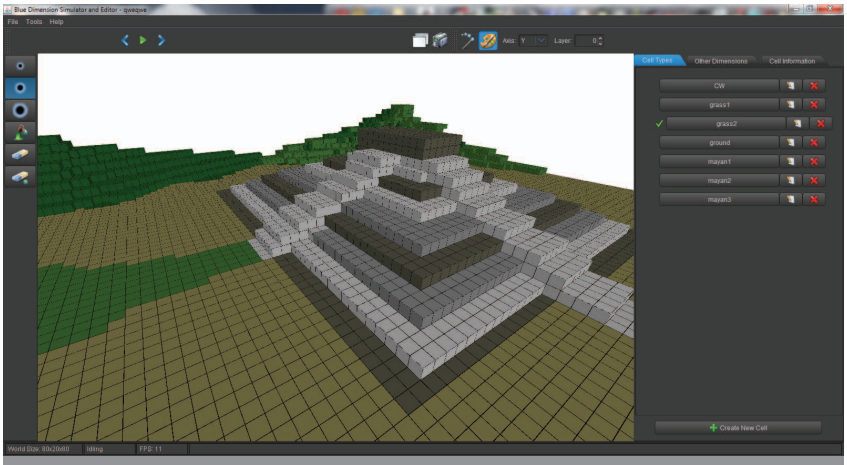
Blue Dimension: A new software editor and simulator for cellular automata

Rishal Hurbans

A cellular automaton (CA) is a collection of cells consisting of at least one definite value (or internal state). All cells collectively constitute a world. The cells are classified in types, whereby each cell type is associated with various transition rules that can be applied to the instance cells of that type. These rules are fixed (per cell type) and are applied consecutively to complete a generation of cells throughout a world.

The rules are applied using the cell's current value, as well as the cell's neighbour's current values of the same generation to compute the new value for the next generation.

addressed in one generation of rule processing; therefore the addressing of cells and neighbours must be speedy to lower the latency before the next generation is available for display.



The Blue Dimension simulator and editor cell types.

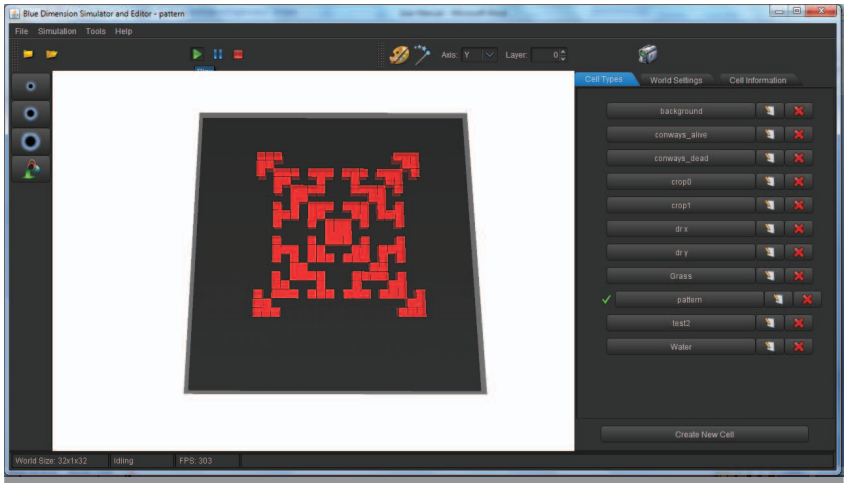
On the basis of such rather simple low-level rules on a cellular automaton world, one may observe complex and useful behaviour on a global scale.

The Blue Dimension software system

The Blue Dimension cellular automata simulator and editor had the following main requirements:

- The program should support n-dimensional cellular automaton worlds. It should not only render worlds visually in two or three dimensions, but also use other means (such as sound) to represent higher dimensions that cannot be visualised.
- The program should support heterogeneous worlds: the cells in the world do not need to be of the same type.
- The cells should be addressed efficiently: the cellular automaton worlds that are to be created with the application are potentially large. Millions of cells are
- The program should allow the user to create and edit cell types, author and edit rules, and create and edit cellular automaton worlds by placing cells of various types anywhere in the artificial world.
- The program should allow the user to simulate the cellular automaton in time, based on the rules and initial cell states. A simulation must be able to run continuously until the evolution stagnates or until there is a user-triggered interrupt. Slow step-by-step simulation, which pauses after each new generation, should also be possible.
- The program should allow the user to create cell transition rules in a simple and intuitive manner.

The Blue Dimension CA software supports a five-dimensional world, of which three dimensions can be seen on screen. A fourth dimension is introduced by a concept of 'gravity' to which every cell in the artificial world can be subjected.



The Blue Dimension simulator and editor pattern.

A fifth dimension is simulated using summation or average values of the entire world's joint internal state, and this value can then be translated into audible output with varying tones.

Users may play, pause and take snapshots of the world at any time. Snapshot simulations may be saved and reloaded at a later stage and run from that point. Because cellular automaton runs may have very long durations, the save and reload feature is important from a practical perspective.

A new rule definition language was designed to allow a simpler and more intuitive way for lay users to make cell transition rules. The aim of the rule definition language was to not restrict the user to overly simplistic cell types, but to also allow the user to use traditional, explicit referencing of neighbour cells. Furthermore, the language should be sufficiently intuitive for easy learning and applying the tool.

To represent an evolving cellular automaton world efficiently (at high speed), a particular voxel engine

```
if  NOT  SUM of all  n              >          x  AND/OR
         AVG         [n1, n2, ...]  <          x  AND/OR
         MIN                        >=         x  AND/OR
         MAX                        <=         x  AND/OR
         COUNT                      between    x, y
         EXACT                      ==

then self   +- SUM of all n                or CONST
            -= AVG        [n1, n2, ...]
            ?- MIN
            /- MAX
            -  COUNT
               EXACT
```

Snippet of the definition language.

The software supports a five-dimensional world, of which three dimensions can be seen on screen. A new rule definition language was designed to allow a more intuitive way for lay users to make cell transition rules.

was created. Essentially, the representation engine does OpenGL matrix manipulation, which moved processing of the view from the computer's central processing unit (CPU) to the graphics processing unit (GPU).

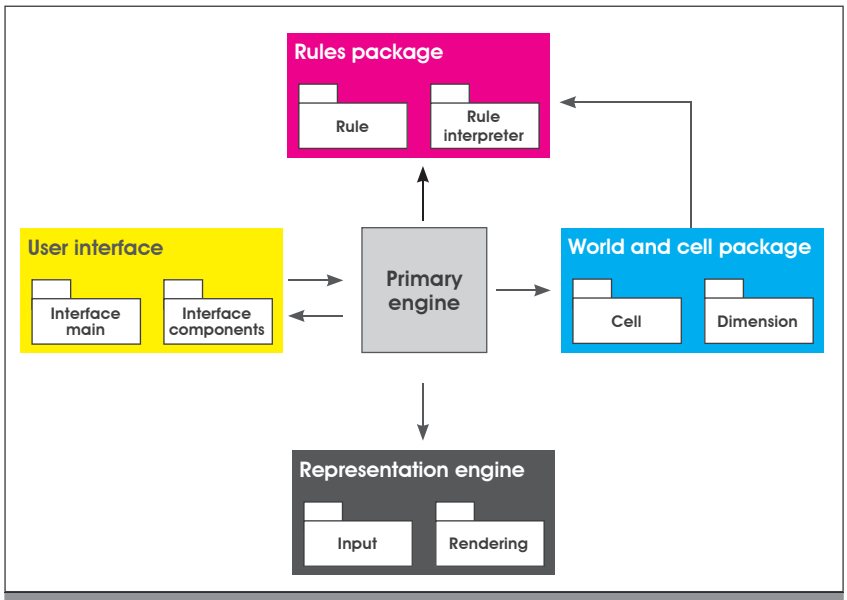
Software architecture

The software architecture of the system is a variation of the MVC pattern, which separates the model (M), the view (V) and the control (C) parts of the software system. For M, extensible markup language (XML) files were used, representing a database schema. The controller's main purpose is to run the simulations step by step, whereas the view part consists of the graphical user interface. All components of the system are only loosely coupled for the sake of software maintenance and adaptability.

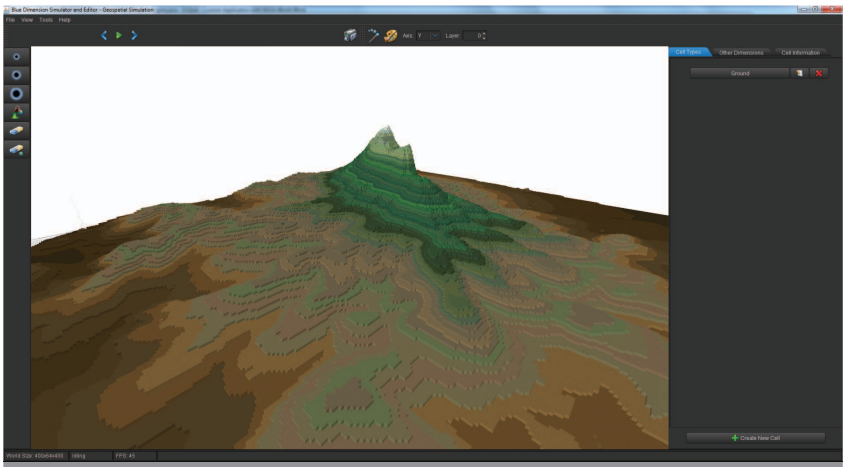
Applications

The intention with creating the custom applications was to show that the theory of cellular automata has many useful applications, both scientific and otherwise. The implementation of these applications also demonstrates the ease with which external libraries and subsystems can be plugged into the primary cell transition engine.

The National Aeronautics and Space Administration (NASA) World Wind is such an example. This software includes maps of the entire planet in detail with regard to elevation and other useful data related to the land and sea. These features were harnessed and combined with the concepts of cellular automata, using the new tool. Users are thus able to select portions of the NASA World Wind map and create detailed three-dimensional CA worlds where rules can be applied to simulate scenarios such as floods and fires.



A graphic representation of the software architecture.



The NASA World Wind map software.

The CA program for shape detection reaches into the field of artificial intelligence. The concept of artificial neural networks was applied to the theory of cellular automata. The behaviour of artificial neural networks relies on communication among neighbouring cells. This is also the case (although in a somewhat different manner) for cellular automata. An attempt was made to join these two concepts. It allowed for the construction of a CA program that was able to detect and follow a specific coloured object. This was achieved by capturing frames from a web camera as images, transforming the images to two-dimensional cellular automaton worlds and, finally, applying the cellular neural network rules and mechanisms. The same is done for every frame of the camera. After much training and learning, the program was able to detect the movement of a red ball across the screen in any direction.

Rishal Hurbans *is a computer science student working towards his honours degree in the University of Pretoria's Department of Computer Science. His ongoing project on cellular automata is supervised by Prof Stefan Gruner.*