**Spreading the flu in a classroom:**

I would prefer to model this as a Cellular Automaton. Students are generally confined to one place in a classroom, that being their desk. I think it would be better to represent a classroom using cells. Each cell could contain a desk with a student, since desk don’t usually move, this would align with Cellular Automaton’s cells that don’t move. The students would be also be in simple states. There is also no need for an environment in this model since the environment has no impact on the spread of the flu.

States: Healthy – The student is not infected with the flu

Infected – The student is sick with the flu and spreading it

Recovered – The student has had the flu and cannot catch it anymore

Initial Configuration: By default, I would have a class of 25 and all the students would be healthy. However, 10% of the students would be made infected. None of the students would be recovered. There would be only 10% of infected to represent the handful of students represented in the description. There would be no recovered since the flu hasn’t occurred in the classroom before the model begins.

Rules: Healthy students can become sick based on their neighbors. If none of their neighbors are infected, then the students would not have any chance of becoming infected. However, if the healthy student has a neighbor that is infected, then the healthy student’s chance to become would be 20% \* n, n being the number of neighbors that are infected. I chose 20% as the baseline because of the high spread of the flu, in addition, even if every neighbor of a healthy student is infected, the healthy student will never have a 100% chance of becoming infected. Using n allows the infection rate to increase with each infected student a healthy student comes into contact with. Infected students would become recovered after 3 ticks. Recovered students can never become infected again since they are now immune to the flu.

Stop Running: The model would stop running when every student is either healthy or recovered. This is done since the flu can no longer spread in the classroom.

Figure:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**BLUE = HEALTHY RED = INFECTED**

**Spreading HIV on campus:**

I would use an Agent Base Model for this model. The main reason being that the agents aren’t going to only interact with their neighbors, they are going to move around and interact with any of the other agents. The agents will also have characteristics that affect the infection rate. Although the environment in this case is unimportant, the ability of the agents to move around is pivotal.

Agents: Healthy – Students who have not contracted HIV and having unprotected sex

Newly Infected – Students who have contracted HIV and are extremely infectious and are unaware of their new disease

Being Treated – Students who are undergoing treatment for HIV and have stopped having sex

Treated – Students who have been cured of HIV but continue to not use protection

Protected – Students who use protection when having sex

Characteristics: Healthy – Chance of sex when they meet another agent, chance of infection

Newly Infected – Chance of sex when they meet another agent

Being Treated – Chance to forget to take medicine and have to restart

Treated – Chance of sex when they meet another agent, chance of infection, chance to start using protection

Protected - Chance of sex when they meet another agent

Environment: The environment would simply be blank patches. The agents would simply move around in them. The environment has to effect on the spread of HIV in this model.

Initial Configuration: By default, I would spawn 100 agents and all of our agents would be healthy. About 10% of the agents would be set to protected, we don’t need to see a large number of protected students. This model is tracking HIV spread, not safe sex occurrences. Another 10% of our healthy students would be set to newly infected to introduce HIV to the campus. No being treated students or treated students would need to be set in the beginning.

The chance of sex characteristic for all agents would be set to 50%. I’m going to assume our group of students are quite promiscuous. The chance of infection for all agents who can contract would be set to 75%. HIV is, after all, very infectious. The chance of forget to take medicine for the being treated students would be set to 10% because hopefully only a small number of people have little disregard for their health. The chance to start using protection for the being treated students would be set to 75%.

Rules: At each tick, all of the agents will choose a random direction and move one patch. If they are on the same patch as another student, they will use the chance of sex characteristic to determine if they have sex with the other agent, unless the other agent is being treated. If a healthy or treated agent interacts with a newly infected agent, then they use the chance of infection characteristic to determine if they will become newly infected. An agent will spend 3 ticks as newly infected agents before becoming being treated agents. Being treated agents will spend 3 ticks before becoming treated or protected agents. During those 3 ticks, if the being treated agents roll the chance of forget characteristic and do forget, their timer will reset to the beginning. After completing treatment, being treated agents will roll the chance to start using protection characteristic to determine if they become protected or treated.

Stop Running: The model would run for a semester of a school year, or about 100 days. The model will go for 100 ticks since 1 tick is equal to 1 day.

Figure:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| H |  | H |  | H |  | P | H |
| H |  | H | H |  | H | I | H |
| H | P | H |  | H |  | H |  |
| H |  | H |  | H | H |  | H |
|  | H |  | H |  | H | H | H |
|  | H | H |  | H | H |  |  |
| I | H |  | H |  |  |  |  |
|  |  | H |  | I |  | P | H |

H = HEALTHY I = INFECTED P = PROTECTED

**Landslides:**

I would prefer to model this using Cellular Automaton. Each piece of land could be represented using cells and the cells would have according shrubbery. The cells would only need to interact with those around them, since land would only move from one cell to another. There would also not be any agents that need to move around in this model.

States: Low Stability Low Vegetation – The land is low with low amounts of vegetation

Low Stability High Vegetation – The land is low with high amounts of vegetation

High Stability Low Vegetation – The land is high with low amounts of vegetation

High Stability High Vegetation – The land is high with high amounts of vegetation

Initial Configuration: The cell configuration would be 5x5 for a total of 25 cells. The states would be randomly distributed with each having a 25% to be set for each cell.

Rules: In order for a HSLV to become a LSLV, it would need 2 LSLV neighbors or 3 LSHV neighbors. In order for a HSHV to become a LSHV, it would need 3 LSLV neighbors or 4 LSHV neighbors. If a neighbor of a LSLV or LSHV becomes Low Stability, then they become High Stability.

Stop Running: The model would stop running once there are no Low Stability cells and only High Stability cells. If this never occurs, the model would stop running after 50 ticks.

Figure:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LSLV | HSLV | LSHV | HSLV | LSLV |
| LSHV | HSLV | LSLV | LSLV | LSHV |
| LSLV | HSHV | HSLV | HSHV | HSHV |
| HSHV | LSLV | LSHV | HSLV | LSLV |
| LSHV | HSLV | HSHV | LSHV | HSHV |

ORANGE = LOW STABILITY LOW VEGETATION

RED = LOW STABILITY HIGH VEGETATION

BLUE = HIGH STABILITY LOW VEGETATION

PURPLE = HIGH STABILITY HIGH VEGETATION

**Landslides… revisited:**

I would prefer to make this model using Agent Based Modeling. Even though the landslide nature of this model is better done in Cellular Automaton, with the introduction of animals that are grazing, it is better suited towards ABM. The animals are better represented as agents, they need to move to satisfy the grazing aspect.

Agents: Animals – These are the animals grazing on the land. I see no need to separate them into their own species since their eating habits aren’t specified.

Characteristics: Animals – The rate at which they eat vegetation around them.

Environment: Low Stability – Dirt with a high chance of moving

High Stability - Dirt with a low chance of moving

Vegetation – Grass, shrubs, etc. that would be placed on top of the dirt

Initial Configuration: The patches would be set with either Low Stability or High Stability, with a 50% chance for each. Vegetation would be then randomly distributed around the model. There would be a high chance that vegetation is placed.

I would spawn about 20 animals around the model. Each time they found vegetation near them, they would roll a chance to eat that vegetation. I would set this rate at 25%.

Rules: At each tick the animals would choose a random direction and move one space that way. If they find vegetation on the patch they land on, they will roll the rate to eat to decide if they will consume it.

The rules for landslide spread would be similar to the Cellular Automaton version. In order for a HS to become a LS, it would need 2 LS neighbors with no more than 3 vegetation or 3 LS neighbors with no more than 5 vegetation. If a neighbor of a LS becomes Low Stability, then it becomes High Stability.

Stop Running: The model would stop running once there are no Low Stability cells and only High Stability cells. If this never occurs, the model would stop running after 50 ticks.

Figure:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 Veg Animal | 1 Veg | 4 Veg Animal | 5 Veg | 2 Veg |
| 7 Veg | 3 Veg | 1 Veg | 0 Veg Animal | 1 Veg |
| 4 Veg | 5 Veg | 6 Veg | 3 Veg | 4 Veg |
| 5 Veg | 6 Veg Animal | 2 Veg | 1 Veg | 4 Veg |
| 6 Veg | 1 Veg | 4 Veg | 5 Veg Animal | 3 Veg |

ORANGE = LOW STABILITY RED = HIGH STABILITY

VEG = VEGETATION AND NUMBER ANIMAL = ANIMAL AGENTS

**Up to You:**

The Problem: I want to measure and predict flooding in an area impacted by a hurricane, like the Carolinas recently. In order for flooding to spread from one area to another, the area needs to have a high amount of flooding.

I want to use cellular automaton to display this model. Each area of a city can be represented by a cell, with each cell accompanied by a flood level. There is no need for agents in this model so I find it better to use CA.

States: Not Flooded – The area is not flooded

Light Flooding – The area contains a small amount of flooding, nothing dangerous

Heavy Flooding – The area is heavily flooded and is potentially dangerous

Initial Configuration: At the beginning, all cells would be set to Not Flooded. I would then set 4 random cells to have Heavy Flooding. Each of those 4 cells would have their cells set to Light Flooding. This is done to show the areas of heavy rainfall in susceptible places.

Rules: A Not Flooded cell can become a Light Flooded cell if there are 2 neighboring cells with Light Flooding or 1 neighboring cell with Heavy Flooding. A Light Flooded cell can become a Heavy Flooding cell if there are 2 neighboring cells with Heavy Flooding. If a Light Flooding or Heavy Flooding’s neighbor floods, then their level would go down one, to Not Flooded and Light Flooding respectively. After 5 ticks at Light Flooding, the Light Flooding cell will become a Not Flooded cell. After 5 ticks at Heavy Flooding, the Heavy Flooding cell will become a Light Flooding cell.

Stop Running: This model would stop running once there are no more Heavy Flooding or Light Flooding cells left.

Figure:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

GREEN = NOT FLOODED BLUE = LIGHT FLOODING PURPLE = HEAVY FLOODING