Wilderness Exploration and Pathway Formation

Konrad Aust
Dept of Computer Science
University of Calgary
ktaust@ucalgary.ca

Mark Barley
Dept of Computer Science
University of Calgary
mpbarley@ucalgary.ca

***Abstract*—In this project, we will explore the emergent behaviors of destructive agents moving through randomly generated wilderness terrain. Specifically, we will investigate how pathway networks develop in the landscape, and the extent to which a network of paths eases the difficulties of traversing terrain.**

I. Introduction

Wilderness terrain is rarely uniformly wild. Most often, it is found carved up with many pre-formed paths and animal trails. How did these paths form? Clearly, they were not engineered, but were the product of some sort of emergence. In this project, we attempt to explore this process by simulating the movement of entities across procedurally generated terrain.

Our simulation models a section of wilderness terrain, through which many different agents move. These agents can represent hikers, all-terrain vehicles, animals, or anything else. Each of these agents has nothing more than a general direction they are traveling, and knowledge of their immediate surroundings. Their goal is to reach their destination through the easiest path they can find. Agents do not have any organization amongst themselves, nor do they communicate directly with one another. Ultimately, the purpose is to see if coherent and complex networks of pathways can emerge from the actions of agents that are not cooperating with one another.

Another topic of investigation is how easy traversing the terrain becomes as the path network reaches maturity. A wild wilderness should presumably be a difficult area to move through. Just how much of an effect does an established network of paths have on minimizing this difficulty?

Also, just what is the effect on the terrain itself when these agents come tearing through, trampling everything in their path? Is the destruction they cause enough to destroy ecosystems, or can a wild wilderness be made traversable without compromising its various ecological balances?

We have created a complex system crafted out of relatively simple rules, so there is a lot of potential for completely unforeseen properties to emerge. We hope to find unexpected behaviors in the simulation. As with many experiments, the secondary goal of this simulation is to be happily surprised by it.

II. Simulation Overview

The simulation was written in NetLogo 5.1. It consists of two major parts: the terrain, and the agents. The terrain is a procedurally generated, grid-based map, and agents are mobile objects that traverse this map.

A. Terrain Generation

Terrain is generated through a mix of Perlin noise [2] and Voronoi tessellation [1]. The Voronoi tessellation divides the map into a number of regions, each corresponding to a different biome. This biome dictates the difficulty of moving through the tile, as well as the resistance of the tile to being trampled. The perlin noise governs the initial integrity of the tile, which determines how intact the terrain is, and thus how difficult it is to traverse. For example, a snow tile with high integrity can be thought of as deep snow, which is very difficult to navigate, but rapidly becomes very easy to navigate as agents pass through it.

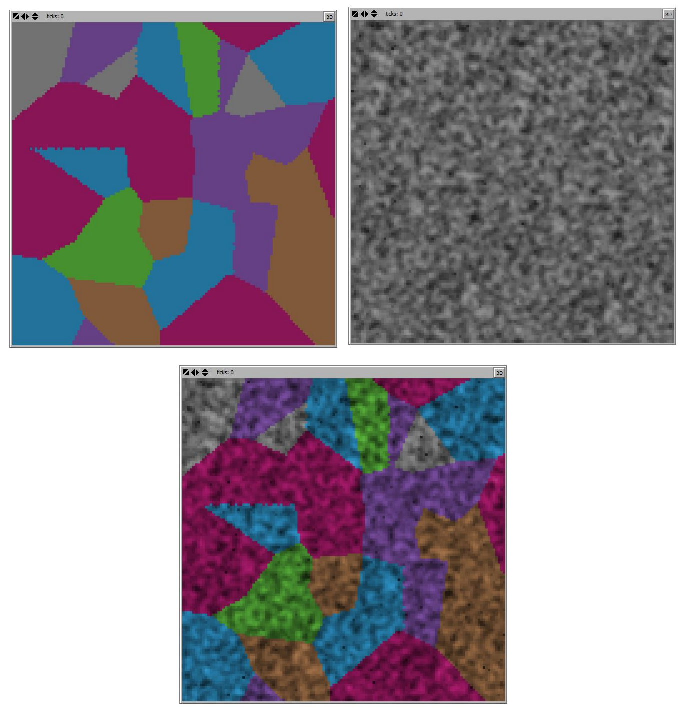

Fig. 1. Map Generation Layers. Voronoi Tessellation (top left) and Perlin Noise (top right) are combined to create our final terrain patch. (Bottom)

Each type of biome has a difficulty multiplier, and the actual difficulty of a tile is given by

```
difficulty = integrity * diff_mult
```

Each type of biome also has a deterioration value. When an agent steps on the patch, its integrity is set according to:

```
integrity = old_integrity * det_value
```

A darker colored patch in the simulation has a greater integrity value than a lighter colored patch, so when a path is formed, generally it shows up as a line of white patches surrounded by regular terrain.

See table 1 for an overview of the different terrain types we used in our simulation.

Type	Color	Description
Forest	Green	Normal movement. Normal Durability
Rock	Brown	Easy to move through. Very durable.
Snow	Gray	Difficult to move through. Not durable.
Underbrush	Blue	Easy to move through. Not durable.
Jungle	Purple	Difficult to move through. Durable.
Swamp	Pink	Difficult to move through. Moderately durable.

TABLE I
BIOME TYPES

B. Agent Behavior

Agents are spawned in at fixed intervals. They appear on one edge of the simulation, and are given a random destination point on another edge of it. They attempt to traverse the terrain to get to their destination, at which point they leave the simulation, and their total weighted distance traveled is plotted.

Agents use A* pathfinding with a limited sight radius to decide which path to take through the terrain. Agents can only determine the difficulty of terrain tiles that are within their sight radius, and assume that all tiles beyond their sight radius have equal difficulty. Each tick, an agent takes a step along its pre-computed path, and at fixed intervals, an agent will re-compute its path.

As was mentioned above, as agents move across the terrain, they trample that terrain, reducing its integrity and thus its difficulty to pass through for future agents. The destruction of terrain creates paths that other agents are more likely to follow, not because of any inter-agent communication, but because the paths are now easier to traverse. We can analyze the structure of these paths, along with the weighted distance that each agent travels as the path network is developed.

C. Extension: Terrain Regrowth

As an extension to the simulation, we implemented a system where, rather than being trampled permanently, terrain can slowly regenerate itself when left alone. Terrain regrows at a user specified rate. This rate scales with the missing integrity of the terrain tile, so regrowth is faster at low integrity, and slower at high integrity. Also, terrain will not regrow for a certain number of ticks after it has been trampled.

We believed that over the long term, this may create a more accurate representation of how paths generate in the wilderness. If agents cease to use a certain path, it can be reclaimed by nature, and so the network of paths can not only grow, it can also shrink.

III. RESULTS

Our simulation does in fact produce an emergent network of pathways through the terrain. As you can see, agents tend to create simple networks from edge to edge, as well as forming pathways around the edge.

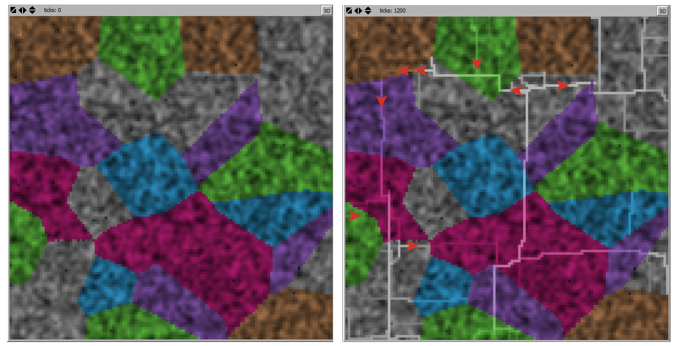

Fig. 2. On the left, an untouched map. On the right, a map with some established pathway networks.

Once an established pathway is found that leads in the same direction as our destination, the agents will walk along it instead of trampling nearby terrain. By graphing the travel distances of each turtle, and the average trip distance over time, we are able to see a clear logarithmic drop off of difficulty.

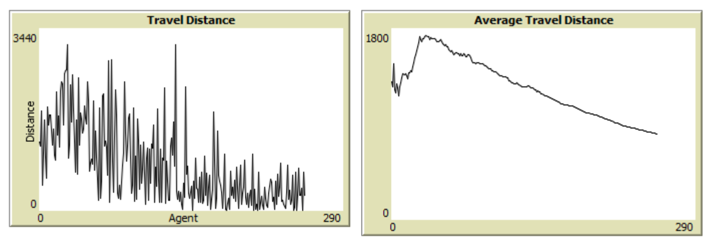

Fig. 3. Travel Distance and average travel distance as simulation progresses

Next I will go through how each type of parameter changes its behavior.

A. Turtle Spawn Rates and Population Cap

The spawning frequency and maximum number of turtles affects the maps formation almost exclusively in the early stages of simulation. Once a network of pathways connects all four edges agents quickly converge to walking across them. Essentially once the pathway has formed these parameters amount to throughput. See figure 4.

In the early stages of the simulation this affects the density of our path network, essentially, how many paths through the center of the network are created. This higher density of paths reduce the average cost of traveling over the map as we create more direct pathways. See figure 5.

B. Agent Sight parameters: Sight Radius - Backtracking

Backtrack is something that never worked as well as we would have liked. We found that by letting agents turn around they were often stuck in two or three patch loops because

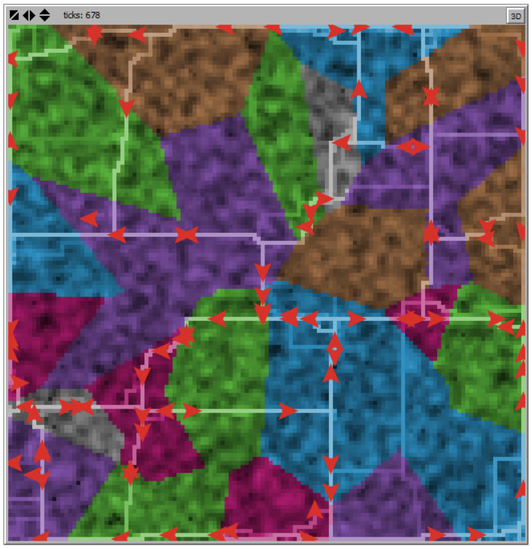

Fig. 4. Spawn Rates and Population Cap

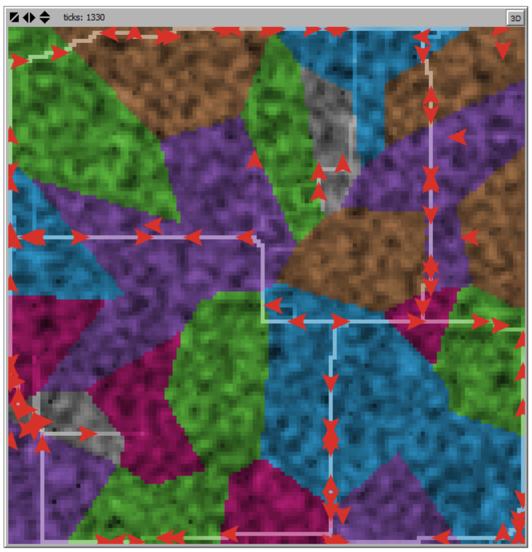

Fig. 5. Alternate path network for the same map.

as they trample the ground it becomes easier to walk on. I think we could remedy this, but we would need to change our movement system to work based on vectors instead of discrete movements. This would allow our agents velocity to respond to the difficulty of their terrain, which we could use to determine how fast an agent can turn around or even just stop.

Sight radius affects how quickly the agents converge to the already built pathway networks. Having a very low sight radius will generate a more dense pathway network, because they will not be able to find paths that are far apart. With a sight radius of 1, agents just move to the square closer to the destination with lower difficulty. This just tramples the entire map. (See figure 6.)

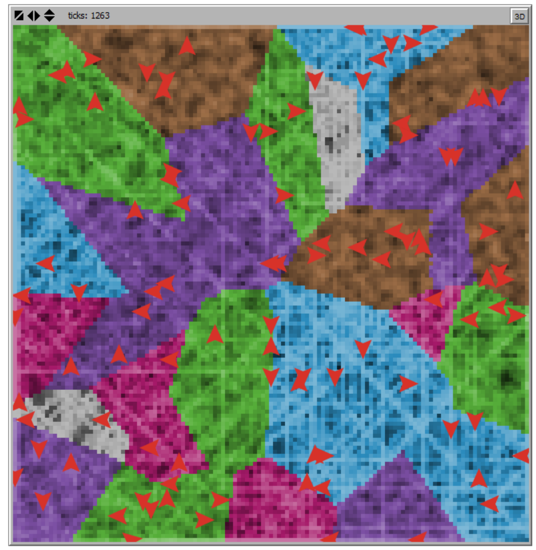

Fig. 6. An established network of paths from agents with very low sight radius

C. Terrain Generation Parameters

These effects were less pronounced on the system's behavior. Noise octaves provided stochastic levels of terrain difficulty, however the higher octaves are affecting the difficulties by such a small margin when compared to other aspects of pathway generation that they have little to no noticeable effect on paths. This could be refined.

Biome count had similar effects. Each biome that must be traversed can be considered a sub-problem of our simulation. Having more biomes means that there are more biomes for a pathway to travel through, generally speaking at least. In a low density network this can have negligible effects because there are only a few pathways anyways, however in higher density path networks this shows structured pathways within each biome.

D. Extension: Terrain Regrowth

The terrain regrowth effect either does not affect the system very much, if the regrowth is slow enough to not destroy paths; or it forces agents to make and remake new paths if it's growing faster than the agents can wear it down. This is as expected, we figured it wouldn't have much effect on the regularly used paths, especially considering the rate at which it recovers in the real world; and would more serve to rebuild the nature next to paths.

IV. Discussion

Our inspiration for this project was 'off the beaten path' trails that are often found to emerge in places where humans and/or animals are presented with some non-descriptive landscape to traverse. Whether it's the lawn in front of math science, or a trail through the woods to the campsite latrine, it's clear that when a rudimentary pathway is formed agents in real life tend to walk the road more traveled. Additionally, every agent in our real world system is acting independent of each other, which makes modeling this system very easy in NetLogo.

I think our simulations results show several interesting things about the way humans think and work together. Since our simulation can be thought of as graph traversal, it tends to optimize the distance based on difficulty, however this is only a half truth, because as agents walk across the map the damage terrain, which makes the 'optimal' solution change. I think a better way of putting it would be to say that our simulation starts with a naive pathway system created by the first agents that is not very efficient. Over time our simulation serves to optimize the difficulties of the patches it exists on, so our pathway becomes the most efficient version of that specific route possible.

Likewise, this is exactly what humans do in real life. For an example consider a group walking through the woods. If the first agent doesn't see the easiest path to their destination, they will pick a less efficient route, deteriorating it slightly in the process. Every agent thereafter will naturally follow this leader, also unaware of the paved highway 20 feet to the left, which is how they would optimize their travel distance. Instead they are optimizing the pass-ability of this secondary pathway, any future agents who walk over it will have an easier time.

This way of doing this actually shows a remarkable level of unconscious forethought and swarm behavior on humanity's behalf, and I find it hilarious how unaware we are of how often these side effects arise from our even most selfish behavior. Everyone who cuts across the Math Science lawn thinks that they are independently trying to optimize their walk to school, this is not considered altruistic, if anything we feel shame for walking on the grass. However this also has the side effect of reducing the difficulty of the short-cut path, and over time we end up with a grass free dirt path that might as well be paved, effectively replacing the old path with a new optimal solution for everyone. Again, the math science lawn is a great example. I do not think I have ever seen a single person leave through the north doors to head east, and walk all the way around the lawn instead of cutting across the dirt path.

Now I want to be clear here that these extrapolations I am making are not rigorously tested, and should not by any means be taken as fact. My extrapolations about human's intentions stems from my deeply held belief that humans are, generally speaking, incredibly naive; especially when it comes to understanding their own behavior. If you agree with this, our simulation can be thought of as to outline the possibility of manipulating selfish agents into working together; which to me seems easier than getting them to willingly cooperate.

V. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank Hugo Elias, for creating a friendly and easy to use guide to the implementation of Perlin Noise [3], Meghendra Singh, for implementing A* pathfinding in netlogo [4], and Uri Wilensky for creating Netlogo, and writing an excellent implementation of Voronoi Tessellation for it [5] [6].

References

[1] Weisstein, Eric W. "Voronoi Diagram." From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/VoronoiDiagram.html

[2] Perlin noise. (2016, March 22). In Wikipedia, The Free Encyclopedia. Retrieved April 28, 2016, from `https://en.wikipedia.org/w/index.php?title=Perlin_noise`

[3] Hugo Elias, "Perlin Noise." (1998, September 14). Retrieved April 28, 2016, from `http://freespace.virgin.net/hugo.elias/updates.htm`

[4] Meghendra Singh (2014). Astardemo1. Retrieved April 27, 2016, from http://ccl.northwestern.edu/netlogo/models/community/Astardemo1.

[5] Wilensky, U. (2006). NetLogo Voronoi model. http://ccl.northwestern.edu/netlogo/models/Voronoi. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

[6] Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.