Volcanic Eruption Simulation

PRESENTED BY :-
GUGULOTHU LINGANNA

Volcanic Eruption Simulation Using C Graphics

This project aims to create a captivating visual simulation of a volcanic eruption using the powerful graphical capabilities of the C programming language. The simulation will bring to life the dynamic and awe-inspiring forces of nature, allowing users to experience the raw power of a volcanic event.

BT

Objectives and Scope

1

Realistic Lava Flow

Accurately model and animate the behavior of molten lava as it flows down the mountainside.

2

Ash Cloud Dispersion

Simulate the complex patterns of ash and smoke dispersing through the air during the eruption.

3

Interactive Visualization

Provide users with an engaging and interactive experience to explore the volcanic eruption.

Graphical Capabilities in C

Drawing Primitives

Leverage C's built-in functions for drawing lines, circles, rectangles, and other shapes to create the volcanic landscape.

Color Manipulation

Utilize color palettes and gradients to accurately depict the fiery reds and oranges of the lava and the billowing grays of the ash cloud.

Animation Techniques

Implement frame-by-frame animation to bring the volcanic eruption to life, with smooth transitions and dynamic movements.

Functions Used

- Setfillstyle()
- Setcolor()
- Floodfill()
- Line()
- Rectangle()
- Drawpoly()
- Ellipse()
- Pieslice()
- Outtextxy()
- Delay()
- Closegraph()

Color Code

0.BLACK
1.BLUE
2.GREEN
3.CYAN
4.RED
5.MAGENTA
6.BROWN
7.LIGHTGRAY
8.DARKGRAY
9.LIGHTBLUE
10.LIGHTGREEN
11.LIGHTCYAN
12.LIGHTRED
13.LIGHTMAGENTA
14.YELLOW
15.WHITE

Representing Lava Flow

1

Initial Eruption

Simulate the initial burst of molten lava from the volcano's crater, with realistic patterns and textures.

2

Downslope Movement

Model the lava's descent down the mountainside, accounting for terrain features, gravity, and fluid dynamics.

3

Cooling and Solidification

Depict the gradual cooling and hardening of the lava as it travels, creating a sense of flow and texture.

Simulating Ash Cloud Dispersion

1

Ash Ejection

Simulate the initial ejection of ash and debris from the volcano's crater, creating a towering column.

2

Atmospheric Dispersion

Model the complex patterns of ash and smoke dispersing through the air, influenced by wind and weather conditions.

3

Ground Deposition

Visualize the gradual settling of ash and debris on the surrounding landscape, creating a layer of volcanic ash.

Techniques and Algorithms

Fluid Dynamics

Apply computational fluid dynamics algorithms to realistically simulate the flow and behavior of the molten lava.

Particle Systems

Utilize particle systems to model the complex patterns and movements of the ash cloud and debris.

Terrain Mapping

Incorporate detailed terrain data to accurately depict the landscape and its impact on the lava and ash flow.

Real-time Rendering

Optimize the simulation for real-time rendering, ensuring a smooth and responsive user experience.

CODE :-

```
#include<graphics.h>
#include<conio.h>
#include<conio.h>
#include<math.h>
#include<stdlib.h>
int main()
{
  int gd=DETECT,gm;
  float i=0,j=0;
  int k=0,l=0;
  int trpz[]={349,380,325,150,324,150,300,380};
  int trpzl[]={140,380,300,380,323,160,300,204,260,248,210,292,160,340,150,360,140,380};
  int trpzr[]={349,380,560,380,510,340,460,290,410,240,350,190,326,160,349,380};
  initgraph(&gd,&gm,"C:\\Turboc3\\bgi");
  setcolor(13);
  outtextxy(50,150,"  VOLCANIC ERUPTION  ");
  setcolor(6);
  rectangle(0,380,300,400);
  setfillstyle(1,6);
  floodfill(1,381,6);
```

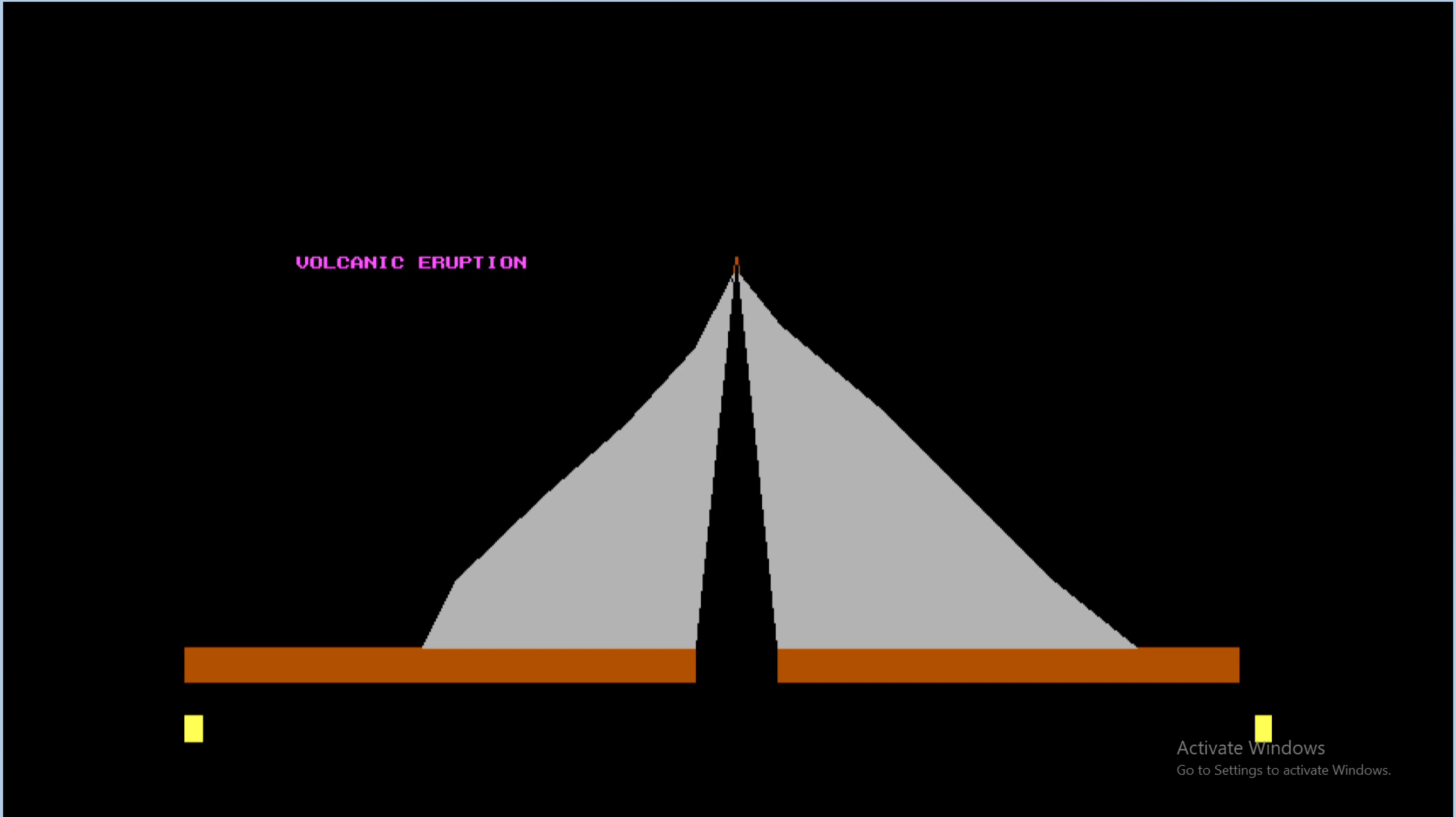
```
rectangle(349,380,620,400);
  setfillstyle(1,6);
  floodfill(350,381,6);
  drawpoly(4,trpz);
  setcolor(7);
  drawpoly(9,trpzl);
   setfillstyle(1,7),
   floodfill(141,379,7);
  drawpoly(8,trpzr);
   setfillstyle(1,7);
   floodfill(350,379,7);
  for(k=0;k<300;k++)
  {
   setcolor(14);
   line(0+k,420,0+k,435);
   line(640-k,420,640-k,435);
   delay(10);
  }
  delay(10);
  setcolor(14);
```

```
ellipse(325,425,0,360,160,25);
setfillstyle(1,14);
floodfill(326,426,14);
delay(5);
for(;i<27 && j<400;)
{
if(j<80)
  setcolor(14);
else if(j>=80 && j<200)
  setcolor(LIGHTRED);

 else if(j>255&&j<400)
 {
  setcolor(4);
  pieslice(325,185,50,130,55);
  setfillstyle(1,4);
  floodfill(326,151,4);
 }
 else
  setcolor(RED);

line(300+i,400-j,350-i,400-j);
delay(20);
i=i+0.1;
j=j+1;
}
```

```
delay(30);
 setcolor(4);
 line(325,125,325,90);
 line(340,125,350,92);
 line(355,130,380,108);
 line(365,150,398,150);
 line(360,165,389,190);
 line(310,125,300,92);
 line(295,130,272,111);
 line(285,150,260,150);
 line(293,165,273,188);
 delay(50);
 for(l=0;l<=5;l++)
 {
  setcolor(8);
  ellipse(325,70-(10*l),0,360,40+(20*l),15);
  setfillstyle(1,8);
  floodfill(326,71-(10*l),8);
  delay(40);
 }
```

```
for(l=1;l<=5;l++)
  {
   setcolor(4);
   pieslice(325,150,244-l,246,15*l);
   pieslice(325,157,310,324-l,14*l);
   setfillstyle(1,4);
   floodfill(320,159,4);
   delay(200);
  }
  getch();
  closegraph();
  return 0;
}
```

Volcanic Eruption Simulation



Challenges and Considerations

Performance

Ensuring the simulation runs efficiently and without lag, even on lower-powered systems.

Realism

Achieving a high degree of visual realism and accuracy in the simulation of natural phenomena.

Interactivity

Providing users with intuitive controls and options to explore and manipulate the simulation.

Data Sources

Incorporating accurate and up-to-date data on volcanic eruptions and their characteristics.

Conclusion and Future Enhancements

Realistic Lava Flow	Achieved through the application of fluid dynamics algorithms and detailed terrain mapping.
Ash Cloud Dispersion	Simulated using particle systems and atmospheric data to create dynamic and authentic dispersal patterns.
Interactive Visualization	Provided users with intuitive controls to explore and manipulate the volcanic eruption in real-time.
Future Enhancements	Incorporate additional volcanic phenomena, such as pyroclastic flows, earthquakes, and magma chamber dynamics, to further expand the scope and realism of the simulation.