

Mushroom Simulation Analysis

By Jerahmy Bindon, Skylar Onstot, Saiful Salim, Jonathan Acoltzi

Project: Fairy Circle Growth

This project develops a CA model of fungal spore growth over various constraints. The original model, derived from Shiflet & Shiflet, is expanded and includes wider functionality and poses more simulation questions.



Fig. 1 Real fairy ring on flat turf

Overview:

Our model uses a 2D grid as the baseline for fungal spore growth. Physically, this is assumed to be a flat, unconstrained, plane which contains only healthy and impermeable (inert) material. Model cell states represent different spore/hyphae lifecycle stages at a given point in time, and state change behaviors were governed by various stochastic and procedural methods. Currently, each runtime step is significantly high because each of the $N \times N$ cells gets redrawn individually without graphics optimization.

Questions:

- Why does the shape look like of the fairy circle?
- How sensitive is the model based on its spore spawning, mushroom growth, and hyphae spreads?
- How accurate is the growth pattern in relation to the real world?
- What happens if we initialize different amount of spore/s in a different grid environment?
- What happens if we change the probability of spore spreading from constant, to dynamics based on the number of neighbors that contain young hyphae?
- How would weather, elevation, and presence of semi-permeable materials affect apical growth?

- Who would directly benefit from this simulation?
- Is there any possibility to model the downward (third) dimension?

The Cell States

State	Color
EMPTY	Light Green
SPORE	Black
YOUNG	Dark Grey
MATURING	Light Grey
MUSHROOMS	White
OLDER	Light Grey
DECAYING	Tan
DEAD1	Brown
DEAD2	Dark Green
INERT	Yellow

State Changes

The model is built off the following state diagram and adds multiple other behaviors:

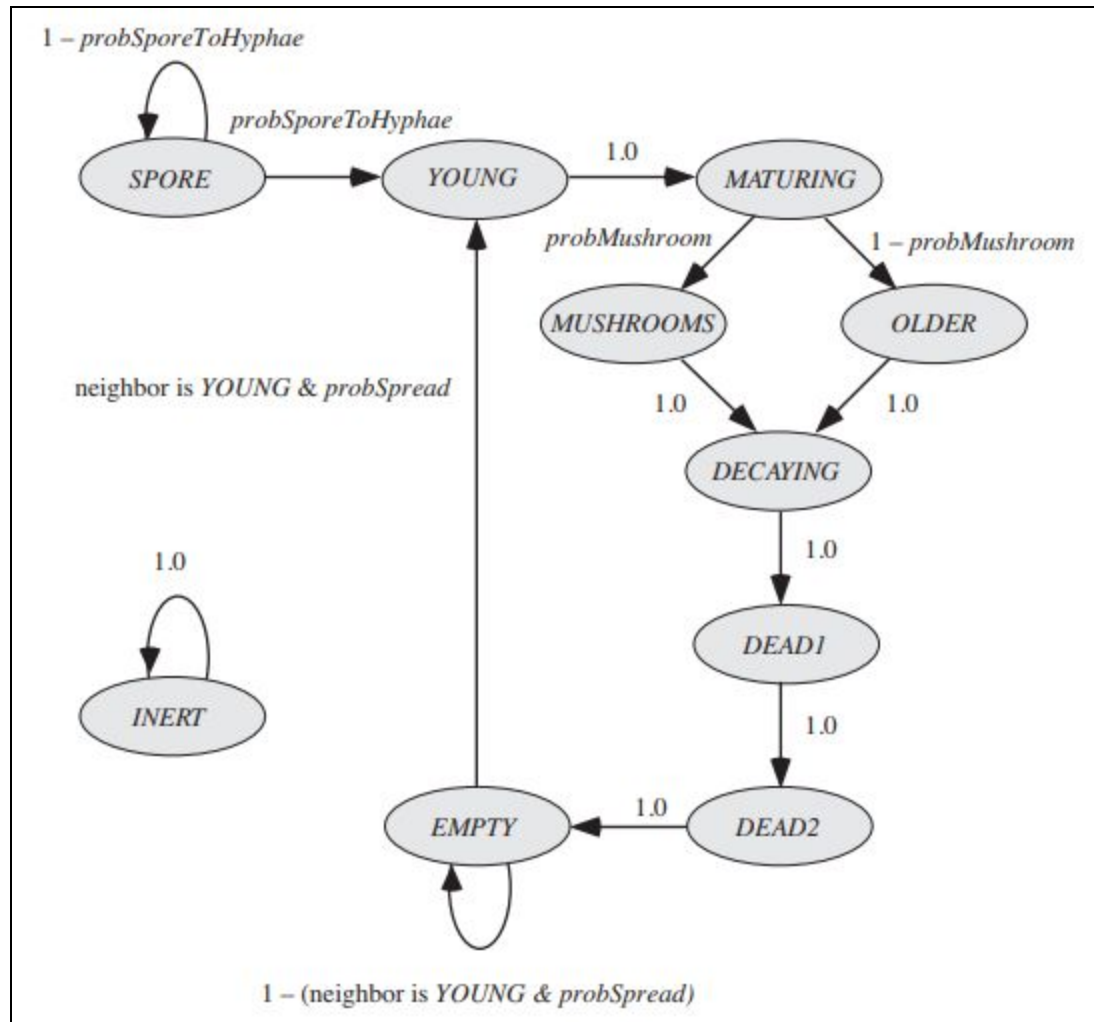


Fig. 2 State diagram from S&S p.717

```

def changeState(copyGrid, i, j):
    global probSpore
    global numMushrooms
    # Check if neighbor cells have mushrooms and update spore spawn probability.
    if mushroomSporeSpawnRule == 1 and grid[i,j] != MUSHROOMS:
        probSpore = (grid[i - 1:i + 1, j - 1:j + 1] == MUSHROOMS).sum() / 8

    if copyGrid[i,j] == SPORE:
        if random.random() < probSporeToHyphae:
            grid[i, j] = YOUNG

    # Modification of state diagram to include spawning of new spores on
    # any kind of ground except inert ground. Code branch applies to both
    # spore spawn rules.
    elif random.random() < probSpore and copyGrid[i,j] != INERT:
        grid[i,j] = SPORE

    # Turn all young hyphae into maturing hyphae after a single timestep
    elif copyGrid[i,j] == YOUNG:
        grid[i,j] = MATURING

    # A maturing hyphae has the chance of fruiting a mushroom. Otherwise, it ages.
    elif copyGrid[i,j] == MATURING:
        if random.random() < probMushroom:
            grid[i, j] = MUSHROOMS

            # Increment mushroom counter if spore spawning is based on mushroom count.
            if mushroomSporeSpawnRule == 0:
                numMushrooms = numMushrooms + 1

        else:
            grid[i, j] = OLDER

    # All mushrooms and old hyphae start to decay after one timestep.
    elif copyGrid[i,j] == MUSHROOMS or copyGrid[i,j] == OLDER:
        grid[i,j] = DECAYING

    # All decaying hyphae will die after one timestep.
    elif copyGrid[i,j] == DECAYING:
        grid[i,j] = DEAD1

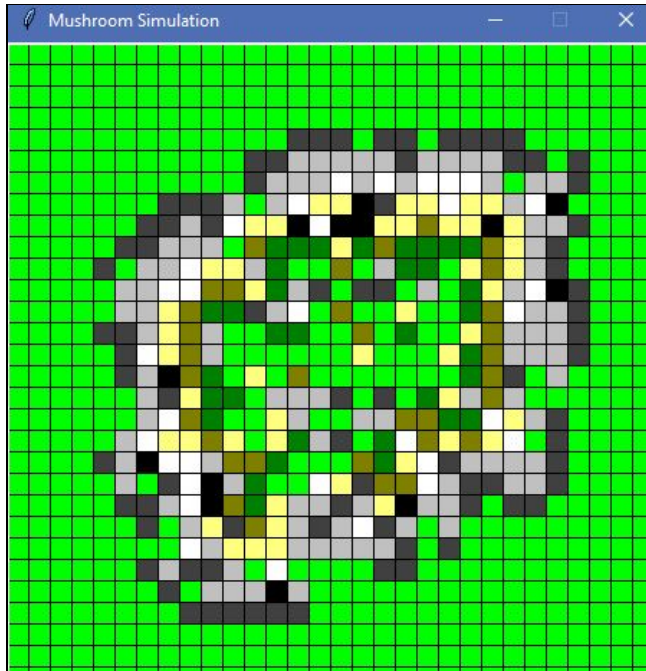
```

Fig. 3 A snippet of state change model code

Model Validation

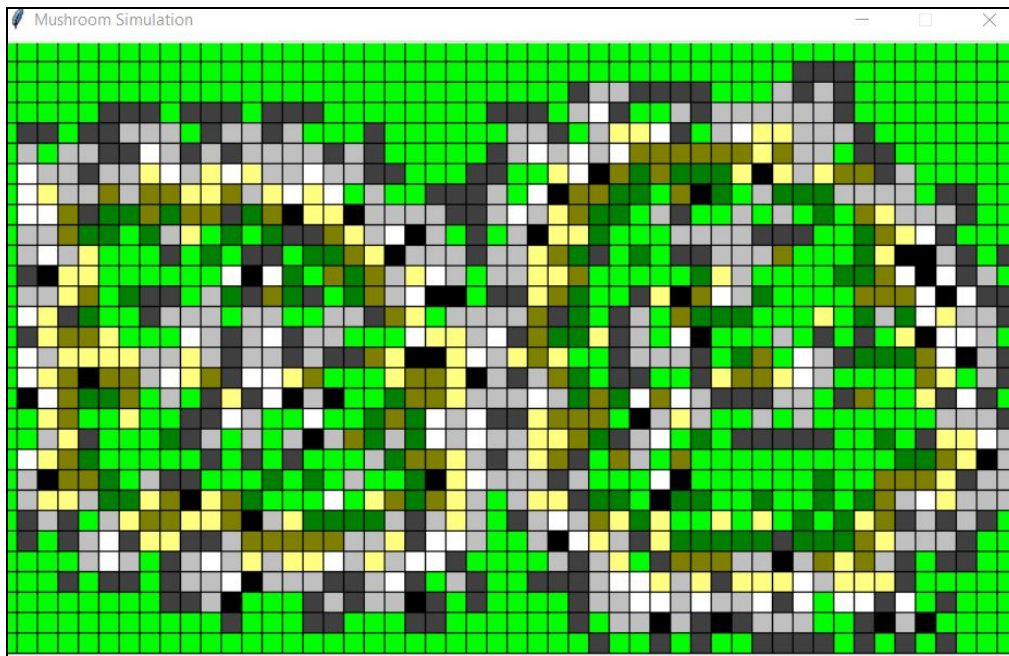
The model was validated using graphical analyses of multiple growth scenarios. The scenarios we tested are listed below:

Single Spore, all probabilities = 60%



We initialize the starting spore to be in the center of the grid. Looking at the above figure, we can see that the mushrooms (WHITE) grow in a circular pattern around the center of the grid and is spreading outwards. We say it is spreading outwards because of the decay(TAN) and dead(DARK GREEN) on the inside of the circle and a lot of young/hyphae(DARK GREY) and maturing(LIGHT GREY) mushrooms at the outer layer/linings. The above figure was taken when timeSteps=12.

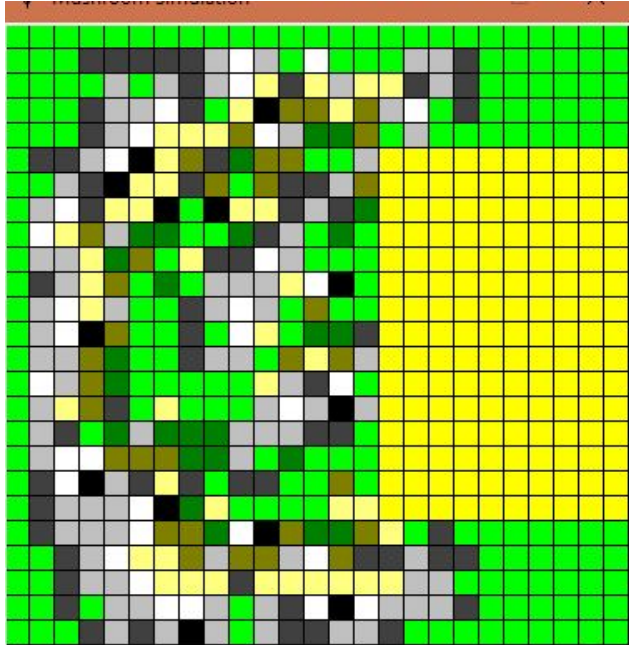
Two Spore, all probabilities = 60%



The starting spore was initialized with two separate spores that are several cells apart toward the middle. In the above figure, we can see the mushrooms (WHITE) to be in a pattern of a rotated figure "8", with some in the center of the starting spores location. The mushrooms intersect with each other in the middle of the grid while spreading outwards.



Simulating a single spore with a barrier. All probabilities = 60%



The starting spore was initialized in the center of the grid. Inert cells(YELLOW) was initialized to create barriers that fungi can't cross over or grow.

We can see the mushrooms (WHITE) created some kind of arc over the barrier while spreading outwards. We verify this simulation with an example from the real world:



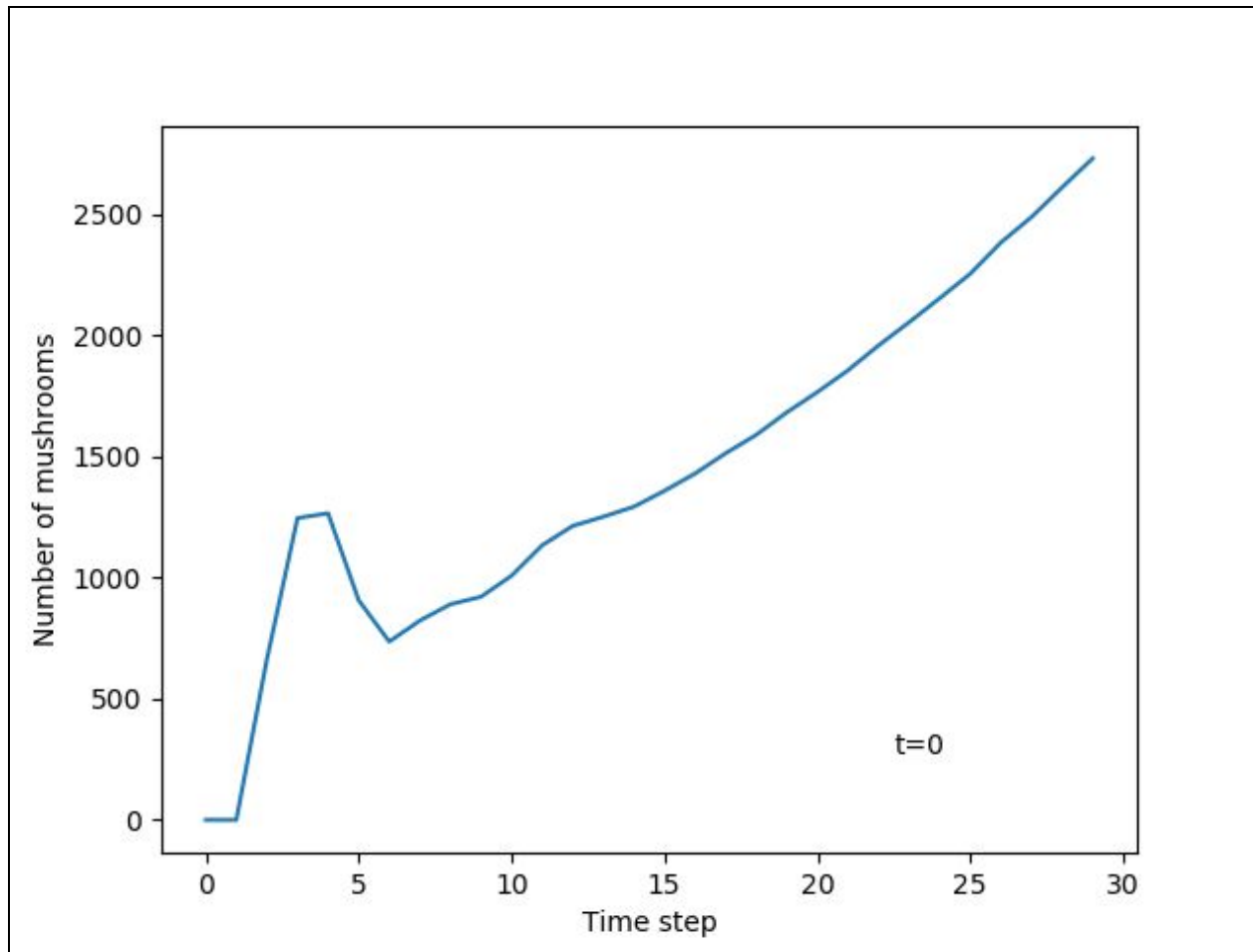
The mound in the above figure is represented by the inert cells.

Random walk inside a CA model

When a spore turns into hyphae and grows like a tree root, it grows out in many random directions. Our model modified the S&S model by replacing the numerical probability with one that better simulates a potential apical growth pattern. Thus, the larger the random walk, the more precise the state changes will be.

Trends found

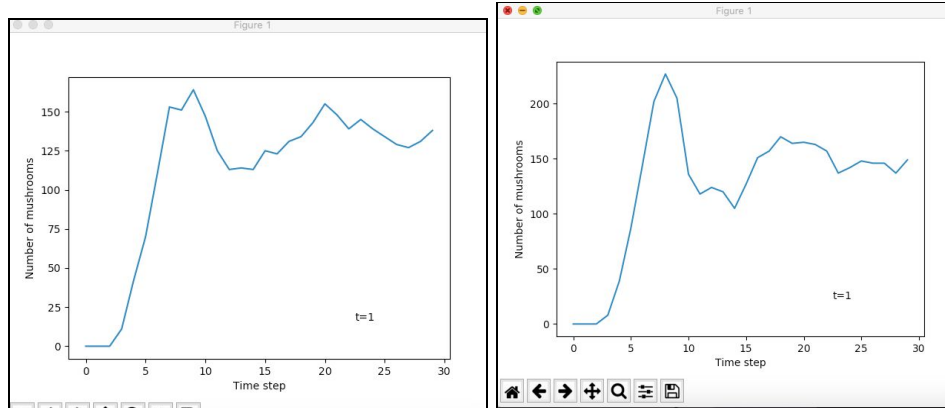
We found from a graph showing the amount of mushroom on a plot per time step. For many simulations runs we kept on getting a dip in our mushroom numbers around time step 6-9. We looked at our model and notice that this could be due to mushrooms reaching a time where they transfer to another stage but the simulation still has not run long enough for more mushrooms to appear. Figure 2.0 shows a plot of mushrooms per time step where the probability of a spore becoming hyphae is 100%



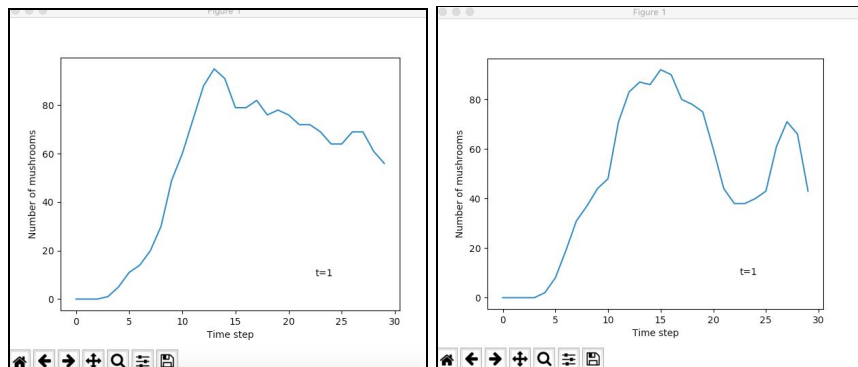
Sensitivity check

It seems like the code breaks whenever the probability gets too low an example is when the probability of hyphae is decreased to .001 is implemented as opposed to 0.01. The first two graphs show the probability being at .01, where two different simulations still give the same general number of mushrooms. With the next pair, the probability is .001 which starts to show signs of not being constant.

Mushroom numbers when probhyphae is .01



Number of mushrooms per time step when probhyphae is .001



As seen above when probability gets too low the number diverge from simulation to simulation. This is due to if a spore has a lower chance of becoming hyphae, there are fewer hyphae that can infect other cells and become hyphae. Since spores can never die unless they become hyphae and evolve, there is a dip for the time that many spores do not reach the young stage. The increment is when these spores finally do evolve into young and later become mushrooms and infect more cells.