Fire in the Bosque- A Basic Comprehension of Fire in its Environment

Supercomputing Challenge Final Report

March 31, 2005

Team 050
Rio Rancho High School
Team Members
Christopher Morrison
Teachers
Dale Henderson

Introduction

Every year about 5 million acres of woodland forests burn in the U.S. In recent years New Mexico has been a victim in these fires. Specifically the Cerro Grande fire rampaged through Los Alamos and in more recent times the New Mexico Rio Grande Bosque has been a victim. The Bosque was hit by fire on two occasions last year and each one brought great damage to the wildlife and humans living in the area.

In order to keep losses to a minimum, New Mexico should, if it doesn't have a plan already, have a way to predict fire occurrences and their behavior in the Bosque and across the state. With this in mind, team 050 came up with this problem for the 2005 New Mexico AiSC Challenge-

"What problems would a fire in the Bosque pose to the environment and its people."

In order to achieve this goal , several objectives must be laid down:

1. Have an algorithm to map heat flow
2. Research fire characteristics and simulate fire
3. Research and simulate how heat flow changes under different circumstances
    a. wind, rain, elevation,
4. Research and simulate how fire acts with varying fuel sources and circumstances
5. Accurately map the Bosque and its surroundings for variables related to objective three and four.
6. Put all these aspects into a program and decipher environmental damage to nature and its rural surroundings.

This won't be easy, however and a lot of work must be put into making this project a success.

Research

Fire

Fire is one of man's best friends, however that can quickly change when fire gets out of control when this happens how do you predict how the fire will act in order that you can slow or even stop the fire when they get out of control However, in order to predict how fire will act you first must try to understand fire..

Fire is a combination of three main ingredients, heat, fuel, and oxygen and can be expressed on the chemistry level in this equation

6C(10)H(15)O(7)+Heat==C(50)H(10)O+10CH(2)O

Wood + Heat is Char + GassesThis is only part one of the combustion. As the Gasses are produced they react to the oxygen and produce heat.

CH(2)+O(2)==H(2)O +CO(2) + CO + C + N(2)

gasses Oxygen water carbon dioxide, monoxide and nitrogen

This heat in turn causes the heated carbon atoms to rise over the cooler air and produce light and color at a magnitude of the fuel that is burned. . The fire will then continue to burn until it has ran out of either heat oxygen or fuel gasses. As the flame spreads it spreads opposite to gravity and follows wind patterns across the landscape. The flame is hotter than the surrounding air so it will travel in a direction opposite of gravity or in other words up. The then fire then burns and spreads in a self-perpetuating manner or in other Words the fire continues to burn as a result of its own heat. The Fire gains heat as it breaks the bond of compounds and burns the basic flammable elements-carbon and hydrogen.

Everything has two phases of flammability

1. Piloted ignition temperature; phase where compound will ignite/break down if lit is exposed to a spark
2. Unpiloted ignition temperature; phase where the compound will instantaneously combust/break down

The unpiloted ignition temperature of fire is much hotter than piloted ignition is commonly known as the flash point. Every compound has a different flash and piloted flash point and no matter what they always have one.

-Common wood's (carbon based) flash point is 300 degrees Celsius or 573 Kelvin

-Common wood gives off 9,894kilocalories per Kg burned or 6880 Btu per pound

-Common wood has a high concentration of water, especially living wood, which under normal exposed conditions greatly reduces the Btu of calorie factor (www.)

The mass density and shape of the fuel also effects the rate of fuel consumption. The higher the mass the less oxygen there is the slower a fuel will burn. Dense materials burn slower just because there is more to burn and the fire uses up the oxygen in the area. Lastly an object with more surface area will burn faster than an object of the same substance with less surface area.

Heat production depends much on these three factors listed above.

To sum it all up the more oxygen available the hotter the fire

Bosque

The Bosque is an area of dense foliage along the Rio Grande River that is the habitat to a diverse community of wetland animals. However today humans are included on the list. The Bosque contains many fuels that could start a fire including

Trees; Cottonwood, Russian Olive, and elm trees

Brush; oak, and other brush

Ground fuel; A thick layer of leaves1-5 in deep covering the ground under and around trees

The brush in the Bosque is quite thick from centuries of human protection the Bosque hasn't been cleared out by a fire.

Recently the Bosque has experienced fires that have damaged its ecosystem and endangered human interests. The Atrisco (150 acre) and Montano (113 acre) fires were devastating to the Albuquerque Bosque burning down several structures. They caused the launch of an effort to safeguard the Rio Grande Bosque from the effects of fire.

Methodology

Description of program

The program that is being developed is a basic graphical heat flow program for the Bosque. It will

-imitate heat flow in a three dimensional environment
-imitate fire and its characteristics
-have a semi-accurate setting in the Albuquerque Bosque close to the Montano bridge
-take into account for many environmental situational and heat flow factors
-give recorded results of the program

It is not expected to

-perfectly model fire
-have an **exact** Bosque setting (semi-exact)
-model human intervention (possibly if there's time)

Factors in the Program

For this project, it was decided that heat flow (fire) should be very simple. This program operates on sensible conjectures based on how fire works. The heat flow mathematical model, at its simplest form has two rules-

1. Heat flow is an average of the area around it per unknown unit in time.

2. If the temperature of the area represented by the patch goes over 300 degrees Celsius (the piloted flash point), then there is a fire where that patch is, and a fire multiplier will be added to the equation.

Then there are environmental factors to complicate the fire's heat flow including

1. Fuel; The fire will continue to burn until all fuel is exhausted and then will comply with normal heat flow rules.
2. Different fuel types; Different fuel types (brush, forest, open, water) have different amounts of fuel

3. Elevation; Lower ground level spreads fire to higher ground faster than level ground and vice versa.

After that there are situational factors that effect the flow of heat to another extent

1. Wind ; Wind spreads fire faster in its direction, and slower in the opposite direction.

2. Humidity/ground; The more the humidity (wet ground) the less the fire spreads.

3. Rain; increases humidity to 100 percent and cools fire to the rain's temperature.

Lastly in this program there are the variables that you can edit in this program.

Note; The program was set up so that the program variables can be edited; this program is based on simple logical assumptions about fire, and no constants are known for the heat flow, but the variables can be edited to the point to where the program can be quite accurate. There was also an added variable that lets you place fire and the numbers of fires you want.

Mathematical Model

For the basic heat flow the mathematical model is very simple.

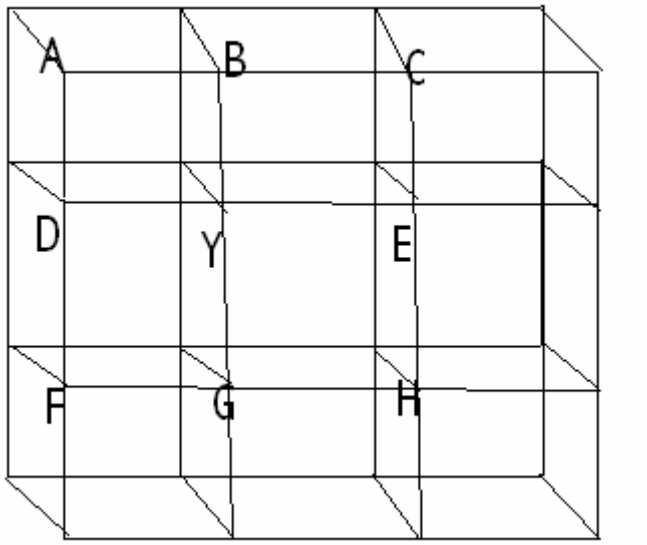
As everyone knows a cube has eight corners. For every corner that a surrounding cube touches on the original cube, more heat flows from the surrounding cube into the original cube. Now every corner of an adjacent cube that is touching the original cube (z), being averaged represents a variable multiplied by z (for example a(variable)*(z)) and every variable represents

another adjacent cube. Here is the

equation

$$y=\frac{(a*z+b*z+c*z+d*z+e*z+f*z+g*z+16*\text{ambient temperature}+8y)}{48}$$

The Y equation



So say A,B,C,D,E,F, and G =283 degrees Kelvin H=633 ambient temperature=281 y=288

Y(per unit in

time)=(283*2+283*4+283*2+283*4+283*4+283*2+283*4+623*2+16(281)+8*288)/48

A B C D E F G H ambient Y #z's

Y=14272K/48 Y=297.33K = 24.33 degrees Celsius

So accounting for heat flow per unit of time when time =2; y=24.77 degrees Celsius from an original 15 degrees Celsius.

Note

I am using Kelvin because it is a linear degree function starting from absolute zero rather than Celsius at 273 Kelvin. Also the program uses a coordinate value system rather than a variable based system to store values

When an area is over 300 degrees Celsius (573 Kelvin) it is at wood's(carbon based) piloted ignition point, or in other words a fire (www.howstuffworks.com). So in this program when an area, such as y, is on fire the heat generated by the rapid

oxidization of the fuel is represented by a multiplier Fire.. Fire is a variable in the program that can be edited basically it is a function of how much heat the fuel gives off when it is burned which varies with the quality of fuel being burned..

If y<300 degrees Celsius and fuel>0

$$\left(\frac{y=(\quad a*z+b*z+c*z+d*z+e*z+f*z+g*z+16*\text{ambient temperature}+8y)}{48}\right)*\text{Fire}$$

Y fire equation

Fire has a limit in the program so that when the fire burns it burns fuel and when the amount of fuel left in the area is equal to zero then the heat flow equation is used again. Also in the program is included starting temperatures that reflect the availability of fuel and in the case of water no fuel.

The environmental types one for each area are;

	Starting temperature
Forest: fuel=40	300C 573K
Brush: fuel=20	300C 573K
Open: fuel=10	325C 598K
Water: fuel=0	N/A

For elevation the rule is that heat and fire travel upward faster than down, so the program finds the y averaging area and finds the elevation of all the area around it and stores a temporary value to all of them that increases the lower area's temperature by a variable multiplied by the difference and vice versa.

(A(ele)-Y(ele))*A(temp)*multiplier=temporary temperature

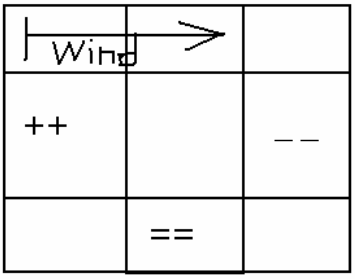
F Y equation(temperory temp)=new temp

Basically all the environmental and heat flow factors are constants for this program due to the fact that they are the base logic of this program and also because it's a fire in the Bosque not any place (that would take way too long to program in to be a variable) .

The situational factors are not solid. This is because of the fact that the situational change from different situations is unknown. So all of the situational factors are variables that can be changed on both the situation, and how much the situation affects the outcome.

So here they are

Wind: For wind the program takes the temperature values of the coordinates around the area, takes the direction of the wind, puts them into a temporary value chart, decreases the temperature of the temporary temperature value pointing away from the wind in accordance to the area it is averaging for and increases the temperature of the patch in the direction of the wind. The areas don't maintain their temporary values. The wind speed is a multiplies of the percentages that dictate temporarily value heat loss or heat gain.



Humidity: Humidity/ground wetness is basically slows heat flow. It makes it to where the temperature gain for temperature<100 is divided by a variable divisor times the humidity amounts.

Rain: Rain raises the humidity, but also lowers the temperature of the fire surrounding it by dividing by a number >1 controlled by a variable multiplied by the rains intensity.

I realize that this is very confusing for you to follow so maybe the program will make more sense to you.

Program

Fire in the Bosque

v.1 This version was the original “learning program” that was made only slightly after C++ was learned.

v.2 This version incorporates most of the variables that are supposed to be in the program however the MSDOS output is unacceptable, so better graphics were needed.

v.3 This version isn’t done yet. However, it looks very promising due to the fact that heat flow and basic fire work.

V.1

Note; If you want to see the more advanced code please look at version two or version three

```
//Fire in the Bosque v.1
#include<iostream.h>

void ambientTemperature();
void precipation();
void wind();       //all the functions up to here basically establish situational varibles
void fireTransfer();//The hard coded junk in the program
void forest();     //an attempt at establishing enviornmental varibles
void water();
void open();

 void one(); void two();void three(); void four(); void five(); void sixx();
 void seven(); void eight(); void nine(); void teno(); void eleven();
 void twelve(); void thirteen(); void fourteen(); void fifteen();void sixteen();
 void seventeen(); void eighteen(); void ninteen(); void twenty(); void twentyone();
 void twentytwo(); void twentythree(); void twentyfour();void twentyfive();

int burnTime;
int timeh;
int type;
int heatGain;
int topHeat;
```

```
int burntime;
int ambientTemp;
int speed;
int direction;
int rain;
int temp;

int on=25;  int tw=25;  int th=25;  int fo=25;  int fi=25;  int si=25;  int se=25;  int ei=30000;  int ni=25; int
ten=25;  int ele=25;int twe=2500;  int thi=25;int fou=25;  int fif=25; int six=25;  int sev=25;  int eig=25;  int
nin=25;  int twen=25;int twon=25;  int twtw=25;  int twth=25;  int twfo=25;  int twfi=25;

//*************************************************************************
main()
{
   fireTransfer();
   system("Pause");
}

//*************************************************************************
/*void ambientTemperature()
{

          cout<<"What temperature is it outside";
          cin>>ambientTemp;

}
//*************************************************************************
void precipation()
{

    cout<<"is it raining outside rate 1-10 10 being more rain 1 less 0 none";
    cin>>rain;


}
//*************************************************************************
void wind()
{
   int blah;

 cout<<"what is the speed of the wind?";
 cin>>speed;
for(blah=1;blah>=100;blah++)
 {cout<<"what is the direction of the wind?";
 cin>>direction;
 {
 switch(direction)
 {
  case ('n'):{ direction=1;
             break;  }
  case ('w'): { direction=2;
              break;  }
  case ('s'): {direction=3;
           break;   }
  case ('e'): {direction=4;
           break;    }
  default: {cout<<"please retype";}
}
}
}
}*/
//*************************************************************************
```

```
void fireTransfer()
{


  int continu;
  cout<<'\n'<<"how many minuites do you want to iterate for?";
  cin>>continu;

  for(timeh=0;timeh<=continu;timeh++)
  {
    cout<<on<<"f    "<<tw<<"f    "<<th<<"w ###  #####"<<"     "<<fo<<"w    "<<fi<<'f'<<'\n'<<'\n'
       <<si<<"f    "<<se<<"f    "<<ei<<"w ##########"<<"    "<<ni<<"w    "<<ten<<'f'<<'\n'<<'\n'
       <<ele<<"f    "<<twe<<"o    "<<thi<<"w #######"<<"       "<<fou<<"w    "<<fif<<'f'<<'\n'<<'\n'
       <<six<<"o    "<<sev<<"o    "<<eig<<"w ######"<<"       "<<nin<<"w    "<<twen<<'o'<<'\n'<<'\n'
       <<twon<<"o    "<<twtw<<"o    "<<twth<<"w ####"<<"          "<<twfo<<"w
"<<twfi<<'f'<<'\n'<<'\n'<<'\n'<<'\n';

    system("Pause");
     one();
     two();
     three();
     four();
     five();
     sixx();
     seven();
     eight();
     nine();
     teno();
     eleven();
     twelve();
     thirteen();
     fourteen();
     fifteen();
     sixteen();
     seventeen();
     eighteen();
     ninteen();
     twenty();
     twentyone();
     twentytwo();
     twentythree();
     twentyfour();
     twentyfive();
}
}
 //*****************************************************************************
     // long evil monster of functions the
     void one()
      {          temp=on; void forest();


                   if(temp<300)
                  {on=(on+tw+25+25+si)/(5)*1;}
                    else
                  { cout<<"fire on one!";

                  on=on+20;

                  }
                  return;
      }
      void two()
```

```
 {
         temp=tw; void forest();
                 tw=(on+tw+th+se+25)/(5)*1;
                 return;
  }
void  three()
{       temp=th; void water();
              th=(tw+25+15+ei+th)/(5)*.8;
                 return;
}
void four()
{         temp=fo; void water();
                 fo=(25+15+fo+ni+fi)/(5)*.8;
                 return;
 }
 void five()
 {        temp=fi; void forest();
                 fi=(25+25+fo+fi+ten)/(5)*1;
                 return;
 }
 void sixx()
 {        temp=si; void forest();
                 si=(si+25+ele+se+on)/(5)*1;
                 return;
  }
  void seven()
  {       temp=se; void forest();
                 se=(se+si+ei+tw+twe)/(5)*1;
                 return;
   }
   void eight()
   {      temp=ei; void water();
                 ei=(ei+se+th+thi+15)/(5)*.8;
                 return;
    }
    void nine()
    {     temp=ni; void water();
                 ni=(ni+ten+15+fo+fou)/(5)*.8;
                 return;
     }
     void teno()
     {    temp=ten; void forest();
                  ten=(ten+25+fi+ni+fif)/(5)*1;
                 return;
      }
      void eleven()
      {   temp=ele; void forest();
                  ele=(ele+twe+25+six+si)/(5)*1;
                 return;
       }
       void twelve()
       {   temp=twe; void open();
                  twe=(twe+ele+thi+se+sev)/(5)*1.2;
                 return;
        }
        void thirteen()
        {  temp=thi; void water();
                  thi=(thi+twe+15+ei+eig)/(5)*.8;
                 return;
         }
         void fourteen()
         { temp=fou; void water();
```

```
            fou=(fou+15+fif+nin+ni)/(5)*.8;
           return;
   }
 void fifteen()
 {  temp=fif; void forest();
            fif=(fif+ten+twen+fou+25)/(5)*1;
           return;
 }
 void sixteen()
 {   temp=six; void open();
            six=(six+sev+25+ele+twon)/(5)*1.2;
           return;
  }
  void seventeen()
  {  temp=sev; void open();
            sev=(sev+six+eig+twe+twtw)/(5)*1.2;
           return;
   }
   void eighteen()
   { temp=eig; void water();
            eig=(eig+15+sev+thi+twth)/(5)*.8;
           return;
    }
    void ninteen()
    {temp=nin; void water();
            nin=(nin+twen+15+fou+twfo)/(5)*.8;
           return;
    }
    void twenty()
    {temp=twen; void open();
             twen=(twen+nin+twfi+fif+25)/(5)*1.2;
           return;
    }
    void twentyone()
    {temp=twon; void open();
             twon=(twon+25+25+twtw+six)/(5)*1.2;
           return;
    }
    void twentytwo()
    {temp=twtw; void open();
             twtw=(twtw+twth+twon+25+sev)/(5)*1.2;
           return;
   }
   void twentythree()
    {temp=twth; void water();
             twth=(twth+twtw+eig+15+25)/(5)*.8;
           return;
    }
  void twentyfour()
  {temp=twfo; void water();
             twfo=(twfo+twfi+nin+15+25)/(5)*.8;
           return;
  }
  void twentyfive()
  {  temp=twfi; void forest();
             twfi=(twfi+twfo+twen+25+25)/(5)*1;
           return;
  }
```

```
//*********************************************************************************
```

```
void forest()
{
		burnTime=100;
		topHeat=300;
		type=1;
		heatGain=1;
		return;


}

//*****************************************************************************

void water()
{
		  burnTime=15;
		  topHeat=350;
		  type=2;
		  heatGain=2;
		  return;

}

//*****************************************************************************

void open()
{
		burnTime=35;
		topHeat=225;
		type=3;
		heatGain=1;
		return;

}
//*****************************************************************************
```

V.2

Note; This version still has bugs but it works in all the areas described in the methodology section.

```
#include<math.h>
#include<iostream.h>
#include<iomanip.h>


/*AiSC 2005 Fire in the Bosque v.2
This code was written right after I decided that my v.1 was complete junk
*/

void setValues();     //sets enviornmental varibles for the Bosque
void cxyout();        // af function to cout the XY coordinate plane
void heatflowXY();    //the upper function that checks every area
void fireOrNot();     // finds if the temperature and fuel are great enough to start a fire
void fire();          // Uses the y equation and the fire multiplier to decide the new value of the area
void startFire();     // set up function that places the fire wherever at whatever temperature
void heatTransferEquation(); //if the temperature and fuel are not good enough to have a fire the area temp is
calculated here
void winds();

double fuel[7][16];         //holds the amount of fuel each area has
double Bosque[7][16];       //Holds Bosque temperature values I sometimes call it area
double elevation[7][16];    // holds the Bosque elevation
double fuelType[7][16];     //holds the Bosque fuel type (forest open brush)
double BosqueE[7][16];      //temporary place holding varible for the wind and elevation


char wind;
char direction[3];

float N,S,E,W,Sw,Nw,Ne,Se;
float heatA;
const int water=10;
int humidity;
int set;
int x;
int y;
double windspeed;
 main()
{  cout<<"What is the ambient temperature?"<<'\n';
     cin>>set;      //semi-nested  semi top down
    setValues();
              cxyout();
```

```
            startFire();
    heatflowXY();
    return 0;
}
//*****************************************************************************
void cxyout()
{                                                                       // y=1 x++ 1,1 1,2 1,3 1,4 ect
for(y=14; y>=1; y--)                    //this is what draws to the screen through cout commands
   {  for(x=1; x<=5; x++)
     { cout<<setprecision(4)<<Bosque[x][y]<<"\t\t";
     }
}   cout<<"\n\n";
}
//*****************************************************************************
void heatflowXY()
{            int time;
             int go;
  cout<<"how many minuites do you want to go for?";
                      cin>>go;
 for(time=0; time<=go; time++)
 { for(y=14; y>=1; y--)          //checks one area at a time
       {for(x=1; x<=5; x++)
                                          { fireOrNot();
                                          }
       }
 Bosque[2][4]=water; Bosque[2][3]=water; Bosque[3][3]=water;
Bosque[2][2]=water;Bosque[3][2]=water;Bosque[3][1]=water;Bosque[3][0]=water;Bosque[2][6]=water;Bos
que[2][5]=water;
 cxyout();
 system("Pause");
 }return;
}
//*****************************************************************************
void fireOrNot()
{             heatA=7;
             if((Bosque[x][y]>300)&&(fuelType[x][y]==3))
                                  {  fire(); return;}              // checks to see if the temperatue of the different
areas fueltype is suffeicent to start a fire
   if((Bosque[x][y]>350)&&(fuelType[x][y]==1))
        {   fire(); return;}
   if((Bosque[x][y]>250)&&(fuelType[x][y]==2))
        {   fire(); return;}

 heatTransferEquation();        //if the heats not suffeicent then it sends it to the normal heat flow y equation
return;
}
//*****************************************************************************
void fire()
{
            double rapidOxyidize=7.0; //this is the fire varible or rapid oxyidize
            float OIO=10; //This is the varible that decides how much elevation effects heat flow

        /*This part of the code is confusing but bear with me
        It stores a BosqueE [x+-a}[y+-b] every area to every side and changes the temporary BosqueE
        value in its application in the y equation for Bosque[x][y] . It changes their value based on elevation
        differences and on how the wind is blowing */
            BosqueE[x-1][y]=Bosque[x-1][y]*(1+((elevation[x][y]-elevation[x-
1][y])/OIO))*(1+(E*windspeed));
            BosqueE[x+1][y]=Bosque[x+1][y]*(1+((elevation[x][y]-
elevation[x+1][y])/OIO))*(1+(W*windspeed));
            BosqueE[x][y+1]=Bosque[x][y+1]*(1+((elevation[x][y]-
elevation[x][y+1])/OIO))*(1+(S*windspeed));
```

```
            BosqueE[x-1][y+1]=Bosque[x-1][y+1]*(1+((elevation[x][y]-elevation[x-
1][y+1])/OIO))*(1+(Se*windspeed));
            BosqueE[x+1][y+1]=Bosque[x+1][y+1]*(1+((elevation[x][y]-
elevation[x+1][y+1])/OIO))*(1+(Sw*windspeed));
            BosqueE[x][y-1]=Bosque[x][y-1]*(1+((elevation[x][y]-elevation[x][y-
1])/OIO))*(1+(N*windspeed));
            BosqueE[x-1][y-1]=Bosque[x-1][y-1]*(1+((elevation[x][y]-elevation[x-1][y-
1])/OIO))*(1+(Ne*windspeed));
            BosqueE[x+1][y-1]=Bosque[x+1][y-1]*(1+((elevation[x][y]-elevation[x+1][y-
1])/OIO))*(1+(Nw*windspeed));

            if(fuel[x][y]>0)
            {           Bosque[x][y]=((4*BosqueE[x+1][y]+4*BosqueE[x-
1][y]+4*BosqueE[x][y+1]+4*BosqueE[x][y-1]+8*BosqueE[x][y]
             +2*BosqueE[x+1][y+1]+2*BosqueE[x+1][y-1]+2*BosqueE[x-1][y+1]+2*BosqueE[x-1][y-
1]+16*set)/48)*rapidOxyidize;
            fuel[x][y]--;
            }
            else
            {heatTransferEquation();} //if there is not enough fuel it kicks into the normal y equationheat flow
            return;
}
//*****************************************************************************
void startFire()
{           int fireCount,
                        time,
                        heat;
            cout<<"how many fires do you want to start?";
                        cin>>fireCount;

            for(time=1; time<=fireCount; time++)
            {           cout<<"location x is on fire "<<time<< "is\n";
                        cin>>x;
                        cout<<"location y is on fire "<<time<<"is \n";
                        cin>>y;
                        cout<<"how hot is the fire?\n";
                        cin>>heat;
                        Bosque[x][y]=heat; //places any temperature to any patch thus starting a possible fire
            }
   cout<<"What is the humidity\n";
   cin>>humidity;
   cout<<"Is the wind blowing y or n";
   cin>>wind;
          switch(wind)
          {case ('y'): {cout<<"from which direction?(No So Ea We NW SW ect.\n";
          cin>>direction;
          cout<<"what is the windspeed\n";
          cin>>windspeed;
           break;break;}
          case  ('n'): {break; break;}
          default  : {cout<<"Huh?";}//if there is wind and what direction it is coming from
   return;
        }
}
//*****************************************************************************************
*
void winds()
{    float away=-.03;
     float twards=.03;
     float between=-.005;
     float foward=.01;
     float backwards=-.01;
```

```
    char No[10]="North",
So[10]="South",We[10]="West",Ea[10]="East",NW[3]="NW",NE[3]="NE",SW[3]="SW",SE[3]="SE";
    // basically what this function is doing is it gives biased heat flow based on which way the wind is blowing
    if(!strcmp(direction,So))
    {N=away; S=twards; E=between; W=between; Sw=foward; Se=foward; Ne=backwards;
Nw=backwards;}
    if(!strcmp(No,direction))
    {N=twards; S=away; E=between; W=between; Sw=backwards; Se=backwards; Ne=foward;
Nw=foward;}
    if(!strcmp(direction,Ea))
    {N=between; S=between; E=twards; W=away; Sw=backwards; Se=foward; Ne=foward;
Nw=backwards;}
    if(!strcmp(direction,We))
    {N=between; S=between; E=away; W=twards; Sw=foward; Se=backwards; Ne=backwards;
Nw=foward;}
    if(!strcmp(direction,SW))
    {N=backwards; S=foward; E=backwards; W=foward; Sw=twards; Se=between; Ne=away;
Nw=between;}
    if(!strcmp(direction,SE))
    {N=backwards; S=foward; E=foward; W=backwards; Sw=between; Se=twards; Ne=between;
Nw=away;}
    if(!strcmp(direction,NW))
    {N=foward; S=backwards; E=backwards; W=foward; Sw=between; Se=away; Ne=between;
Nw=twards;}
    if(!strcmp(direction,NE))
    {N=foward; S=backwards; E=foward; W=backwards; Sw=away; Se=between; Ne=twards;
Nw=between;}
  return;
}
//*****************************************************************************************
*
void setValues()
{
    int w=0,
                        o=1,
        b=2,
        f=3,
                t=2;
              //grand temp scale of Bosque
Bosque[0][15]=set;Bosque[1][15]=set;Bosque[2][15]=water;Bosque[3][15]=set;
Bosque[4][15]=set;Bosque[5][15]=set;Bosque[6][15]=set;
Bosque[0][14]=set;Bosque[1][14]=set;Bosque[2][14]=water;Bosque[3][14]=set;
Bosque[4][14]=set;Bosque[5][14]=set;Bosque[6][14]=set;
Bosque[0][13]=set;Bosque[1][13]=set;Bosque[2][13]=water;Bosque[3][13]=set;
Bosque[4][13]=set;Bosque[5][13]=set;Bosque[6][13]=set;
Bosque[0][12]=set;Bosque[1][12]=set;Bosque[2][12]=water;Bosque[3][12]=set;
Bosque[4][12]=set;Bosque[5][12]=set;Bosque[6][12]=set;
Bosque[0][11]=set;Bosque[1][11]=set;Bosque[2][11]=water;Bosque[3][11]=set;
Bosque[4][11]=set;Bosque[5][11]=set;Bosque[6][11]=set;
Bosque[0][10]=set;Bosque[1][10]=set;Bosque[2][10]=water;Bosque[3][10]=set;
Bosque[4][10]=set;Bosque[5][10]=set;Bosque[6][106]=set;
Bosque[0][9]=set; Bosque[1][9]=set; Bosque[2][9]=water; Bosque[3][9]=set;
Bosque[4][9]=set;Bosque[5][9]=set;  Bosque[6][9]=set;
Bosque[0][8]=set; Bosque[1][8]=set; Bosque[2][8]=water; Bosque[3][8]=set;
Bosque[4][8]=set;Bosque[5][8]=set;  Bosque[6][8]=set;
Bosque[0][7]=set; Bosque[1][7]=set; Bosque[2][7]=water; Bosque[3][7]=set;
Bosque[4][7]=set;Bosque[5][7]=set;  Bosque[6][7]=set;
Bosque[0][6]=set; Bosque[1][6]=set; Bosque[2][6]=water; Bosque[3][6]=set;
Bosque[4][6]=set;Bosque[5][6]=set;  Bosque[6][6]=set;
Bosque[0][5]=set; Bosque[1][5]=set; Bosque[2][5]=water; Bosque[3][5]=set;
Bosque[4][5]=set;Bosque[5][5]=set;  Bosque[6][5]=set;
```

```
Bosque[0][4]=set; Bosque[1][4]=set; Bosque[2][4]=water; Bosque[3][4]=set;
Bosque[4][4]=set;Bosque[5][4]=set;  Bosque[6][4]=set;
Bosque[0][3]=set; Bosque[1][3]=set; Bosque[2][3]=water; Bosque[3][3]=water;
Bosque[4][3]=set;Bosque[5][3]=set;  Bosque[6][3]=set;
Bosque[0][2]=set; Bosque[1][2]=set; Bosque[2][2]=water; Bosque[3][2]=water;
Bosque[4][2]=set;Bosque[5][2]=set;  Bosque[6][2]=set;
Bosque[0][1]=set; Bosque[1][1]=set; Bosque[2][1]=set;   Bosque[3][1]=water;
Bosque[4][1]=set;Bosque[5][1]=set;  Bosque[6][1]=set;
Bosque[0][0]=set; Bosque[1][0]=set; Bosque[2][0]=set;   Bosque[3][0]=water;
Bosque[4][0]=set;Bosque[5][0]=set;  Bosque[6][6]=set;
   //grand Bosque elevation of area
elevation[0][15]=3 ;elevation[1][15]=1; elevation[2][15]=0; elevation[3][15]=1; elevation[4][15]=1;
elevation[5][15]=3; elevation[6][15]=4;
elevation[0][14]=3 ;elevation[1][14]=1; elevation[2][14]=0; elevation[3][14]=1; elevation[4][14]=1;
elevation[5][14]=3; elevation[6][14]=4;
elevation[0][13]=3 ;elevation[1][13]=1; elevation[2][13]=0; elevation[3][13]=1; elevation[4][13]=1;
elevation[5][13]=3; elevation[6][13]=4;
elevation[0][12]=3 ;elevation[1][12]=1; elevation[2][12]=0; elevation[3][12]=1; elevation[4][12]=1;
elevation[5][12]=3; elevation[6][12]=4;
elevation[0][11]=3 ;elevation[1][11]=1; elevation[2][11]=0; elevation[3][11]=1; elevation[4][11]=1;
elevation[5][11]=3; elevation[6][11]=4;
elevation[0][10]=3 ;elevation[1][10]=1; elevation[2][10]=0; elevation[3][10]=1; elevation[4][10]=1;
elevation[5][10]=3; elevation[10][6]=4;
elevation[0][9]=3  ;elevation[1][9]=1;  elevation[2][9]=0;  elevation[3][9]=1;  elevation[4][9]=1;
elevation[5][9]=3;  elevation[6][9]=4;
elevation[0][8]=3  ;elevation[1][8]=1;  elevation[2][8]=0;  elevation[3][8]=1;  elevation[4][8]=1;
elevation[5][8]=3;  elevation[6][8]=4;
elevation[0][7]=3  ;elevation[1][7]=1;  elevation[2][7]=0;  elevation[3][7]=1;  elevation[4][7]=1;
elevation[5][7]=3;  elevation[6][7]=4;
elevation[0][6]=3  ;elevation[1][6]=1;  elevation[2][6]=0;  elevation[3][6]=1;  elevation[4][6]=1;
elevation[5][6]=3;  elevation[6][6]=4;
elevation[0][5]=2  ;elevation[1][5]=2;  elevation[2][5]=0;  elevation[3][5]=1;  elevation[4][5]=1;
elevation[5][5]=2;  elevation[6][5]=3;
elevation[0][4]=2  ;elevation[1][4]=1;  elevation[2][4]=0;  elevation[3][4]=1;  elevation[4][4]=1;
elevation[5][4]=2;  elevation[6][4]=3;
elevation[0][3]=3  ;elevation[1][3]=1;  elevation[2][3]=0;  elevation[3][3]=0;  elevation[4][3]=1;
elevation[5][3]=2;  elevation[6][3]=2;
elevation[0][2]=3  ;elevation[1][2]=1;  elevation[2][2]=0;  elevation[3][2]=0;  elevation[4][2]=1;
elevation[5][2]=2;  elevation[6][2]=2;
elevation[0][1]=4  ;elevation[1][1]=2;  elevation[2][1]=1;  elevation[3][1]=0;  elevation[4][1]=1;
elevation[5][1]=2;  elevation[6][1]=2;
elevation[0][0]=4  ;elevation[1][0]=3;  elevation[2][0]=1;  elevation[3][0]=0;  elevation[4][0]=1;
elevation[5][0]=2;  elevation[6][6]=3;
      //assigns a fuel type to any every place in the Bosque
fuelType[0][15]=f; fuelType[1][15]=b; fuelType[2][15]=w; fuelType[3][15]=o; fuelType[4][15]=o*t;
fuelType[5][15]=b*t; fuelType[6][15]=f*t;
fuelType[0][14]=f; fuelType[1][14]=b; fuelType[2][14]=w; fuelType[3][14]=o; fuelType[4][14]=o*t;
fuelType[5][14]=b*t; fuelType[6][14]=f*t;
fuelType[0][13]=f; fuelType[1][13]=b; fuelType[2][13]=w; fuelType[3][13]=o; fuelType[4][13]=o*t;
fuelType[5][13]=b*t; fuelType[6][13]=f*t;
fuelType[0][12]=f; fuelType[1][12]=b; fuelType[2][12]=w; fuelType[3][12]=o; fuelType[4][12]=o*t;
fuelType[5][12]=b*t; fuelType[6][12]=f*t;
fuelType[0][11]=f; fuelType[1][11]=b; fuelType[2][11]=w; fuelType[3][11]=o; fuelType[4][11]=o*t;
fuelType[5][11]=b*t; fuelType[6][6]=f*t;
fuelType[0][10]=f; fuelType[1][10]=b; fuelType[2][10]=w; fuelType[3][10]=o; fuelType[4][10]=o*t;
fuelType[5][10]=b*t; fuelType[6][10]=f*t;
fuelType[0][9]=f;  fuelType[1][9]=b;  fuelType[2][9]=w;  fuelType[3][9]=o;  fuelType[4][9]=o*t;
fuelType[5][9]=b*t;  fuelType[6][9]=f*t;
fuelType[0][8]=f;  fuelType[1][8]=b;  fuelType[2][8]=w;  fuelType[3][8]=o;  fuelType[4][8]=o*t;
fuelType[5][8]=b*t;  fuelType[6][8]=f*t;
fuelType[0][7]=f;  fuelType[1][7]=b;  fuelType[2][7]=w;  fuelType[3][7]=o;  fuelType[4][7]=o*t;
fuelType[5][7]=b*t;  fuelType[6][7]=f*t;
```

```
fuelType[0][6]=f;  fuelType[1][6]=b;  fuelType[2][6]=w;  fuelType[3][6]=o;  fuelType[4][6]=o*t;
fuelType[5][6]=b*t;  fuelType[6][6]=f*t;
fuelType[0][5]=f;  fuelType[1][5]=f;  fuelType[2][5]=w;  fuelType[3][5]=o;  fuelType[4][5]=o*t;
fuelType[5][5]=f*t;  fuelType[6][5]=f*t;
fuelType[0][4]=b;  fuelType[1][4]=b;  fuelType[2][4]=w;  fuelType[3][4]=o;  fuelType[4][4]=o*t;
fuelType[5][4]=f*t;  fuelType[6][4]=b*t;
fuelType[0][3]=b;  fuelType[1][3]=b;  fuelType[2][3]=w;  fuelType[3][3]=w;  fuelType[4][3]=o*t;
fuelType[5][3]=b*t;  fuelType[6][3]=b*t;
fuelType[0][2]=f;  fuelType[1][2]=f;  fuelType[2][2]=w;  fuelType[3][2]=w;  fuelType[4][2]=o*t;
fuelType[5][2]=b*t;  fuelType[6][2]=f*t;
fuelType[0][1]=f;  fuelType[1][1]=f;  fuelType[2][1]=o;  fuelType[3][1]=w;  fuelType[4][1]=o*t;
fuelType[5][1]=b*t;  fuelType[6][1]=o*t;
fuelType[0][0]=f;  fuelType[1][0]=f;  fuelType[2][0]=o;  fuelType[3][0]=w;  fuelType[4][0]=o*t;
fuelType[5][0]=b*t;  fuelType[6][6]=o*t;
      // assigns every area a fuel amount
fuel[0][15]=f*t; fuel[1][15]=f*t; fuel[2][15]=w*t; fuel[3][15]=b*t; fuel[4][15]=b*t; fuel[5][15]=b*t;
fuel[6][15]=b*t;
fuel[0][14]=f*t; fuel[1][14]=f*t; fuel[2][14]=w*t; fuel[3][14]=b*t; fuel[4][14]=b*t; fuel[5][14]=b*t;
fuel[6][14]=b*t;
fuel[0][13]=f*t; fuel[1][13]=f*t; fuel[2][13]=w*t; fuel[3][13]=b*t; fuel[4][13]=b*t; fuel[5][13]=b*t;
fuel[6][13]=b*t;
fuel[0][12]=f*t; fuel[1][12]=f*t; fuel[2][12]=w*t; fuel[3][12]=b*t; fuel[4][12]=b*t; fuel[5][12]=b*t;
fuel[6][12]=b*t;
fuel[0][11]=f*t; fuel[1][11]=f*t; fuel[2][11]=w*t; fuel[3][11]=b*t; fuel[4][11]=b*t; fuel[5][11]=b*t;
fuel[6][11]=b*t;
fuel[0][10]=f*t; fuel[1][10]=f*t; fuel[2][10]=w*t; fuel[3][10]=b*t; fuel[4][10]=b*t; fuel[5][10]=b*t;
fuel[6][10]=b*t;
fuel[0][9]=f*t;  fuel[1][9]=f*t;  fuel[2][9]=w*t;  fuel[3][9]=b*t;  fuel[4][9]=b*t;  fuel[5][9]=b*t;
fuel[6][9]=b*t;
fuel[0][8]=f*t;  fuel[1][8]=f*t;  fuel[2][8]=w*t;  fuel[3][8]=b*t;  fuel[4][8]=b*t;  fuel[5][8]=b*t;
fuel[6][8]=b*t;
fuel[0][7]=f*t;  fuel[1][7]=f*t;  fuel[2][7]=w*t;  fuel[3][7]=b*t;  fuel[4][7]=b*t;  fuel[5][7]=b*t;
fuel[6][7]=b*t;
fuel[0][6]=f*t;  fuel[1][6]=b*t;  fuel[2][6]=w*t;  fuel[3][6]=o*t;  fuel[4][6]=o*t;  fuel[5][6]=b*t;
fuel[6][6]=f*t;
fuel[0][5]=f*t;  fuel[1][5]=f*t;  fuel[2][5]=w*t;  fuel[3][5]=o*t;  fuel[4][5]=o*t;  fuel[5][5]=f*t;
fuel[6][5]=f*t;
fuel[0][4]=b*t;  fuel[1][4]=b*t;  fuel[2][4]=w*t;  fuel[3][4]=o*t;  fuel[4][4]=o*t;  fuel[5][4]=f*t;
fuel[6][4]=b*t;
fuel[0][3]=b*t;  fuel[1][3]=b*t;  fuel[2][3]=w*t;  fuel[3][3]=w*t;  fuel[4][3]=o*t;  fuel[5][3]=b*t;
fuel[6][3]=b*t;
fuel[0][2]=f*t;  fuel[1][2]=f*t;  fuel[2][2]=w*t;  fuel[3][2]=w*t;  fuel[4][2]=o*t;  fuel[5][2]=b*t;
fuel[6][2]=f*t;
fuel[0][1]=f*t;  fuel[1][1]=f*t;  fuel[2][1]=o*t;  fuel[3][1]=w*t;  fuel[4][1]=o*t;  fuel[5][1]=b*t;
fuel[6][1]=o*t;
fuel[0][0]=f*t;  fuel[1][0]=f*t;  fuel[2][0]=o*t;  fuel[3][0]=w*t;  fuel[4][0]=o*t;  fuel[5][0]=b*t;
fuel[6][6]=o*t;

return;
}
//*****************************************************************************
void heatTransferEquation()
{    winds();
     float el=100;
     float humfactor=.001;
     /* Again this is a temperorary biaes in the actual temperatures, according to Bosque[x][y]
      because of the elevation and Wind
      */
   BosqueE[x-1][y]=Bosque[x-1][y]*(1+((elevation[x][y]-elevation[x-1][y])/el))*(1+(E*windspeed));//E;
            BosqueE[x+1][y]=Bosque[x+1][y]*(1+((elevation[x][y]-
elevation[x+1][y])/el))*(1+(W*windspeed));//W;
```

```
            BosqueE[x][y+1]=Bosque[x][y+1]*(1+((elevation[x][y]-
elevation[x][y+1])/el))*(1+(S*windspeed));//S;
            BosqueE[x-1][y+1]=Bosque[x-1][y+1]*(1+((elevation[x][y]-elevation[x-
1][y+1])/el))*(1+(Se*windspeed));//SE;
            BosqueE[x+1][y+1]=Bosque[x+1][y+1]*(1+((elevation[x][y]-
elevation[x+1][y+1])/el))*(1+(Sw*windspeed));//SW;
            BosqueE[x][y-1]=Bosque[x][y-1]*(1+((elevation[x][y]-elevation[x][y-
1])/el))*(1+(N*windspeed));//N;
            BosqueE[x-1][y-1]=Bosque[x-1][y-1]*(1+((elevation[x][y]-elevation[x-1][y-
1])/el))*(1+(Ne*windspeed));//NE;
            BosqueE[x+1][y-1]=Bosque[x+1][y-1]*(1+((elevation[x][y]-elevation[x+1][y-
1])/el))*(1+(Nw*windspeed));//NW;
            // The basic Y heat transfer equation
            Bosque[x][y]=((4*BosqueE[x+1][y]+4*BosqueE[x-1][y]+4*BosqueE[x][y+1]+4*BosqueE[x][y-
1]+8*BosqueE[x][y]
             +2*BosqueE[x+1][y+1]+2*BosqueE[x+1][y-1]+2*BosqueE[x-1][y+1]+2*BosqueE[x-1][y-
1]+16*set)/48)*(1-(humidity*humfactor));

            return;
}
//*****************************************************************************
/*  started Fire in Bosque v.2 2-17-05*/

/*heat transfer working 2-20-05*/

/*fire working 2-29-05*/

/*added ambient temp, wind 3-1-05*/

/*Added fuel types fire rendering 3-2-05*/

/* This program calculates
  wind, elevation, humidity, fueltype  (brush, tree, open, water,) fuel remaining
  fire is: limited to burn for a number of iterations depending on fuel type starts at different temperatures
depending on fueltype
             is humidity sensitive and is calculated by the double rapidoxidize
  It needs now a map of the bosque and a SDL rendering 3-25-05
*/
/* I realize that this code still has errors however I have decided to move on to version 3*/
```

V.3

I am now working on fire in the Bosque v.3 I started it as of March 27 excellent progress has been made in using a SDL rendering of heat flow. With the exception of the situational factors, the program is basically done,. Here is the code;

```
#include <stdio.h>
#include <stdlib.h>
#include <SDL/SDL.h> //SDL library! works!
#include<iostream.h>
void setUp();       //sets all the values in this program
void DrawScene();
void DrawBosque();
void heatFlowXY();
void fireOrNot();
```

```
void fire();
void noFire();
void startFire();
struct area
{
      int fuelType;
      int fuel;          // Declares a structure to rpresent the Bosque
      double long temperature;
      int elevation;
}Bosque[42][27];
struct varibles
{
      int start; int fire;
      float heatLoss;
      float rapidOxyidize;    //I have decided (for now) to make all the
     double ambientTemp;      //varibles be declared in the program
     double fireTemp;
     int time;
     double waterTemp;
}varible={20,12,7.00,2,12,850,100,10};

SDL_Surface *key;                    // pointers to the BMP files might replace with SDL_RECT
SDL_Surface *mouse;
SDL_Surface *screen;
SDL_Surface *zero;
SDL_Surface *ten;
SDL_Surface *twenty;
SDL_Surface *thirty;
SDL_Surface *fourty;
SDL_Surface *fifty;
SDL_Surface *sevenfive;
SDL_Surface *hundred;
SDL_Surface *onefifty;
SDL_Surface *twohun;
SDL_Surface *twofifty;
SDL_Surface *threehun;
SDL_Surface *fourhun;
SDL_Surface *fivehun;
SDL_Surface *sixhun;
SDL_Surface *sevenhun;
SDL_Surface *eighthun;
SDL_Surface *ninhun;
SDL_Surface *thousand;
SDL_Surface *twelvefifty;
SDL_Surface *thousandfive;
SDL_Surface *twothou;
SDL_Surface *twothoufive;
SDL_Surface *greatertwothouf;


int xpos=-75,ypos=200;
int xa,ya;
long timer=0;
//*****************************************************************************************
****
int InitBosque()
{
  key = SDL_LoadBMP("Bosquebackground.bmp");  // establishes BMP files might replace with SDL_RECT
  zero= SDL_LoadBMP("-0.bmp");
  ten = SDL_LoadBMP("1-10.bmp");
  twenty = SDL_LoadBMP("11-20.bmp");
  thirty = SDL_LoadBMP("21-30.bmp");
```

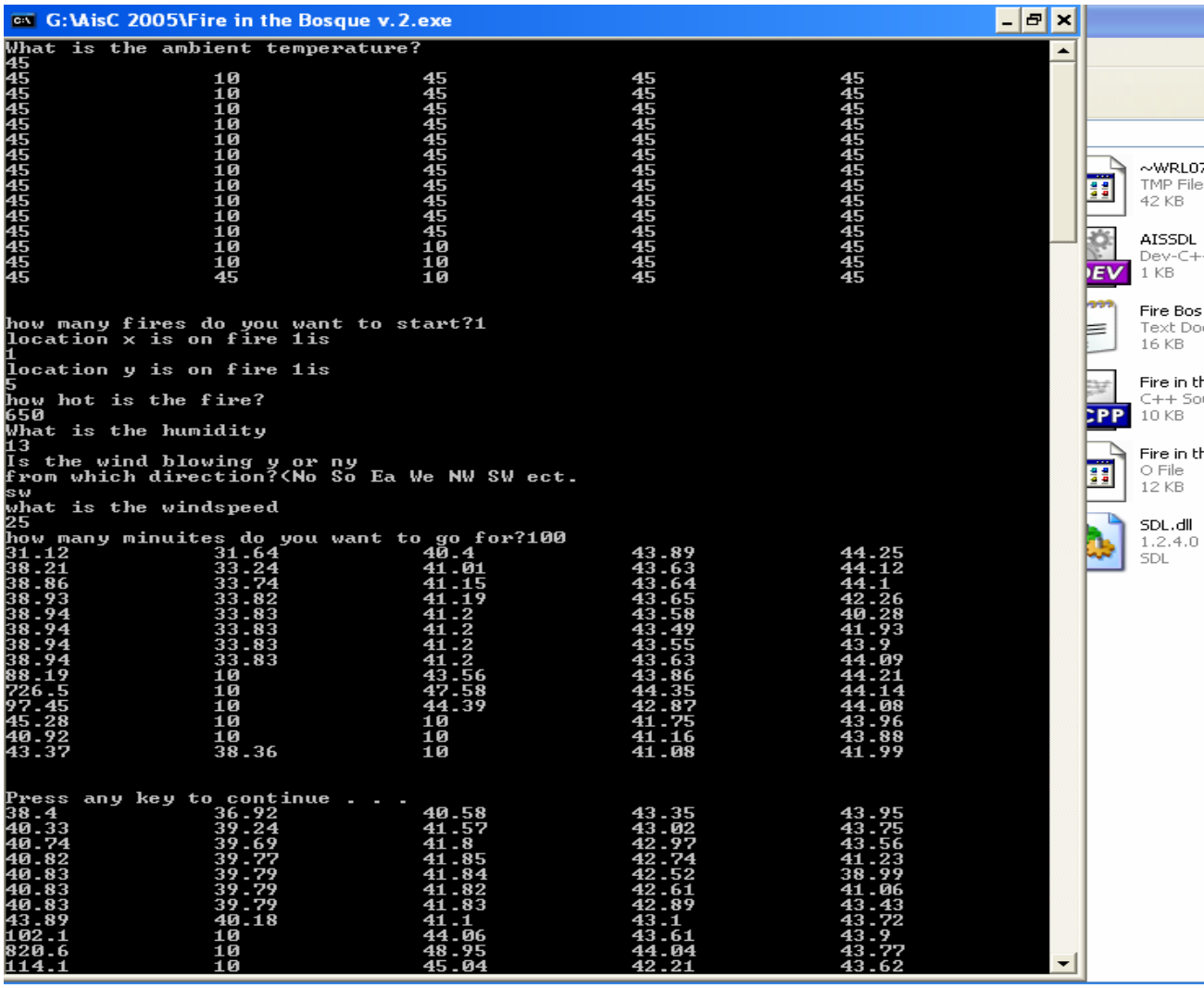
```
 fourty = SDL_LoadBMP("31-40.bmp");
 fifty= SDL_LoadBMP("41-50.bmp");
 sevenfive = SDL_LoadBMP("51-75.bmp");
 hundred = SDL_LoadBMP("76-100.bmp");
 onefifty = SDL_LoadBMP("101-150.bmp");
 twohun = SDL_LoadBMP("151-200.bmp");
 twofifty = SDL_LoadBMP("201-250.bmp");
 threehun = SDL_LoadBMP("251-300.bmp");
 fourhun = SDL_LoadBMP("301-400.bmp");
 fivehun = SDL_LoadBMP("401-500.bmp");
 sixhun = SDL_LoadBMP("501-600.bmp");
 sevenhun = SDL_LoadBMP("601-700.bmp");
 eighthun = SDL_LoadBMP("701-800.bmp");
 ninhun = SDL_LoadBMP("801-900.bmp");
 thousand = SDL_LoadBMP("901-100.bmp");
 twelvefifty = SDL_LoadBMP("1001-1250.bmp");
 thousandfive = SDL_LoadBMP("1251-1500.bmp");
 twothou = SDL_LoadBMP("1501-2000.bmp");
 twothoufive = SDL_LoadBMP("2001-2500.bmp");
 greatertwothouf= SDL_LoadBMP("gt2500.bmp");
 return 0;
}
//*****************************************************************************************
**
void DrawIMG(SDL_Surface *img, int x, int y)
{
 SDL_Rect dest;        // draws BMPs
 dest.x = x;
 dest.y = y;
 SDL_BlitSurface(img, NULL, screen, &dest);
}
//*****************************************************************************************
****
void DrawIMG(SDL_Surface *img, int x, int y, int w, int h, int x2, int y2)
{
 SDL_Rect dest;
 dest.x = x;
 dest.y = y;
 SDL_Rect dest2;
 dest2.x = x2;
 dest2.y = y2;
 dest2.w = w;
 dest2.h = h;
 SDL_BlitSurface(img, &dest2, screen, &dest);
 return;
}
//*****************************************************************************************
****
void DrawBosque()
{           if(timer==0)
          {DrawIMG(key, 0, 0);}
    for(xa=1; xa<=40; xa++)
     {
       for(ya=1; ya<=25; ya++)
              {      // if statements on where to place the BMP SDL_RECT

                if(Bosque[xa][ya].temperature<=0)
                   DrawIMG(zero, 12*xa+50,12*ya+50);
                if((Bosque[xa][ya].temperature<=10)&&(Bosque[xa][ya].temperature>0))
                  {DrawIMG(ten, 12*xa+50,12*ya+50);}
                if((Bosque[xa][ya].temperature<=20)&&(Bosque[xa][ya].temperature>10))
                  {DrawIMG(twenty, 12*xa+50,12*ya+50);}
```

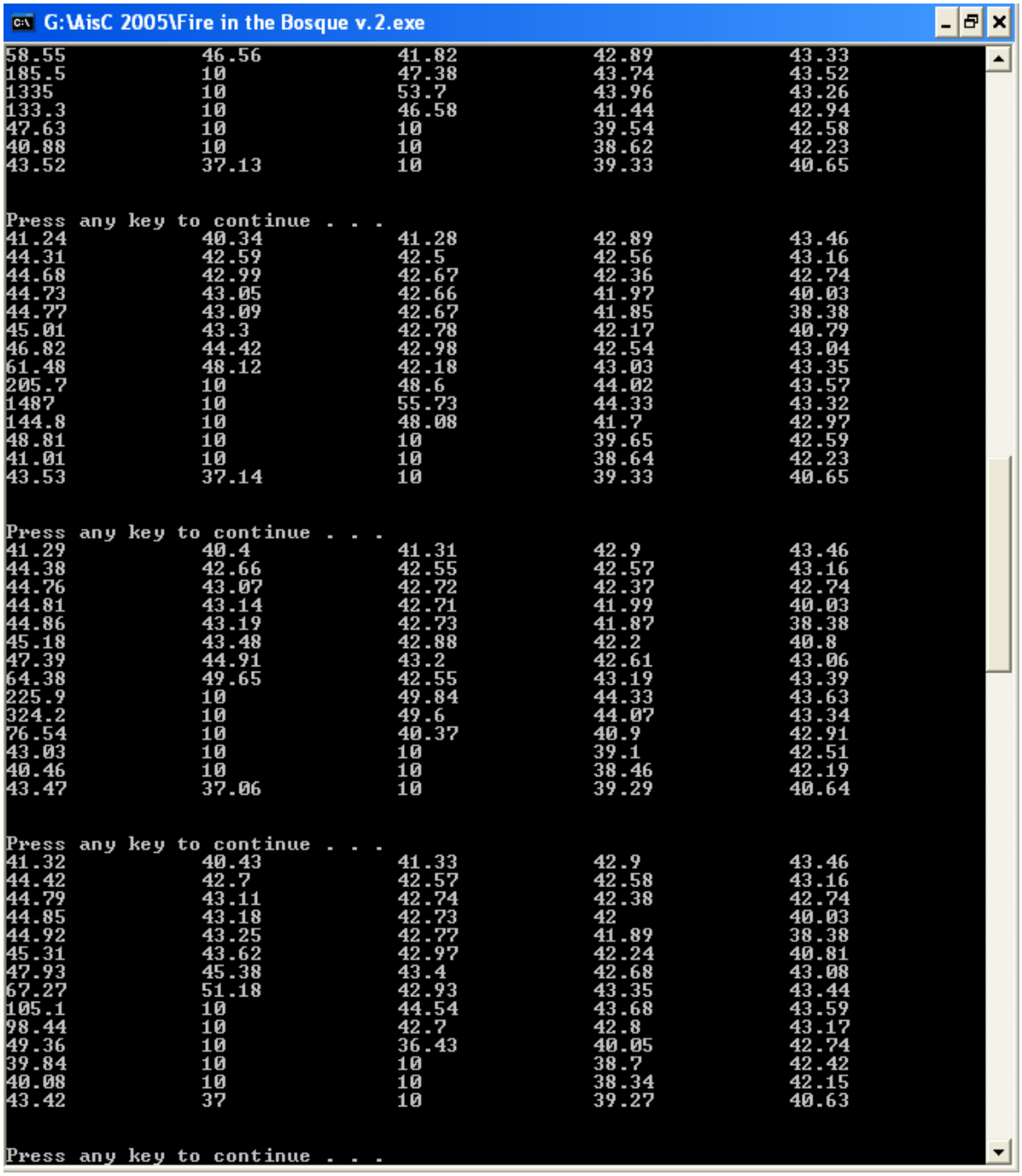
```
                if((Bosque[xa][ya].temperature<=30)&&(Bosque[xa][ya].temperature>20))
                  {DrawIMG(thirty, 12*xa+50,12*ya+50);}
                if((Bosque[xa][ya].temperature<=40)&&(Bosque[xa][ya].temperature>30))
                  {DrawIMG(fourty, 12*xa+50,12*ya+50);}
                if((Bosque[xa][ya].temperature<=50)&&(Bosque[xa][ya].temperature>40))
                  {DrawIMG(fifty, 12*xa+50,12*ya+50);}
                if((Bosque[xa][ya].temperature<=75)&&(Bosque[xa][ya].temperature>50))
                  {DrawIMG(sevenfive, 12*xa+50,12*ya+50);}
                if((Bosque[xa][ya].temperature<=100)&&(Bosque[xa][ya].temperature>75))
                  {DrawIMG(hundred, 12*xa+50,12*ya+50);}
                if((Bosque[xa][ya].temperature<=150)&&(Bosque[xa][ya].temperature>100))
                  {DrawIMG(onefifty, 12*xa+50,12*ya+50);}
                if((Bosque[xa][ya].temperature<=200)&&(Bosque[xa][ya].temperature>150))
                  {DrawIMG(twohun, 12*xa+50,12*ya+50);}
                if((Bosque[xa][ya].temperature<=250)&&(Bosque[xa][ya].temperature>200))
                  {DrawIMG(twofifty, 12*xa+50,12*ya+50);}
                if((Bosque[xa][ya].temperature<=300)&&(Bosque[xa][ya].temperature>250))
                  {DrawIMG(threehun, 12*xa+50,12*ya+50);}
                if((Bosque[xa][ya].temperature<=400)&&(Bosque[xa][ya].temperature>300))
                  {DrawIMG(fourhun, 12*xa+50,12*ya+50);}
                if((Bosque[xa][ya].temperature<=500)&&(Bosque[xa][ya].temperature>400))
                  {DrawIMG(fivehun, 12*xa+50,12*ya+50);}
                if((Bosque[xa][ya].temperature<=600)&&(Bosque[xa][ya].temperature>500))
                  {DrawIMG(sixhun, 12*xa+50,12*ya+50);}
                if((Bosque[xa][ya].temperature<=700)&&(Bosque[xa][ya].temperature>600))
                  {DrawIMG(sevenhun, 12*xa+50,12*ya+50);}
                if((Bosque[xa][ya].temperature<=800)&&(Bosque[xa][ya].temperature>700))
                  {DrawIMG(eighthun, 12*xa+50,12*ya+50);}
                if((Bosque[xa][ya].temperature<=900)&&(Bosque[xa][ya].temperature>800))
                  {DrawIMG(ninhun, 12*xa+50,12*ya+50);}
                if((Bosque[xa][ya].temperature<=1000)&&(Bosque[xa][ya].temperature>900))
                  {DrawIMG(thousand, 12*xa+50,12*ya+50);}
                if((Bosque[xa][ya].temperature<=1250)&&(Bosque[xa][ya].temperature>1000))
                  {DrawIMG(twelvefifty, 12*xa+50,12*ya+50);}
                if((Bosque[xa][ya].temperature<=1500)&&(Bosque[xa][ya].temperature>1250))
                  {DrawIMG(thousandfive, 12*xa+50,12*ya+50);}
                if((Bosque[xa][ya].temperature<=2000)&&(Bosque[xa][ya].temperature>1500))
                  {DrawIMG(twothou, 12*xa+50,12*ya+50);}
                if((Bosque[xa][ya].temperature<=2500)&&(Bosque[xa][ya].temperature>2000))
                  {DrawIMG(greatertwothouf, 12*xa+50,12*ya+50);}
                if(Bosque[xa][ya].temperature>2500)
                  {DrawIMG(greatertwothouf, 12*xa+50,12*ya+50);}
              timer++;
              }
        }
return;}
//*****************************************************************************************
****
int main(int argc, char *argv[])
{
   setUp();
   startFire();
  Uint8* keys;

  if ( SDL_Init(SDL_INIT_AUDIO|SDL_INIT_VIDEO) < 0 )
  {
    printf("error running SDL: %s\n", SDL_GetError());
    exit(1);
  }
  atexit(SDL_Quit);

  screen=SDL_SetVideoMode(640,480,32,SDL_HWSURFACE|SDL_DOUBLEBUF);
```

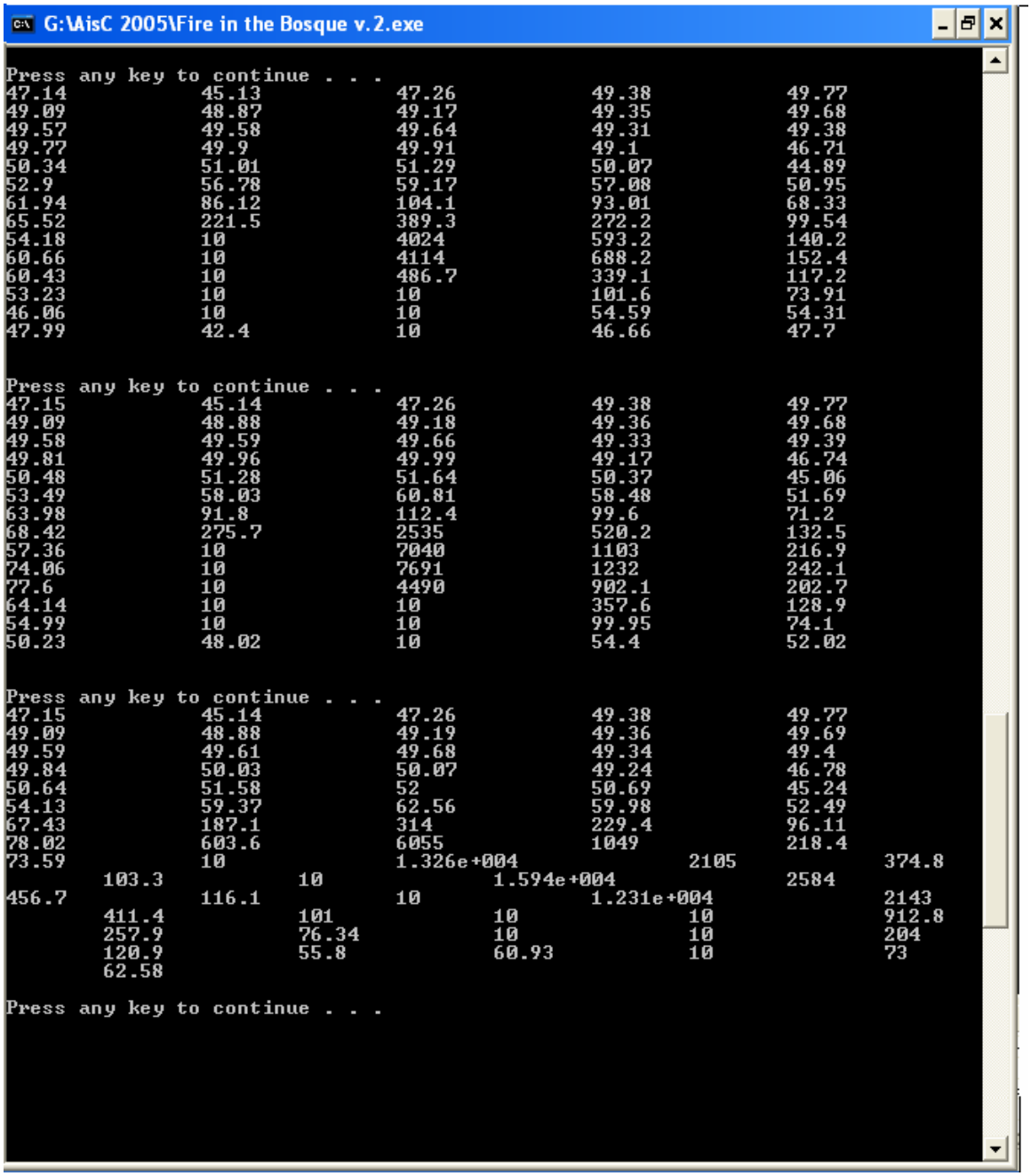
```
 if ( screen == NULL )
 {
  printf("Unable to set graphics video: %s\n", SDL_GetError());
  exit(1);
 }
 InitBosque();
 DrawBosque();


 int done=0;

 while(done == 0)
 {
  SDL_Event event;

  while ( SDL_PollEvent(&event) )
  {
   if ( event.type == SDL_QUIT )  {  done = 1;  }

   if ( event.type == SDL_KEYDOWN )
   {
    if ( event.key.keysym.sym == SDLK_ESCAPE ) { done = 1; }
   }
  }
  //heatFlowXY();
  keys = SDL_GetKeyState(NULL);
  if ( keys[SDLK_UP] ) { ypos -= 1; }
  if ( keys[SDLK_DOWN] ) { ypos += 1; }
  if ( keys[SDLK_LEFT] ) { xpos -= 1; }
  if ( keys[SDLK_RIGHT] ) { heatFlowXY();}
  if ( keys[SDLK_PLUS]  ) {xpos +=1;}

 }

 return 0;
}
//*****************************************************************************************
***********
void heatFlowXY()
{
    for(xa=1; xa<=40; xa++)
     {
         for(ya=1; ya<=25; ya++)
         {    fireOrNot();      //checks each individual Struct [xa][ya]
         }
     }

   DrawBosque(); return;
}
//*****************************************************************************************
***********
void fireOrNot()
{
 if((Bosque[xa][ya].fuelType==1)&&(Bosque[xa][ya].temperature>=275))
    { fire();return; }
 if((Bosque[xa][ya].fuelType==2)&&(Bosque[xa][ya].temperature>=300))
      { fire();return;}
 if((Bosque[xa][ya].fuelType==3)&&(Bosque[xa][ya].temperature>=325))
    { fire();return; }
    //checks to see if conditions are right for a fire
    //if not then there is no fire
      noFire(); return;
```

```
}
//*****************************************************************************************************
***********
void fire()
{
    if(Bosque[xa][ya].fuel>0)       //kicks back if there is no fuel
    {
Bosque[xa][ya].temperature=((Bosque[xa][ya].temperature+ Bosque[xa-1][ya].temperature+Bosque[xa][ya-
1].temperature+
                        Bosque[xa+1][ya].temperature+Bosque[xa][ya+1].temperature+

Bosque[xa][ya].temperature+varible.ambientTemp+varible.ambientTemp)/varible.heatLoss)*varible.rapidOx
yidize;

                Bosque[xa][ya].fuel--; return;
    }
    else {noFire(); return;}
}
//*****************************************************************************************************
***********
void noFire()
{
Bosque[xa][ya].temperature=(Bosque[xa][ya].temperature+ Bosque[xa-1][ya].temperature+Bosque[xa][ya-
1].temperature+
                        Bosque[xa+1][ya].temperature+Bosque[xa][ya+1].temperature+

Bosque[xa][ya].temperature+varible.ambientTemp+varible.ambientTemp)/varible.heatLoss;
}
//*****************************************************************************************************
************
void startFire() //starts a fire controlled by struct varible
{     Bosque[varible.start][varible.fire].temperature=varible.fireTemp;
  return;
}
//*****************************************************************************************************
***********
void setUp() //sets the whole Bosque to be an elevation 1 fueltype 2 I will add more to later
{
    for(xa=0; xa<=41; xa++)
    {
      for(ya=0; ya<=26; ya++)
        {
              Bosque[xa][ya].temperature=varible.ambientTemp;
              Bosque[xa][ya].fuel=10;
              Bosque[xa][ya].fuelType=2;
              Bosque[xa][ya].elevation=1;
        }
    }
}
//*****************************************************************************************************
**************
/* v.3 started 3-27 however took until 3-30 to get any where with ( library linker problem)*/
/*4-1 finally figured out how to use basic SDL and created a heatflow and fire program*/
```

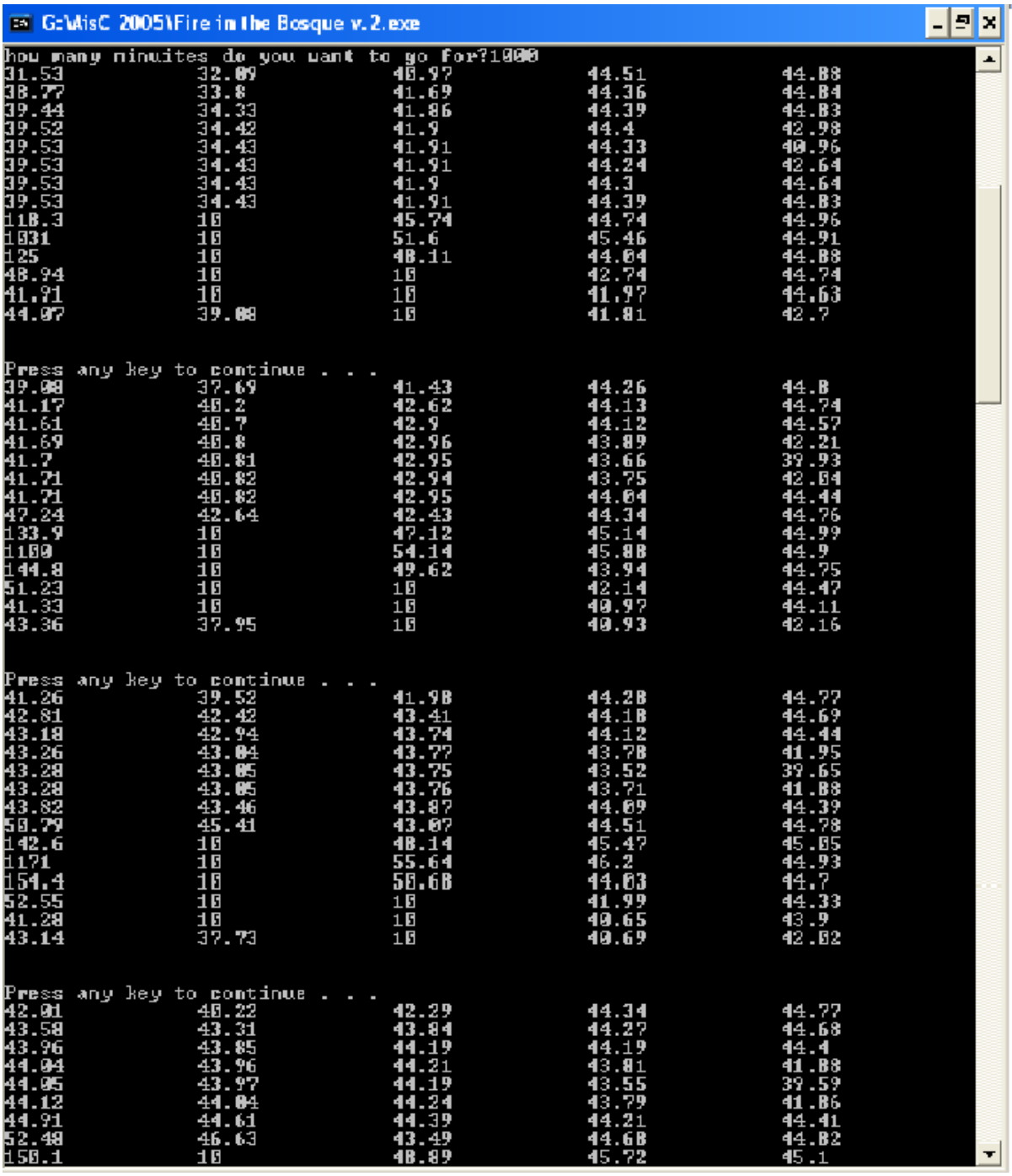
Results- V.2


G:\AisC 2005\Fire in the Bosque v.2.exe
What is the ambient temperature?
45
45 10 45 45 45
45 10 45 45 45
45 10 45 45 45
45 10 45 45 45
45 10 45 45 45
45 10 45 45 45
45 10 45 45 45
45 10 45 45 45
45 10 45 45 45
45 10 45 45 45
45 10 45 45 45
45 10 10 45 45
45 10 10 45 45
45 45 10 45 45
how many fires do you want to start?1
location x is on fire 1is
1
location y is on fire 1is
5
how hot is the fire?
650
What is the humidity
13
Is the wind blowing y or ny
from which direction?<No So Ea We NW SW ect.
sw
what is the windspeed
25
how many minuites do you want to go for?100
31.12 31.64 40.4 43.89 44.25
38.21 33.24 41.01 43.63 44.12
38.86 33.74 41.15 43.64 44.1
38.93 33.82 41.19 43.65 42.26
38.94 33.83 41.2 43.58 40.28
38.94 33.83 41.2 43.49 41.93
38.94 33.83 41.2 43.55 43.9
38.94 33.83 41.2 43.63 44.09
88.19 10 43.56 43.86 44.21
726.5 10 47.58 44.35 44.14
97.45 10 44.39 42.87 44.08
45.28 10 10 41.75 43.96
40.92 10 10 41.16 43.88
43.37 38.36 10 41.08 41.99
Press any key to continue . . .
38.4 36.92 40.58 43.35 43.95
40.33 39.24 41.57 43.02 43.75
40.74 39.69 41.8 42.97 43.56
40.82 39.77 41.85 42.74 41.23
40.83 39.79 41.84 42.52 38.99
40.83 39.79 41.82 42.61 41.06
40.83 39.79 41.83 42.89 43.43
43.89 40.18 41.1 43.1 43.72
102.1 10 44.06 43.61 43.9
820.6 10 48.95 44.04 43.77
114.1 10 45.04 42.21 43.62
~WRL07
TMP File
42 KB
AISSDL
Dev-C++
1 KB
Fire Bos
Text Do
16 KB
Fire in th
C++ So
10 KB
Fire in th
O File
12 KB
SDL.dll
1.2.4.0
SDL


G:\AisC 2005\Fire in the Bosque v.2.exe
58.55 46.56 41.82 42.89 43.33
185.5 10 47.38 43.74 43.52
1335 10 53.7 43.96 43.26
133.3 10 46.58 41.44 42.94
47.63 10 10 39.54 42.58
40.88 10 10 38.62 42.23
43.52 37.13 10 39.33 40.65
Press any key to continue . . .
41.24 40.34 41.28 42.89 43.46
44.31 42.59 42.5 42.56 43.16
44.68 42.99 42.67 42.36 42.74
44.73 43.05 42.66 41.97 40.03
44.77 43.09 42.67 41.85 38.38
45.01 43.3 42.78 42.17 40.79
46.82 44.42 42.98 42.54 43.04
61.48 48.12 42.18 43.03 43.35
205.7 10 48.6 44.02 43.57
1487 10 55.73 44.33 43.32
144.8 10 48.08 41.7 42.97
48.81 10 10 39.65 42.59
41.01 10 10 38.64 42.23
43.53 37.14 10 39.33 40.65
Press any key to continue . . .
41.29 40.4 41.31 42.9 43.46
44.38 42.66 42.55 42.57 43.16
44.76 43.07 42.72 42.37 42.74
44.81 43.14 42.71 41.99 40.03
44.86 43.19 42.73 41.87 38.38
45.18 43.48 42.88 42.2 40.8
47.39 44.91 43.2 42.61 43.06
64.38 49.65 42.55 43.19 43.39
225.9 10 49.84 44.33 43.63
324.2 10 49.6 44.07 43.34
76.54 10 40.37 40.9 42.91
43.03 10 10 39.1 42.51
40.46 10 10 38.46 42.19
43.47 37.06 10 39.29 40.64
Press any key to continue . . .
41.32 40.43 41.33 42.9 43.46
44.42 42.7 42.57 42.58 43.16
44.79 43.11 42.74 42.38 42.74
44.85 43.18 42.73 42 40.03
44.92 43.25 42.77 41.89 38.38
45.31 43.62 42.97 42.24 40.81
47.93 45.38 43.4 42.68 43.08
67.27 51.18 42.93 43.35 43.44
105.1 10 44.54 43.68 43.59
98.44 10 42.7 42.8 43.17
49.36 10 36.43 40.05 42.74
39.84 10 10 38.7 42.42
40.08 10 10 38.34 42.15
43.42 37 10 39.27 40.63
Press any key to continue . . .

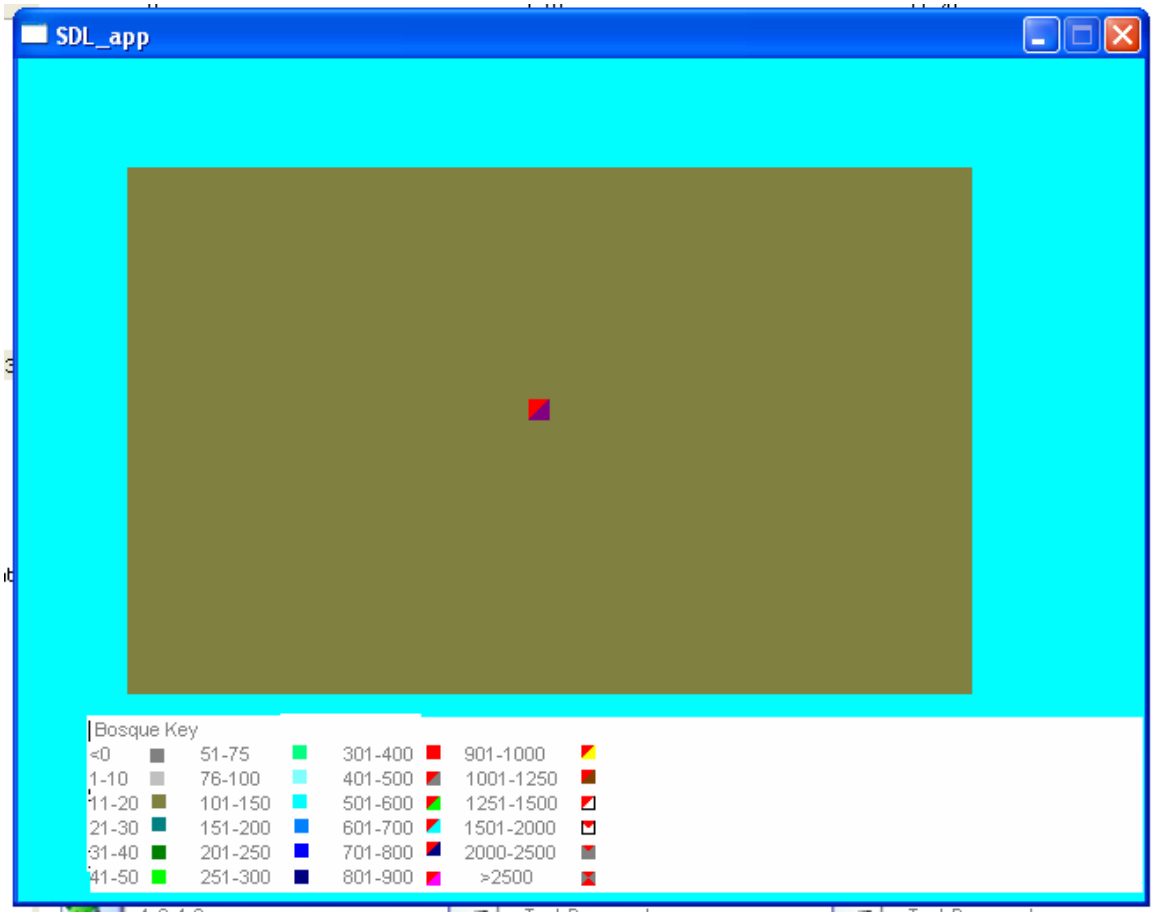


Horrible cout messes up when the fires reach over 10000 degrees C



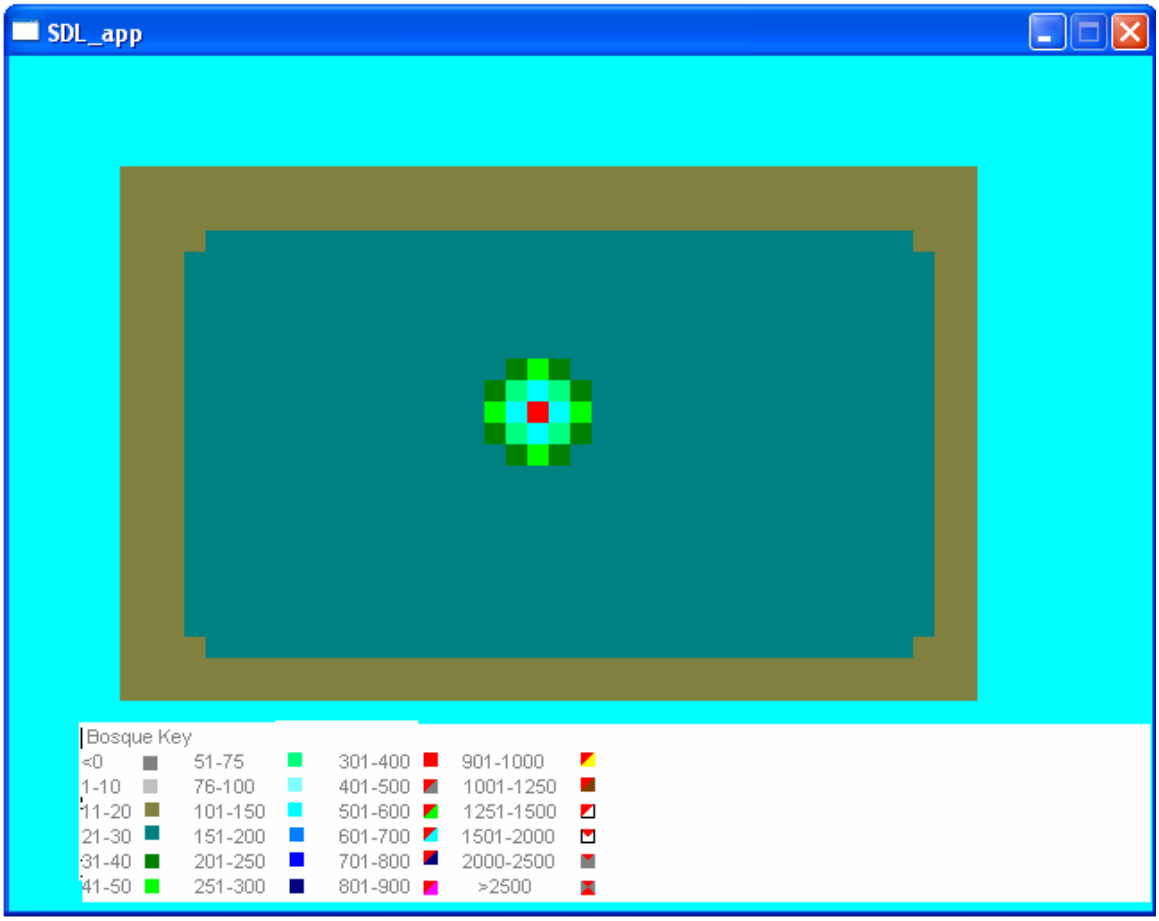
These are results from version two. Really they aren't much to speak of in their MSDOS forms. However before you hold version two for too much scrutiny you must be reminded that this is not the final program. Version three will be ready for the expo.

Results V.3

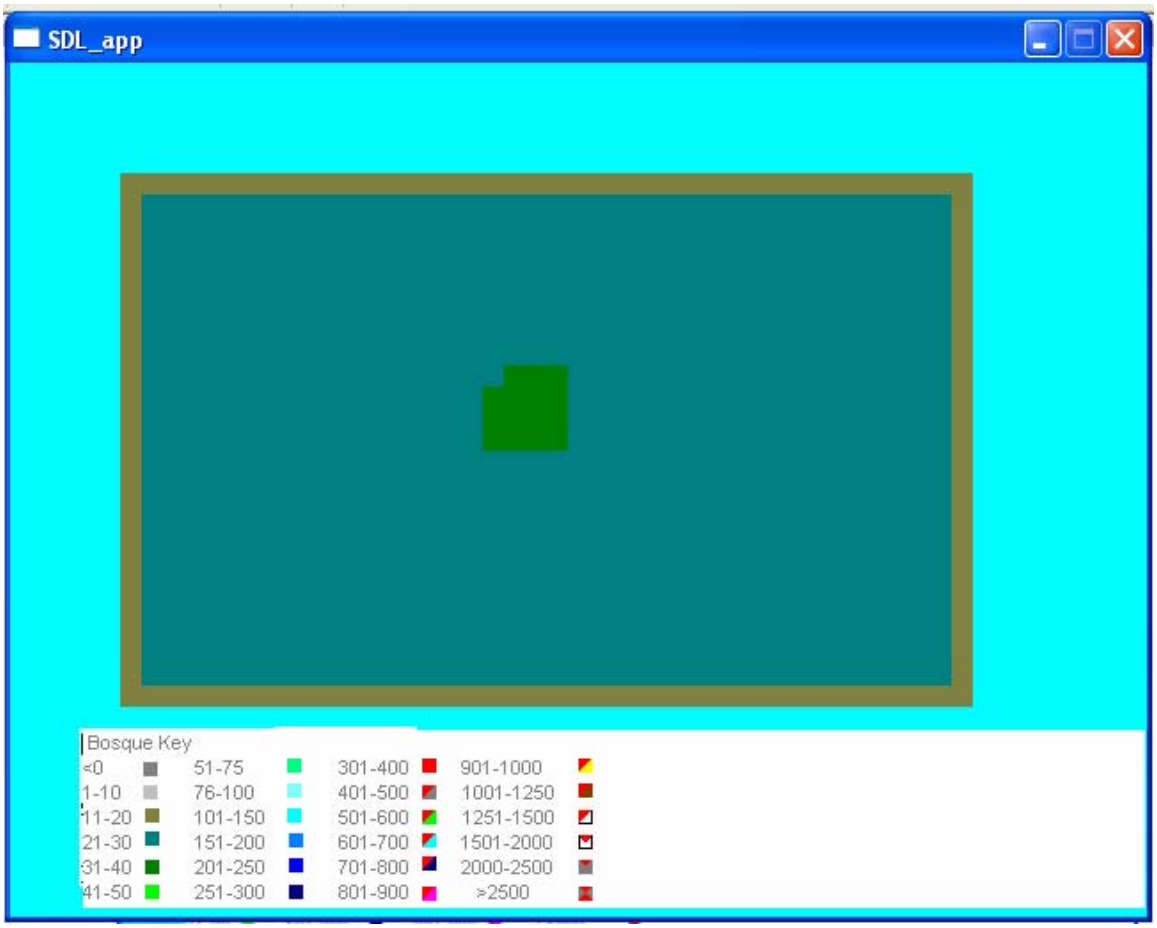


This is the standard screen for version three. What it is, is a large area of patches (26*40) and program shows the temperature of these patches in a patch coding system with a key at the bottom as to what temperatures the patches represent.


SDL_app
Bosque Key
<0
1-10
11-20
21-30
31-40
41-50
51-75
76-100
101-150
151-200
201-250
251-300
301-400
401-500
501-600
601-700
701-800
801-900
901-1000
1001-1250
1251-1500
1501-2000
2000-2500
>2500



This is showing the fire trying to start other. The light blue patches are representing a temperature of 191 to 150 degrees Celsius. That is roughly half the temperature needed to start a fire.



The heat is fading out into the ambient temperature and the fire didn’t spread..

Conclusion

The conclusion to this final report is that the results that were obtained are somewhat inconclusive. This is due to the fact that version three isn't done yet. However, the results obtained by version two were sufficient to say that I accomplished many of the objectives set out at the beginning;

1 .Have an algorithm to map heat flow;

Both the existing version three and version do this quite well.

2. Research fire characteristics and simulate fire;

This was accomplished by logical assumptions, and could have been done slightly better.

3. Research and simulate how heat flow changes under different circumstances

b. wind, rain, elevation,

This was accomplished in version two, however not in version three. There is much room for improvement in these areas by version two and there is a lack of them in version three. However, when version three is done it will incorporate all these variables in its programming and come out with sensible answers.

4. Research and simulate how fire acts with varying fuel sources and circumstances

This objective is easily accomplished, just go into version two and change the variables..

5. Accurately map the Bosque and its surroundings for variables related to objectives three and four.

This objective was never accomplished in the coding, however it all the information to do this has been collected. It will be included in the finished version three.

6. Put all these aspects into a program and decipher environmental damage to nature and its rural surroundings.

Thus far version two can only tell you in its output text file the recorded areas where the fire has been , and wherever the fire has been that area is completely turned to ash. Version three will have a view to where you can actually see the damage the fire has done.

However the best thing that has happened while I was working on the AiSC 2005 challenge is how I have progressed my programming skills. If it wasn't for the AiSC I would have no interest for computer programming at all. Just from taking one class C++ computer science I have progressed from almost complete beginner to a decent programmer with skills beyond MSDOS. As you can probably tell this wasn't a team project (in the beginning it was). So for one person I have concluded that although I may not be the best writer, I believe I have worked hard and would like to show the world what I have done in my three versions of Fire in the Bosque. Even though the work I have done this far isn't outstanding, I still have until the expo in Los Alamos to improve. I will improve so when you see me at the AISC expo come over to my board and view the progress I have made since today.

Really I have no official conclusion for now except that I will continue to work to make version three as great as it can be.

Bibliography

www.howstuffworks.com (great help)

www.bloodshed.net (great compiler)

www.cone3d.gamedev.net

http://www.bosquerevive.com/FireRestoration/reports.htm

http://fire.nist.gov/

http://mb-soft.com/juca/print/311.html

Acknowledgements

Dale Henderson – Helped make this year's AiSC possible!

Debra Loftin

Executive Summary-Fire in the Bosque-a Basic Comprehension of Fire in its Environment

This year the theme for AISC was emergency evacuation. It seemed only proper that team 050 follow along with their project titled “Fire in the Bosque “. In the summer of 2003 two major fires ravaged the Bosque, The Atrisco and Montano fires. Together they burned 263 acres of prime Bosque area. The intent of team 050 is to create a program to model heat flow in the Bosque environment in order to model a fire in the Bosque..

The time and effort of this project was tremendous, especially because the in January nothing was known about programming, and now an SDL rendering is in the works, and all this was done completely independent with little help from anyone else. The product of honing programming skills is; Fire in the Bosque version, one two, and three. All three model heat flow, version two includes all the situational , and environmental factors mentioned in the Methodology of this program and Version three is not complete , however you can see the basic idea of the SDL programming in what has already been programmed. Version two is mostly complete, but was abandoned after SDL was found to be a better way to go for it’s graphical appliances. My results are off in many extents, however I set out to make fire modeling a very basic process, rather than complex. Nonetheless my conclusion is that my incomplete version three needs to be completed before I can draw any major conclusions about a fire in the Bosque. Even though version three is incomplete, version three will be done before the expo. If you would look back at version one and then to version three you will see how very much has been progressed and improved just since February. That is what is team 050’s greatest achievement is- progression in computer programming while working on this years AiSC.

Christopher Morrison