Conservation Strategy Simulator User Manual

1. Introduction

The *Conservation Strategy Simulator* runs Monte Carlo simulations of adversarial contests between conservationists ("Greens") and developers ("Farmers") over land plots with agricultural and environmental value. Users can replicate the results of *Weinhold & Andersen (2025)* or set different combinations of the parameter space, exploring how alternative conservation siting strategies (e.g., **Maximize Environment**, **Hot Spot**, **Block Farmers**) perform with different:

- Developer behavior
- Leakage
- Correlation between Agricultural and Environment values
- Asymmettrical political/economic power
- Grid size

Audiences:

- **Regular users (students, colleagues):** run experiments via popup menu.
- **Super users (researchers):** run from the command line, automate batches, or extend the code.

Outputs:

- **Overlay plots** (static vs leakage; dynamic trajectories).
- **Effects tables (CSV)** with Pure Strategy Effect (PSE), Displacement–Leakage Effect (DLE), and conservation scores.

---

2. Installation

Download *conservation_game_sim.py* and *requirements.txt* from:
https://github.com/dmweinhold/Conservation_Strategy_Simulation)

Requirements:

- Python 3.10+ (or python compiler (IDE) such as PyCharm)
- Libraries: `numpy`, `pandas`, `matplotlib`, `pyyaml`

Install with: `pip install -r requirements.txt`

---

3. Launching the Simulator

Run the script from terminal or an IDE:

```
python conservation_game_sim.py
```

- **Default**: opens a popup menu UI.
- **Super users**: disable the UI (`USE_UI_DEFAULT = False` in the script) and use CLI flags.

---

4. Popup Menu (Regular Users)

Each option corresponds to paper concepts:

- **Replications**: Monte Carlo runs (default =500).
- **Correlation (ρ)**: between environmental & agricultural values (–1 to +1, default=0).
- **Allocation**: Default=Equal (50/50) or Political (choose Farmer %).
- **Green strategies**: Select siting rules (defaults: Max Env, Hot Spot, Block Farmers).
- **Farmer strategies**: Naïve or Strategic.
- **Leakages**: e.g. `1.0,0.5,0.0`.
- **Grid size (n)**: number of rows/columns (max 30), Default=10.
- **Rounds**: `"auto"` = one claim per side per round; integer < max(claims) triggers batching.
- **Output folder**: where CSV and plots are saved.
- **Mode**: all / static / dynamic / effects.
- **Seed**: random seed for reproducibility.

---

5. Outputs

Saved automatically into `outputs/`:

- **Static overlay plots**: outcomes vs leakage (Conservation, Additionality, Welfare Loss).
- **Dynamic overlay plots**: time paths; leakage shown by lighter→darker shades.
- **Effects tables (CSV)**: Strategy × Leakage × Farmer type, with PSE, DLE, and final conservation scores.

---

6. Advanced Use (Super Users)

Run with CLI flags, e.g.:

```
python conservation_game_sim.py --mode all --farmer strategic \
  --multi-per-round on --risky-rule farmer_claims
```

Key flags:

- `--greens`, `--farmer`, `--alloc`, `--farmer_pct`
- `--rho`, `--grid`, `--rounds`, `--reps`
- `--leakages`, `--risky-rule`
- `--outdir`, `--seed`

---

7. Troubleshooting

- **Plots not showing** → check `matplotlib` is installed.
- **UI won't open** → Tkinter not supported in headless servers; use CLI mode.
- **Slow runs** → reduce replications, grid size, or strategies.

---

8. Citation

If you use this tool, please cite:

Weinhold, D. & Andersen, L. E. (2025).
Conservation Strategies in Contested Environments: Dynamic Monte Carlo Simulations and a Bolivian Case Study.
London School of Economics & Universidad Privada Boliviana. Working Paper.
DOI: 10.5281/zenodo.17114490