The Mathematical Modeling of Forest Fires

Clarice Shaw

Carthage College

cshaw1@carthage.edu

October 14, 2010

Abstract

With worsening severity of forest fires in America, it has become a necessity to understand the behavior of these phenomena. Not only will this understanding assist land managers with preventing the start and spread of fires, but they will also be better able to protect the people living in areas experiencing forest fires. There are many ways to model these fires, but one of the simplest is through weather variables such as temperature, wind, and humidity. Through the creation of a differential equation using these parameters, one is able to determine many behavioral patterns of wildfires. By implementing this equation in a C++ computer program, it was found that fires will burn more of a forest if the wind is blowing in multiple directions, increasing the probability of fire spread. Additionally, it was found that fires which originate closer to the center of a forest will, in the end, burn more trees than those originating near the outer edge. Many other behaviors could be determined with the addition of fire-specific weather parameters as well as fire-fighting measures to the model. This type of information is vital for the prevention and extinguishing of forest fires.

1 Introduction

For those living in the western part of the United States, summer brings more than barbecues, pool parties, and days at the beach. Summer brings fire season. Not only is the west home to some of the most forested areas in America; the climate is ideal for the start of wildfires. A rainy season promotes vegetation growth and is followed by conditions of drought, optimal for the spread of wildfires. As of October 23, 2015, 9,391,601 acres of U.S. land had been burned by wildfires, over triple the year-to-date acreage in 2014. Additionally, Colorado experienced what many believe to be its most destructive fire in history, killing 2 people, destroying 500 homes, and forcing almost 40,000 people to relocate.

Increasing severity of wildfires is pushing many researchers to develop a better understanding of the fires, their causes, and behaviors. Through mathematical modeling, one can predict many of the patterns wildfires display ranging from direction and speed to intensity. This knowledge can prove extremely beneficial to emergency crews, responsible for land management during wildfires, allowing them to prevent continual fire spread and better protect those who are affected by the blaze. Throughout this paper, a mathematical model to determine the affects of wildfires is created as well as a computer simulation to implement the model, allowing the testing of several parameters to determine general behaviors of wildfires. Combined, these tools will help determine the most beneficial course of action in times of wildfire.

2 Definitions and Development

Many factors such as weather, geographical location, and fuel type determine the probability of a wildfire spreading. For this model, we will focus on different elements of weather. Not only are these parameters specific and readily available, but they can be applied to any type of forest, regardless of location and the type of trees it contains. A differential equation which tracks the change in how many trees are burned over time is created. The variable *x* represents the number of trees in a forest while *p* is the probability that a tree will catch fire. To examine the number of trees burnt over time during a wildfire, classifications must be created to determine what states a tree can be in at the conclusion of the fire. Intuitively, we assert that trees can either be affected, or unaffected. The parameter *f* represents the number of affected trees while *x* represents those that are unaffected.

Definition 1 A tree is classified as **affected** if at any point during a wildfire, the tree was on fire.

Definition 2 A tree is classified as **unaffected** if the tree was not on fire at any point during a fire.

Using the elements above, we derive our model.

$$\frac{df}{dt} = xfp - f \tag{1}$$

The first term represents the newly burning trees: this should be proportional to both the unaffected trees *(x)* and the currently burning trees (*f*). The second term represents the currently burning trees who extinguish. Taken together, these give the net change in the number of trees that are burning.

The parameter *p*, representing the probability that fire will spread from one tree to another, is determined using three different weather indicators; temperature, humidity, and wind. For our model, we will assume that higher temperatures *(T)* will indicate a higher probability that fire will spread. Additionally, we assume that higher humidity *(h)* levels will indicate a lower probability that fire will spread. Applying these rules, the simplest form for p is

$$p = \frac{T}{h}.$$

We assume that higher wind speed *(w)* will indicate a higher probability of fire spread. Hence, the term will be found in the numerator of our p-value. However, to introduce wind to the *p* term already derived, one must consider the direction in which the wind is blowing.

Definition 3 Wind is classified as **single-directed** if the wind direction is North, South, East, or West.

Definition 4 Wind is classified as **multiple-directed** if the wind direction is Northeast, Northwest, Southeast, or Southwest.

If wind is blowing in a single-directed manner, the tree to the direction of the blowing wind will have a higher likelihood of catching fire, and thus we scale the probability, *p* by the value *w*, finding our first equation. This scaling will only affect the direction in which the wind is blowing, while others remain unchanged.

$$p = \frac{T+w}{h} \tag{2a}$$

However, if the wind is multi-directed, it will be blowing in a diagonal manner. Therefore, the probability of fire spread in the directions the wind is blowing will not be equal to the wind speed as it is dispersed through multiple directions. This follows from the rules of special triangles, as seen in Figure 1.

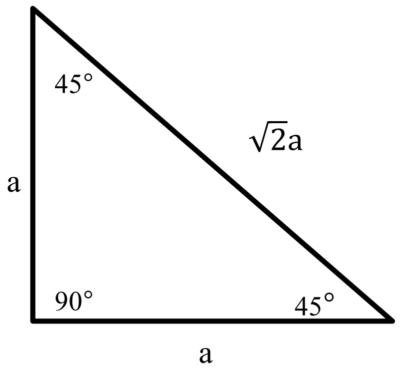

Figure 1: Special Isosceles Right Triangle

Thus, if wind is blowing in a multi-directed fashion, the term w must be scaled by $\sqrt{2}$, revealing the other equation to mathematically model wildfires.

$$p = \frac{T+\frac{w}{\sqrt{2}}}{h} \tag{2b}$$

Computer Programming Implementation

Many professionals utilize computer programming to determine the behavior of wildfires, employing mathematical models to simulate fire conditions. Found in Appendix 1, a C++ program was created to test the behavior of wildfires using the probabilities 2a and 2b. To begin, this program requires four weather parameters to be entered by the user: temperature, humidity level, wind speed, and wind direction. Additionally, it takes two forest-size variables, length and width, which work to determine the size of the forest affected by a wildfire and, in the end, how much of the forest is burned. Lastly, one must enter values which represent where the fire started, dependent upon the size of the forest. Built like a table of nodes, these values mimic coordinates. For example, if one wanted to experiment with a 1000 tree forest, size 10 trees by 100 trees, and desired to begin the fire at the exact middle tree, they would either enter the values 5 and 50 or 4 and 49. Because this forest has even values for both the length and width, one must choose the values which most precisely approximate the middle.

Once started, a function is called that acts to “spread” the virtual fire. Iterating through each tree one at a time, this function finds trees that are on fire and tests to see whether or not it will spread to the trees directly below, above, to the left, and to the right. Next, a random number between 1 and 100 is generated. This value is compared to the probability, derived using the weather criteria entered by the user. If the random number is less than or equal to the specific probability, the fire will spread. Once the last tree in the forest is reached, the program iterates through the forest again. This top-down process continues until each tree that is on fire has been checked. Depending on user preference, this program can return two types of data at the conclusion of running. First, the program can simply return the number of trees burnt during the span of the fire. Additionally, the program is able to return the number of trees burnt between each iteration of the program. This allows for evaluation of the fire over time. Using programs such as this one, professionals are able to determine characteristics of wildfires, allowing for better prevention and understanding.

Model Application

With the introduction of a mathematical model of wildfire spread as well as a program to implement this model, many questions can be answered about the behavior of these fires. One concept we will address is whether or not the location at which a fire starts has an effect on the final percentage of the forest affected. This is done using two computer simulations on identical forests, one of which will simulate an edge-fire, while the other simulates a middle-fire.

Definition 3 An **edge-fire** refers to any fire which begins on a tree located on the absolute right, left, bottom, or top edge of a forest.

Definition 4 A **middle-fire** refers to any fire which begins on a tree located in the exact middle of a forest. If the forest has an even length or width, one will use a number that closest approximates the middle.

The number of trees affected by each fire were recorded for twenty unique probabilities, ranging from .05 to 1 for both forests. Each probability was tested 100 times. By comparing the average number of trees burned for each probability between middle and edge-fires, one is able to determine whether or not start location has any affect on the final number of trees burnt by a wildfire.

We also examine how different types of wind affect the final number of trees burned during a fire. Using three identical forests, we simulate a fire when there is no wind, one with single-directed wind, and one with multi-directed wind. These simulations are done 100 times across ten unique probabilities ranging from .05 to 1. By recording and averaging the number of trees burned in each case, one can see the differences that occur when different wind patterns are present during a fire.

3 Results

In order to validate the results of each experiment, it is important to compare the simple logistic growth Equation 1 with the results from the C++ program found in Appendix 1. When Equation 1 is plotted with respect to time and probability of 1 in Mathematica using the NDSolve command, we create Figure 2. Similarly, by running the program over 10 probabilities ranging from .05 to 1 100 times, and averaging the number of time steps and trees burnt, we create Figure 3.

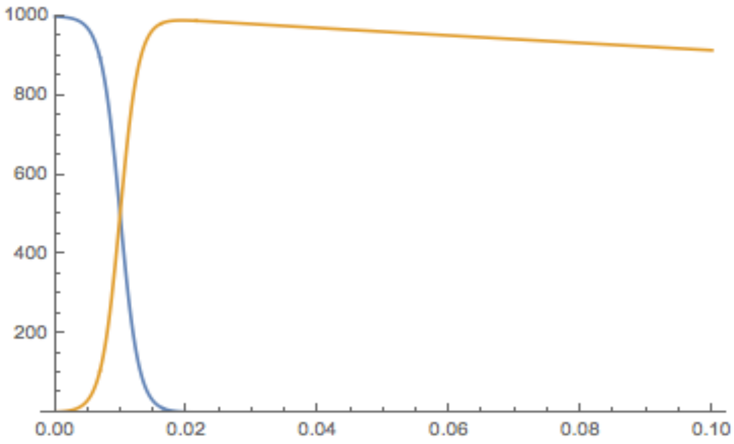

Figure 2: Solution to the system of differential equations $\frac{df}{dt} = xfp$ and $\frac{dx}{dt} = xfp - f$ with respect to time.

Figure 2 plots the solution to the system of linear equations representing the number of affected and unaffected trees at a given point in time. Plotting the number of trees along the *y*-axis and time along the *x*-axis, it can be seen that as the number of unaffected trees (represented by the blue curve) decreases, the number of affected trees (represented by the orange curve) decreases.

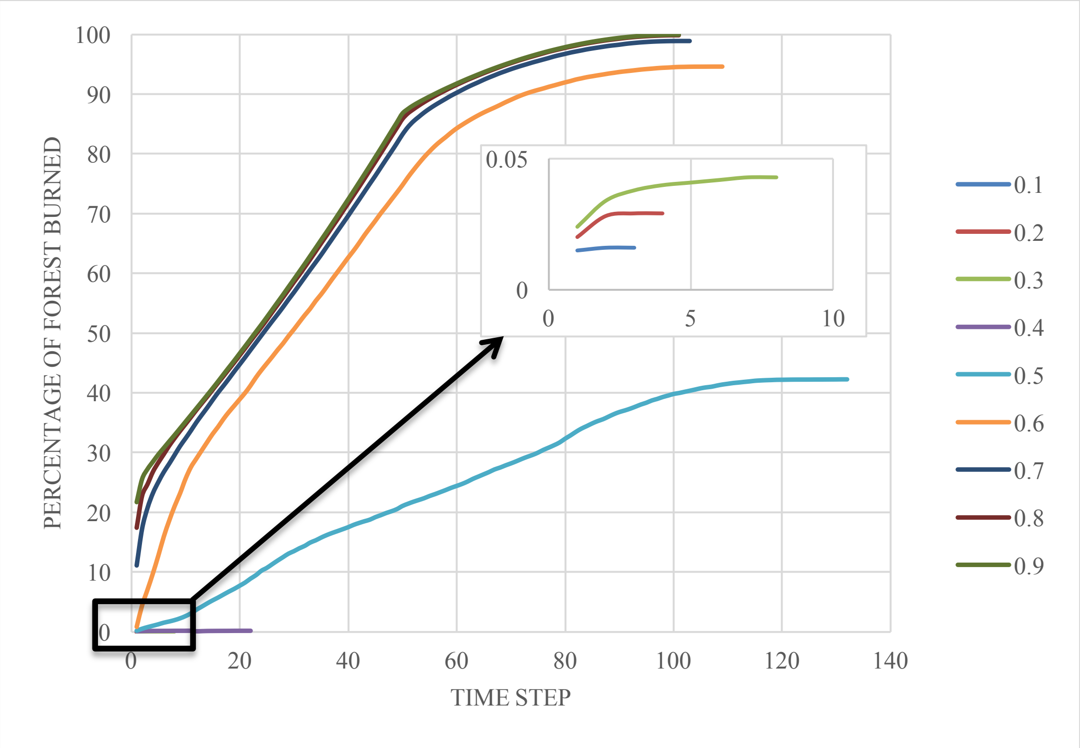

Figure 3: Average Percentage of Forest Burned Over Time by Probability

Although Figure 2 and Figure 3 display graphs with essentially the same shape, Figure 2 shows a more rapid increase than Figure 3. This is due to the disregard for spatial relationships in Mathematica. In the C++ program, trees are only able to catch fire if their neighbor is on fire. However, using Mathematica allows any tree to catch fire, regardless of their neighbor. Hence, the number of burning trees is able to increase at a more rapid rate. In addition to this, Figure 2 begins to decrease at a certain point. This is because of the *f* parameter which is subtracted at the end of Equation 1, representing trees that were once on fire, but later were either extinguished or went out naturally. The action of going out or being extinguished is not included in the computer application of this mathematical model, and therefore results from computer simulations will not show a decrease in the number of trees burnt at the end.

Figure 3 exposes many interesting results which are helpful when considering further experimentation. First, note that the curves have differing lengths, or the fires burned for differing amounts of time dependent upon the probability of fire spread. The longest fire was for *p*=0.5. The fires below it burned for short periods of time because the probability of fire spread was not high enough to sustain a long-burning blaze. Similarly, the curves above p=0.5 were short because their probabilities were high enough to burn trees at a fast rate. Above this threshold at *p*=0.5, higher probabilities also display a similar burn pattern, noticed through similarity in the shape of their curves. For probabilities .7 and higher, the forest burns in its entirety.

Probability (p)	Avg % of Forest Burned
0.05	0.0127
0.1	0.0178
0.15	0.022
0.2	0.0367
0.25	0.0527
0.3	0.0645
0.35	0.1538
0.4	0.3957
0.45	2.856
0.5	42.8192
0.55	82.014
0.6	90.94
0.65	94.601
0.7	98.8804
0.75	99.4821
0.8	99.8142
0.85	99.9483
0.9	99.9874
0.95	99.9992
1	100

Table 4: Average Percentage of Forest Burned by Middle-Fire

Probability (p)	Avg % of Forest Burned
0.05	0.0129
0.1	0.0151
0.15	0.0175
0.2	0.0251
0.25	0.0286
0.3	0.0386
0.35	0.1022
0.4	0.2016
0.45	0.5702
0.5	22.963
0.55	58.125
0.6	72.98
0.65	83.8749
0.7	95.864
0.75	95.517
0.8	98.8095
0.85	97.9452
0.9	99.9913
0.95	99.9995
1	100

Table 5: Average Percentage of Forest Burned by Edge-Fire

Through the use of the C++ Program in Appendix 1, 100 wildfire simulations were run for each probability of fire spread on two different forests, one of which contained an edge-fire, the other a middle fire. Recorded in Tables 4 and 5, the percentage of the forest burned in each run were then averaged.

By simply examining these numbers, it seems as though they only vary slightly, with middle-fires burning more than edge-fires. This is likely because with an edge fire, there are only three neighbor trees to spread fire throughout the entire forest rather than three. Although not included in the study, this pattern would likely continue for corner trees, creating an even smaller percentage burned after a fire. The difference is most obvious in simulations using a probability between .45 and .9. This variation can also be seen in Figure 6, a graphical representation of Tables 4 and 5.

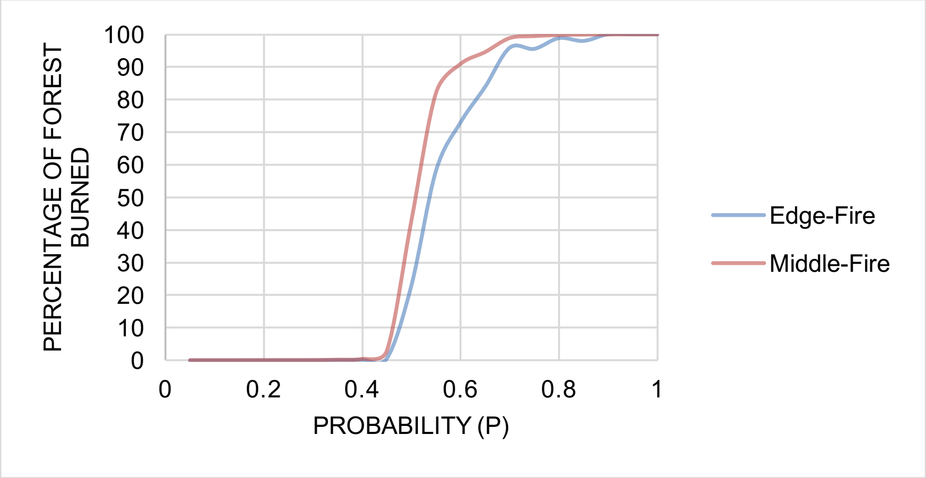

Figure 6: Average Percentage of Forest Burned by Edge and Middle-Fires from P=0 to P=1

Next, using the C++ program in Appendix 1, 100 wildfire simulations were run for each probability of fire spread on three different forests. One of these forests was experiencing no wind, while another had single-directed wind and the last multi-directed wind. Recorded in Figure 7 are the average percentages of the forest burned in each run.

Probability (p)	No Wind	Single-Directed	Multi-Directed
0.05	0.01	0.01	0.02
0.10	0.02	0.02	0.02
0.15	0.02	0.02	0.05
0.20	0.03	0.03	0.12
0.25	0.04	0.05	1.31
0.30	0.10	0.08	15.18
0.35	0.12	0.16	35.42
0.40	0.44	0.52	75.54
0.45	2.53	5.50	98.22
0.50	40.26	61.36	99.86
0.55	83.68	82.49	99.96
0.60	88.12	92.78	99.98
0.65	96.54	97.97	99.99
0.70	97.86	98.11	100.00
0.75	98.52	99.60	100.00
0.80	99.82	99.85	100.00
0.85	99.95	99.96	100.00
0.90	99.99	100.00	100.00
0.95	100.00	100.00	97.00
1.00	100.00	100.00	100.00

Table 7: Average Percentage of Forest Burned with No Wind, Single-Directional Wind, and Multi-Directional Wind

Analyzing the values in Table 7, it appears that multi-directed wind causes the greatest number of trees to be burned during a forest fire. Thus, although the wind speeds are scaled by $\frac{1}{\sqrt{2}}$ due to the rules of special triangles, illustrated in Figure 1, the presence of multi-direction wind leads to a larger percentage of trees burned by forest fires. This result was expected, since wind increases the likelihood of fire spreading from tree to tree. Following multi-directed wind, single-directed wind shows the second largest percentage of trees burned, with forests experiencing no wind having the least percentage of trees burned. This information is displayed graphically in Figure 8.

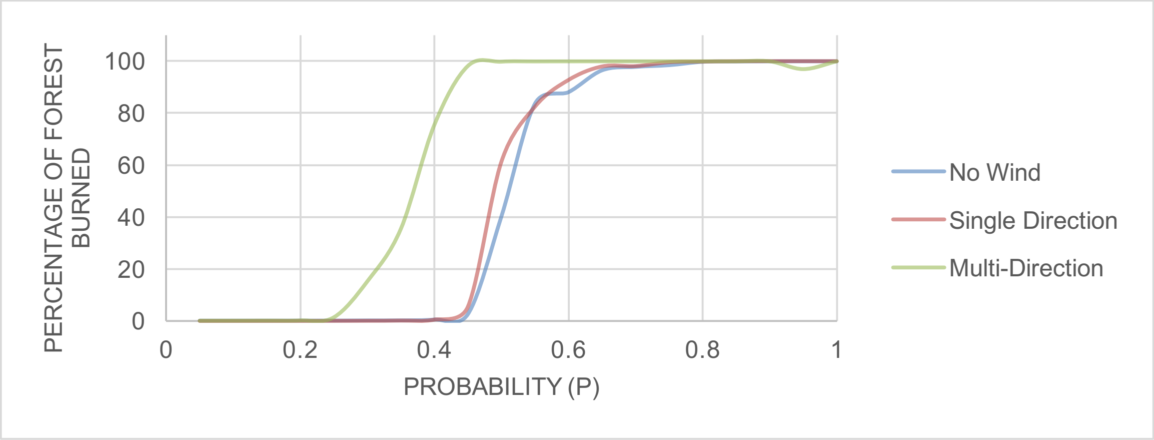

Figure 8: Average Percentage of Forest Burned with No Wind, Single-Directional Wind, and Multi-Directional Wind

4 Conclusion and Directions for Further Research

Through the creation of a mathematical model of forest fires and a computer program to implement this model, one is able to study how wind affects the spread of wildfires, as well as how the start location of fires can change the final percentage of forests burned. First, we see that the closer a fire is started to the interior of a forest, the greater percentage of the forest will burn throughout the fire. Additionally, it was found that multi-directed wind conditions allow for the greatest percentage of forest burned by fires, followed by single-directed wind and no wind. This information can prove very helpful for land managers. Not only will this make them better-able to prevent these blazes from spreading so rapidly, but they will have the ability to protect people and their property from harm.

Many additions can be made to this model to further enhance its benefits. To begin, this model does not involve any actions to thwart the spread of fire such as fire-fighters, or the addition of firebreaks. Because there is always action taken to contain forest fires in real life, this addition would make the model much more realistic while also allowing experimentation with placement of firefighting strategy. Additionally, this model does not account for the weather caused by fires such as increased temperatures in certain areas and fire whirls, which increase the wind speed within burning trees. Lastly, it would be extremely beneficial to model which direction the wildfire spreads in, allowing for more simplified evacuation of people living in affected areas. With increasing severity of wildfires in recent years, studies such as these which can illustrate the behavior of wildfires are very helpful to professionals working to prevent the spread of fires, and protect those affected by these fires.

References

[1] United States. National Park Service. "Wildland Fire: Understanding Fire Danger | U.S. National Park Service." *National Parks Service*. U.S. Department of the Interior, n.d. Web. 02 Mar. 2016.

Appendices

Appendix 1 – C++ Program

Main.cpp

```
#include <iostream>
#include "Tree.h"
#include "Forest.h"
using namespace std;

int main()
{
   srand(time(NULL));

      Forest f(100, 100, 0,"NONE", 180, 2);
      f.startFire(50, 50);
      f.spreadFireCredentials();

   return 0;
}
```

```
//length, width, wind speed, wind direction, temp, humidity
```

Tree.h

```
#pragma once
#include <vector>
#include <string>

using namespace std;

class Tree
{
public:
    Tree();
    bool isBurning;
    bool isChecked;
    char isBurnt;

private:


};
```

Tree.cpp

```
#include "Tree.h"
Tree :: Tree()
{
    isBurning=false;
    isChecked=false;
    isBurnt='O';

}
```

Forest.h

```
#pragma once
#include "Tree.h"
#include <vector>
#include <string>

using namespace std;

class Forest
{
public:
    Forest(int l, int w, int s, string d, int t, int h);//
    Forest(int l, int w, int p);
    void setForestName(string n);//
    void setForestSize();//
    int getForestSize();//
    int getLength();//
```

```
    int getWidth();//
    vector<vector<Tree>> getForest();//
    void startFire(int column, int row);
    void spreadFireCredentials();
    int percentageBurned();
    bool isFinished();


private:
    vector<vector<Tree>> forest;
    int length;
    int width;
    int ForestSize;
    int windSpeed;
    int temperature;
    int humidity;
    string windDirection;

};
```

Forest.cpp

```
#include "Forest.h"
#include "Tree.h"
#include <iostream>
#include <ctime>
#include <cstdlib>
#include <cmath>

using namespace std;

//--
Forest :: Forest(int l, int w, int s, string d, int t, int h)
{
    length=l;
    width=w;

    forest.resize(l);
    for (int i=0; i<l; i++)
    {
        forest[i].resize(w);
    }

    for(int i=0; i<length; i++)
    {
        for(int j=0; j<width;j++)
        {
```

```
            forest[i][j].isBurning=false;
            forest[i][j].isChecked=false;
            forest[i][j].isBurnt='O';
        }
    }

    ForestSize=l*w;
    windDirection=d;
    windSpeed=s;
    temperature=t;
    humidity=h;
}
//--
int Forest:: getForestSize()
{
    return ForestSize;
}
//--
int Forest::getLength()
{
    return length;
}
//--
int Forest::getWidth()
{
    return width;
}
//--
vector<vector<Tree>> Forest :: getForest()
{
    return forest;
}
//--
void Forest::startFire(int column, int row)
{
    if(row<=length && column <= width)
    {
        forest[column][row].isBurning=true;
        forest[column][row].isBurnt='X';

    }
    else
    {
        cout << "The tree you wish to strike is out of range!" << endl;
    }
}
//--
void Forest:: spreadFireCredentials()
{
    while(!isFinished())
    {
```

```
for(int i=0; i<length; i++)
{
   for(int j=0; j<width;j++)
   {
      if(forest[i][j].isBurning==true && forest[i][j].isChecked==false)
      {
         forest[i][j].isChecked=true;
         forest[i][j].isBurning=true;
         forest[i][j].isBurnt='X';
         //checkAbove(column, row-1, burnProbability);
         if(j!=0 && forest[i][j-1].isBurning!=true)
         {
            double weatherVariable;
            if(windDirection=="N")
            {
               weatherVariable=((temperature+windSpeed)/(humidity));
            }
            else if(windDirection=="NW" || windDirection=="NE")
            {
               weatherVariable=((temperature+(windSpeed*(1/(sqrt(2))))/(humidity)));
            }
            else
            {
               weatherVariable=(temperature/(humidity));
            }

            int x;
            x= (rand()%100);

            if(x<=weatherVariable)
            {
               forest[i][(j-1)].isBurning=true;
               forest[i][j-1].isBurnt='X';
            }

         }
         //checkBelow(column, row+1, burnProbability);
         if(j!=width-1 && forest[i][j+1].isBurning!=true)
         {
            double weatherVariable;

            if(windDirection=="S")
            {
               weatherVariable=((temperature+windSpeed)/(humidity));
            }
            else if(windDirection=="SW" || windDirection=="SE")
            {
                weatherVariable=((temperature+(windSpeed*(1/(sqrt(2))))/(humidity)));
            }
            else
            {
```

```
        weatherVariable=(temperature/(humidity));
    }

    int x;
    x= (rand()%100);

    if(x<=weatherVariable)
    {
        forest[i][(j+1)].isBurning=true;
        forest[i][j+1].isBurnt='X';
    }


}
//checkLeft(column-1, row, burnProbability);
if(i!=0 && forest[i-1][j].isBurning!=true)
{
    double weatherVariable;

    if(windDirection=="W")
    {
        weatherVariable=((temperature+windSpeed)/(humidity));
    }
    else if(windDirection=="SW" || windDirection=="NW")
    {
        weatherVariable=((temperature+(windSpeed*(1/(sqrt(2))))/(humidity)));
    }
    else
    {
        weatherVariable=(temperature/(humidity));
    }

    int x;
    x= (rand()%100);

    if(x<=weatherVariable)
    {
        forest[i-1][j].isBurning=true;
        forest[i-1][j].isBurnt='X';
    }

}
//checkRight(column+1, row, burnProbability);
if(i!=length-1 && forest[i+1][j].isBurning!=true)
{
    double weatherVariable;

    if(windDirection=="E")
    {
        weatherVariable=((temperature+windSpeed)/(humidity));
                      }
```

```
                        else if(windDirection=="NE" || windDirection=="SE")
                        {
                             weatherVariable=((temperature+(windSpeed*(1/(sqrt(2))))/(humidity)));
                        }
                        else
                        {
                            weatherVariable=(temperature/(humidity));
                        }

                        int x;
                        x= (rand()%100);

                        if(x<=weatherVariable)
                        {
                            forest[i+1][j].isBurning=true;
                            forest[i+1][j].isBurnt='X';
                        }


                    }

                }

            }
        }

            cout << percentageBurned() << endl;

    }
}
//--
int Forest:: percentageBurned()
{
    int numBurned=0;

    for(int i=0; i<length; i++)
    {
        for(int j=0; j<width;j++)
        {
            if(forest[i][j].isBurning==true)
            {
                numBurned ++;
            }
        }
    }

    return numBurned;
}
//--
bool Forest::isFinished()
{
```

```
    bool retVal=true;
    for(int i=0; i<length; i++)
    {
        for(int j=0; j<width;j++)
        {
            if(forest[i][j].isBurning==true && forest[i][j].isChecked==false)
            {
                retVal=false;
            }

        }
    }
    return retVal;
}
//--
```