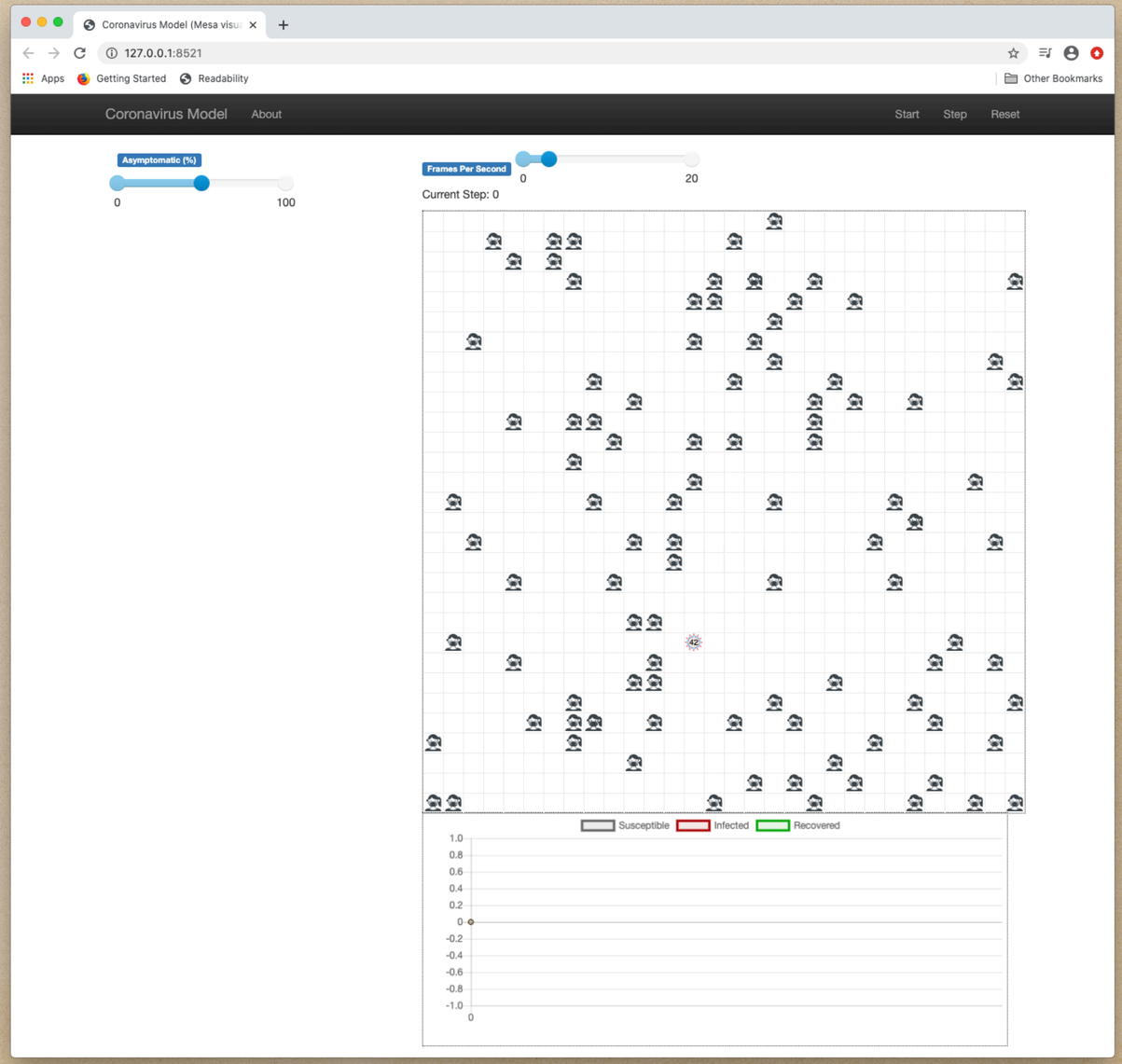


Examples of Types of Spaces

Grid (https://github.com/trobey/coronavirus)

In the top coronavirus directory

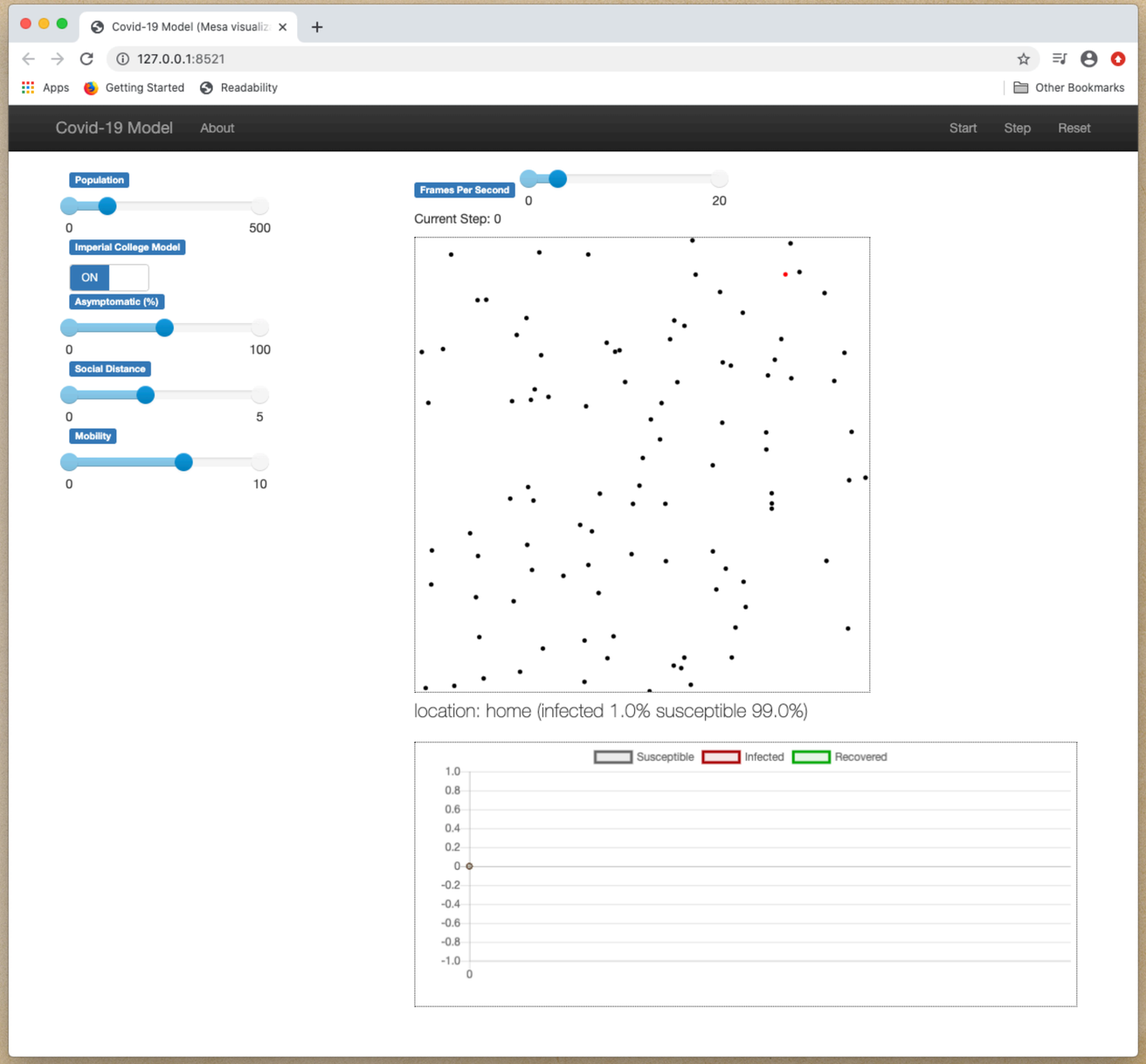
- pip install -r requirements.txt
- mesa runserver



Continuous (https://github.com/trobey/covid)

In the top covid directory

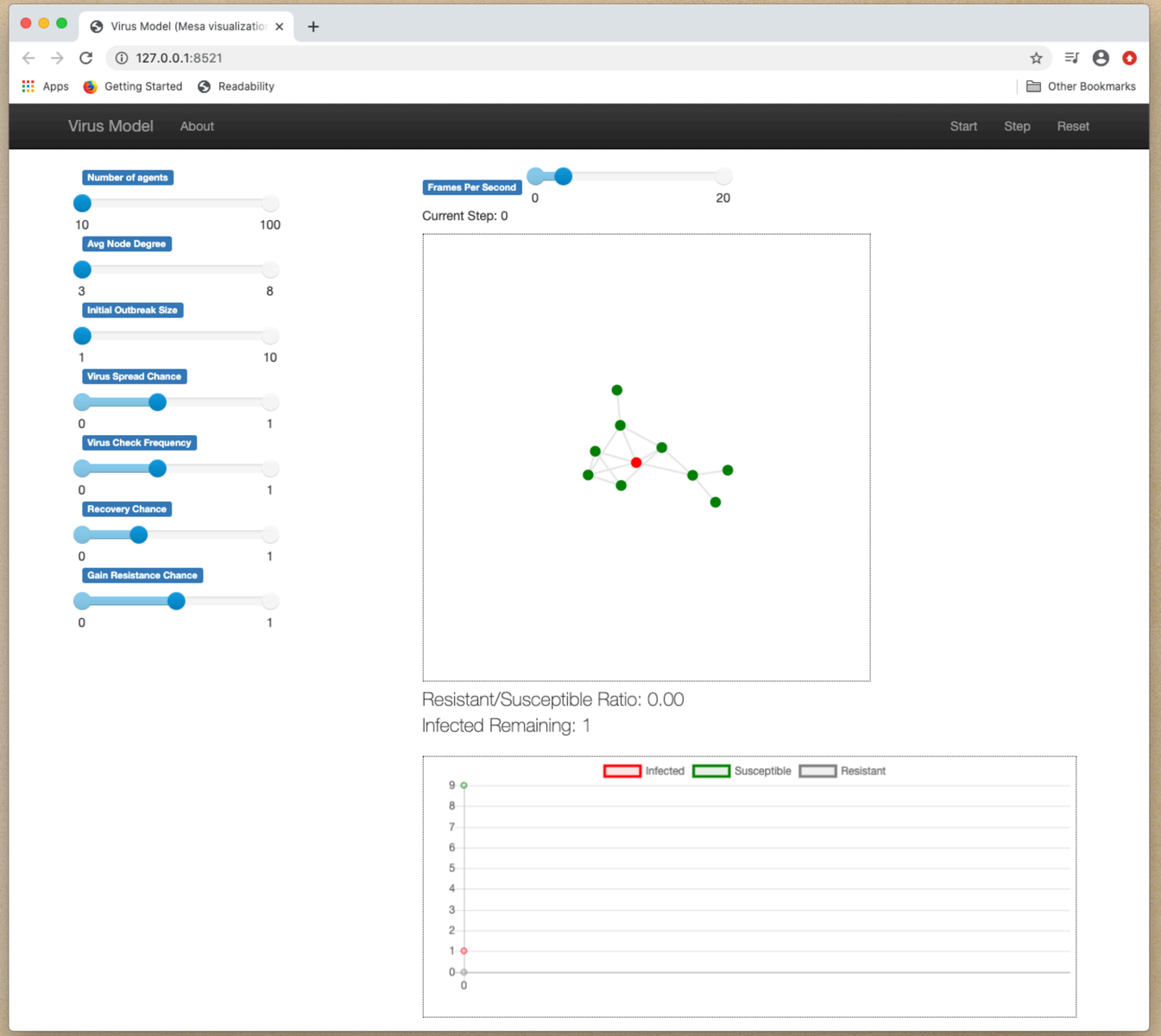
- pip install -r requirements.txt
- mesa runserver



Network (examples/virus_on_network)
https://github.com/projectmesa/mesa

In the top virus_on_network directory

- pip install -r requirements.txt
- mesa runserver



Python Mesa - Boltzmann Wealth Model

https://mesa.readthedocs.io/en/stable/tutorials/intro_tutorial.html

Log into your Google account and then open the link above. There is a link on the webpage to open this in Google Colab.

Using classes is particularly suited for agent based modeling. A ***`class`*** is a template for creating an ***`object`***. So a class is like a recipe and the objects are the meals that are created by the recipe.

```
agent = Agent(unique_id, model)
```

An agent is an object created from the class Agent. An agent has ***`properties`*** and ***`methods`*** or actions. The properties of an agent are its unique ID and the model containing the agent. The reason to include the model as a property of an agent is that it makes it easier to access the properties and methods of the model. Discussion of methods will be deferred for a bit.

Classes can inherit properties and methods. So a class Secret is a particular type of agent that has all the properties and methods of an agent but also can define properties and methods that only belong to Secret.

```
from mesa import Agent

class Secret(Agent):
    …
```

When an object is created from a class, the properties of the object need to be initialized. There is a special function __init__(). The first argument is a pointer to itself that by convention is "self." There can be additional arguments. Optional arguments can provide a default using an equals sign.

```
class Agent():
    def __init__(self, unique_id, model):
        self.unique_id = unique_id
        self.model = model
```

For classes that inherit from Agent, they will need to also call the Agent __init__(). This is done using super().

```
class Secret(Agent):
    def __init__(self, unique_id, model, pos):
        super().__init__(unique_id, model)
        self.name = "Secret"
        self.color = "Green"
        self.pos = np.array(pos)
```

Every agent should have at least one method. This is the method step(). For every step of the model the scheduler calls this function for every agent in the model.

```
def step(self):
    angle = 360.0 * self.model.random.random();
    x = math.sin(angle);
    y = math.cos(angle)
    new_pos = self.pos + np.array((x, y)) * self.model.random.random() *
self.model.mobility

    self.model.space.move_agent(self, new_pos)
```

Note that mobility is the maximum distance an agent can go in a step and it is a property of the model.

There is some additional code that makes it easier to debug a model. It helps when an agent is printed out that all their properties are printed. So a function is added to the class to convert the object to a string in the way we want.

```
def __str__(self):
    return str(self.__class__) + ": " + str(self.__dict__)
```

Models

Models have to

- Initialize themselves
- Initialize all of the agents
- Have the model take a step
- Have the agents perform a step

Other things that can be done in a model

- Collect data
- Count the numbers of each type of agent

Appendix I: Installing Python 3

It is easy to find instructions by searching on the web. Some are more complicated than necessary but if you get stuck just look around for instructions. Usually typing python gets the default version and python3 (or pip3) runs just the Python 3 version in case Python 2 is also installed. Python 2 is no longer supported. Note that the long dash — is typed as two short dashes.

If an Integrated Development Environment (IDE) is desired, then PyDev is a plugin for Eclipse.

Windows

- Go to https://www.python.org/downloads/windows. Find the 32 bit or 64 bit executable installer for Python 3 and download it. There is a bug in Python 3.8.
- Run the installer. Make sure to check to add Python to the path. You may get a message to disable the path length limit. Select this if it appears.
- Type python —version to verify install.
- Launch from a terminal by typing python.

PIP is included with Python 3.4+. Check by typing pip —version.

Linux

Python comes already installed on Linux. Although Python 2 is no longer supported check by typing

- python —version
- python3 —version

If the default version is Python 2 then you may need to use python3 and pip3 instead of just python and pip.

Mac

Python is installed on a Mac but this is for the system to use. Leave this alone to avoid breaking things. Homebrew will be used. It is a package manager for Macs. First, Xcode will need to be installed. It can be found in the app store. It is a large download. Once Xcode is installed, then open up a terminal

- /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"

Then install Python

- brew install python3

Check to verify installation

- python —version

Appendix II: Installing Python Mesa

- pip install mesa

Python also has requirements.txt files so if an example lists mesa in this file then it will automatically be installed using

- pip install -r requirements.txt

The two coronavirus examples have mesa listed so it will be automatically installed.