# WOFOST-Gym

DISCLAIMER: This repository is still actively under development. Please email Anonymous Authors ([anon@anon.edu](mailto:anon@anon.edu)) with use-case questions and bug reports.

## Description

This package provides the following main features:

1. A crop simulation environment based off of the WOFOST8 crop simulation model which simulates the growth of various crops in N/P/K and water limited conditions. The model has been modified to interface nicely with a Gymnasium environment and to provide support for perennial crops and multi-year simulations.
2. A Gymansium environment which contains the WOFOST8 crop simulation environment. This wrapper has support for various sized action spaces for the application of fertilizer and water, and various reward functions which can be specified by the user. We provide easy support for multi-year simulations across various farms.
3. We support the training of Deep RL agents with PPO, SAC, and DQN based on [cleanRL](https://github.com/vwxyzjn/cleanrl) implementations. We provide various visualizations for different state features and to visualize the training of RL agents.
4. The generation of historical data across years, farms and crops to support offline RL and off-policy evaluation methods.

Our aim with this project is to researchers, by enabling easy evaluation of decision making systems in the agriculture environment.

## Getting Started

### Dependencies

* This project is entirely self contained and built to run with Python 3.10.9
* Install using miniconda3
* NOTE: This repository contains git submodules that must be installed using --recurse-submodules (see below)

### Installing

Recommended Installation Method:

1. Navigate to desired installation directory
2. git clone [git@github.com](mailto:git@github.com):wofostgym/wofostgym.git
3. conda create -n python=3.10.9
4. conda activate
5. pip install -e pcse
6. pip install -e pcse\_gym
7. pip install -r requirements.txt

For Gail/BC experiments: 8. pip install -e imitation 9. pip install -e stable-baselines3

These commands will install all the required packages into the conda environment needed to run all scripts in the wofostgym package

## Executing Programs

After following the above installation instructions:

1. Navigate to the base directory ../wofost-gym/
2. Run the testing domain with: python3 test\_wofost.py --save-folder logs/test/. This will generate a sample output using default configurations
3. This may take a few seconds initially to configure the weather directory

### Use Cases:

* To generate data for Offline RL, Off-Policy Evaluation Problems, or Transfer Learning Problems:
  1. To generate data with a specified policy from pcse\_gym/policies.py:
     + python3 gen\_data.py --save-folder <Location> --data-file <Filename> --policy-name <Name of Policy>
  2. To generate data with a trained RL Agent based policy:
     + python3 gen\_data.py --save-folder <Location> --data-file <Filename> --agent-type <PPO | DQN | etc> --agent-path <Location/agent\_name.pt>
  3. NOTE: Use --load-config-fpath to load an environment configuration from a config.yaml file. If `None`, the default configuration will be used.
* To train an RL Agent:
  1. python3 train\_agent.py --save-folder <Location> --agent-type <PPO | DQN | etc>
  2. Use --<Agent\_Type: PPO|DQN|etc>.<Agent\_Specific\_Args> to specify algorithm specific arguments
  3. To track using Weights and Biases add --<Agent\_Type: PPO|DQN|etc>.track
  4. NOTE: Use --load-config-fpath to load an environment configuration from a config.yaml file. If `None`, the default configuration will be used. This works for loading previous Agent Configurations as well

## Help

Initial configuration for the Gym Environment parameters (Note: NOT the actual crop simulation) can be modified in the utils.py file.

* The default path is the current working directory. All other paths are relative based on this working directory.
* The env\_config/ folder contains all default .yaml files for configuring the Crop Simulator. For further information, please see the following READMEs:
  + env\_config/README\_agro.md - overview of how to configure a crop simulation.
  + env\_config/site\_config/README\_add\_site.md - overview of how to add a new site with all required parameters.
  + env\_config/site\_config/README\_site\_paramters.md - an overview of all configurable site parameters
  + env\_config/site\_config/README\_site\_states.md - an overview of all site state and rate variables available for output with corresponding units.
  + env\_config/crop\_config/README\_add\_crop.md - overview of how to add a new crop with all required parameters.
  + env\_config/crop\_config/README\_crop\_paramters.md - an overview of all configurable crop parameters
  + env\_config/crop\_config/README\_crop\_states.md - an overview of all crop state and rate variables available for output with corresponding units.
  + pcse/README.md - an overview of the Python Crop Simulation Environment (PCSE) and available resources to learn more.
  + rl\_algs/README.md - an overview of the available Reinforcement Learning agents available
  + pcse\_gym/README.md - an overview of the Gymnasium wrapper and available configurations

Email [anon@anon.edu](mailto:anon@anon.edu) with any further questions

## Authors

Anonymous Authors ([anon@anon.edu](mailto:anon@anon.edu)) - Principle Developers

## Version History

* 1.0.0
  + Initial Release

## License

This project is licensed under the MIT License - see the LICENSE.md file for details

## Acknowledgments

The Original PCSE codebase and WOFOST8 Crop Simulator can be found at:

* [PCSE](https://github.com/ajwdewit/pcse)

While we made substantial modifications to the PCSE codebase to suit our needs, a large portion of the working code in the PCSE directory is the property of Dr. Anonymous.

The original inspiration for a crop simulator gym environment came from the paper:

* [CropGym](https://arxiv.org/pdf/2104.04326)

We have since extended their work to interface with multiple Reinforcement Learning Agents, have added support for perennial fruit tree crops, grapes, multi-year simulations, and different sowing and harvesting actions.

The Python Crop Simulation Environment (PCSE) is well documented. Resources can be found here:

* [PCSE Docs](https://pcse.readthedocs.io/en/stable/)

The WOFOST crop simulator is also well documented, and we use the WOFOST8 model in our crop simulator. Documentation can be found here:

* [WOFOST Docs](https://wofost.readthedocs.io/en/latest/)