Muhammad Salman – 100995527 – SYSC5104F – November 21st, 2021

Assignment 2 – Wildfire Terrain Simulation using CD++

Part 1: Model Selection

This assignment begins with the selection of a potential Cell-DEVS model, which can then be implemented in CD++ and visualized with the provided CD++ visualization webpage. The proposed model must be categorizable under a Cellular Automata model, meaning it can be represented in a cellular space, while also following the DEVS model formalism. The proposed model will be based off the research paper: *Using cellular automate to simulate wildfire propagation and to assist in fire management* [1].

The paper presents a model that was used to simulate a wildfire in Portugal in mid-2012, where approximately 25 000 ha of terrain was burned. Interestingly, the researchers base the probability of fire spreading from one cell to the next, on various factors: such as the type of vegetation, density of vegetation, type of topography, and wind fields. Each one of these factors carries its own probability, and when combined, we receive p_{burn}, the probability of a neighboring cell burning based on these provided factors. A burning cell can spread in eight directions, marking each one of the adjacent cells on a cellular grid (N, E, S, W, NE/NW, SE/SW), also known as a Moore's neighborhood.

The paper proposes some sample probabilities for different terrains based on their likelihood of burning within in the next time step. Once inputted into the burning probability formula below, the probability that any adjacent cells will burn can be calculated. The formula, along with the listed probabilities of fire spread based on vegetation and land density are presented below [1].

$$p_{\text{burn}} = p_0(1 + p_{\text{veg}})(1 + p_{\text{den}})p_{\text{w}}p_{\text{s}}.$$

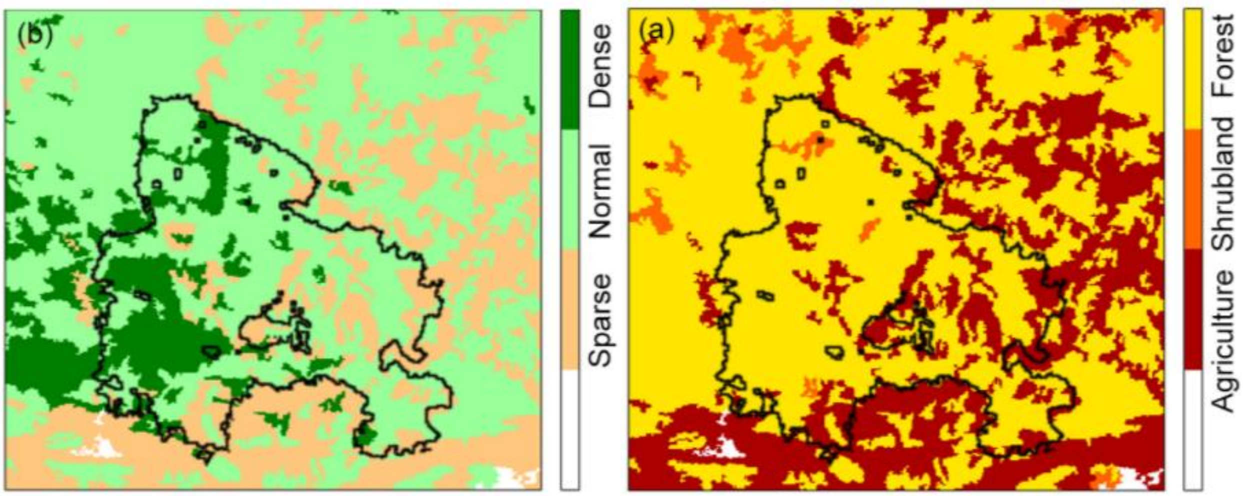
Categories	p_{veg}
No vegetation	-1
Agriculture	-0.4
Forests	0.4
Shrubland	0.4

Categories	p_{dens}
No vegetation	-1
Sparse	-0.3
Normal	0
Dense	0.3

The rules for the state changes for the cellular automata are mentioned below:

- Cells that are unburnable will not be affected by fire (realistically this could be a water or rocky terrain, or terrain lacking vegetation)
- A cell that begins to burn will be completely burnt once one time step is complete
- Cells cannot re-burn, once burnt, the wildfire will not return
- As mentioned before, the fire can spread to other cells based on p_{burn}

The paper provides formulae to additionally calculate the probability of wind (p_w) and the probability of terrain slope (p_s), however these will be predefined per cell for our simulation purposes. For our simulation purposes, we will also propose a predefined cell space with different terrains to simulate wildfire spreading based on probability of different factors. A sample simulation space from the paper can be seen below.



Part 2: Model/State Definitions

When defining our models according to CD++ rules, we can make a table to predict the burning probability. The variable p_0 represents a true or false, representing if a burning cell is in the neighborhood as defined above. As such, we will make a table of states that corresponds to certain combinations of environments. Since probability is difficult to represent in CD++, we will allow p_{burn} to represent the burn rate as well as the strength of the fire. Once the fire hits a certain strength, it will eventually burn out completely and be marked as a fully burnt area that cannot re-burn. The initial states for each layer will be pre-defined in an external file.

The table below shows the various values for each category (1 for steep terrain and windy means true, while 0.5 means not as steep or windy).

Vegetation	Density	Steep Terrain?	Windy?
-1	-1	1	1
-0.4	-0.3	0.5	0.5
0.4	0		
	0.3		

However, according to the research and map shown in the paper, most agricultural land is also sparse, a forest could be dense or normal, while shrublands are mostly normal density. The wind and steep terrain will increase the strength and rate of growth of the fire. This gives us 6 pre-defined environmental states, which will each yield a different p_{burn} from the formula above. Two additional states, for fire and burn out are also added, giving a total of 8 states that will be used in our Cell-DEVS atomic model definition in the next section.

State	Vegetation	Density	Steep Terrain?	Windy?	Comments	Burn Rate
-1	X	X	X	X	-water, cannot burn	0
0	-1	-1	X	X	-no veg. or den., cannot burn	0
1	-0.4	-0.3	0.5	1	-agric. is not steep or dense	0.21
2	0.4	0	1	0.5	-steep forest is not dense	0.7
3	0.4	0.3	0.5	1	-dense forest is not steep	0.91
4	0.4	0	0.5	0.5	-shrubland	0.35
5+	X	X	X	X	-fire begins	
6+	X	X	X	X	-burnout	

When a neighboring cell is on fire next to one of the predefined cells, the rate of burn in terms of time will be accelerated based on the burn rate. The cell's state will change to a fire state based on the burn rate, and the intensity of the fire will grow based on that. After the state is already on fire, it will take one standard time delay for it to burn completely as mentioned earlier, turning black thereafter (this was later made more complex in the final simulation). Once completely burnt, the state will go to 6 (or slightly above such as 6.1 or 6.2), since it cannot be burnt again, or the vegetation has been wiped out. A state that is on fire begins at state 5, and anything larger than that defines the intensity of the fire. There were three simulations that were done, with the third being the most complex. The atomic model formalism below is shown for the third simulation.

CD = < X, Y, I, S, θ, N, d, δ_{int}, δ_{ext}, τ, λ, D >

X = Y = {∅}

S = {s ∈ {0,1,2, 3, 4, [5, 6.2]}}

N= {(-1,-1),(-1,0),(-1,1),(0,-1),(0,0),(0,-1),(1,-1),(1,0),(1,1)}

d (transport) =

60000ms (standard),

0.79*60000 ms if S = 1,

0.3*60000 ms if S = 2,

0.09*60000 ms if S = 3,

0.65*60000 ms if S = 4

*The time delay can be calculated as (1- burn_rate) * 1 minute. The greater the burn rate, the faster the cell will burn.

τ: N → S is defined by the following rules:

cell (0,0) =5.21 if cell (0,0) = 1 and #macro(checkfire)

cell (0,0) =5.7 if cell (0,0) = 2 and #macro(checkfire)

cell (0,0) =5.91 if cell (0,0) = 3 and #macro(checkfire)

cell (0,0) =5.35 if cell (0,0) = 4 and #macro(checkfire)

cell (0,0) = value at (0,0) + 0.2 if cell (0,0)>=5 and (0,0)<6 and #macro(checkfire)

cell (0,0) = value at (0,0) + 0.2 if cell (0,0)>=5 and (0,0)<6

*The new cell value is generated by 5 + burn_rate, where 5 represents the weakest fire strength, while 6 is burnout. If a cell is already on fire and not burnt out yet, then the strength of the fire will increase by 0.2 if there are neighborhood cells on fire, or by 0.1 if there are not.

The macro above can be defined as:

```
#BeginMacro(checkfire)
((1,1)>=5 or (1,0)>=5 or (1,-1)>=5 or (0, 1)>=5 or (0,-1)>=5 or (-1,-1)>=5 or (-1,0)>=5
or (-1,1)>=5) and not ((1,1)>=6 and (1,0)>=6 and (1,-1)>=6 and (0, 1)>=6 and (0,-1)>=6
and (-1,-1)>=6 and (-1,0)>=6 and (-1,1)>=6)
#EndMacro
```

This macro checks if any neighboring cell is on fire (greater than or equal to 5) and while also making sure that all the neighbors are not completely burnt out (all neighbors are not greater than or equal to 6).

The top coupled model formalism can be seen below:

M= < Xlist, Ylist, η, I, N, X, Y, {m,n}, B, c, Z, select >

Xlist=∅

Ylist=∅

η=9

I=<P^X,P^Y>,with P^X={∅},P^Y={∅};

N= {(0,0), (1,0), (-1,0), (0,1), (0,-1), (1,1), (1,-1), (-1,-1), (-1,1))}

X=Y={-1, 0, 1, 2, 3, 4, [5, 6.2]};

m=50; n=50;

B={nowrapped};

C= {C_{ij}/i ∈ [0,49], j ∈ [0,49]}

Z:

$P_{ij}{}^{Y}{}_{1} \rightarrow P_{i,j-1}{}^{X}{}_{1}$ $\quad P_{i,j+1}{}^{Y}{}_{1} \rightarrow P_{ij}{}^{X}{}_{1}$

$P_{ij}{}^{Y}{}_{2} \rightarrow P_{i,j+1}{}^{X}{}_{2}$ $\quad P_{i,j-1}{}^{Y}{}_{2} \rightarrow P_{ij}{}^{X}{}_{2}$

$P_{ij}{}^{Y}{}_{3} \rightarrow P_{i-1,j}{}^{X}{}_{3}$ $\quad P_{i-1,j}{}^{Y}{}_{3} \rightarrow P_{ij}{}^{X}{}_{3}$

$P_{ij}{}^{Y}{}_{4} \rightarrow P_{i+1,j}{}^{X}{}_{4}$ $\quad P_{i+1,j}{}^{Y}{}_{4} \rightarrow P_{ij}{}^{X}{}_{4}$

$P_{ij}{}^{Y}{}_{5} \rightarrow P_{i,j}{}^{X}{}_{5}$ $\quad P_{i,j}{}^{Y}{}_{5} \rightarrow P_{ij}{}^{X}{}_{5}$

$P_{ij}{}^{Y}{}_{6} \rightarrow P_{i-1,j-1}{}^{X}{}_{6}$ $\quad P_{i-1,j-1}{}^{Y}{}_{6} \rightarrow P_{ij}{}^{X}{}_{6}$

$P_{ij}{}^{Y}{}_{6} \rightarrow P_{i-1,j+1}{}^{X}{}_{7}$ $\quad P_{i-1,j+1}{}^{Y}{}_{6} \rightarrow P_{ij}{}^{X}{}_{7}$

$P_{ij}{}^{Y}{}_{8} \rightarrow P_{i+1,j+1}{}^{X}{}_{8}$ $\quad P_{i+1,j+1}{}^{Y}{}_{8} \rightarrow P_{ij}{}^{X}{}_{8}$

$P_{ij}{}^{Y}{}_{9} \rightarrow P_{i+1,j-1}{}^{X}{}_{9}$ $\quad P_{i+1,j-1}{}^{Y}{}_{9} \rightarrow P_{ij}{}^{X}{}_{9}$

Select= {(0,0), (1,0), (-1,0), (0,1), (0, -1), (1,1), (1,-1), (-1,-1), (-1,1))};

Additionally, the color palette for different environments can be seen below:

```
[-1.0;0.0] 15 236 247 %water

[0.0;1.0] 255 255 255 %no vegetation

[1.0;2.0] 232 213 39 %agriculture

[2.0;3.0] 161 250 65 %steep/sparse forest

[3.0;4.0] 103 148 55 %dense forest
```

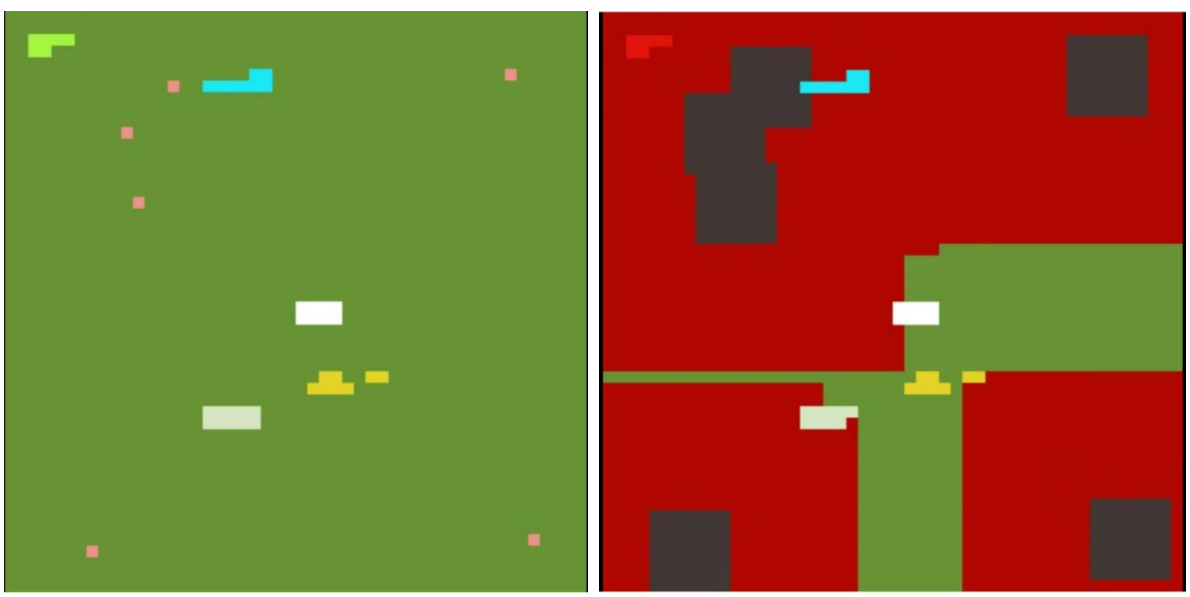
```
[4.0;5.0] 213 230 195 %shrubland
[5.0; 5.2] 245 142 142 %fire level 1 (weakest level)
[5.2; 5.35] 242 121 121 %fire level 2
[5.35; 5.7] 242 99 99 %fire level 3
[5.70; 5.9] 242 15 15 %fire level 4
[5.9; 6.0] 189 2 2 %fire level 5 (strongest level)
[6.0; 7.0] 66 54 54 %burnout - no more fire
```

Part 3: Model Testing

All simulations were done on a 50x50 grid with pre-defined terrain formations. By the end of the simulation, all vegetated areas should be burnt down, only leaving water bodies or areas of no vegetation. The standard time unit is 60000 ms, which is one minute, however, as defined above, it will be sped up by a factor of the burn rate depending on the environment. The fire strength is also dependent on the burn rate.

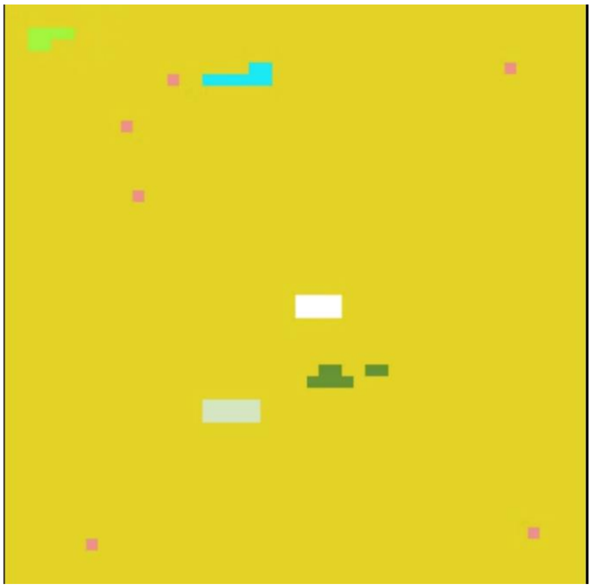
First Simulation:

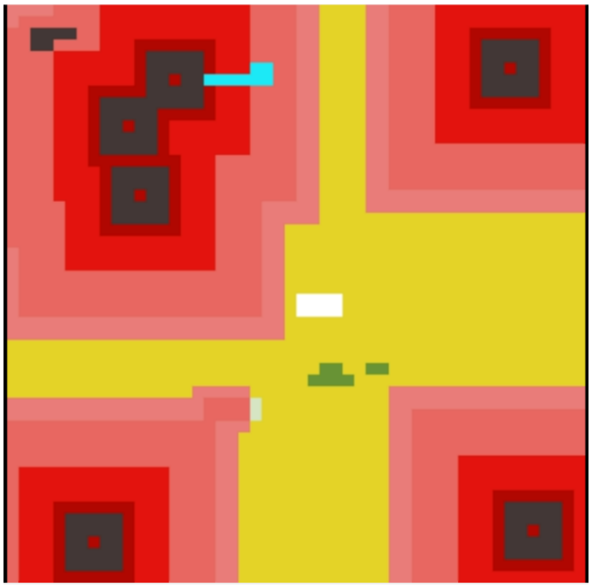
In the first simulation, most of the area was formed of a dense forest, which is a darker green. The light green represents a sparse forest, while yellow is agriculture, greyish green is shrublands, blue is water, and white is no vegetation. The light pink cells are the initial spots that are on fire. The fire gradually grows and once a cell is on fire, it will completely burnout, leaving it a dark greyish color. When a cell catches on fire, the type of fire (strength and time of fire spread) depends on the cell specification (its original state).



Second Simulation:

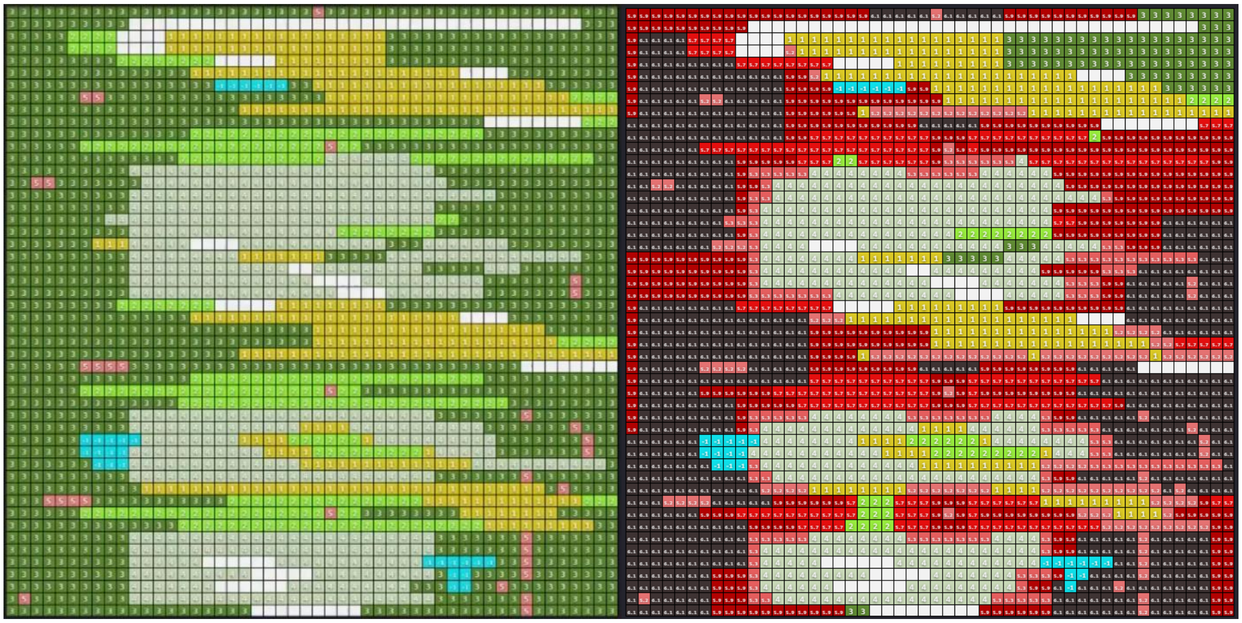
In the second simulation, the environmental states were the same, however the main difference was the simulation was done over a mostly agricultural environment. This takes the longest to burn with the weakest fire intensity other than water/no vegetation. Also, to add more complexity, instead of just burning out, the intensity of the fire will increase each time unit until burnout.





Third Simulation:

In this simulation, the land will be more complex, including more possible environment alternatives, building off the second simulation. The checkfire macro is also improved, to make sure if the entire neighborhood of a cell is burnt, then the cell cannot catch on fire. This simulation includes a dense forest area that also contains areas of agriculture and no vegetation, with some lakes. As the images below show, the fire spread is much more detailed now with various state values at play. The rate at which the fire in a burning cell becomes stronger also became more complex, with the intensity growing by 0.2 if a neighboring cell is also on fire, else it only grows by 0.1.



Part 4: Conclusion

In conclusion, there were three simulations done, based on the formula and conceptual environments provided in [1]. The simulations were designed in CD++ and became gradually more complex each time. Each simulation has a video included with it in its respective folder.

References

[1] J. G. Freire and C. C. DaCamara, “Using cellular automata to simulate wildfire propagation and to assist in fire management,” *Natural Hazards and Earth System Sciences*, 22-Jan-2019. [Online]. Available: https://nhess.copernicus.org/articles/19/169/2019/#section11. [Accessed: 08-Nov-2021].