

Lab 5 Hands-on

Conway's Game of Life

1 Introduction

Conway's Game of Life is a cellular automaton devised by the British mathematician John Horton Conway in 1970. It's not a game in the traditional sense — there are no players, no scores, and no direct decisions. Instead, it's a mathematical simulation that demonstrates how simple local rules can produce incredibly complex and lifelike behavior.

The "world" of the Game of Life is a two-dimensional grid of cells, each of which can be in one of two states: alive or dead. The grid evolves in discrete steps called generations, with every cell's state at the next step determined entirely by its eight neighbors (the cells that touch it horizontally, vertically, and diagonally).

The rules are simple:

1. Survival – A living cell with two or three living neighbors survives.
2. Death by isolation – A living cell with fewer than two neighbors dies.
3. Death by overcrowding – A living cell with more than three neighbors dies.
4. Birth – A dead cell with exactly three living neighbors becomes alive.

Despite these minimal rules, the system exhibits emergent complexity. Some patterns remain static ("still lifes"), some oscillate periodically ("blinkers"), and others move across the grid ("gliders"). Entirely new and intricate behaviors can appear from just a few starting cells.

Conway's Game of Life has become a classic example of how simple rules can create complex systems. It has deep connections to mathematics, computer science, and theoretical biology, and even demonstrates Turing completeness — meaning it can, in principle, perform any computation given enough space and time.

2 Goal

You are required to develop a program that simulates and visualizes the evolution of Conway's Game of Life, starting from a randomly generated initial configuration. The computation of each successive generation must be performed using a Compute Pipeline.

The implementation may be carried out in the programming language of your choice. Note that WebGPU can be conveniently used from Python, JavaScript, Rust, and various other languages.