Clint Olsen

Chase Heck

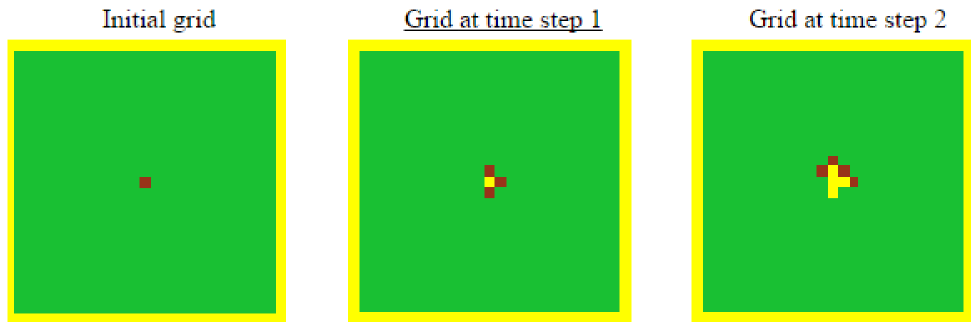
ECEN 4313/CSCI 4830

Project Proposal

For this project, we would like to investigate simulating the spread of fire using concurrency. Present models of this phenomenon are complex and difficult to model mathematically. Instead, we would like to simulate the behavior of a spreading fire using a concurrent program that take into account several factors to model an algorithm. These factors could potentially include the effects of wind, season, and precipitation. This is an interesting problem due to the fact that fire spreads in numerous directions at once and thus lends itself to a multithreaded approach. If handled linearly, this simulation would most likely be inefficient due to the large number of variables. The main difficulty with this problem will be the probabilistic factors that determine whether surrounding areas will catch fire.

The use of this program would be to help model the behavior of forest fires to help park service members better understand the potential behaviors/impacts of fires. This could lead to a greater understanding of where fires will propagate if started in particular area and could also be used as forecasting to quickly mitigate fires if started suddenly.

Programmatically, the basis of the simulation will be modeled using a 2D matrix, composed of “trees.” Each tree will be an element in the array which holds a value (burning/not burning) and monitors its adjacent trees. The adjacent trees hold a probability value which will determine the likelihood of it catching fire. This probability will likely have to be a complex algorithm comprised of multiple factors. The “fire” will start in a random cell and will propagate outwards in accordance to this algorithm.



If approached linearly this program would model spread slowly at best and would not reflect the true propagation of the fire from its origin. Furthermore, spread could only be modeled one cell at a time. Concurrency would allow for multiple adjacent cells to evaluated and changed simultaneously, meaning multiple cells could be spreading and burning. This would result in a faster simulation and more effectively simulate the real behavior of a large-scale fire. We could imagine four threads spreading from the initial cell, creating more threads to handle the cells adjacent to them. Conversely, each cell could be allocated a thread that waits until the fire reaches them. Mutual exclusion will also be important in the case that multiple fires want to spread to the same cell, which could potentially be resolved by lock free marking.

In terms of quantitatively benchmarking the performance of this simulation, we could potentially benchmark a linear version of the program against the proposed concurrent solution. Another option would be to calculate how parallelizable this solution is, after solidifying an algorithm to build the simulation.