Carleton University

Department of Systems and Computer Engineering

Assignment 2:

Cell-DEVS: Procedural Content Generation of a Game Map

SYSC 4906 G

Winter 2025

Professor: Dr. Gabriel Wainer

Submitted By:

Michael De Santis: 101213450

March 24, 2025

Part I: Conceptual Model

1 Background

Procedural Content Generation (PCG) is a technology that has found varied applications, including in the design of video games, where it is used to “produce infinite game maps without any human effort” [1]. In many genres of games, such as rogue-likes, the player is provided with a set of generated levels or maps to move through, with these maps formed dynamically at runtime through the procedural application of some algorithm. In this way, a player’s virtual world remains varied and new with each played iteration, avoiding the staleness of static and singularly designed maps or levels - and avoiding the need for game designers to manually generate such maps.

One prominent technology used in PCG applications is Cellular Automata (CA). Using CA, game maps are modeled using a celled grid space, and automation rules are applied to mathematically differentiate terrain and therefore procedurally generate maps. Fig. 1 below provides one example of a PCG game map generated from a multi-leveled CA [1].

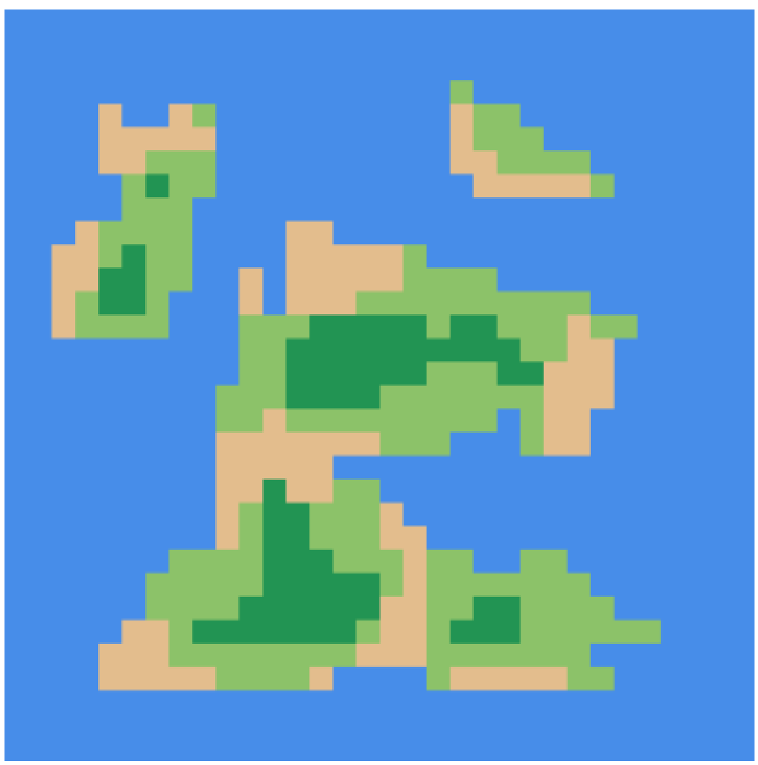

Figure 1: PCG Game Map Generated from a Multi-leveled CA [1]

2 Model Overview

The theory of CA may be combined with Discrete Event System Specification (DEVS) to form the Cell-DEVS methodology [2], which allows additional application of event-based modeling and simulation techniques to enhance PCG. This report documents the design and use of a PCG Map Generation tool, where the underlying CA model and simulation are formed using the Cell-DEVS methodology [2] and implemented using the Cadmium V2 library. Please see the Map Generation repository for source code, documentation, and examples.

The basis for the CA comes from Wu et. al.'s "Procedural Game Map Generation Using Multi-Leveled Cellular Automata by Machine Learning" (2021) [1], itself an extension of Cook's single-leveled PCG CA work, "Generate random cave levels using cellular automata [3]. Each of these works begin from Conway's Game of Life [4], using modified rules to achieve more globular, connected subspaces within the cell space to represent different types of discrete terrain: water, land, forest, and sand.

Wu et. al.'s original formulation of the CA [1] relies on multiple and discrete rounds of automation, where only one type of terrain is populated at a time before the final map is then composited together. In the model presented herein, the CA model has been adapted to perform all terrain differentiation simultaneously, allowing a composite map to be generated with all terrain layers in a single simulation. Additionally, the automation rules have been slightly adapted to permit additional behaviour: forest terrain formation is restricted to interior land (ie. land not adjacent to water), and receives a multiplicative bonus for each adjacent forest cell; sand formation is restricted to coastal land (ie. land adjacent to water), and receives a multiplicative bonus for each adjacent water cell. The base probabilities and multipliers for forest and sand terrain are also configurable for increased variation in experimentation.

The CA model herein uses the same Moore's neighbourhood as Wu et. al.'s [1], which is visualized in Fig. 2 below.

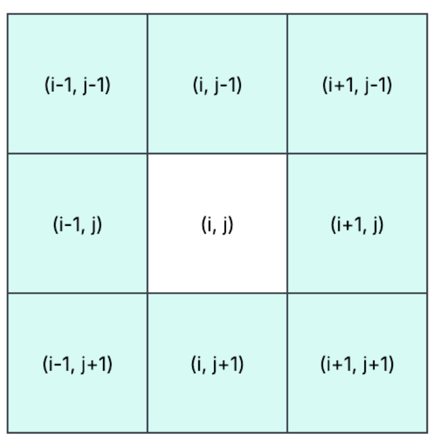

Figure 2: Moore's Neighbourhood for Cell (i, j)

The state of a cell corresponds to the type of terrain it is classified and represented as, with different types of terrain possessing different properties that may affect game play. For this model, terrain is visually represented through colouration as per the scheme provided in Fig. 3 below.

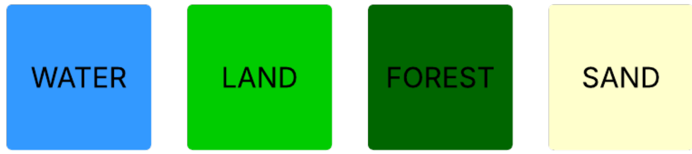

Figure 3: Terrain Colouration Scheme

Based on the state of a cell (i, j), the state of its neighbours, and the automation rules defined in the local computing function (formally presented in Part II below), each cell may transition to another type of terrain during a round of automation. Water can become land, and land can become water, which has the potential to modify the topologies of the land and water masses within each map. Land (not water) may differentiate into forest or sand terrain; correspondingly, forest or sand terrain may revert to general land terrain (but not water). In brief, the possible transformations (which are expounded in the formal specifications of Part II) are as follows:

- Water $\longrightarrow$ { Water, Land }
- Land $\longrightarrow$ { Water, Land, Forest, Sand }
- Forest $\longrightarrow$ { Land, Forest }
- Sand $\longrightarrow$ { Land, Sand }

By configuring an initial cell space with only water and land cells and applying several rounds of automation, different PCG game maps may be generated. As several automation rules rely on random number generation and probability rates and multipliers, even the same initial configuration can result in different PCG map results.

Part II: Cell-DEVS Formal Specification and Implementation

This section introduces Cell-DEVS formal specifications [2] for the atomic models of the system under study, corresponding to the conceptual model description of Part I. It additionally defines an experimental frame that is exercised under simulation using the Map Generation utility, with each experiment supplemented by results and analysis.

1 Cell-DEVS Formal Specification

Cell-DEVS formal specifications for atomic model components (ie. map cells) are presented according to the following formal definition [2]:

$$\langle TDC = X, Y, S, N, delay, d, \delta_{int}, \delta_{ext}, \tau, \lambda, D \rangle$$

where:

- X = the set of input external events
- Y = the set of output external events
- S = the state set
- N = the set of input values
- d = the type of delay for each cell
- $delay$ = the delay duration for each cell
- δ_{int} = the internal transition function
- δ_{ext} = the external transition function
- τ = the local computing function
- λ = the output function
- D = the delay function

1.1 Atomic Model: Map Cell

The Cell-DEVS formal specification for each map cell atomic model is as follows:

- $X = \emptyset$
- $Y = \emptyset$
- $S = \{0, 1, 2, 3\}$ where:

  0 = water terrain (blue)

  1 = land terrain (light green)

  2 = forest terrain (dark green)

  3 = sand terrain (yellow)
- $delay$ = inertial
- N = Moore's Neighbourhood
- $d = 1$ time unit
- $\tau = \{$

```
/* Local Variables */

// Accumulators for all neighbor types (differentiated by terrain)
int water_neighbors = 0;
int land_neighbors = 0;
int forest_neighbors = 0;
int sand_neighbors = 0;
int non_water_neighbors = 0;

/* Canvas the Neighborhood */

// Canvas this cell's neighborhood to categorize neighbor types
// ie. tally neighbor cells by terrain type
for (neighbor : neighborhood) {
    // WATER neighbors
    if(neighbor.state.terrain == WATER) {
        water_neighbors++;
    }
    // LAND neighbors
    if(neighbor.state.terrain == LAND) {
        land_neighbors++;
    }
    // FOREST neighbors
    if(neighbor.state.terrain == FOREST) {
        forest_neighbors++;
    }
    // SAND neighbors
```

```
    if(neighbor.state.terrain == SAND) {
        sand_neighbors++;
    }
}

/* Mutate State Based on Rules and Return */

// Case: WATER cell
if(state.terrain == WATER) {
    // Uncount this cell from its own neighborhood tally
    water_neighbors--;
    // Tally the alive (ie. non-WATER neighbors) for rule use
    non_water_neighbors = land_neighbors + forest_neighbors + sand_neighbors;

    // Case: WATER --> LAND
    if(non_water_neighbors > state.land_birth_limit) {
        state.terrain = LAND;
    }
}

// Case: LAND cell
else if (state.terrain == LAND) {
    // Uncount this cell from its own neighborhood tally
    land_neighbors--;
    // Tally the alive (ie. non-WATER neighbors) for rule use
    non_water_neighbors = land_neighbors + forest_neighbors + sand_neighbors;

    // Case: LAND --> WATER
    if(non_water_neighbors < state.land_death_limit) {
        state.terrain = WATER;
    }

    // Case: LAND --> SAND | LAND
    // SAND can only form near WATER (ie. at least one WATER neighbor)
    else if (water_neighbors) {
        // Get random number in [0, 1] to test against SAND rules
        double r = randomProbability();
        // Get base threshold for becoming SAND
        double sand_threshold = state.sand_base_rate;
        // SAND is more likely to form near WATER.
        // Apply multiplier to increase chance of becoming SAND
        // based on number of WATER neighbours
        sand_threshold += state.sand_multiplier * water_neighbors;
        // Case: LAND --> SAND
        if (r <= sand_threshold) {
            state.terrain = SAND;
        }
        // Case: LAND --> LAND
    }

    // Case: LAND --> FOREST | LAND
```

```
        // Forests only grow in interior (ie. no WATER neighbors)
        else {
            // Get random number in [0, 1] to test against FOREST rules
            double r = randomProbability();
            // Get base threshold for becoming FOREST
            double forest_threshold = state.forest_base_rate;
            // FOREST is more likely to form near FOREST.
            // Apply multiplier to increase chance of becoming FOREST
            // based on number of FOREST neighbors.
            forest_threshold += state.forest_multiplier * forest_neighbors;
            // Case: LAND --> FOREST
            if (r <= forest_threshold) {
                state.terrain = FOREST;
            }
            // Case: LAND --> LAND
        }

    }

    // Case: FOREST cell
    else if (state.terrain == FOREST) {
        // Uncount this cell from its own neighborhood tally
        forest_neighbors--;
        // Tally the alive (ie. non-WATER neighbors) for rule use
        non_water_neighbors = land_neighbors + forest_neighbors + sand_neighbors;

        // Case: FOREST --> LAND
        // If too much WATER nearby, revert to LAND
        if(water_neighbors > state.forest_death_limit) {
            state.terrain = LAND;
        }
        // Case: FOREST --> FOREST
    }

    // Case: SAND cell
    else if (state.terrain == SAND) {
        // Uncount this cell from its own neighborhood tally
        sand_neighbors--;
        // Tally the alive (ie. non-WATER neighbors) for rule use
        non_water_neighbors = land_neighbors + forest_neighbors + sand_neighbors;

        // Case: SAND --> LAND
        // If not enough WATER nearby, revert to LAND
        if(water_neighbors < state.sand_death_limit) {
            state.terrain = LAND;
        }
        // Case: SAND --> SAND
    }

    return state;
}
```

- D = inertial delay of 1 time unit
- $\delta_{int}, \delta_{ext}, \lambda$ are defined according to the Cell-DEVS specifications [2].

2 Experimentation

The model defined above was implemented in C++ using the Cadmium V2 library to create the Map Generation tool. The Map Generation tool allows for configuration of the initial state of the cell space to provide customizable starting points for simulation; it additionally allows for the configuration of other cell state parameters that influence the local computing function. Three sets of experiments are discussed below, with each set corresponding to an initial cell space that is simulated under variable sets of configuration parameters that are presented as a variation of the experiment. Simulation outputs for each experimental variation were logged to a file for analysis, and given to the Cell-DEVS Visualizer to generate images of the initial and final states of the map, as well as a video showing the simulation dynamically.

2.1 Experiment 1: Island with Inner Bodies of Water

The following experiment was conducted using an initial map configuration representing a large island, surrounded on all sides by water, and itself containing smaller inner bodies of water. All simulations were run with a duration of 20 time units. The initial map, before any automation, is represented in Fig. 4 below. Final maps for each experimental variation are presented in the following sections.

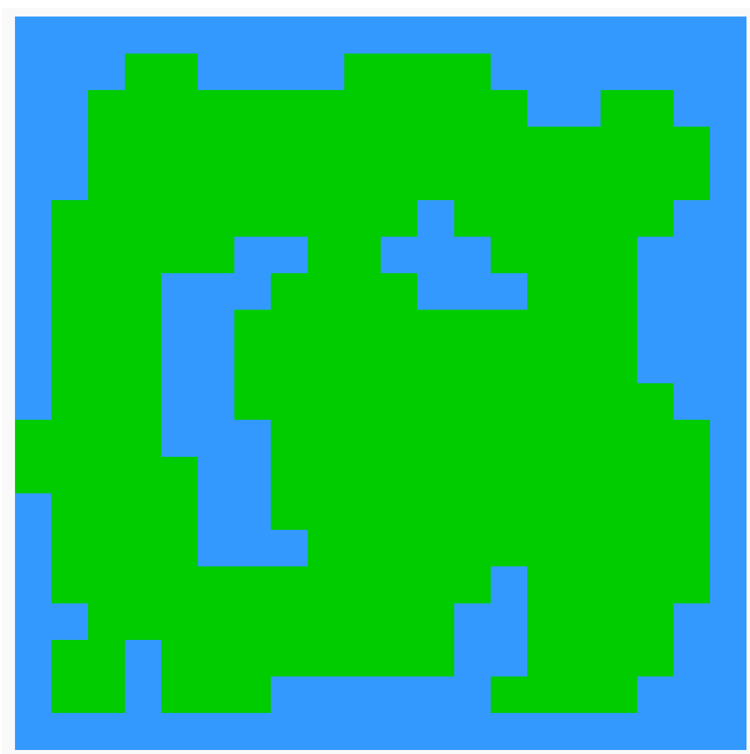

Figure 4: Experiment 1 Initial Map

Experiment 1: Variation 1
In this variation, forest and sand terrain base probability rates and multipliers were moderately set. Fig. 5 below shows the final map after a simulation duration of 20 time units.

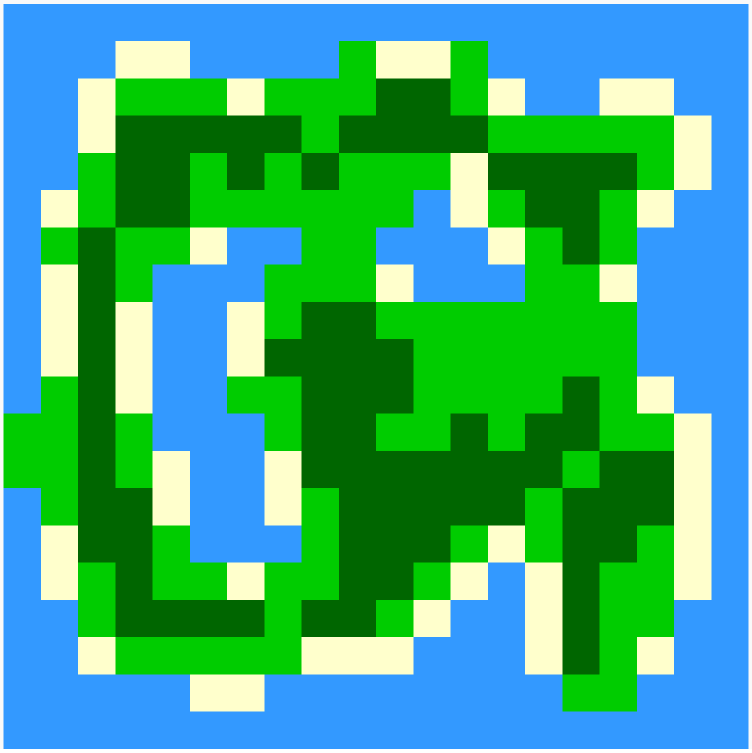

Figure 5: Experiment 1 Variation 1 Final Map

Please see the following files in the Map Generation repository for additional information:

- Configuration File: `/config/ex01/ex01_v01_mapgen_config.json`
- Visualization File: `/config/ex01/ex01_v01_mapgenVisualization_config.json`
- Output Log: `/example-output/ex01/ex01_v01_mapgen_grid_log.csv`
- Initial Map: `/example-output/ex01/ex01_mapgen_initial.png`
- Final Map: `/example-output/ex01/ex01_v01_mapgen_end.png`
- Simulation Video: `/example-output/ex01_v01_mapgen_video.webm`

Experiment 1: Variation 2
In this variation, forest and sand terrain base probability rates and multipliers were set to increase forest formation and decrease sand formation. Fig. 6 below shows the final map after a simulation duration of 20 time units.

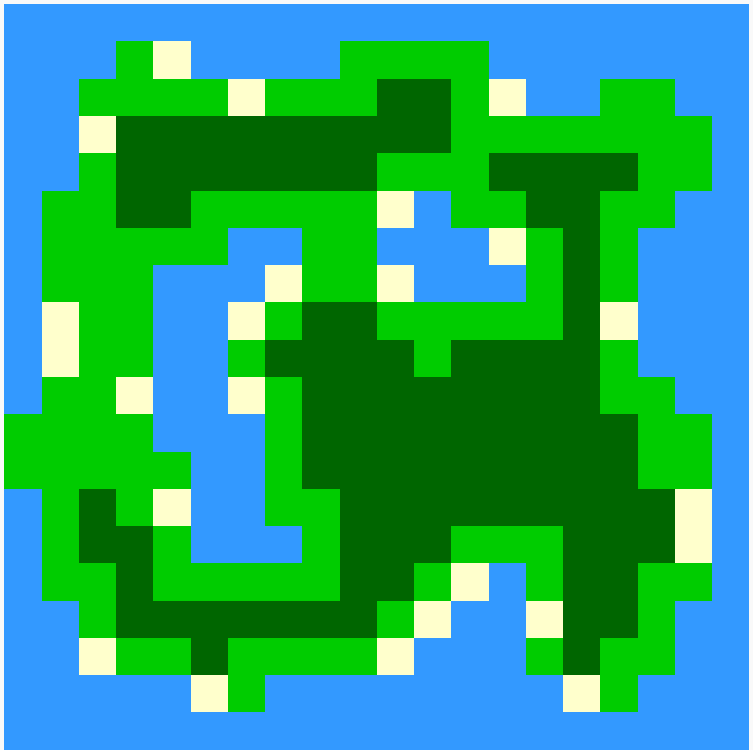

Figure 6: Experiment 1 Variation 2 Final Map

Please see the following files in the Map Generation repository for additional information:

- Configuration File: `/config/ex01/ex01_v02_mapgen_config.json`
- Visualization File: `/config/ex01/ex01_v02_mapgenVisualization_config.json`
- Output Log: `/example-output/ex01/ex01_v02_mapgen_grid_log.csv`
- Initial Map: `/example-output/ex01/ex01_mapgen_initial.png`
- Final Map: `/example-output/ex01/ex01_v02_mapgen_end.png`
- Simulation Video: `/example-output/ex01_v02_mapgen_video.webm`

Experiment 1: Variation 3

In this variation, forest and sand terrain base probability rates and multipliers were set to decrease forest formation and increase sand formation. Fig. 7 below shows the final map after a simulation duration of 20 time units.

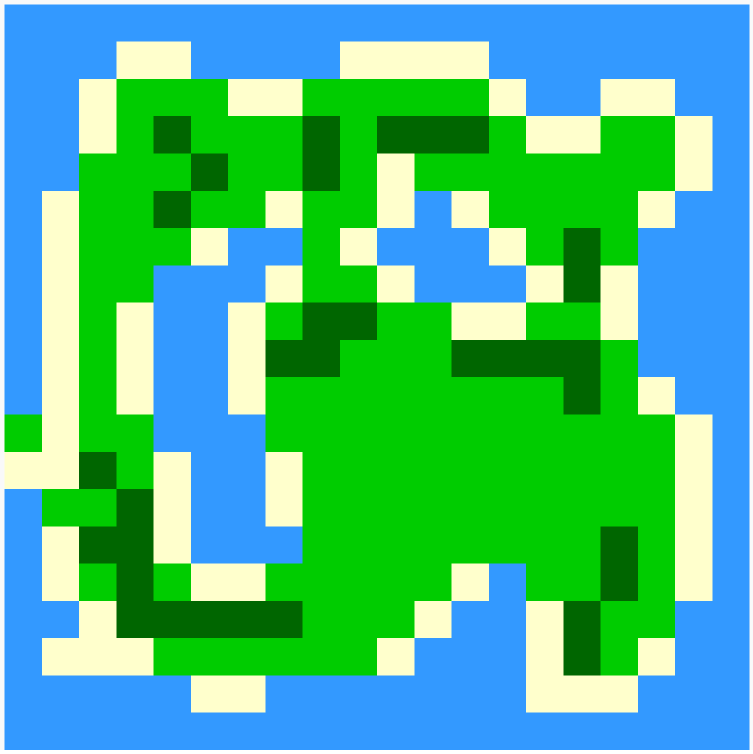

Figure 7: Experiment 1 Variation 3 Final Map

Please see the following files in the Map Generation repository for additional information:

- Configuration File: `/config/ex01/ex01_v03_mapgen_config.json`
- Visualization File: `/config/ex01/ex01_v03_mapgenVisualization_config.json`
- Output Log: `/example-output/ex01/ex01_v03_mapgen_grid_log.csv`
- Initial Map: `/example-output/ex01/ex01_mapgen_initial.png`
- Final Map: `/example-output/ex01/ex01_v03_mapgen_end.png`
- Simulation Video: `/example-output/ex01_v03_mapgen_video.webm`

2.2 Experiment 2: Lakes within a Consolidated Land Mass

The following experiment was conducted using an initial map configuration representing a large consolidated land mass containing smaller inner bodies of water, some of which additionally contain inner land masses. All simulations were run with a duration of 20 time units. The initial map, before any automation, is represented in Fig. 8 below. Final maps for each experimental variation are presented in the following sections.

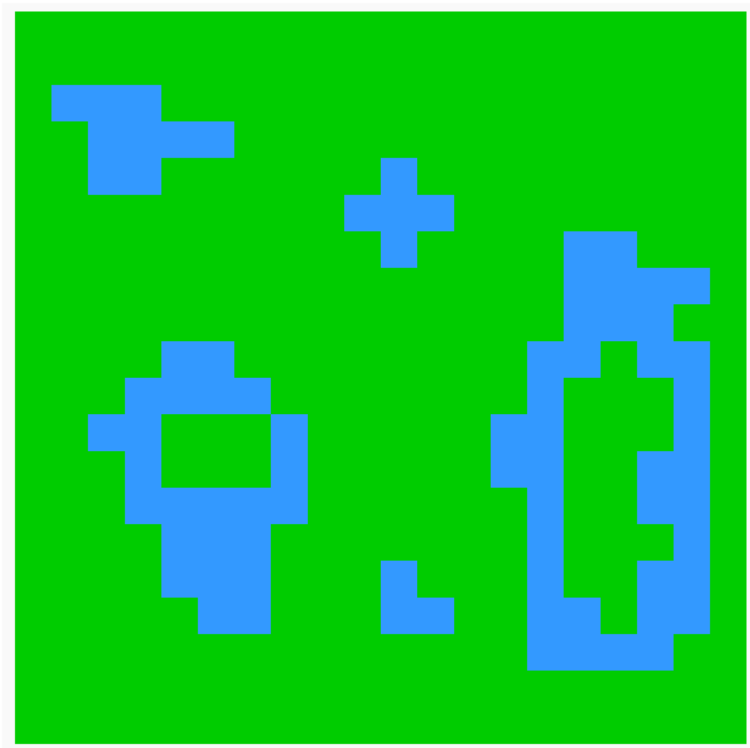

Figure 8: Experiment 2 Initial Map

Experiment 2: Variation 1
In this variation, forest and sand terrain base probability rates and multipliers were moderately set. Fig. 9 below shows the final map after a simulation duration of 20 time units.

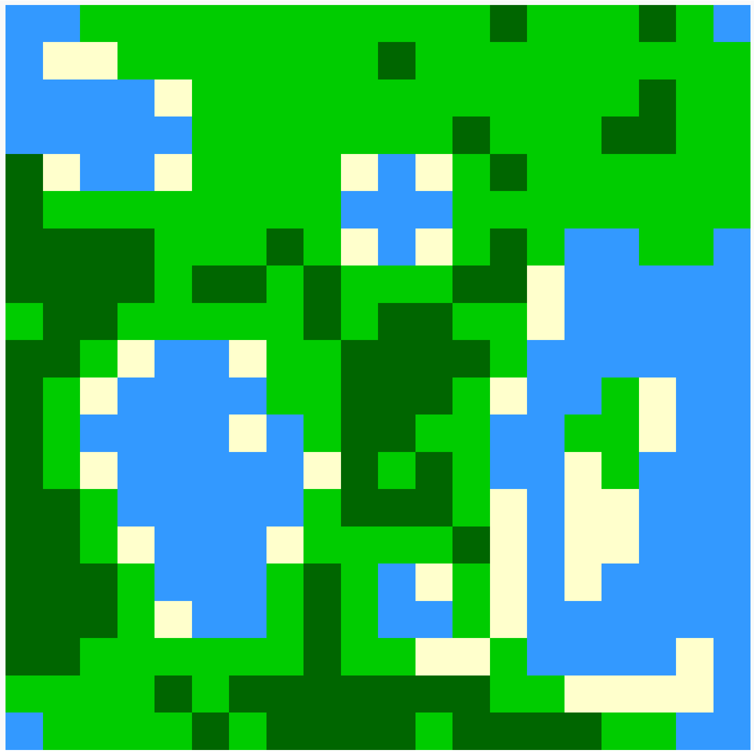

Figure 9: Experiment 2 Variation 1 Final Map

Please see the following files in the Map Generation repository for additional information:

- Configuration File: `/config/ex02/ex02_v01_mapgen_config.json`
- Visualization File: `/config/ex02/ex02_v01_mapgenVisualization_config.json`
- Output Log: `/example-output/ex02/ex02_v01_mapgen_grid_log.csv`
- Initial Map: `/example-output/ex02/ex02_mapgen_initial.png`
- Final Map: `/example-output/ex02/ex02_v01_mapgen_end.png`
- Simulation Video: `/example-output/ex02_v01_mapgen_video.webm`

Experiment 2: Variation 2

In this variation, forest and sand terrain base probability rates and multipliers were set to increase forest formation and decrease sand formation. Fig. 10 below shows the final map after a simulation duration of 20 time units.

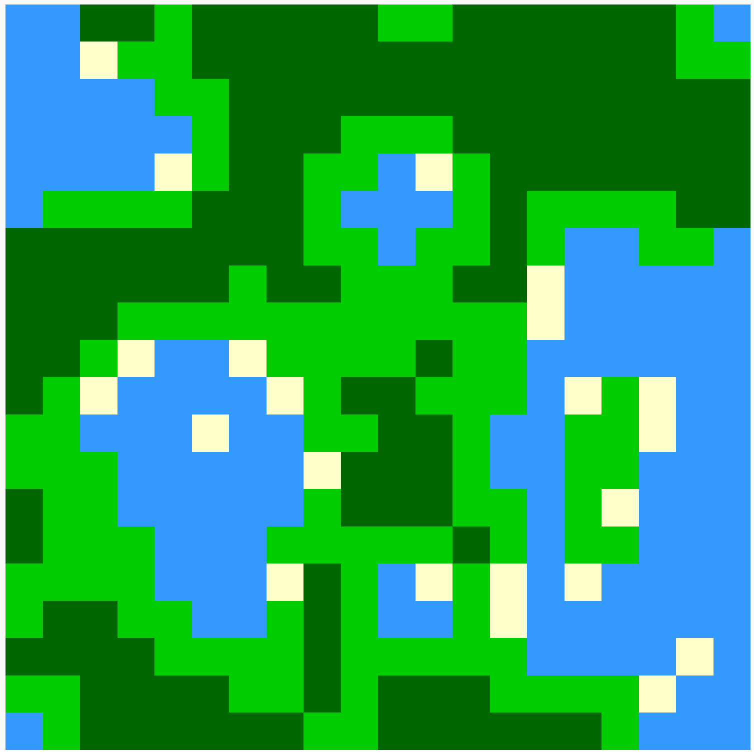

Figure 10: Experiment 2 Variation 2 Final Map

Please see the following files in the Map Generation repository for additional information:

- Configuration File: `/config/ex02/ex02_v02_mapgen_config.json`
- Visualization File: `/config/ex02/ex02_v02_mapgenVisualization_config.json`
- Output Log: `/example-output/ex02/ex02_v02_mapgen_grid_log.csv`
- Initial Map: `/example-output/ex02/ex02_mapgen_initial.png`
- Final Map: `/example-output/ex02/ex02_v02_mapgen_end.png`
- Simulation Video: `/example-output/ex02_v02_mapgen_video.webm`

Experiment 2: Variation 3
In this variation, forest and sand terrain base probability rates and multipliers were set to decrease forest formation and increase sand formation. Fig. 11 below shows the final map after a simulation duration of 20 time units.

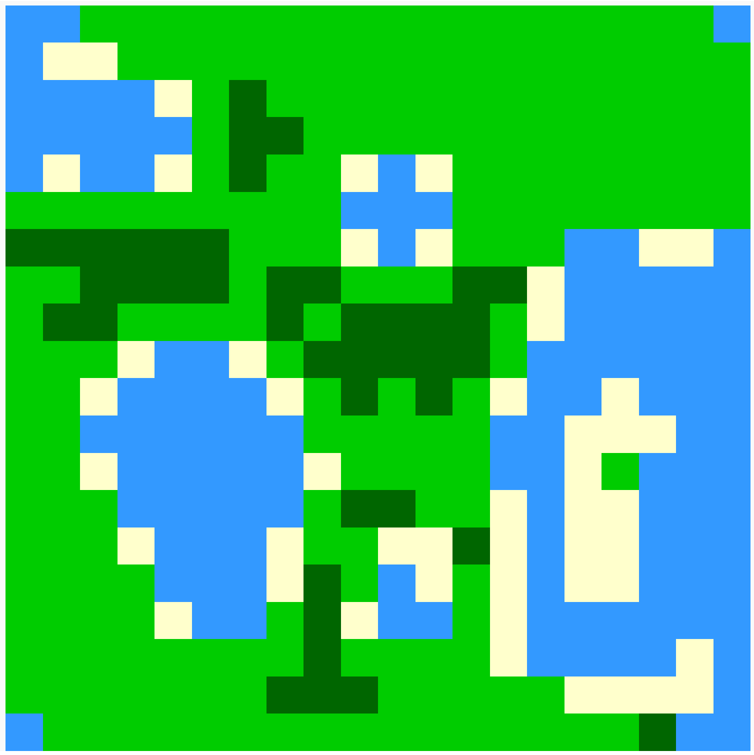

Figure 11: Experiment 2 Variation 3 Final Map

Please see the following files in the Map Generation repository for additional information:

- Configuration File: `/config/ex02/ex02_v03_mapgen_config.json`
- Visualization File: `/config/ex02/ex02_v03_mapgenVisualization_config.json`
- Output Log: `/example-output/ex02/ex02_v03_mapgen_grid_log.csv`
- Initial Map: `/example-output/ex02/ex02_mapgen_initial.png`
- Final Map: `/example-output/ex02/ex02_v03_mapgen_end.png`
- Simulation Video: `/example-output/ex02_v03_mapgen_video.webm`

2.3 Experiment 3: Islands at Sea

The following experiment was conducted using an initial map configuration representing several islands of various shapes within a sea of water. All simulations were run with a duration of 20 time units. The initial map, before any automation, is represented in Fig. 12 below. Final maps for each experimental variation are presented in the following sections.

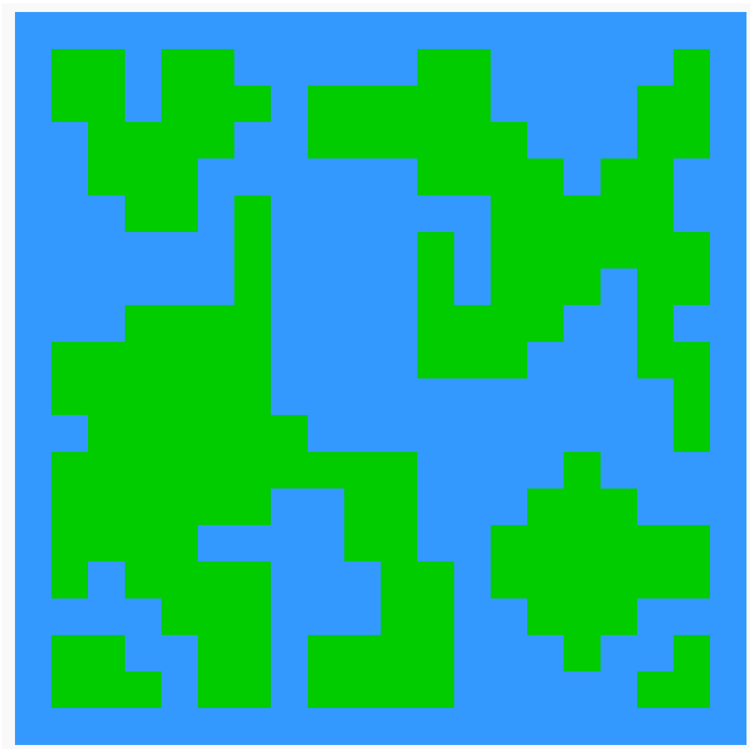

Figure 12: Experiment 3 Initial Map

Experiment 3: Variation 1

In this variation, forest and sand terrain base probability rates and multipliers were moderately set. Fig. 13 below shows the final map after a simulation duration of 20 time units.

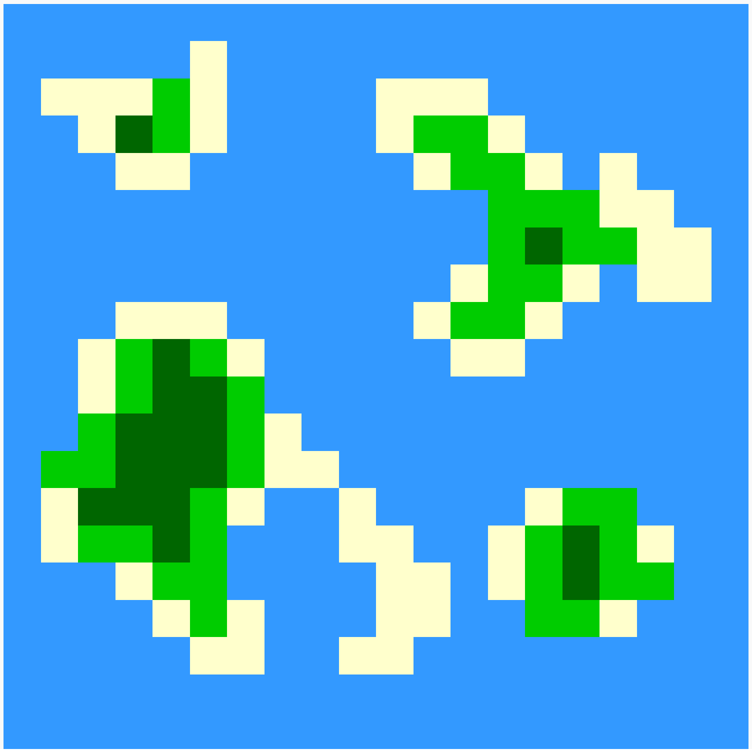

Figure 13: Experiment 3 Variation 1 Final Map

Please see the following files in the Map Generation repository for additional information:

- Configuration File: `/config/ex03/ex03_v01_mapgen_config.json`
- Visualization File: `/config/ex03/ex03_v01_mapgenVisualization_config.json`
- Output Log: `/example-output/ex03/ex03_v01_mapgen_grid_log.csv`
- Initial Map: `/example-output/ex03/ex03_mapgen_initial.png`
- Final Map: `/example-output/ex03/ex03_v01_mapgen_end.png`
- Simulation Video: `/example-output/ex03_v01_mapgen_video.webm`

Experiment 3: Variation 2
In this variation, forest and sand terrain base probability rates and multipliers were set to increase forest formation and decrease sand formation. Fig. 14 below shows the final map after a simulation duration of 20 time units.

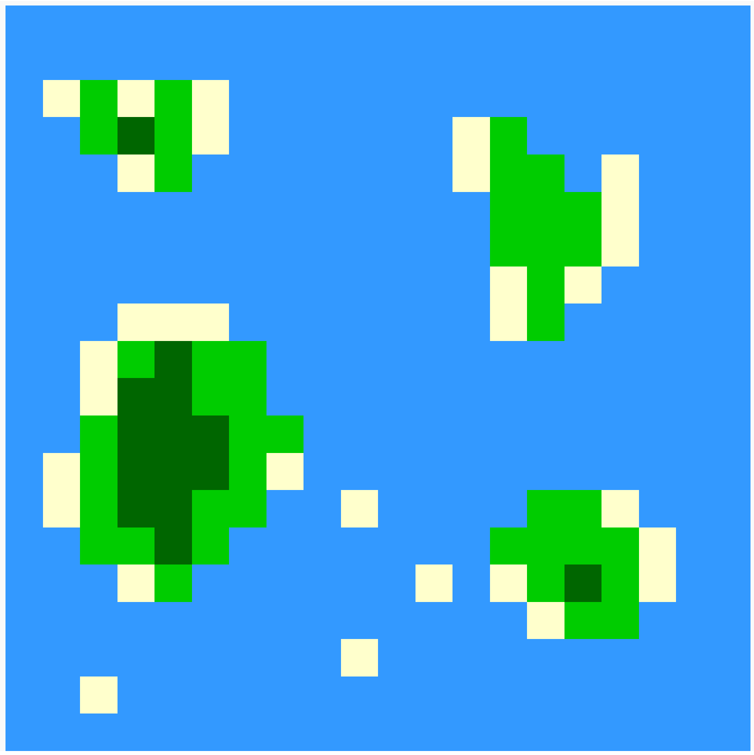

Figure 14: Experiment 3 Variation 2 Final Map

Please see the following files in the Map Generation repository for additional information:

- Configuration File: `/config/ex03/ex03_v02_mapgen_config.json`
- Visualization File: `/config/ex03/ex03_v02_mapgenVisualization_config.json`
- Output Log: `/example-output/ex03/ex03_v02_mapgen_grid_log.csv`
- Initial Map: `/example-output/ex03/ex03_mapgen_initial.png`
- Final Map: `/example-output/ex03/ex03_v02_mapgen_end.png`
- Simulation Video: `/example-output/ex03_v02_mapgen_video.webm`

Experiment 3: Variation 3
In this variation, forest and sand terrain base probability rates and multipliers were set to decrease forest formation and increase sand formation. Fig. 15 below shows the final map after a simulation duration of 20 time units.

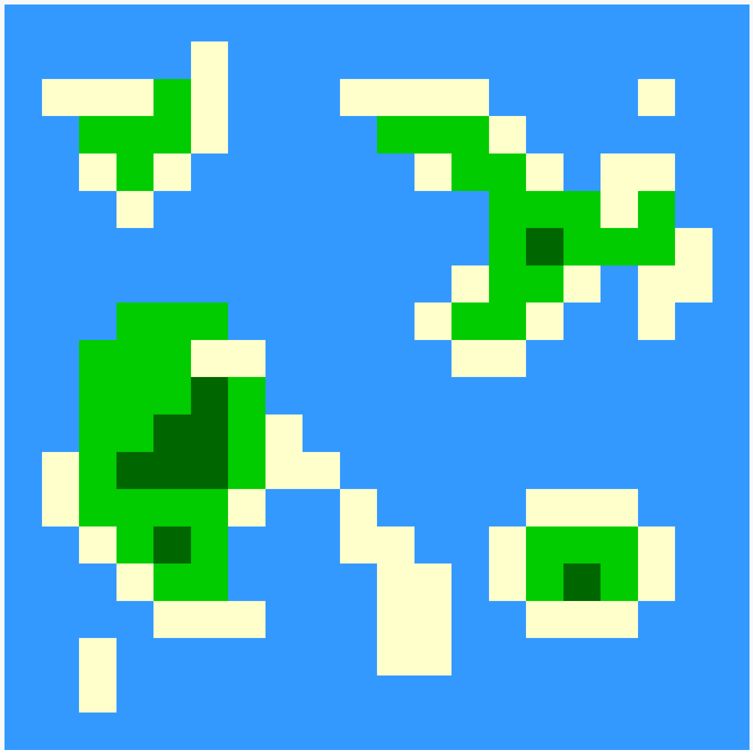

Figure 15: Experiment 3 Variation 3 Final Map

Please see the following files in the Map Generation repository for additional information:

- Configuration File: `/config/ex03/ex03_v03_mapgen_config.json`
- Visualization File: `/config/ex03/ex03_v03_mapgenVisualization_config.json`
- Output Log: `/example-output/ex03/ex03_v03_mapgen_grid_log.csv`
- Initial Map: `/example-output/ex03/ex03_mapgen_initial.png`
- Final Map: `/example-output/ex03/ex03_v03_mapgen_end.png`
- Simulation Video: `/example-output/ex03_v03_mapgen_video.webm`

2.4 Analysis

The experimental results of the simulations presented above agree with the expectations of the formally defined model, and also present opportunities to observe certain emergent behaviour engendered by the interaction of the cells within the model.

The initial map configuration of each experiment is a dominant factor in determining the shape and terrain composition of the resultant map after simulation. While some minor changes to water and land mass shapes along coastal areas can be noted between initial and final maps, final maps generally resemble initial maps in terms of the shapes of their land masses and bodies of water. In coastal areas of sparse land with mostly water neighbourhoods, such as in the islands at sea of experiment 3 or along the eastern border of experiment 2, land and water are more likely to change.

From the different experiments, it can also be seen that different initial maps favour particular types of terrain. In experiment 2, which is predominantly land terrain with smaller enclosed lakes, land and forest are much more prevalent than sand terrain due to the reduced coastline and expansive interior. As the coastlines are enclosed, coastal cells also tend to have less water neighbours, and therefore sand formation is decreased due to lower probability multipliers. In contrast, the islands of experiment 3 have much more relative coastline, and sand formation is additionally encouraged by increased water neighbours. Since there is little interior space on each island in experiment 3, interior forest growth is highly discouraged. In experiments 1 and 2, where there is much interior land, forest growth dominates the land masses.

From the experimental variations, it can also be seen that modifying the base probability rate and multipliers of forest and sand terrains has a significant effect on terrain differentiation. When forest formation is encouraged (variation 2), final maps show increased forest growth in land interiors; when it is discouraged (variation 3), forests are sparse and general land terrain is more pervasive. Similarly, when sand growth is encouraged (variation 3), coastal regions are dominated by sand terrain; when discouraged (variation 2), coastal areas remain as general land terrain.

Given the results of the simulations above, Cell-DEVS is a very useful methodology when applied to the field of PCG and the task of map generation. Although the models and experiments presented here are trivial, more complex models may be designed and implemented to generate more complex game maps. Even beyond the realm of video games, Cell-DEVS may find application in other PCG or environmental technology and tooling, as it is capable of modeling and simulating real-world as well as game-world terrain formation.

References

1. Wu, Zhixuan, et al. "Procedural Game Map Generation Using Multi-Leveled Cellular Automata by Machine Learning." *Proceedings of the 2nd International Symposium on Artificial Intelligence for Medicine Sciences*, ACM, 2021, pp. 168–72, https://doi.org/10.1145/3500931.3500962.

2. Wainer, Gabriel A. *Discrete-Event Modeling and Simulation: A Practitioner's Approach.* Boca Raton: CRC Press, 2009. Print.

3. Cook, M. "Generate random cave levels using cellular automata," *Envato Tuts+*. [Online]. Available: https://code.tutsplus.com/generate-random-cave-levels-using-cellular-automata–gamedev-9664t. Accessed: Mar. 21, 2025.

4. M. Gardner, "The fantastic combinations of John Conway's new solitaire game 'life'," *Mathematical Games, Scientific American,* vol. 223, no. 4, pp. 120–123, Oct. 1970. [Online]. Available: doi: 10.1038/scientificamerican1070-120. Accessed: Mar. 21, 2025.

Appendix A: Cell-DEVS Model Data Form

Title: Map Generation

Type: Cell-DEVS Model

Acronym/Short name: Map Generation

Purpose for which Developed:

To apply Cell-DEVS techniques in the procedural generation of a game map.

Other Applications for which it is Suitable:

This model and simulation may serve as a basis for other map generation techniques.

Date Developed/Implemented: March 24, 2025.

Domain: Procedural Content Generation

Current Version: 1.0.0

URL: Map Generation

Description: Tool for procedurally generating a game map with differentiated terrain according to the Cell-DEVS formalism and implemented in Cadmium.

Links to Related Documents: None

Keywords: Cell-DEVS, procedural content generation, PCG, video game, map, terrain

Developer:

Name: Michael De Santis	**Student Number**: 101213450
Address 1: —	**e-mail:** michaeldesantis@cmail.carleton.ca
Address 2: —	
City: Ottawa	**Province/State-Country:** Ontario, Canada
Zip: —	**Phone:** —