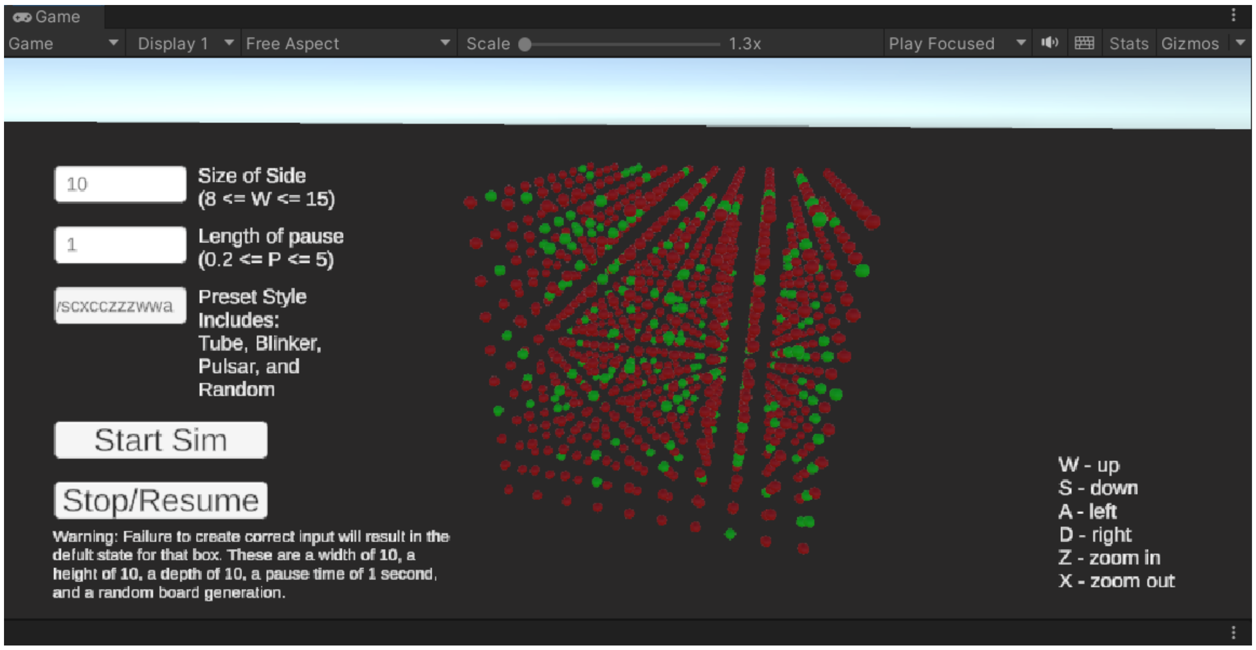
Three Dimensional Game of Life Implementation

Ames Rupnick

In the ever-evolving realm of computational simulations, the Game of Life stands as a captivating model that explores the emergence of patterns and complexity from simple rules. This paper delves into the process of transforming the traditional two-dimensional Game of Life into a dynamic and visually immersive three-dimensional model. Building upon the foundation of the class's initial 2D Game of Life project, this implementation not only extends the dimensionality but introduces a novel game logic tailored to a three-dimensional space. Through this exploration, this study not only figures out the challenges of adapting the cellular automaton to a higher dimension but also learns three unique patterns; a blinker, a pulsar, and a tube. These newfound structures not only showcase the adaptability of the Game of Life concept but also highlight the potential for discovering new patterns when exploring uncharted computational territories.



10 Size of Side (8 <= W <= 15)

1 Length of pause (0.2 <= P <= 5)

/scxcczzzwwa Preset Style Includes: Tube, Blinker, Pulsar, and Random

Warning: Failure to create correct input will result in the defult state for that box. These are a width of 10, a height of 10, a depth of 10, a pause time of 1 second, and a random board generation.

W - up
S - down
A - left
D - right
Z - zoom in
X - zoom out

The transition from the 2D model to the 3D one proved difficult. While the conversion of nodes, views, and mappings from two to three dimensions was straightforward, involving changing 2D arrays into 3D ones and all Unity vectors into 3 dimensions, unanticipated complexities arose during the handling of neighbors, addressing edge cases for toroidal topology, and managing texture transparency.

The challenge of increased computational load during the pathfinding algorithm, compounded by the transition from 8 to 26 neighbors in three dimensions rather than two, caused a reduction in the number of nodes per side from a maximum of 75 to 15. Despite this limitation, the resulting environment remained fine for testing basic patterns.

Another complication came about in establishing a toroidal graph to create wrapping around, simulating infinity. Unlike the 2D model's simpler corner-ignoring approach, the 3D implementation required the exclusion of both corners and the entire line spanning the "edges" of the three-dimensional box. Eventually, this was fixed and implemented in the final product, which was tested with a functioning blinker that was wrapped around the edge of the graph.

Finally, my final major issue occurred with node transparency. Although changing the NodeViews from tiles to spheres was relatively simple, and the code to change the alpha value of the shader was correct, the nodes would not change. After further research, the problem was revealed to be that the default shader that unity provides does not support changes to alpha values, and an entirely new shader would have to be written in HLSL in order for the transparency to work. In the end, this unseen complexity led to the pragmatic decision to shrink nodes for improved spatial clarity rather than using transparency.

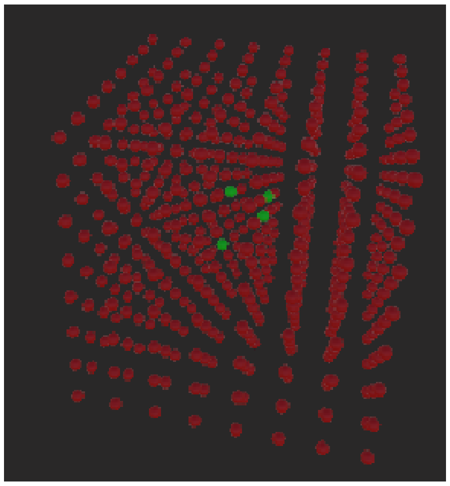
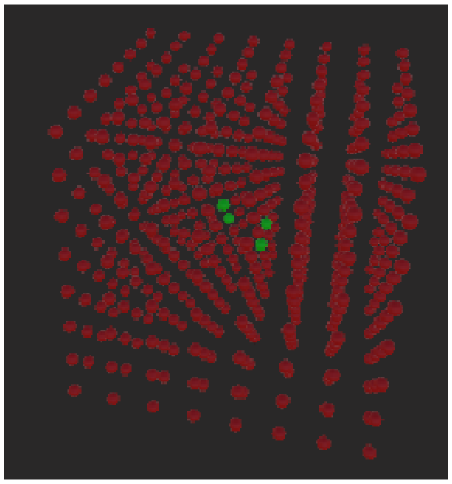
The actual algorithm had very little change, except that the original 2-3 neighbors rule created poor patterns. To compensate for this, the rules described below were implemented:

1. If a dead node has 4 neighbors, it becomes alive.
2. If a live node has 5 or 6 neighbors, it stays alive.
3. All other amounts lead to death, either through starvation or overcrowding.

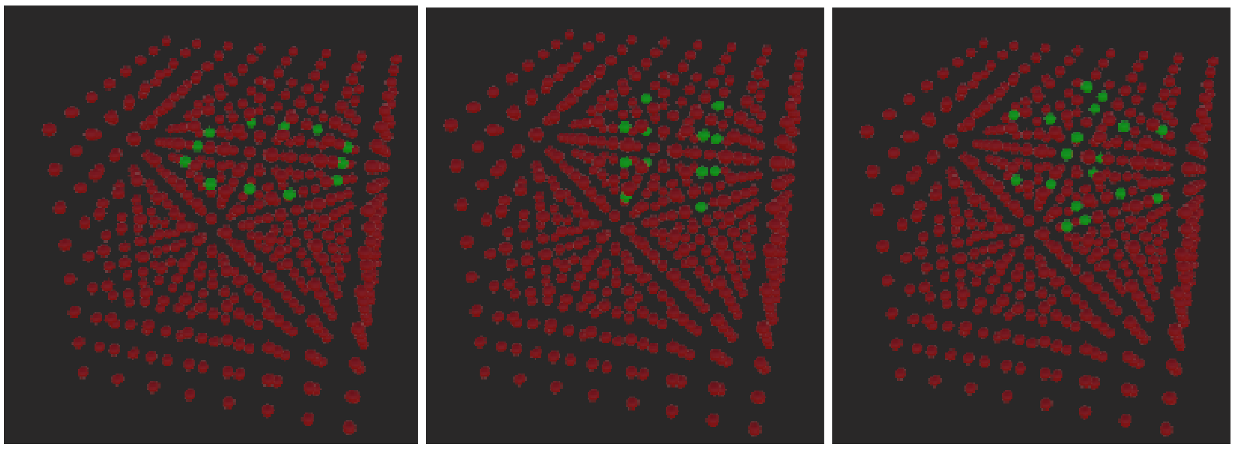
The exploration of various starting shapes for live nodes unveils a spectrum of configurations. These include a randomized pattern, a blinker pattern, a pulsar pattern, a tube pattern.The patterns are described below:

The random pattern involves activating approximately one-third of the nodes in a randomized fashion with each iteration. This adjustment diverges from the half-and-half distribution of the 2D implementation, a modification prompted by the realization that an even split causes premature termination of most nodes after the initial iteration, leading to monotonous and uninteresting patterns. An example of this can be seen in the first photo of this paper.

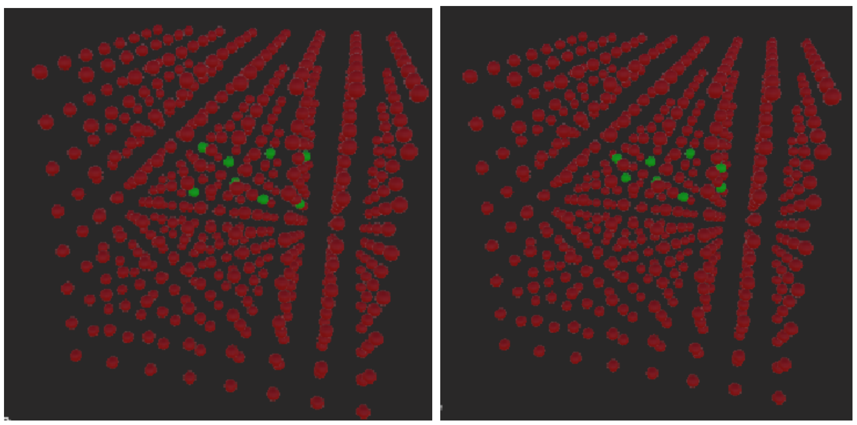
The blinker was built originally by Chris Evans, who calls this pattern the stairs. The stairs involve 3 nodes in an L shape, and one node slightly behind. This creates a never ending pattern of 2 states which iterate back and forth.

Moving on to the pulsar, crafted by Chris Evans and technically called the “tube”. This is not to be confused with my tube implementation, and frankly this should be named the circle. The pulsar starts with a circular arrangement of three nodes per side, cyclically reverting to the original circular pattern after completing three iterations.



Finally, in search of a block implementation, I stumbled on what I call the tube. This is the closest to a block seemingly possible, since both the increase in neighbors and increase in live neighbors needed to become alive meant that static shapes are exceptionally difficult. The tube involves an almost spiral pattern, where the center is fully static, and only the ends of the tube move in a blinking pattern.



In conclusion, the transformation of the traditional two-dimensional Game of Life into a dynamic and visually immersive three-dimensional model has been challenging yet rewarding. The adaptation process from 2D to 3D required addressing unforeseen complexities, such as node mapping, handling neighbors in a toroidal topology, and navigating challenges with texture transparency, ultimately leading to a pragmatic decision to adjust node sizes for enhanced visibility. Despite these technical challenges, the core algorithm underwent few and minor modifications, simply refining the rules to foster more robust patterns. The introduction of new patterns, including the blinker, pulsar, and tube, showcases the adaptability and creative potential inherent in extending the Game of Life into three-dimensional spaces. This exploration not only advances our understanding of cellular automata in higher dimensions, but also opens avenues for further inquiry of other patterns that are largely uncharted.

Source:

Building Conway's game of life in 3D. (2020, July 27).

https://chrisevans9629.github.io/blog/2020/07/27/game-of-life