

# Open Project Management

from an “open” perspective

## UNIT 5

Instructor: Dr. Bradly Alicea

<http://bradly-alicea.weebly.com>



Lecture 18

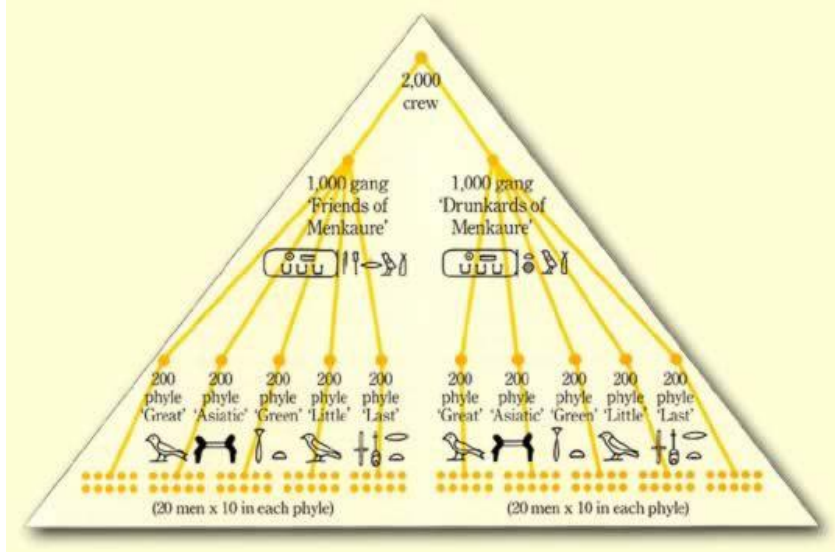
All content



**Welcome!**

# Project Lifecycle and Sustainability





## Egyptian Pyramid workforce organization

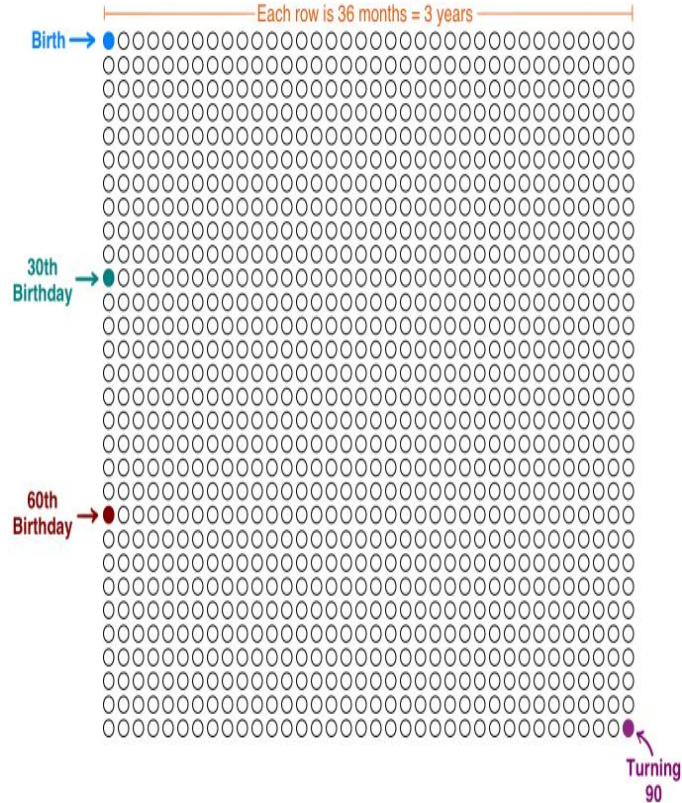
<http://www.toureygypt.net/featurestories/pyramidworkforce.htm>



Amazonian Geoglyphs,  
Stonehenge (England)



# A 90-Year Human Life in Months



<https://waitbutwhy.com/2014/05/life-weeks.html>

# Hype Cycle for Open-Source Software, 2020

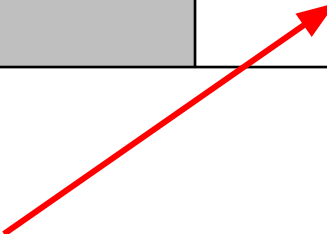


Plateau will be reached:

- less than 2 years
- 2 to 5 years
- 5 to 10 years
- ▲ more than 10 years
- ⊗ obsolete before plateau

## Four Modes of Result from Work

	FAILURE	SUCCESS
OBVIOUS	OBVIOUS FAILURE	OBVIOUS SUCCESS
NON-OBVIOUS	RECOUP FAIL	AMBIGUOUS SUCCESS



No such thing as “abject failure”. Even obviously bad (or null) results are to be learned from.

# Stewardship of open resources

Open-source communities are an opportunity to maintain the software commons

- many necessary components to making software platforms both *relevant* and *usable*.
- contributors can also help to keep platforms relevant over time (as technology and social needs change).
- what does sustainable effort entail and look like in terms of effort?



**Maintaining (not planning)  
projects over the long-term**

# Open-source Project Maintenance

What happens if our open source project falls out of sync with the current state of the art (due to lack of contributors)?



<https://www.fastruby.io/blog/open-source/guide-for-unmaintained-open-source-projects.html>



<https://logging.apache.org/log4j/2.x/security.html>



## DevoLearn

Accelerate research in developmental biology with deep learning models

4 followers | Worldwide | <http://devoworm.weebly.com>



Release schedule is out of sync with Github repository, version of Python is out of sync with project code.

We discussed this in Unit 2.

### Release history

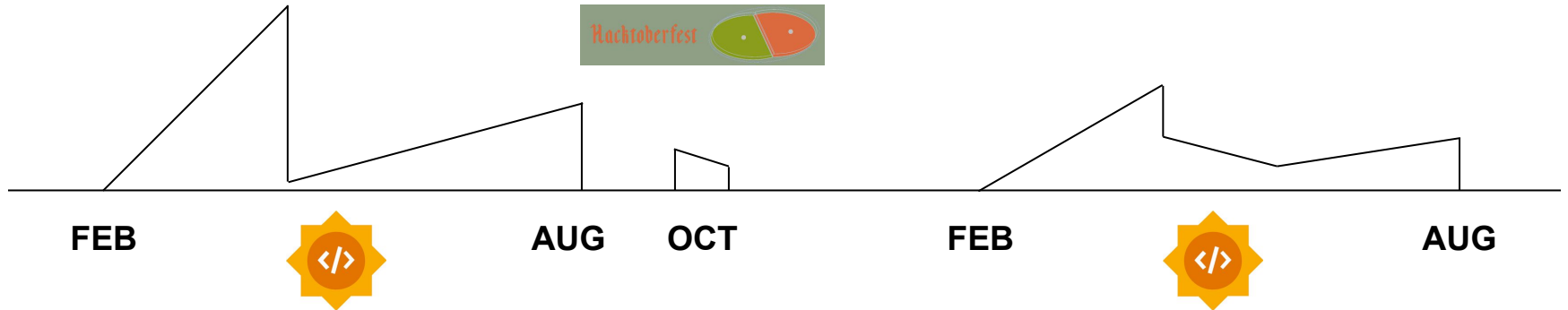
[Release notifications](#) | [RSS feed](#)

<b>THIS VERSION</b>	<b>0.3.0</b> Feb 22, 2021
	<b>0.2.3</b> Feb 6, 2021
	<b>0.2.2</b> Feb 6, 2021
	<b>0.2.1</b> Jan 14, 2021
	<b>0.2.0</b> Aug 19, 2020
	<b>0.1.9</b> Aug 6, 2020
	<b>0.1.8</b> Aug 6, 2020
	<b>0.1.7</b> Aug 5, 2020
	<b>0.1.6</b> Aug 5, 2020

<https://pypistats.org/packages/devolearn>

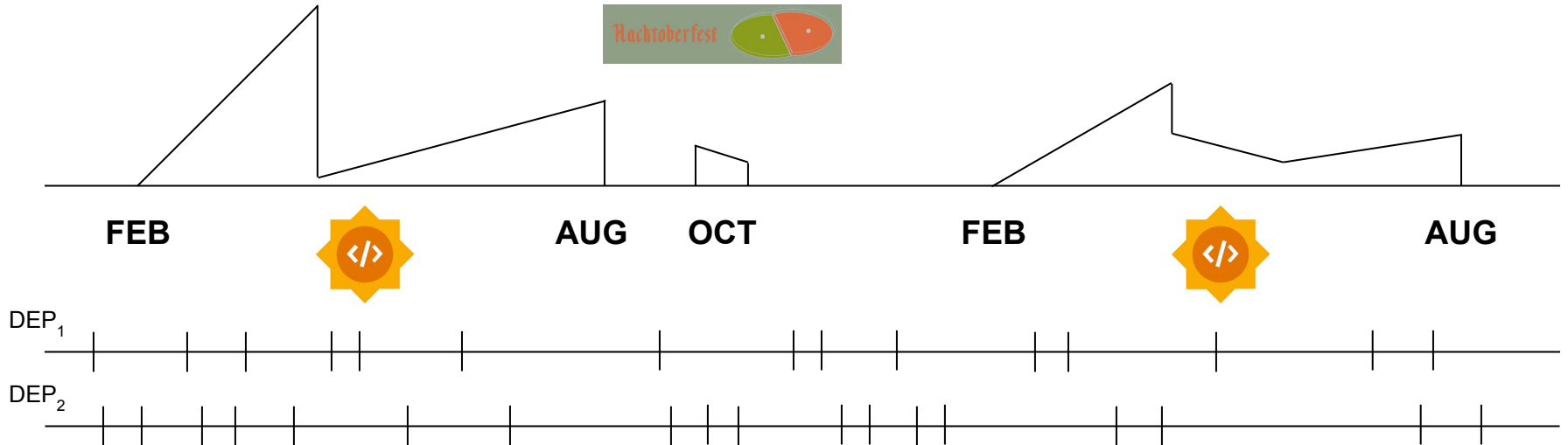
# Seasonal Updates and Maintenance

Uneven contribution history over the course of 18 months.

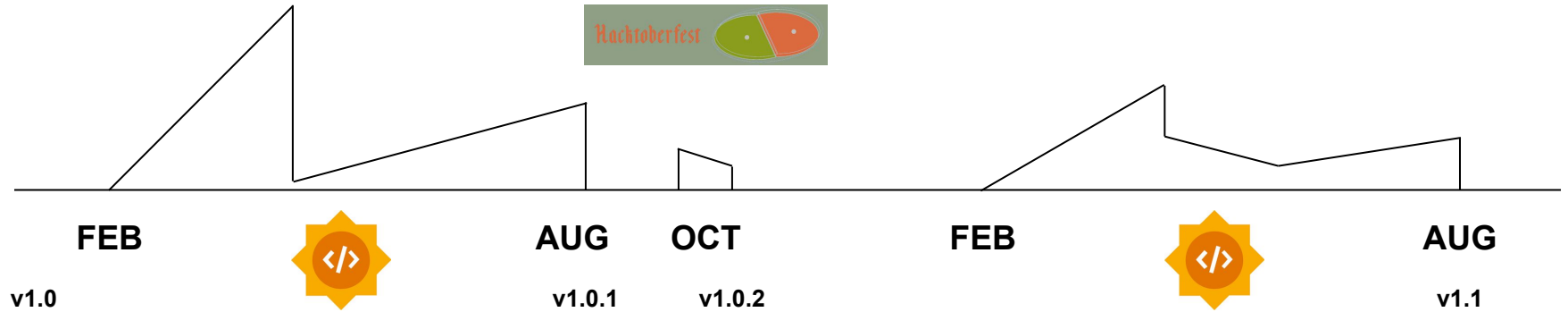


# Seasonal Updates and Maintenance

Uneven contribution history over the course of 18 months. Dependencies are updated at variable rates.



# Season Example of Releases



Search

All packages

Top packages

Track packages

PyPI page

Home page

Author: Mayukh Deb, Ujjwal Singh, Brady Alicea

License:

Summary: Accelerate data driven research in developmental biology with deep learning models

Latest version: 0.3.0

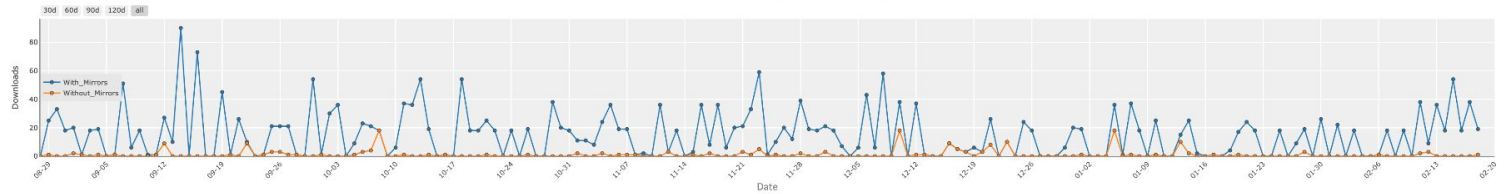
Required dependencies: cyclical | decorator | efficientnet-pytorch | future | imageio | imgaug | imutils | joblib | kielisolver | matplotlib | munch | networkx | numpy | opencv-python | pandas | pillow | pretrainedmodels | pyarsing | python-dateutil | pytz | pywavelets | scikit-image | scikit-learn | scipy | segmentation-models-pytorch | six | sklearn | threadpoolctl | tifffile | timm | torch | torchvision | tqdm | typing-extensions | wget

Downloads last day: 0

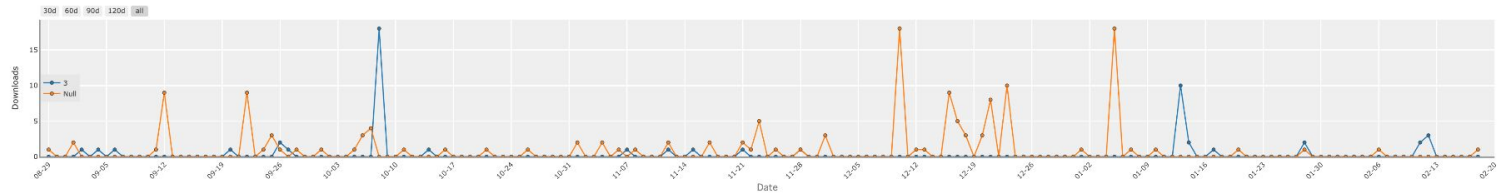
Downloads last week: 1

Downloads last month: 10

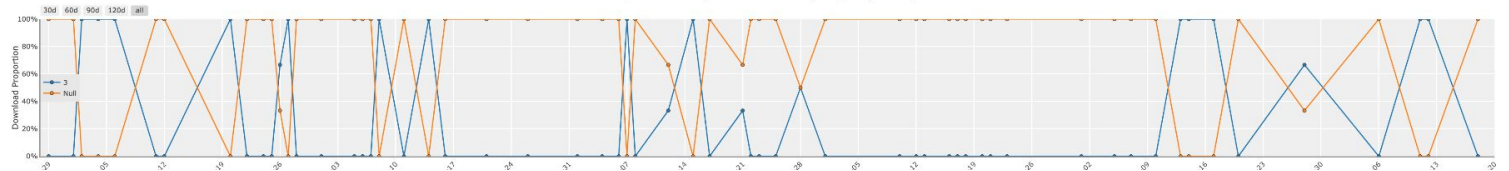
Daily Download Quantity of devolearn package - Overall



Daily Download Quantity of devolearn package - Python Major



Daily Download Proportions of devolearn package - Python Major



API  
About  
FAQs

<https://pypistats.org/packages/devolearn>

# Best Practices for Maintainers

Making your life easier as an open source maintainer, from documenting processes to leveraging your community.

Table of Contents ▾



<https://opensource.guide/best-practices/>



## **Documenting process (develop community standards)**

- getting it out of your head.

## **Communicate expectations (avoid wasted work)**

- keep this public.

## **Learning how to say no (avoiding burnout)**

- enforce rules, maintain project scope, maintain personal boundaries.

## **Leverage the power of community (avoid overreliance on key people)**

- share the workload and fractional participation.

## **Best Practices for Maintainers**

Making your life easier as an open source maintainer, from documenting processes to leveraging your community.

# OpenWorm Foundation

<http://openworm.org>



## Four types of new contributor

- 1) walk-up inquiries: get excited by online content or related talk, fill out contributor form, contribute to various extents.
- 2) formal internship participants: make a formal proposal, follow through.
- 3) get involved through class projects.
- 4) more formal academic connections: part of an OpenWorm-related connection (Senior Contributor's student, PhD-level scientist).

## Dealing with Open Source Conflict

<https://codeengineered.com/blog/2018/open-source-conflict/>

### **Sometimes conflict can be good.**

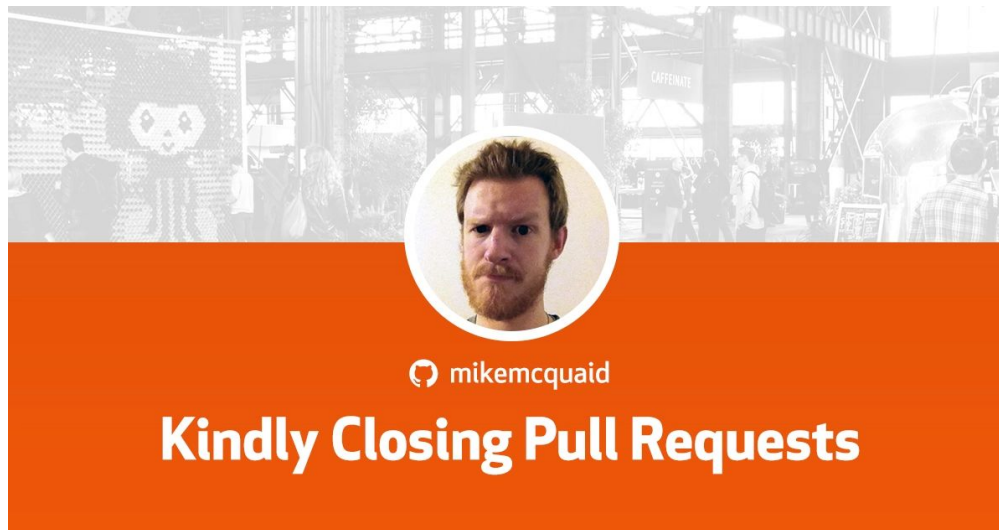
- argumentation and adversarial design (more engaging meetings, more creative outcomes).
- active listening and be understanding of other people's points of view.

## Dealing with Open Source Conflict

<https://codeengineered.com/blog/2018/open-source-conflict/>

Sometimes conflict can be good.

- argumentation and adversarial design (more engaging meetings, more creative outcomes).
- active listening and be understanding of other people's points of view.
- reference code of conduct and community standards.
- focus on the ideas rather than the personalities.



<https://github.blog/2016-03-15-kindly-closing-pull-requests/>

What to do if people submit subpar or out of scope work?

How do you keep contributor workflows moving?

# Effects and Antecedents of Conflict

<https://dl.acm.org/doi/10.1145/2818048.2820018>

RESEARCH-ARTICLE



## The Effects and Antecedents of Conflict in Free and Open Source Software Development

**Authors:**  [Anna Filippova](#),  [Hichang Cho](#) [Authors Info & Claims](#)

CSCW '16: Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing • February 2016 • Pages 705–716 • <https://doi.org/10.1145/2818048.2820018>

Empirical study of FOSS organizations: conflict has varying effects on the long-term viability of the organization, mostly negative.

**Projects as a series of goals over many  
years and contributors**



Jen Kagan

Jan 3, 2017 · 12 min read ·  Listen

## What Does “Open Source” Even Mean?

p5.js edition

When I really didn't know what I was doing, it was just: do something that is more than nothing. And then do something more. And it just kept growing.

<https://medium.com/@kenjagan/what-does-open-source-even-mean-p5-js-edition-98c02d354b39>



# Open Mentorship Model

Inspired by Distributed Open Source project development

Small contributions can move along larger contributions and/or projects. Participant (mentee) makes as many contributions as they would like.

# Open Mentorship Model

Inspired by Distributed Open Source project development

Small contributions can move along larger contributions and/or projects. Participant (mentee) makes as many contributions as they would like.

- short talks, toy projects, skill-building tasks.
- integrate these activities into larger project.

# Open Mentorship Model

Inspired by Distributed Open Source project development

Small contributions can move along larger contributions and/or projects. Participant (mentee) makes as many contributions as they would like.

- short talks, toy projects, skill-building tasks.
- integrate these activities into larger project.
- important to assign when interaction is difficult (beginning of participation, lull in the action, point of potential failure).
- mentor assigns the broad outlines of task, mentee defines the details of the task.

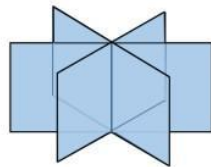


Ziyi Gong



Stefan  
Dvoretzkii

2019 Google Summer of Code



## Developmental Braitenberg Vehicles for Neurosimulation

Preprint! *arXiv*, 2003.07689  
Publication! *Artificial Life*, 28(3), 369-395.



Jesse Parent



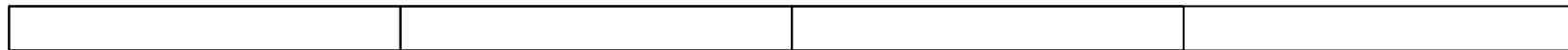
Ankit Gupta

Develop a problem for  
people to address

Present and foster further  
development around preprint

Lead a study group: open  
software and preprint

Published version of work,  
recruit open-source maintainers



**2019**

**2020**

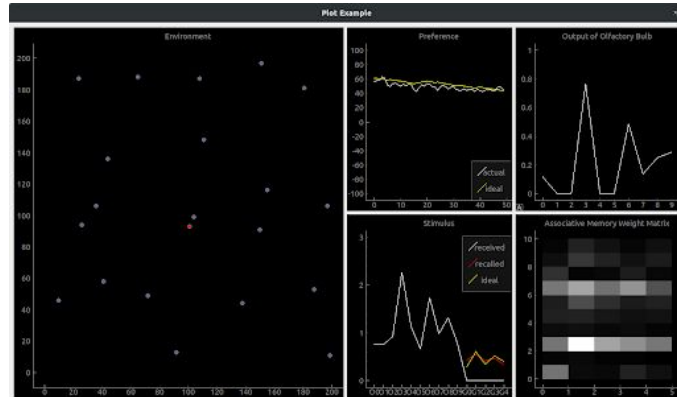
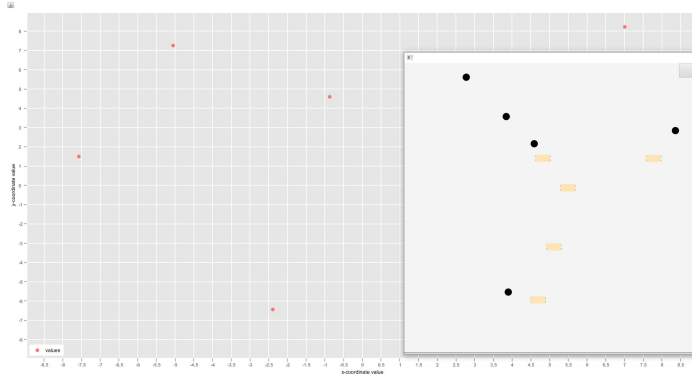
**2021**

**2022**

**2023**

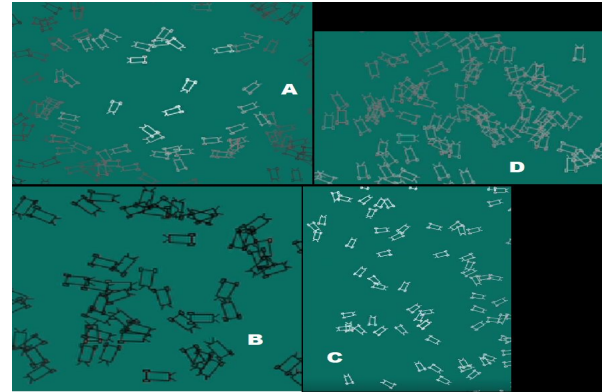
# Additional Open-source Initiatives

## BraGenBrain



## MultiLearn

## BV Collectives





**Present and foster further development around preprint**



Discussion about Log4J  
and open-source

Funded projects from  
adversarial approaches

Preprint on adversarial  
project development

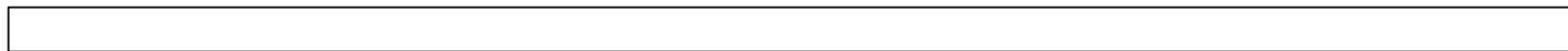
Develop a problem for  
people to address

Seek funding for  
maintainer projects

**2021**

**2022**

**2023**

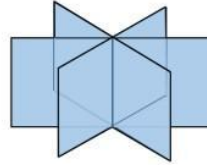






**Hussain Ather**

## 2022 Google Summer of Code



**Brian McCorkle**



**Himanshu Chougule**

### **Preprint!**

[https://www.researchgate.net/publication/369143414\\_Open-source\\_Community\\_Sustainability\\_using\\_Agent-based\\_Models](https://www.researchgate.net/publication/369143414_Open-source_Community_Sustainability_using_Agent-based_Models)

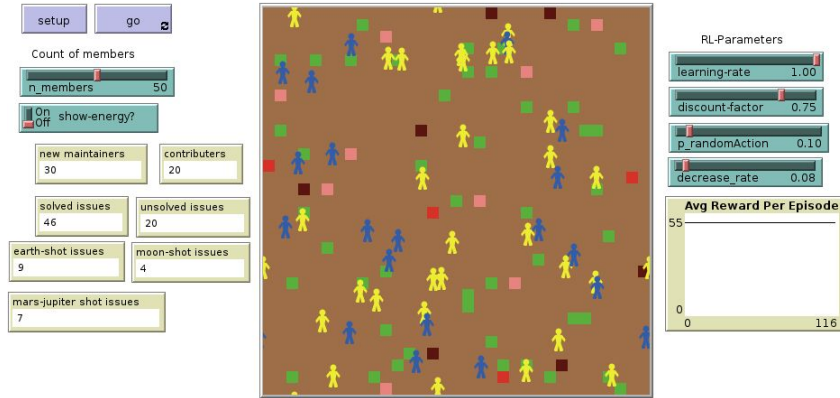
### **Open-source Software!**

<https://github.com/OREL-group/GSoC/tree/main/Open%20Source%20Ethics>

### **Auditing Tool!**

<https://sustainabilityauditingtool.herokuapp.com/>

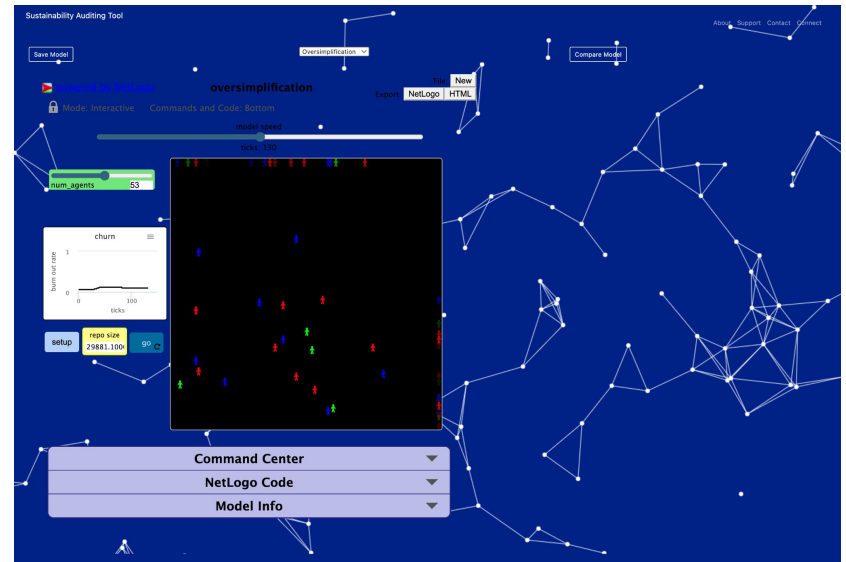
# Funded Projects From Adversarial Approaches



Reinforcement Learning for Open Source Community Sustainability (Open-source Software)

## Sustainability Auditing Tool

<https://sustainabilityauditingtool.herokuapp.com/>



# Development Cycles for Open-source

When is open source helpful in achieving development goals?

When is open source a hindrance in achieving development goals?

# Development Cycles for Open-source

When is open source helpful in achieving development goals?

When is open source a hindrance in achieving development goals?

There are several scenarios where open source can potentially be a hindrance with respect to your development cycle.

Development is too rapid for open source affordances (docs, Wiki, general understanding) to keep up.

Missing transparency (issues not clear, information not shared) breaks link between developers/team and contributors.

Openness clashes with privacy and confidentiality. Social mores change over time!