Cellular Automata (CA) are interesting models of computation, where the actual information processing, transmission of information and storage are massively distributed and parallelized, and each component of the system interacts only locally with the closest neighbors. One such example of CA is the Game of Life, which is proven to be computationally universal. Another example, in the case of a continuous CA is Lenia [1]. A recent work by Google is Growing Neural Cellular Automata (NCA) [2], a continuous CA where neural networks are trained to control the growth process of given shapes, and to regenerate when damaged. Our follow-up work uses NCA for reinforcement learning [3]. Cellular substrates may constitute an “AI generating algorithm” [4].

In this project, the goal is to extend such works by investigating how to achieve long-term emergent dynamics in such cellular automata. In particular, two investigations targeting long-term dynamics are [4] and [5].

The project will encompass:

* The identification of two suitable cellular automata architectures (examples are life-like CA, multiple neighbor CA, multiple state CA, continuous CA, neural CA, spiking NCA, others)
* The identification of metrics to measure the long-term dynamics
* The implementation of the setup to run experiments
* A learning mechanism to find suitable update rules (based on neural nets or other mechanisms depending on the chosen CA architecture) such as evolutionary algorithms
* Documenting the results in a scientific paper

[1] Chan, Bert Wang-Chak. "Lenia: Biology of Artificial Life."<https://chakazul.github.io/lenia.html> (an example video here <https://vimeo.com/440386996>)

[2] Mordvintsev, Alexander, et al. "Growing neural cellular automata." Distill 5.2 (2020): e23. <https://distill.pub/2020/growing-ca/>

[3] <https://avariengien.github.io/self-organized-control/>

[4] <https://arxiv.org/pdf/2101.07627.pdf>

[5]<https://dl.acm.org/doi/pdf/10.1145/2463372.2463395>

Additional material:

<https://arxiv.org/pdf/2104.01008.pdf>

<https://arxiv.org/pdf/1911.01086.pdf>

<https://www.cs.swarthmore.edu/~meeden/DevelopmentalRobotics/lehman_ecj11.pdf> (this is long so please have only a brief look for now)

<https://www.oreilly.com/radar/open-endedness-the-last-grand-challenge-youve-never-heard-of/>

<https://www.youtube.com/watch?v=Rs5bly-HJOY>

<https://sebastianrisi.com/self_assembling_ai/>

**Checkpoints:**

1. **About Classical CA, Exiu\_sting models of CA (MNCA, MSCA, CCA-LENIA, NCA)**
2. **More about NCA**
3. **Long-term evolutionary dynamics of emerging behaviour in classical CA and continuous CA or NCA**
4. **Models we choose - Continuous CA and Multi-State CA**
5. **Bridging Continuous CA and Multi-State CA and NCA**
6. **Differentiable CA (1st version of Continuous CA)**
7. **Growing NCA and Self Organization in differentiable CA (ref** [**1**](https://distill.pub/2020/growing-ca/) **and** [**2**](https://arxiv.org/pdf/2101.07627.pdf)**)**
8. **What is long term emergent dynamics and growing complexity.**
9. **Reference** [**1**](https://arxiv.org/pdf/2104.01008.pdf) **and** [**2**](https://arxiv.org/pdf/1911.01086.pdf)
10. **Implementation of 2 Chosen CA architectures / models**
11. **Need to know what will be the update rule:**
    1. **If chosen Continuous CA**
    2. **If chosen NCA**
    3. **If chosen Multi state CA and so on…**