

# Improving the FLAME GPU User Experience through Examples

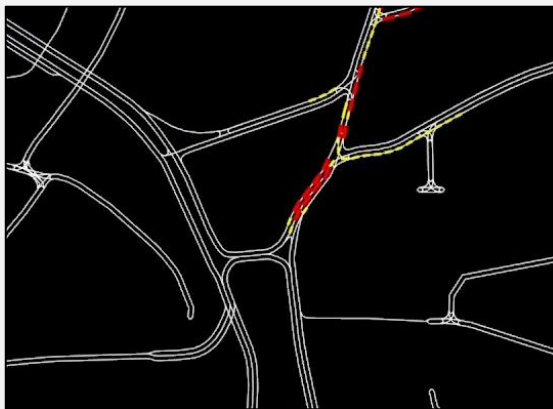
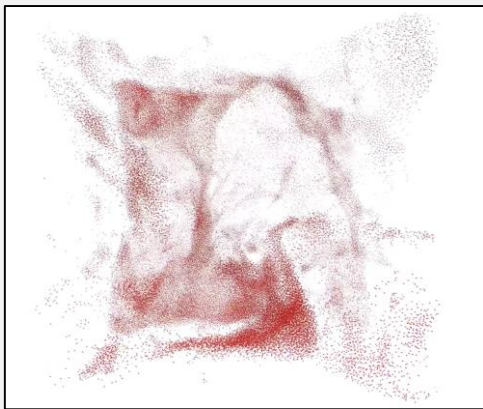
Lewie Fleming

BSc Computer Science, University of Sheffield



# Problem Introduction

---



**FLAME GPU:** Open-source GPU-accelerated software framework for *parallel* Agent-Based Modelling.

**NetLogo:** Open-source software framework for *serial* Agent-Based Modelling.

*Complex agentic models (written in CUDA/C++) executed in parallel on GPUs.*

---

**Goal:** Make agent-based modelling via FLAME GPU more accessible to social science and humanities dRTPs.

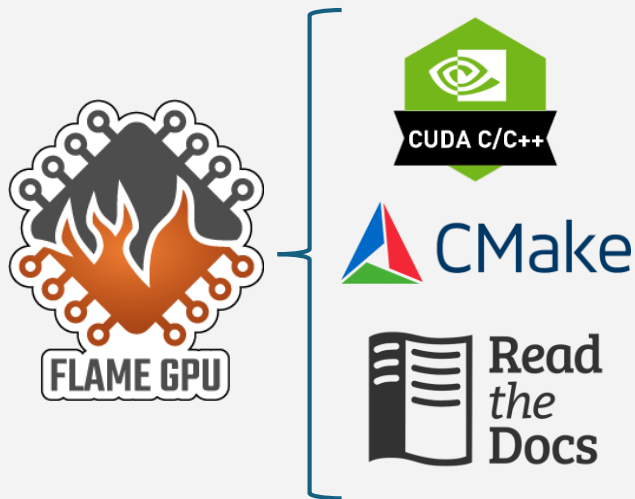


**NetLogo**

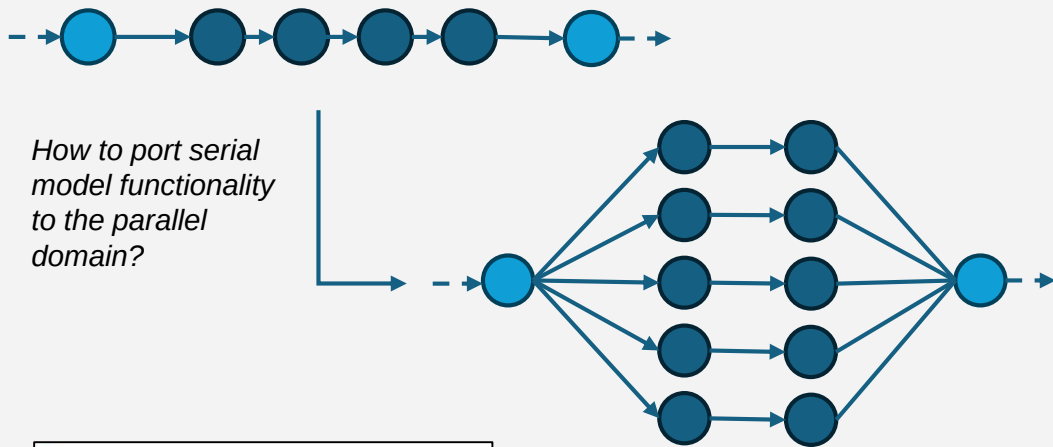
(2-4 models)



# Technical Background



*“The use of native Python agent functions...is currently supported...but classed as an **experimental feature**.”*



```
to setup
clear-all
set-default-shape turtles "square"
;; make some green trees
ask patches with [(random-float 100) < de
[ set pcolor green ]
;; make a column of burning trees
ask patches with [pxcor = min-pxcor]
[ ignite ]
;; set tree counts
set initial-trees count patches with [pco
set burned-trees 0
reset-ticks
end
```

```
# Init fn
class init_fn(pyflamegpu.HostFunction):
    def run(self, FLAMEGPU):
        t_pop = FLAMEGPU.agent("fire")
        for x in range(1, DIM):
            for y in range(1, DIM):
                if np.random.randint(100) < DENSITY:
                    t = t_pop.newAgent()
                    t.setVariableFloat("x", x)
                    t.setVariableFloat("y", y)
                    t.setVariableInt("status", TREE)

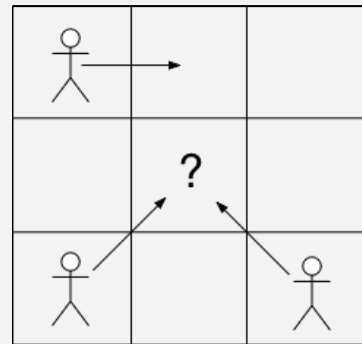
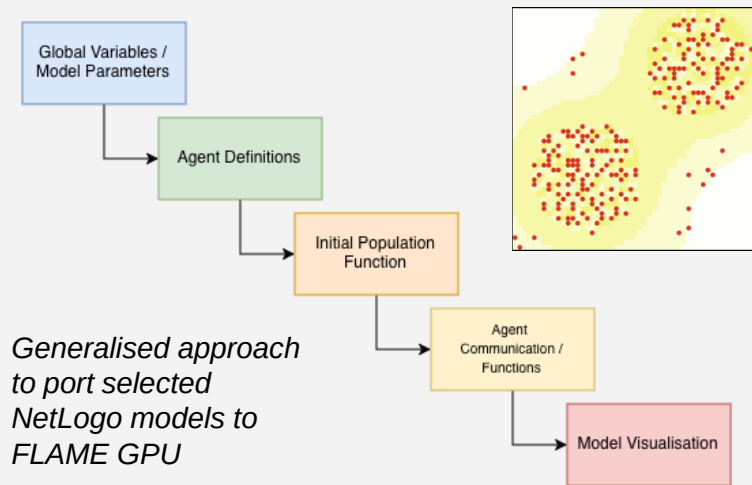
        for i in range(DIM):
            t = t_pop.newAgent()
            t.setVariableFloat("x", 0)
            t.setVariableFloat("y", i)
            t.setVariableInt("status", FIRE)
```

*Which models?*

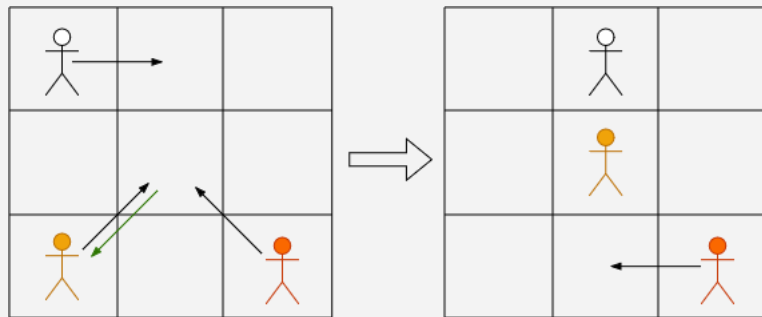
*Previous work?*

# Approach

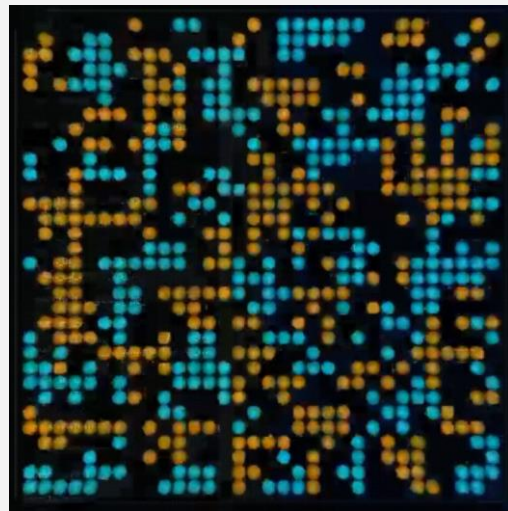
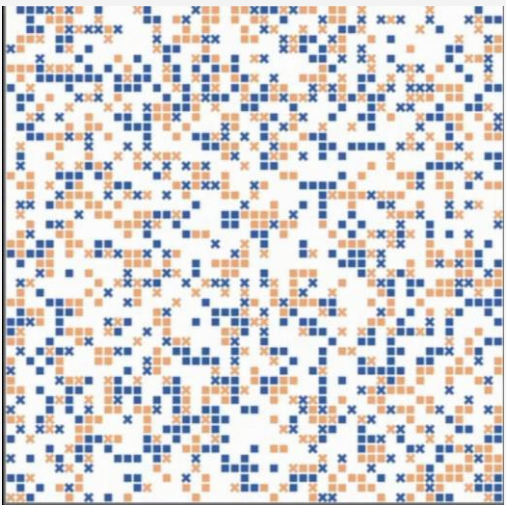
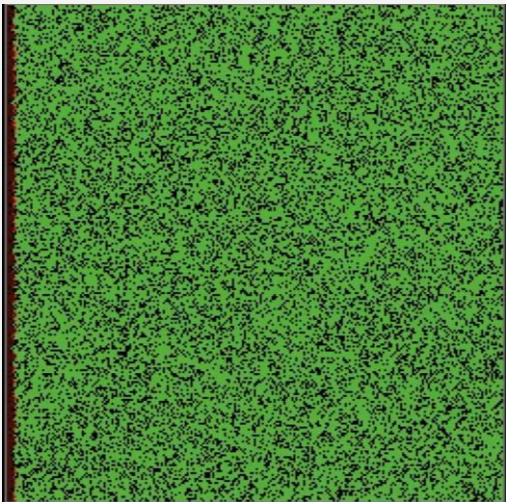
1. Fire (Earth Science)
2. Segregation (Social Science)
3. Bidding Market (Economics)
4. Rebellion (Social Science)

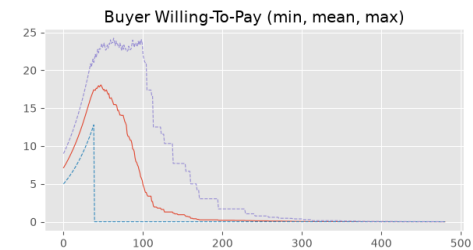
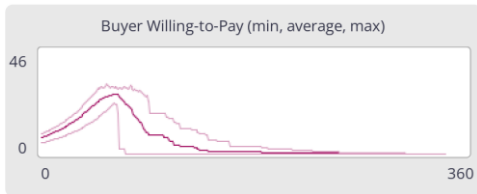
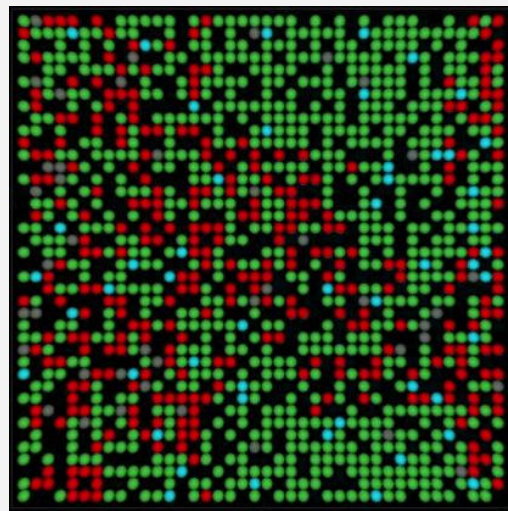
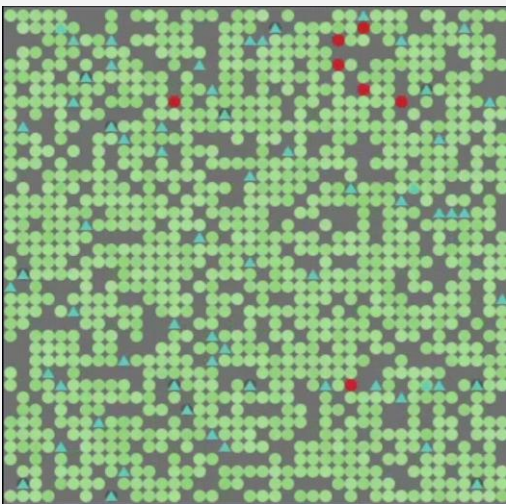


*Determining movement conflicts in parallel ABMs?*



*Bidding approach solution (used as a **submodel**)*

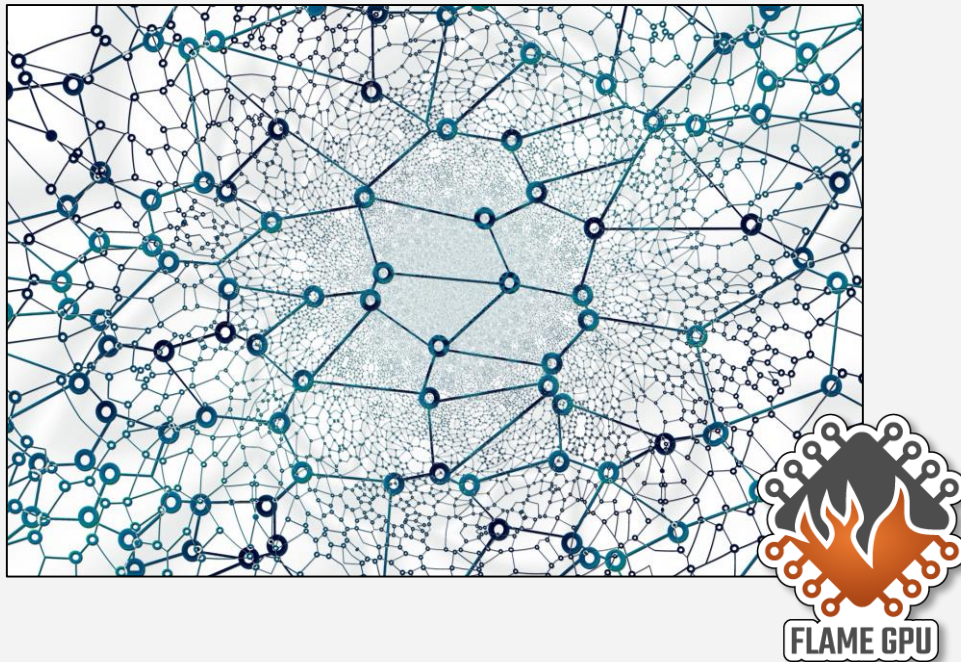




# Applications and Future Work

---

- **Applications:**
  - Reusable and open-source Python FLAME GPU code.
  - Proof of Concept demonstrating framework flexibility.
- **Future Work:**
  - Transpilation of NetLogo code to CUDA/C++/Python
  - Python submodel / template library





# University of Sheffield

**Get in touch:** [l.t.fleming@sheffield.ac.uk](mailto:l.t.fleming@sheffield.ac.uk)

[www.sheffield.ac.uk](http://www.sheffield.ac.uk)