Scenario Generation with Cellular Automaton for Game-based Crisis Simulation System

Pisit Praiwattana
Department of Computer Science
Liverpool John Moores University
Byrom Street, L3 3AF, Liverpool UK
P.Praiwattana@2014.ljmu.ac.uk

Abdennour El Rhalibi
Department of Computer Science
Liverpool John Moores University
Byrom Street, L3 3AF, Liverpool UK
a.elrhalibi@ljmu.ac.uk

***Abstract*— Crisis is infrequent and unpredictable event which is challenging to prepare and resolve. Scenario generation for modeling a crisis using computational approach is preferable due to its flexibility to produce automated variation of scenario based on different setup key factors. Combining crisis generation with game technology, serious-game game can be developed to provide potential support in training and simulation practice of real-world crisis situation to different stakeholders. Therefore, in this paper, we propose a computational framework to provide automated generation of a fire crisis scenario with firefighter-agent simulation using cellular automaton with influence map for agent behavior optimization, and visualization using a 3D game engine. The framework is evaluated based on performance, convergence and realistic behavior of both crisis event (e.g. fire propagation or control) and crisis resolution (e.g. firefighter solving the crisis).**

Keywords*— Crisis Scenario Generation; Cellular Automaton; Influence Map, Planning, Game Engine, Agent, MAS.

I. Introduction

Scenario can be described with hypothetical situation which is providing a necessary environment setup, initial background and a final goal to represent a sense of real-world situation. For Crisis management organization, the management of resource available and facilities to deal with crisis disaster is a challenging task since these resources change over time in different situation. To be prepared and enable understanding of crisis situation under specific environment setup, the simulation of crisis with agent based modeling is a common approach to produce a potential study result. With game engine technology such as Unity3D or Unreal Engine UE4, the visualization of crisis can bring forth more interaction and training of difficult operation during crisis plan and become a practical reference for the actual deployment and crisis resolution.

However, there is still an issue on how to improve a process of delivering a plausible crisis scenario since it is often time-consuming, with limited variations and lacking of a concrete evaluation of scenario quality. A crisis scenario generation framework has been proposed in aiming to consolidate a supporting technique and tool for the delivering of plausible crisis scenario, simulation of crisis plan with multi-agent model on the JADE framework and visualization of result using a game engine [1].

In this paper, we present the design and implementation of one of the scenario generation modules that applies the technique of cellular automaton (CA) to the domain for the proposing framework. The system simulates a behavior of natural crisis such as wild fire with an input of abstract representation of different terrain types with an influence of fuel capacity. It also includes a simulation of crisis agent deployment such as firefighters to resolve the emergency event. Furthermore, the sequence of simulation results is being exported and visualized by a Unity3D or Unreal Engine UE4 to validate and observe the effect of applying game engine to inform the end-user. The remainder of the paper is organized as follows: In section 2, we briefly introduce some related work on existing automated scenario generation systems and cellular automaton; in section 3, we review the design of our proposed framework; in section 4, we present the implementation of applying CA into our scenario generation module; in section 5, we discuss the result; and finally, in section 6, we conclude the paper.

II. Related Work

A. Automated Scenario generation

Computer generated scenario for crisis domain is initially focused on producing a sequence of reasonable events that is considering on: (1) resource management perspective; (2) stakeholders collaboration in solving the situation. With procedural content generation technique, it can faster simulate the scenario setup and constraints over domain-expert manual processing. This approach is not designed to replace the necessity of domain-expert but being used as a supplement tool which applied domain knowledge to deliver a variety of plausible scenarios and ease of customization. Several researches based on planning or genetic algorithms have been carried out and proposed solutions to solve some of these issues.

Hullett and Mateas [2] apply a planning technique to generate a firefighter rescue training scenario in the collapsing building area. The system uses HTNPlanner with building structure data as input while setting a goal to create a situation that satisfies crisis final description. The domain knowledge must be defined for the planner to allow physical consistency and achieving better plausible result.

Martin et al. [3] proposed an automated scenario generation system which aims to be generic and applicable to any domain specific simulation system. The authors introduced a conceptual mapping approach which is based on 1) training objectives; 2) baseline scenario as a pre-defined ideal parameter scenario; and 3) scenario vignettes that is a complexity modification of scenario such as weather, light, and etc. The main objective and additional data such as weather condition (vignette) setting are composed in an XML file and generate the scenario variation using scripted functional L-systems which are similar to shape grammar in procedural modelling but define the syntax rule to represent scenario elements.

Zook et al. [4] introduce a combinatorial optimization approach to scenario generation to deliver the requisite diversity and quality of scenarios while tailoring the scenarios to a particular learner's needs and abilities for military training in virtual environment which is opposite to the planning approach. The main scenario generation is based on a genetic algorithm to search for a best solution; reading in author-specified domain knowledge, the details and the type of possible events and requirement in scenario, and constraint order on events. The process works by considering instant event template for the scenario at random location; mutating the parameter of randomly chosen scenario; and applying cross-over operation, to create new sequence of events to improve the quality of scenario for the next iteration.

Further research focusing on developing a model for generating and simulating the propagation, possibly with cascading effects, have been carried out using powerful computational model such as cellular automaton (CA).

B. Cellular Automaton

Cellular Automaton (CA) is an example of mathematical model first introduced by von Neumann [5], which is describing a behavior of each component in a set having a representation of state using simple rule to control action selection such as destruction and replication. Together within each component functioning on the same rule, the complex pattern or behavior can be observed, thus, it allows modeling of the living ecology and general principle for complex system in nature (Wolfram, 1994) [6]. The most basic CA model implemented in 2D lattice is Conway's Game of Life [7] in which each cell has 2 states and by calculating the total eight neighbor cells, four adjacent orthogonally and for adjacent diagonally, new state can be derived from applying the sum of alive cells using a local rule to governing a death, alive and reproduce behavior. The following is the base rules for Game of Life:

- Survivals. Every counter with two or three neighboring counters survives for the next generation.
- Deaths. Each counter with four or more neighbors dies (is removed) from overpopulation. Every counter with one neighbor or none dies from isolation.
- Births. Each empty cell adjacent to exactly three neighbors--no more, no fewer--is a birth cell. A counter is placed on it at the next move.

From this example, few rules can represent a dynamic evolution system of population of each cell and it is very powerful to simulate a phenomenon whereby its behavior is depending on the interacting of nearby different type of state and basic type of transition function such as fluid, pollution, fire and epidemic outbreak.

There are several applications of applying CA to simulate a model of dynamic crisis phenomenon such as wild fire, lava flow, flooding, and freeway traffic. Other applications include gas phenomena, urban development, immunological processes, and crystallization. More recent applications involve multimedia content generation, cryptography and implementing parallel computer.

Lava flow simulation model SCIARA [8] implements a cellular automata technique with applying a rule to introduce a movement of landslide lava path comparing to the real lava eruption at Etnean eruption in 1991-1993. The model aims to simulate the complexity of phenomenon by describing it in a local transformation of each lava cell using some local cell property such as height, temperature, inflow and outflow calculation. In addition, the initial setup of global parameter is also necessary to ensure flexibility in configuration of specific event i.e. temperature at vent, CA's clock, Cell side length and physical characteristic such as cooling rate and lava adherence parameter. By comparing the simulation result corresponding to real event data, the hexagon representation is more suitable to better approximate the flow path.

André Sabino and Armanda Rodrigues [9] introduced implementation of the flooding threat in dam break emergency management scenario with CA to apply constraints for testing and validating agent operation in the simulation. The model converts the flooding area into spatial representation along with different environment detail for supporting agent decision making process.

The wild-fire spread [10, 11] is another example of extensively applying CA to the aspect of modeling fire disaster. They explain the incorporated of factors affecting a forest fire which are: weather condition; wind direction and speeds; the topography of landscape; and the area property resulting in different fire spreading rates. These factors will allow CA to produce variety of different hypothetical scenario on the environment which is forest in this work. For the proposed model, the simulation still lack an involvement of infrastructure geography representation to further describe the effect of the wild-fire to this type of scenario.

Some research to model and simulate road traffic using CA has been carried out. CA based traffic flow simulations create speed update as a function of location on the street network and obstacle in the adjacent cells. These models can be categorized as microscopic (i.e. as opposed to macroscopic model) and are similar to car following systems, as they take into account only the cell nearby the cell being processed. A very common technique known as the Nagel-Schreckenberg model (i.e the NS model) [12] and centered on the Rule 184 [13] belongs to the class of particle hopping systems. Rule 184 is central to traffic flow modelling with CA. The NS model can be seen as an enhancement of this technique with additional properties such as a discrete speed for vehicle, and an arbitrary propensity to reduce the vehicle speed [14]. These simulations use integer state variables to represent the dynamic features of the system. Traffic flow modelling and simulation is a vast complex topic and the choice of the model to use can be challenging. CA is used as it can produce a large variety of global traffic behaviors using local rules. Each cell of the CA can return individual features and slight alterations in the rules or the cells state can generate important changes [15].

III. The CA model for crisis scenario generation

A. *Methodological approach*

Fundamentally, major classic model is based on classic elemental CA which the basic property is described as: a number of spatial dimension; width and height of each side in array; a width for neighborhood model in each side; cell states; and, finally, a local rule which can be described as a function R. The involvement of global variable is as mentioned in previous section a necessary step to achieve the flexibility of scenario configuration. At least the physical model of desired event must be parameterized for producing realistic result. From the basic CA behavior, the most often used model is 2D square-grid representation. However, for crisis scenario, the rule design requires not to produce an unrealistic behavior especially an oscillating between pattern stages in Figure 1 which is commonly observed in Game of Life example. However, the pattern that generate stable or movement path of the cell state transition is still applicable since it is the aim of generating natural phenomenon of water, fire, lava and other related type. Figure 2 display a common example of stable pattern in game of life which single cell and 4-cells square won't change its shape over any number of consecutive iterations.

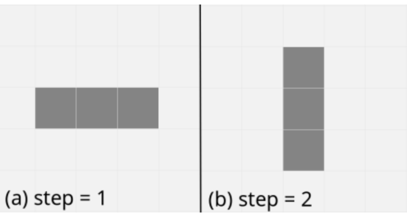


Fig. 1. Cell oscillating bettwen 2 patterns (a) and (b) in consecutive steps

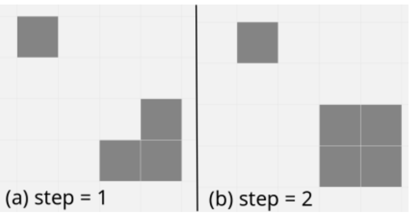


Fig. 2. Cell stable patterns (a) starting alive cells (b) stable patterns

In addition, the involvement of agent model is a necessary stage to produce valuable study case that it evaluates the future prediction of situation outcome based on deployment pattern, capability of agent actions and optimization of agent plan which reflected the real world crisis management preparedness. Agent contains internal state and is able to sense an environment in the grid lattice while assuming the role which has a defined condition on what it is capable to do action influencing the result. As a result, aside from crisis event simulation, the optimization of agent procedure can be observed and designed then recorded for reference in crisis scenario generation process.

B. *The model of Crisis Scenario CA for Fire event*

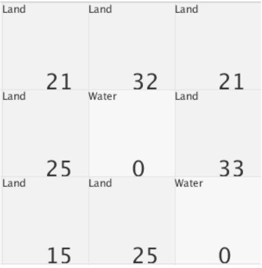
The crisis scenario model using cellular automaton for fire event is a system that adopts the elementary CA rule to simulate the interaction of fire behavior with agent based model to study and measure the possible outcome. While the rules is required for abstraction the physical model of desired phenomenon, over parameterized the model will increase the processing complexity exponentially. The preprocessing of real world data may be required to simplify the representation in CA model. As for state representation, the definition of substate introduces a better description of each individual cell property which is relevant to the evolution process. In our approach, we initially defined an environment with 2D square lattice which each cell's state as S = {Normal, On-fire, Burnt-down} to capture the basic representation of fire situation. The substate variables are also introduced for better description of infrastructure property such as a geological cell type = {Land Terrain, Water Terrain, …, Other Substate}. Finally, we represent a concept of fuel capacity, burning rate, and accumulated heat in the cell to allow calculation of live-step duration and trigger condition of On-fire state without external action influenced from an agent. Thus, each cell will have a description as:

Cell{x, y} = < Type, State, Fc, Br, H>

Where:

- *x* and *y* represents coordinates of identical cell in grids
- *Type* indicates the infrastructure property of cell
- *State* identifies the current situation such that the cell is currently on a burning state or already burnt down
- *Fc* represents the calculated capacity of fuel
- *Br* represents the burning rate of fuel for each cell
- *H* represents accumulated Heat for state change

Calculating a local interaction between cells in 2D grids, we assume that it is corresponding to the cell's neighbor influence. For example, the coordinate {x, y} will have 8 neighbors as *Neighbor(x, y) = {(x-1, y +1), (x, y+1), (x+1, y+1), (x-1, y), (x+1, y), (x-1, y-1), (x, y-1), (x+1, y-1)}*



Land 21	Land 32	Land 21
Land 25	Water 0	Land 33
Land 15	Land 25	Water 0

Fig. 3. Show the fuel capacity calculated by considerating the geolographical type of each cell.

Moreover, we can include external infrastructure information to be mapped and preprocessed into substate property. Depending on the preprocessing formula we initially interpret cell's substate such as geological type to gradually calculate the fuel capacity of each cell. Figure 3 shows an example where the water lake area will have fuel representation as zero as a constraint to not allow fire being spreading on water area.

At each simulation step, rules for state transition will be applied to calculate the next state of each cell according to the previous mentioned parameters:

0. **IF** the cell {*x, y*}'s type is *Land Terrain*:
continue the rule:

1. **IF** the cell {*x, y*} state is *On-fire* **AND** *Fc* > 0
THEN *Fc = Fc – Br, H+= Ongoing Fire rate*

2. **IF** *Fc* is <= 0,
THEN the state is updated to *Burnt-down*.

3. **IF** cell {x. y}'s Fc > 0.
Let N = Count On-fire state from *Neighbor*(x, y).

4. **IF** N > 2 **AND** N < 5
THEN *H = H* + N**Heat TransferRate*

5. **IF** *H > Flame Threshold*
THEN the state is changed to *On-fire*.
ELSE the state is remaining the same.

6. **IF** state is *Normal* or *Burnt-down and unchanged*
THEN *H = H – Cooling down rate*

With these simple rules: the behaviour of fire is easily simulated while it is reflecting on the concept of burning consumption rate, fuel capacity, local relation of heat transfer. The possible addition can include creation of sophisticated movement of fire direction based on how the fuel is plotted in the grid. With manipulation of variable in rule number 4, the velocity of fire propagation can be designed.

Moreover, the existence of agent, which is a firefighter in this model, is similar to an entity existing on valid cell *{x, y}*.

For each Firefighter{*x, y*} = < *State, Health Point (hp)* >, agent will select a suitable action corresponding to its related position in each simulation step. The following are agent's action description:

Extinguish (x, y): He can extinguish any fire within 1 unit distance to any *Neighbor(x, y)* if destination cell's state is *On-fire*. This is the signature action that represents possible interaction between firefighter and fire scenario. The result of action is to reduce accumulated heat in the cell.

MoveTo (x, y): He can move 1 unit distance to any *Neighbor(x, y)* if destination cell's type is *Land Terrain*, thus *Water Terrain* will not be considered a valid destination and it represents basic obstacle for both fire and firefighter simulation.

Similarly, in each simulation step the rule is also applied for firefighter state transition:

0. **IF** Firefighter{*x, y*} state is *alive*: continue the rule.

1. **Let** N = Count *On-fire* state from *Neighbor(x, y).*
IF N > 4 **THEN** Firefighter{x, y}'s hp minus by 1.

2. **IF** Firefighter{*x, y*}'s hp <= 0
THEN Firefighter{*x, y*}'s state is dead.

With this state transition rule, firefighter decision making should avoid moving into a cell that consist of more than 4 Onfire neighbors since it will lead to his dead. Since the calculation of agent's behaviour and fire simulation is ongoing in each frame, applying A* pathfinding from agent to fire cell may consume too much computation time, the implementation of influence map is a cheaper alternative for optimization of firefighter decision making logic and motion

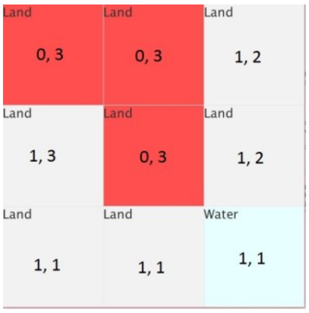
Influence map is a buffer data storing information regarding a specific location it is represented. Its data structure can be varied from spatial square grid to area graph. Map can be calculated for different stakeholder since it will store with various critical information for agent to process and use to make a suitable action. The first step is to set a spatial partition of environment space into corresponding same-size or greater data object, the relationship between each cell is then calculated and being added into the influence variable which can be a single floating number or a vector of parameters. For our application, each cell {x, y} will have an influence {x, y} as the following:

Influence{x, y} = < D, Nf, > Where

- D specifies distance from the current cell to nearest Cell which is On-fire
- Nf is count of Neighbor(x, y) which is On-fire including itself.

Agent will take advantage of this information in their pathfinding so it move to a cell with less distance to fire and with high count of On-fire state resulting in efficient movement to extinguish the most cells that are on fire. We also design that agent may avoid a cell that has too high Nf since it may likely to cause agent to get injured. This design allow more flexibility in agent behavior and ease of additional configuration while represent real-world crisis plan.

Figure 4 illustrates the possible influence value in each cell.



Land 0, 3	Land 0, 3	Land 1, 2
Land 1, 3	Land 0, 3	Land 1, 2
Land 1, 1	Land 1, 1	Water 1, 1

Fig. 4. Influence map on each cell corresponding to its location

We can also control the global behavior of CA and agent by allowing involvement of some global parameters. It can be an initial value of agent's health point, number of maximum firefighter, Burning rate, lethal number of nearby fire to cause agent's injuries, and how much damage it will cause to agent health.

In the following we introduce the implementation of the CA crisis fire generation and simulation framework.

IV. Implementation

A. *Scenario Generation CA Framework Architecture*

The simulation of CA and agent behavior has been implemented using Processing as a 2D visualization framework in Java. The states of each cell for specified number of simulation steps with agent can be exported into an output file which can then be imported into Unity3D or Unreal Engine UE4 Game engine for visualization. Figure 5 displays the overall architecture and Figure 6 illustrates class diagram of scenario generation CA on processing.

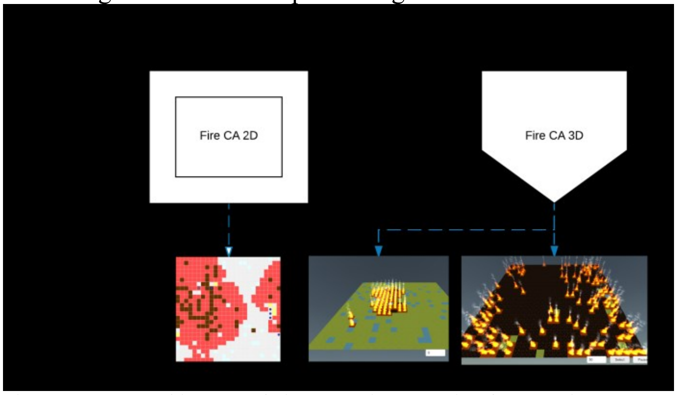


Fig. 5. System architecture of Fire Scenario generation framework

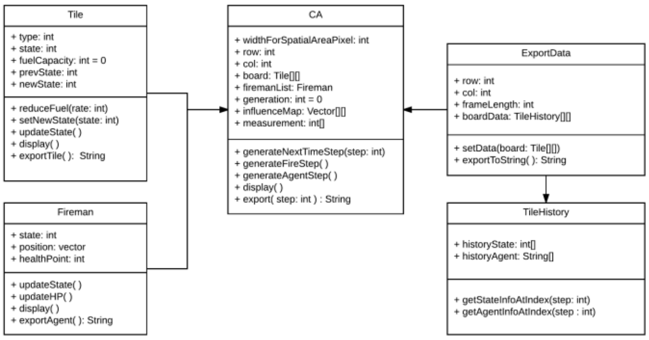


Fig. 6. Class diagram of Fire cellular automata in processing

V. Simulation and Experiment Result

This section explains the simulation results with different setup on a scene of fixed size world. The execution times reported are run in a Laptop with Core i7-5500U @2.40GHz, 2 Cores with RAM 16GB (10GB Available), and Window 10. The environment is divided into 25x25 square tiles representing 625 units of area corresponding to geographical terrain. We have designed it to have only Land or Water terrain for current simulation. The experiments will capture the result of 3 different cases while measuring average statistic variable over same initial setup.

The cells color states are defined below:

⬜	🟦	🟥	🟥	🟦
land	**water**	**fire**	**burnt**	**firefighter**

Fig. 7. Color reference in 2D representation of CA simulation result

Case 1: The fire is spreading in circular pattern until it is out of fuel, and the corresponding CA converges to a stable state after 291 generation, where the majority of the cells are burnt out.

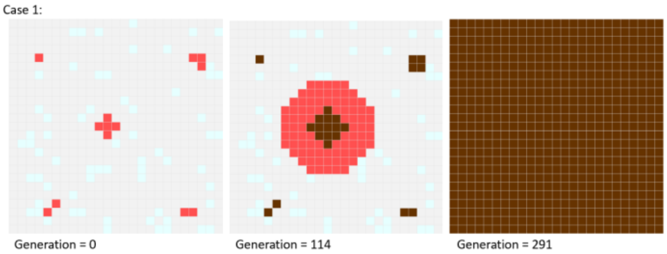


Fig. 8. Basic fire spreading simulation without agents interaction: FPS = 60, Total generation is 291, Burnt-down area is 625 units

Case2: The fire is setup in under perimeter of firefighters. The test result is illustrated in Figure 9 as the case is resolved within only a few simulation steps.

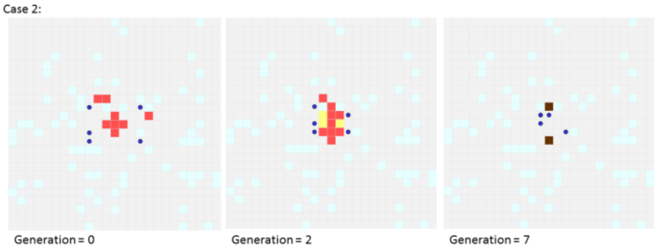


Fig. 9. Basic fire setup surrounded by sufficient firefighters: FPS = 60, Total generation is 7, Burnt-down area is 2 units, Extinguish Actioned is 24 times, Fireman = 5, Alive = 5

Case3: Initial fire location is separated into 2 areas and the number of firefighter assignment is different for both locations. Figure 10 displays one of variation from total 3 runs. From the generation 79, we can observe the disappearance of fire in lower right corner due to the sufficient of firefighters while the fire is getting out of control in upper left corner.

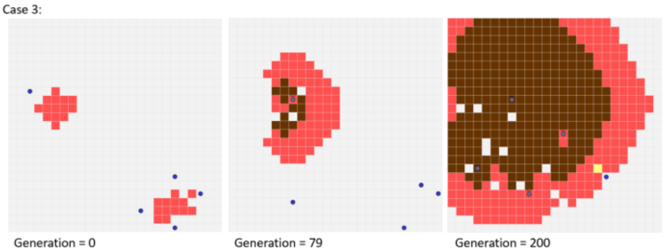


Fig. 10. Separated fire setup with different agent distribution: FPS = 60, Total generation is 200, Burnt-down area is 294 units, Extinguish Actioned is 164 times, Fireman = 6, Alive = 1 at generation 200

In case 2 and 3, the agents acquired information regarding a cell with most fire neighbors including itself from an influence map. Figure 11 displays a normalized visualization mapping between fire CA and its corresponding influence map. Black spot indicates that the cell has less or close to zero influence while white cell is important and required immediate action.

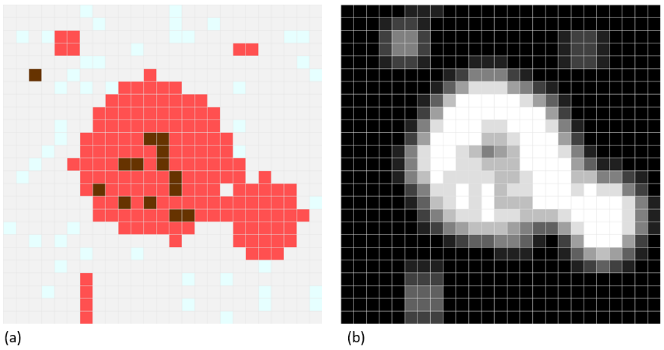


Fig. 11. (a) Simulation on 2D Processing (b) Influence map of each cell

Next, the simulation data is being exported into simple text file format including a history of each generation in CA cells for game engine such as Unity3D can read then render 3D event representation. For now, we currently create a direct mapping between 2D version and 3D to generate an event player. Figure 11 illustrates an example of rendering a simulation result in Unity3D with basic particle effect.

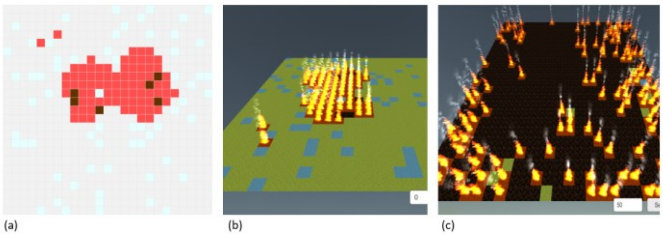


Fig. 12. (a) simulation in processing, (b) and (c) Render result in Unity3D

VI. Conclusion

In this paper, we introduce the design and implementation of crisis scenarios generation using cellular automaton (CA), for fire crisis event creation, simulation and resolution. The system simulates a behavior of natural crisis such as wild fire with an input of abstract representation of different terrain types with an influence of fuel capacity. It also includes a simulation of crisis agent deployment such as firefighters to resolve the emergency event. Furthermore, the sequence of simulation results is being exported and visualized using a game engine.

The simulation results of fire and firefighter behavior in our crisis CA framework have been implemented and show that the framework is an efficient tool to support the modelling and simulation of fire crisis situation involving a variety of situations and possible outcomes, in term of the fire propagation, the success of the firefighters to control the fire, and the risks involved to the firefighter health and life in case there is no enough resources allocated to deal with the crisis event.. It provides several aspects to evaluate the outcome and test crisis personal deployment in simple representation, with the possibility to visualize the outcomes in a 3D environment using a game engine.

In the future work, we will develop different game scenarios using different visualization techniques with game development technology, for better realism of the crisis situation over time. In addition, interface for customization CA simulation parameters such as adding more type of infrastructure, add-and-remove rules, adjusting simulation formula for application different than fire, and finally adjusting different behavior of agent, to include long-term planning, is crucial to deliver better crisis scenario generation process.

References

[1] P. Pisit and E.A, “Survey: development and analysis of a games-based crisis scenario generation system,”, Edutainment Conference, Hangzhou, PRC, 15-17 April 2016, Book Published by LNCS Springer Verlag: E-Learning and Games, 10th International Conference, Edutainment 2016, Hangzhou, China, April 14-16, 2016. Editors: El Rhalibi, A., Tian, F., Pan, Z., Liu, B. (Eds.).

[2] Hullett, K., Mateas, M. (2009). “Scenario generation for emergency rescue training games”. *4th International Conference on Foundations of Digital Games* - FDG '09.

[3] Martin, G., Schatz, S., Bowers, C., Hughes, C., Fowlkes, J., Nicholson, D. (2009). “Automatic Scenario Generation through Procedural Modeling for Scenario-Based Training.” T*he Human Factors and Ergonomics Society Annual Meeting*. 53, 1949-1953.

[4] Zook, A., Riedl, M. O., Holden, H. K., Sottilare, R. A., and Brawner, K. W. (2012). “Automated scenario generation: Toward tailored and optimized military training in virtual environments.” *7th International Conference on the Foundations of Digital Games*.

[5] von Neumann, John., A. Burks, ed. (1966) “The Theory of Self-reproducing Automata”. *Urbana, IL: Univ. of Illinois Press. ISBN 0-598-37798-0*

[6] S. Wolfram, “Universality and complexity in cellular automata”, Physica D 10 (1984) 1-35.

[7] Bays, Carter. "Introduction To Cellular Automata And Conway’S Game Of Life". *Game of Life Cellular Automata* (2010): 1-7. Web. 28 June 2016.

[8] Crisci, Gino M. et al. "The Simulation Model SCIARA: The 1991 And 2001 Lava Flows At Mount Etna". Journal of Volcanology and Geothermal Research 132.2-3 (2004): 253-267. Web.

[9] Sabino, A., Rodrigues, A. “A visual language for spatially aware agent-based model-ing in crisis scenarios”. *In: Proceedings of the 12th AGILE International Conferenceon Gepgraphic Information Science (2009)*

[10] B. Drossel and F. Schwabl, “Self-organized critical forest-fire model,” Physical Review Letters, vol. 69, no. 11, pp. 1629-1632, Sep. 1992.

[11] I. Karafyllidis, “A model for predicting forest fire spreading using cellular automata,” Ecological Modelling, vol. 99, no. 1, pp. 87-97, Jun. 1997.

[12] K. Nagel and M. Schreckenberg. “A cellular automaton model for freeway traffic.” J. Phys. I France, 2:2221–2229, 1992.

[13] Li, Wentian (1987). "Power spectra of regular languages and cellular automata". Complex Systems 1: 107–130.

[14] C. Gershenson and D.A. Rosenblueth. “Modeling self-organizing traffic lights with elementary cellular automata.” arXiv:0907.1925, 2009.

[15] S.C. Benjamin, N.F. Johnson, and PM Hui. “Cellular automata models of traffic flow along a highway containing a junction.” Journal of Physics A: Mathematical and General, 29:3119–3127, 1996.