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.

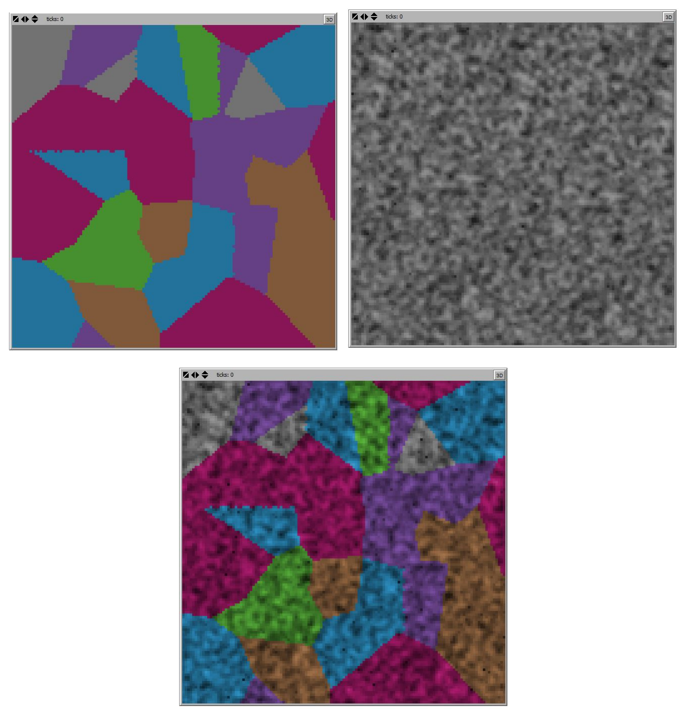

Fig. 1. Map Generation Layers. Voronoi Tessellation (top left) and Perlin Noise (top right) are combined to create our final terrain patch. (Bottom)

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 more complex sections closer to the edges.

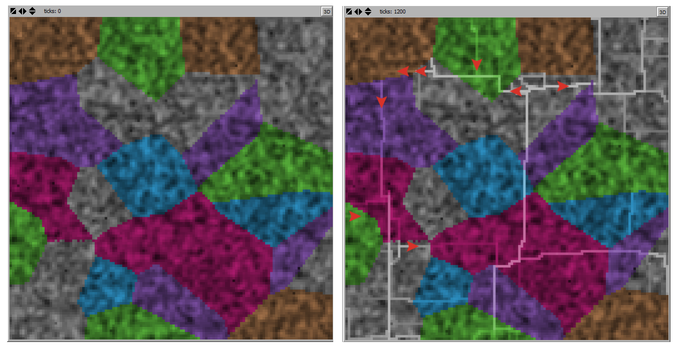

Fig. 2. On the left, an untouched map. On the right, a map with some established pathway networks.

Once an established pathway is found that leads in the same direction as an agent's destination, the agent will walk along it instead of trampling nearby terrain, simply because the trampled section now forms an easier path. By graphing the travel distances of each turtle, and the average trip distance over time, we are able to see a fairly linear decline of difficulty. This means that our simulation had the predicted behavior! The average trip distance is reduced drastically when a mature path network is formed.

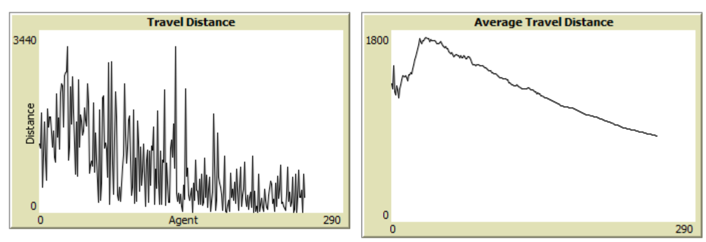

Fig. 3. Travel Distance and average travel distance as simulation progresses

A. Effects of Simulation Parameters

1) Turtle Spawn Rates and Population Cap: The spawning frequency and maximum number of turtles affects the formation of pathways almost exclusively in the early stages of simulation. Once a network of pathways connects all four edges, agents quickly converge to walking across them, rarely taking shortcuts. (See figures 4 and 5). Once the pathway network has become established, parameters amount to throughput.

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 reduces the average cost of traveling over the map as we create more direct pathways.

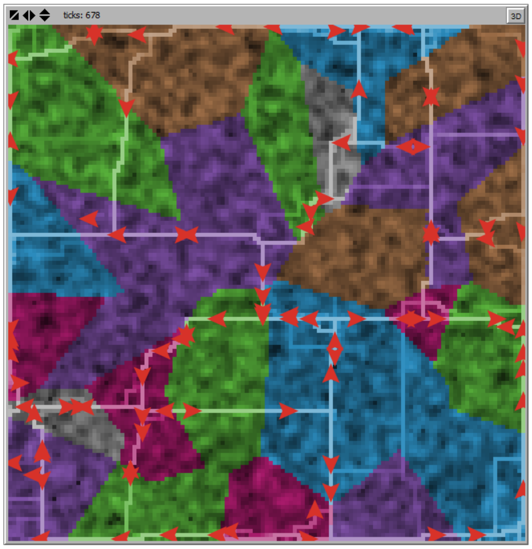

Fig. 4. Spawn Rates and Population Cap

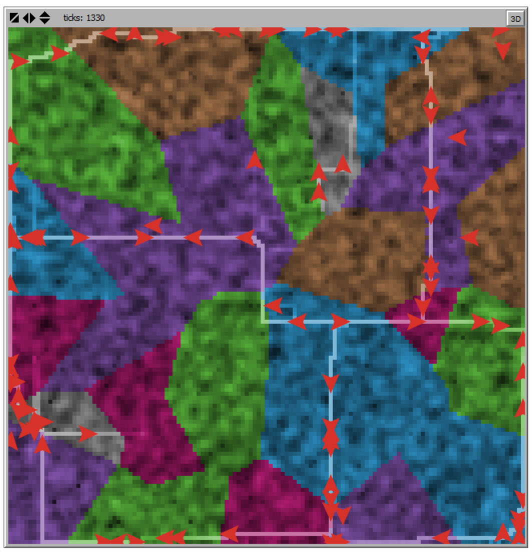

Fig. 5. Alternate path network for the same map.

2) Sight Radius and Backtracking: Sight radius affects how quickly the agents converge to the already built pathway networks. Having a very low sight radius will generate a more cluttered pathway network, because they will not be able to find paths that are far apart. As a result, they will form their own paths a lot more. With a sight radius of 1, agents essentially move directly towards their destination. This just tramples the entire map. (See figure 6).

However, with a larger sight radius, the path network that is formed leaves most of the terrain intact, and rapidly decreases the difficulty in moving through the map.

Backtracking governs how far backwards an agent is willing to go in order to find an alternate route. This is less effective than it sounds, since agents do not have any sort of memory

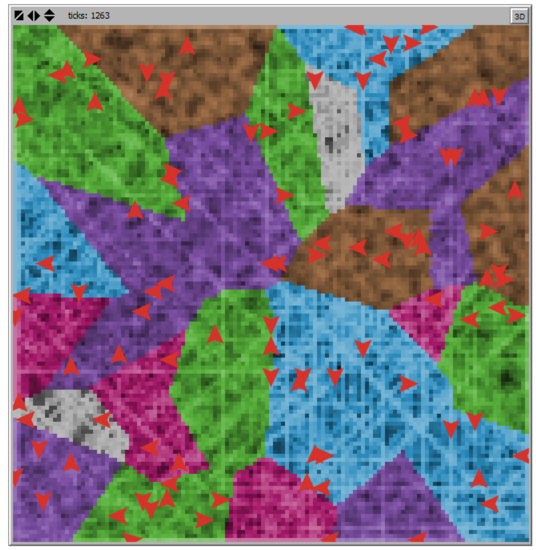

Fig. 6. An established network of paths from agents with very low sight radius

of where they have been. As a result, agents would often go in circles backtracking, trying to find a path around a particularly difficult biome.

If we were to implement a way for agents to remember surroundings they have already traversed, then backtracking could be made useful, and may even result in cleaner path networks. Instead, we kept the simulation simple, and only allowed agents to backtrack one square. The result is that with almost every move, agents must get closer to their destination.

3) Terrain Generation Parameters: The map's noise octaves govern the number of successive noise functions that are added together to create the final perlin noise. The map's noise persistence specifics the amplitude of each octave of noise. These parameters altered the appearance of the perlin noise, but had very few interesting effects on the emergent pathway network. They were included for completeness sake. For more information on these parameters, see the explanation by Hugo Elias, who wrote an excellent tutorial on implementing perlin noise [3].

The biome count determined how many different biomes were generated using Voronoi tessellation. This had a noticeable effect on pathway networks. More biomes means that generated biomes are smaller, and so it is more likely that agents will detour around particularly difficult biomes entirely. When the biomes are large, these types of detours are much less common.

B. Extension: Terrain Regrowth

The terrain regrowth can have interesting effects on the system. If the regrowth rate is very high, pathways can become overgrown and disappear before any other agents have a chance to use them. This results in a patch of terrain that almost never gets any easier to traverse, except when there is a very large number of agents moving through it.

When the rate is lower, the main pathways are almost entirely unaffected, since they have enough traffic that the regeneration is outpaced by the trampling. Interestingly, though, there are situations that arise where pathways that were once heavily used are discarded in favor of newly created, easier pathways. In these cases, it is common for the original pathway to become overgrown, and to disappear entirely.

IV. Discussion

Whether you choose to interpret this simulation as paths in a lawn to a university building, or a trail through the woods to the campsite latrine, it's intuitive that when a rudimentary pathway is formed, agents in real life tend to walk the road more traveled. This means that, indeed, agents tend to cause complex pathways to emerge in terrain even when they are not communicating with one another in any way. Our simulation results show several interesting things about the way agents can cooperate without necessarily meaning to.

Our simulation can be thought of as some sort of graph traversal problem. Each individual agent's goal is to find the optimal path across this graph, but the overall result is that, on a macroscopic scale, these agents end up *creating* this path. Agents walk across the map and damage terrain, which makes the 'optimal' solution change.

It is safe 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 converges on an optimal path, while at the same time trying to minimize the overall terrain that agents need to move over.

This is exactly what humans do in real life. As an example, consider hikers walking through the woods. If the first hiker doesn't see the easiest path to their destination, they will pick a less efficient route to travel on, deteriorating it slightly in the process, but ultimately reaching their goal. Most agents thereafter will naturally follow this path. It does not matter if there is an optimal path 10 meters north if they can not see it. Their secondary pathway will continue to grow until it is almost as good as the optimal one, or becomes conjoined with it.

As another example, consider the lawn in front of the Math Sciences building at the University of Calgary. It is clear that agents (students) take shortcuts off the paved walkways, and cut across the grass to get to their classes. Each student does this presumably because it makes their personal journey that much shorter. Eventually, as more and more students do this, we end up with an established dirt path that is almost as easy to walk on as the paved one, and is used much more often, effectively replacing the old path with a new, shorter path for everyone.

This emergence seems to show a remarkable level of unconscious forethought and swarm behavior on the agent's behalf, even though it is simply a side effect arising from selfish behavior.

V. Conclusion

Ultimately, accepting our assumptions that agents in these simulations act independently and selfishly, and follow certain simple rules, our simulation gives evidence to an interesting idea of cooperation. Specifically, it supports the idea that cooperation need not involve any sort of altruism or communication. Selfish agents can, in fact, cooperate in a way that benefits everyone.

If you wish to experiment with or contribute to this project, please check it out on github, at:
`github.com/Ironykins/wilderness-pathways`

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.