RNS Institute of Technology

Channasandra, Dr. Vishnuvardhan Road, Bangalore – 560098

Department of Computer Science and Engineering

Seminar Report

"Basic Gates Simulations using Conway's Game of Life"

Submitted by:

Sagnik Das – 1RN16CS086

Yash Vora – 1RN16CS123

RNS Institute of Technology

Channasandra, Dr. Vishnuvardhan Road, Bangalore – 560098

Department of Computer Science and Engineering

CERTIFICATE

Certified that the Practicing Project Seminar on topic "*Basic Gates Simulations using Conway's Game of Life*" has been successfully presented at RNS Institute of Technology by Yash Vora and Sagnik Das bearing USN 1RN16CS123 and 1RN16CS086 respectively for the IV Semester, Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi, for the enrichment of knowledge during the academic year 2017-2018.

Dr. G. T. Raju

Dean of Engineering and
Prof. and Head of Dept.

- Mrs. Rashmi M
- Ms. Medha Gowrayya

Project Coordinators

Abstract

The Game of Life, also known simply as Life, is a cellular automaton devised by the British mathematician John Horton Conway in 1970. The "game" is a zero-player game, meaning that its evolution is determined by its initial state, requiring no further input.

One interacts with the Game of Life by creating an initial configuration and observing how it evolves or, for advanced players, by creating patterns with particular properties.

Rules: The universe of the Game of Life is an infinite two-dimensional orthogonal grid of square cells, each of which is in one of two possible states, alive or dead. Every cell interacts with its eight neighbors, which are the cells that are horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:

- Any live cell with fewer than two live neighbors die.
- Any live cell with two or three live neighbors lives on to the next generation.
- Any live cell with more than three live neighbors dies.
- Any dead cell with exactly three live neighbors becomes a live cell.

These rules continue to be applied repeatedly to create further generations.

Acknowledgement

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and the College. I would like to extend my sincere thanks to all of them.

I am highly indebted to RNSIT, CSE Department, for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my special gratitude and thanks to Dr. G.T. Raju for giving me such attention and time.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

Contents

List of Figures

Figures

Screenshots

Introduction

Conway was interested in a problem presented in the 1940s by mathematician John von Neumann, who attempted to find a hypothetical machine that could build copies of itself and succeeded when he found a mathematical model for such a machine with very complicated rules on a rectangular grid. The Game of Life emerged as Conway's successful attempt to drastically simplify von Neumann's ideas. The game made its first public appearance in the October 1970 issue of Scientific American, in Martin Gardner's "Mathematical Games" column. From a theoretical point of view, it is interesting because it has the power of a universal Turing machine: that is, anything that can be computed algorithmically can be computed within Conway's Game of Life.

The game has opened up a whole new field of mathematical research called Cellular Automata. Because of Life's analogies with the rise, fall and alterations of a society of living organisms, it belongs to a growing class of what are called "simulation games" (games that resemble real life processes).

Ever since its publication, Conway's Game of Life has attracted much interest, because of the surprising ways in which the patterns can evolve. Life provides an example of emergence and self-organization. Scholars in various fields, such as computer science, physics, biology, biochemistry, economics, mathematics, philosophy, and generative sciences have made use of the way that complex patterns can emerge from the implementation of the game's simple rules. The game can also serve as a didactic analogy, used to convey the somewhat counter-intuitive notion that "design" and "organization" can spontaneously emerge in the absence of a designer. For example, philosopher and cognitive scientist Daniel Dennett has used the analogue of Conway's Life "universe" extensively to illustrate the possible evolution of complex philosophical constructs, such as consciousness and free will, from the relatively simple set of deterministic physical laws governing our own universe.

Literature Review

In the October 1970 issue of Scientific American, Martin Gardner popularized Conway's Game of Life by describing the game's capacity to "open up a whole new field of mathematical research, the field of cellular automata." Many soon discovered the great power of cellular automata as models of computation. It was determined that cellular automata, as they appear in the Game of Life, have the same computational capacity as Turing machines. The Church-Turing thesis that states:

"No method of computation carried out by a mechanical process can be more powerful than a Turing machine."

Therefore, as the Game of Life is Turing complete, it is one of the most powerful models of computation. In other words, no mechanical form of computation can solve a problem that a Turing machine or cellular automata cannot solve, given sufficient time and space. The following reasons lead researchers to determine the Game of Life has all the computational capability of Turing Machines, meaning it is Turing complete:

- Turing Machines have the ability to loop infinitely. Therefore, because the Game of Life has the ability to simulate an infinite loop or infinite recursion in the form of a Gosper gun or the puffer, people became convinced that the Game of Life has the computational capability of a Turing machine.
- A specific configuration of cellular automata called Rule 110 that uses a cyclic tag system is known to be Turing complete. Through a complex mathematical proof, Stephen Wolfram and his assistant Matthew Cook proved Rule 110 to be capable of supporting universal computation.

Problem Statement

In 1970, mathematician John H. Conway proposed a simulation that he called the Game of Life. Martin Gardner wrote a column about Life in the October 1970 issue of Scientific American that brought widespread attention to the Game of Life. It's not what most people think of as a game; there are no players and there's no way to win or lose the game. Instead, Life is more like a model or simulation in which you can play and experiment.

Life takes place on a two-dimensional grid of square cells. Each square cell can be either alive or dead (full or empty).

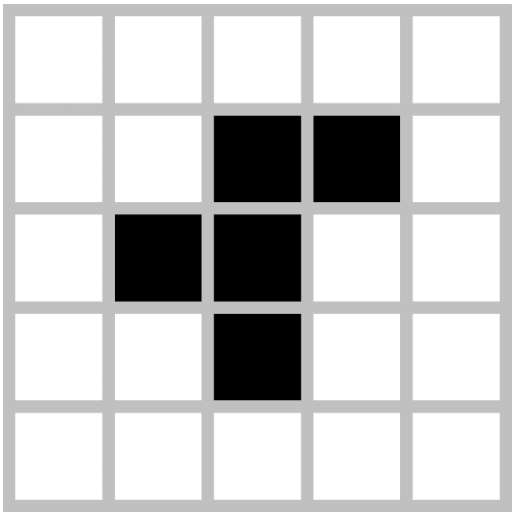


Fig 1.1

The simulation is carried out at fixed time steps; every time step, all the cells on the grid can switch from dead to alive, or alive to dead, depending on four simple rules that only depend on a given cell's eight immediate neighbors. Let's take any cell x in the diagram, whose neighbors are 8 in count.

If the cell is dead:

1. Birth: if exactly three of its neighbors are alive, the cell will become alive at the next step.

If the cell is already alive:

2. Survival: if the cell has two or three live neighbors, the cell remains alive.

Otherwise, the cell will die:

3. Death by loneliness: if the cell has only zero or one live neighbors, the cell will become dead at the next step.
4. Death by overcrowding: if the cell alive and has more than three live neighbors, the cell also dies.

These rules are simple and readily understandable, but they lead to surprisingly complex behavior. (The general term for a simulation carried out on a grid of cells and following some simple rules is a cellular automaton.) For example, here are some patterns that occur after starting with five live cells in a row.

This pattern ends up in a cycle that repeats itself endlessly. Researchers in Life have coined many terms for different types of pattern, and this one is called an oscillator. For example, oscillators go through a cycle of states and return to their initial state; still life patterns are stable and don't change over time at all; spaceships return to their initial configuration but in a different position, and therefore the pattern moves through the grid.

Requirement analysis

Software Requirements

1. Python 3.5.2
2. Default Python Libraries

Hardware Requirements

1. GUI capable display

Detailed Design

Libraries used

1. tkinter
2. random

tkinter

Tk/Tcl has long been an integral part of Python. It provides a robust and platform independent windowing toolkit, that is available to Python programmers using the tkinter package, and its extension, the tkinter.tix and the tkinter.ttk modules.

*The tkinter package is a thin object-oriented layer on top of Tcl/Tk. To use tkinter, you don't need to write Tcl code, but you will need to consult the Tk documentation, and occasionally the Tcl documentation. tkinter is a set of wrappers that implement the Tk widgets as Python classes. In addition, the internal module _tkinter provides a thread safe mechanism which allows Python and Tcl to interact. **

random

*This module implements pseudo-random number generators for various distributions. The functions supplied by this module are actually bound methods of a hidden instance of the random.Random class. You can instantiate your own instances of Random to get generators that don't share state. **

As quoted from official python documentation

Implementation

Algorithm

1. Initialize the cells in the grid.

2. At each time step in the simulation, for each cell (i, j) in the grid, do the following:

    a. Update the value of cell (i, j) based on its neighbors, taking into account the boundary conditions.

    b. Update the display of grid values.

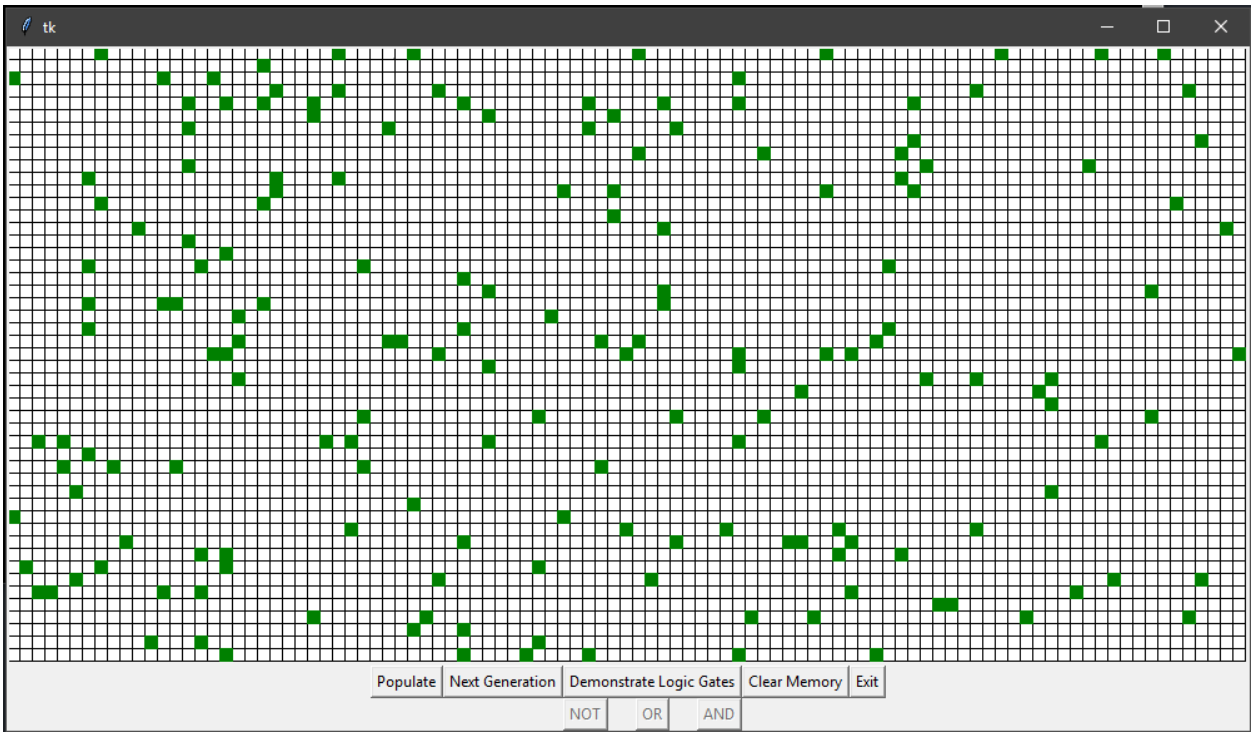
Result Analysis

Results

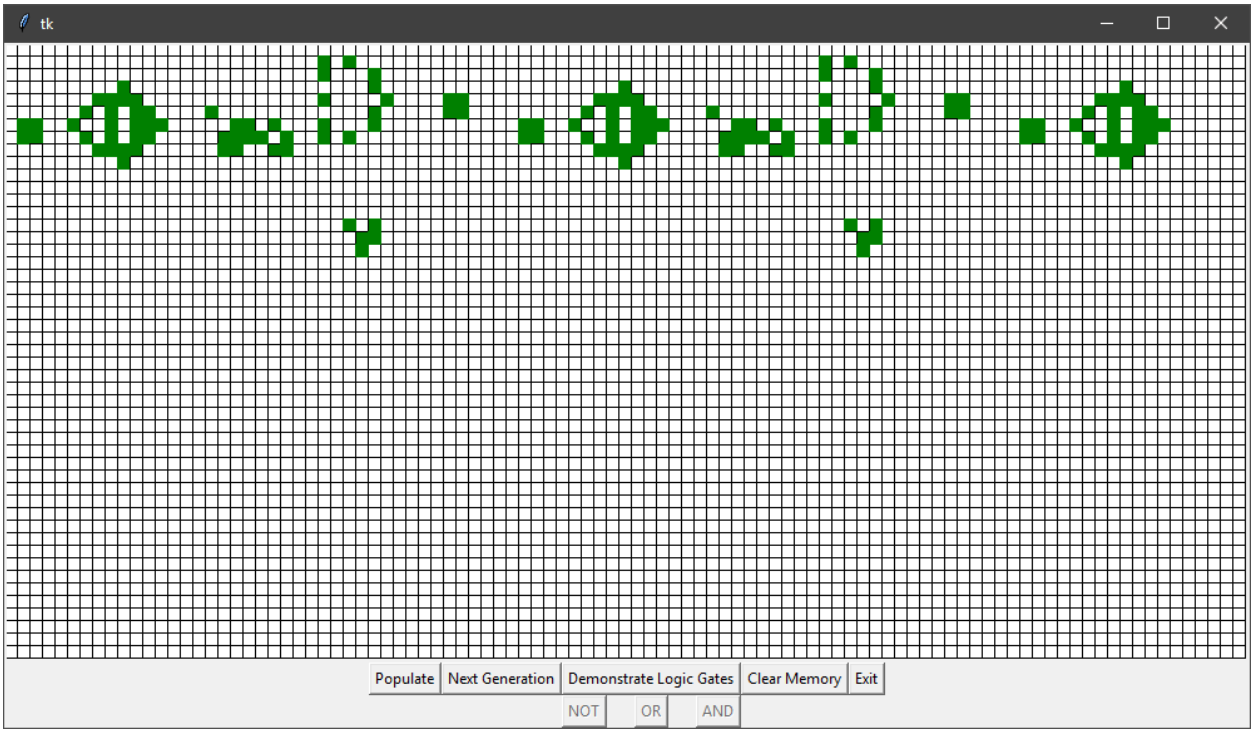
All the logic gates are observed to be working.

Snapshots

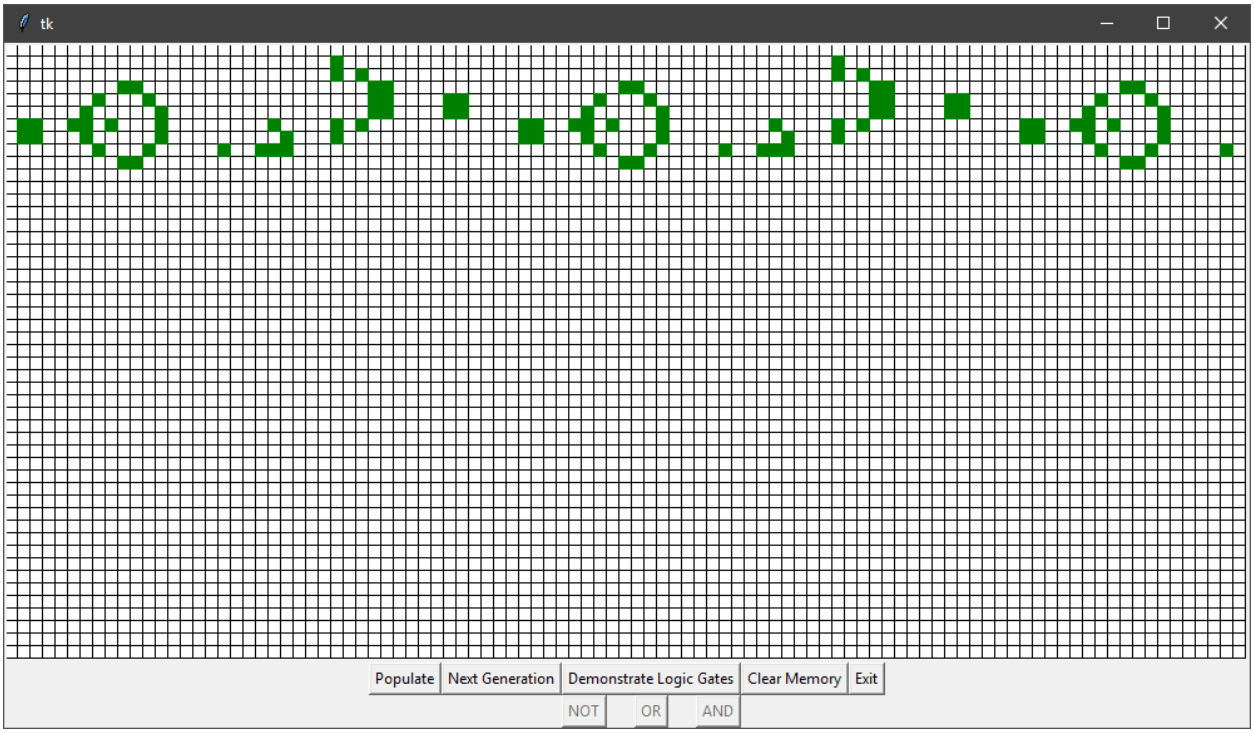
Screenshot 2.1



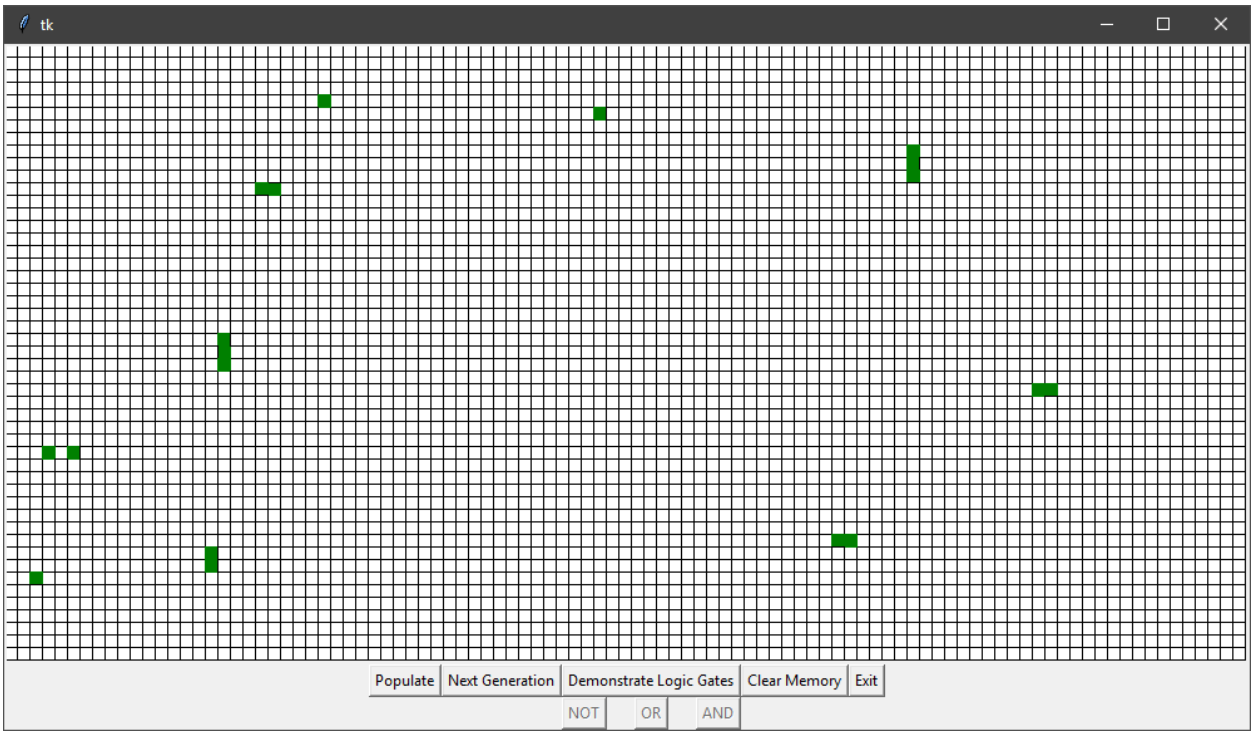
Screenshot 2.3



Screenshot 2.2



Screenshot 2.4



Conclusion

Conclusion

The above demonstration is a working model of a simulated Turing Machine.

Limitations

Extremely slow operation.

Future Work

Introduce a more efficient algorithm and add threaded support.

References

- http://web.stanford.edu/~cdebs/GameOfLife/
- http://home.iitk.ac.in/~tlavanya/14353FinalReport-GameOfLife.pdf
- http://mathworld.wolfram.com/GameofLife.html
- http://www.rennard.org/alife/CollisionBasedRennard.pdf
- https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life
- https://gist.github.com/jasonkeene/2140276

Fin.