# C2SIM

Noah Syrkis

IT University of Copenhagen

Apr. 4, 2024

# 1 | Overview

The project[1] uses JAX[2] throughout, with JaxMARL's[3] SMAX as the main environment. The agents are modelled using behaviour trees (BT) stored in a sqlite3 database (we call it BTBank). The ollama[4] library is used for the language modelling to map game states to human language and BTs, and vice versa.

---

[1]https://github.com/syrkis/c2sim/
[2]https://github.com/google/jax/
[3]https://blog.foersterlab.com/jaxmarl/
[4]https://ollama.com/

# 1 | Overview (cont.)

- ☒ SMAX visual playback (`src/{plot,smax}.py`).

- ☒ BT function constructor (`src/{bt,atomics}.py`).

- ☒ BT based trajectory (`src/smax.py`). (almost done)

    - ▶ Must traverse all leafs always (for array programming)[5].

- ☐ JIT compile the BT traversal. (almost done)

- ☐ Implement the BTBank (`src/bank.py`).

- ☐ Language out (`src/llm.py`).

- ☐ Language in (`src/llm.py`).

- ☐ Smart way to generate atomics (gentic programming)?

[5]Has no effect on performance, as we are always as slow as slowest action

# 2 | SMAX

▶ Extensive work on visual playback of trajectory fig. 1.

  ⊠ Costum SMAX [1] vizualization.

  ⊠ Show unit type, team, health, attacks, and reward.

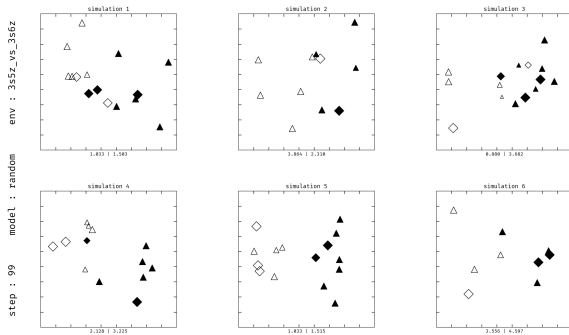  ⊠ Successfully runnning 10K+ parallel environments.

Figure 1: SMAX in parallel

# 2 | SMAX (cont.)

```
1  key = random.PRNGKey(0).split(num_envs)
2  env = make('SMAX', num_allies=n, num_enemies=m)
3  obs, state = vmap(env.reset)(key)
4  for _ in range(num_steps):
5      act = vmap(act_fn)(rng, env, obs, state)
6      obs, state, (_) = vmap(env.step)(act, state)
```

# 3 | Behaviour trees

▶ Behaviour trees (BT) are a way to model the behaviour of agents.

▶ They are used in games and robotics.

# 3 | Atomics

▶ Atomics are the leaves of the tree.

▶ They are the actions that the agent can take.

# 3 | BTBank

- ▶ BTBank is a library for creating and running BTs.
- ▶ It is written in Python.
- ▶ sqlite3 is used to store the trees.

# 4 | Language model

▶ The language model is a transformer model.

▶ I/O architecture.

▶ The output is a sequence of tokens.

# References

[1]   Alexander Rutherford et al. *JaxMARL: Multi-Agent RL Environments in JAX*. Dec. 2023. DOI: 10.48550/arXiv.2311.10090. arXiv: 2311.10090 [cs].