Report On

Tree Structure

Submitted in partial fulfillment of the requirements of the Course project in Semester III of Second Year Computer Engineering

by
Prathamesh Patil (Roll No. 59)
Gaurav Patil (Roll No. 54)
Tirupati Pedagolla (Roll No. 61)
Tanmay Patil (Roll No. 60)

Supervisor
Prof. Akshaya Prabhu

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

(2023-24)

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

CERTIFICATE

This is to certify that the project entitled “Title of the project” is a bonafide work of " Prathamesh Patil (Roll No. 59), Gaurav Patil (Roll No. 54), Tirupati Pedagolla (Roll No.61), Tanmay Patil (Roll No. 60)" submitted to the University of Mumbai in partial fulfillment of the requirement for the Course project in semester III of Second Year Computer Engineering.

Supervisor

Prof. Name Surname

Internal Examiner

External Examiner

Dr Megha Trivedi
Head of Department

Dr. H.V. Vankudre
Principal

Plagiarism Scan Report

Dupli Checker

PLAGIARISM SCAN REPORT

2%
Plagiarised

98%
Unique

Date	2023-10-15
Words	667
Characters	5391

Content Checked For Plagiarism

Abstract

Tree structure is an dynamic animated visual graphic project made using standard graphics library of C programming . It employs recursive functions to create branches, resulting in a visually appealing tree-like structure. The tree() function takes parameters such as starting coordinates, length, and angles to determine the branching pattern. The code utilizes basic graphics functions for drawing lines and circles. Additionally, it includes text output to credit the creators. The main() function initializes the graphics environment, iterates through different angles to create multiple trees, and displays the result in a graphical window. The program is written in C and relies on the Turbo C graphics library.

Index Page

Contents

Abstract

Tree structure is an dynamic animated visual graphic project made using standard graphics library of C programming . It employs recursive functions to create branches, resulting in a visually appealing tree-like structure. The tree() function takes parameters such as starting coordinates, length, and angles to determine the branching pattern. The code utilizes basic graphics functions for drawing lines and circles. Additionally, it includes text output to credit the creators. The main() function initializes the graphics environment, iterates through different angles to create multiple trees, and displays the result in a graphical window. The program is written in C and relies on the Turbo C graphics library.

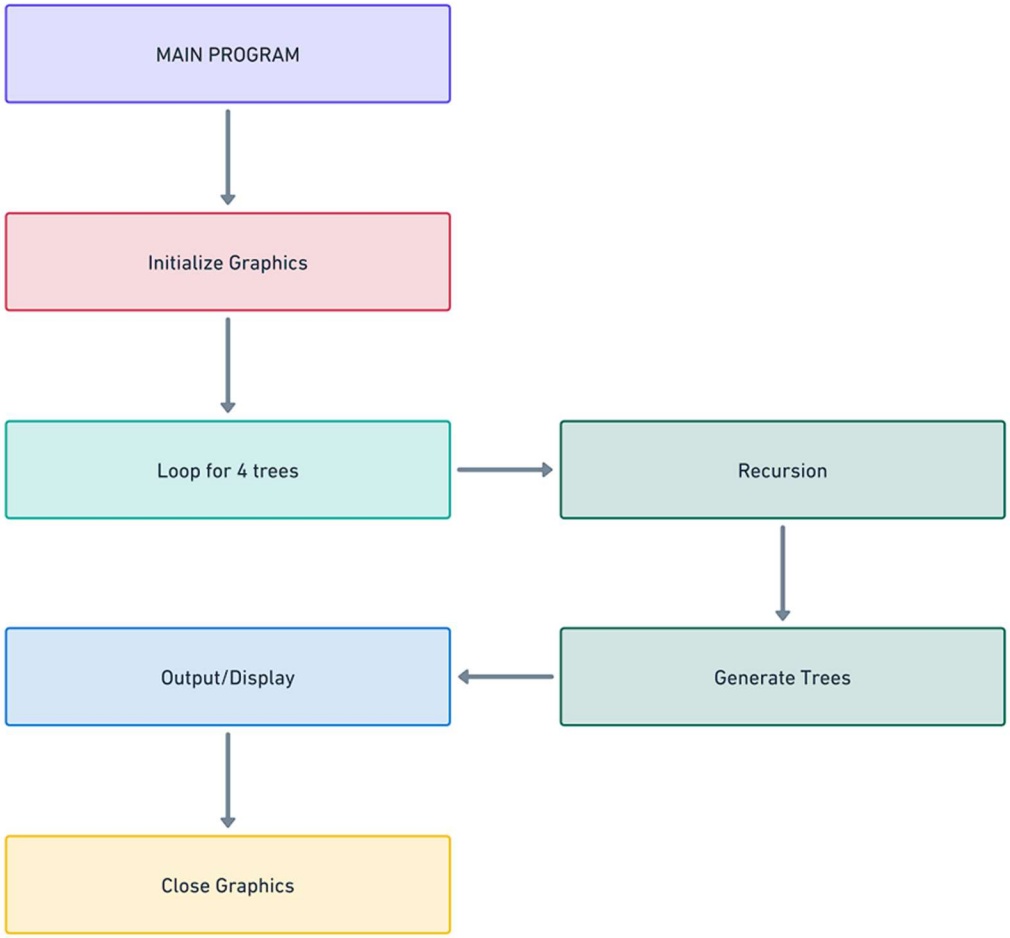
Index Page

Contents

Problem Statement

Design and implement a program in C using the Turbo C graphics library to generate visually appealing recursive tree patterns. The program should allow users to specify parameters such as starting coordinates, initial length, and branching angles. The generated trees should display a natural, aesthetically pleasing structure.

Block Diagram



Working:

1)Main Program: This is the core of the program that orchestrates the entire process.

2)Initialize Graphics: The program begins by initializing the graphics environment, setting up the canvas for drawing.

3)Loop for 4 Trees: A loop runs to generate four different trees. This is achieved by calling the tree() function with different parameters.

4)Generate Tree: The tree() function is responsible for recursively generating the tree structure.

5)Display Result: Once all trees are generated, the program displays the visual output in a graphical window.

6)Close Graphics: Finally, the program ensures the graphics environment is properly closed.

Description

1)Header Files:

stdio.h: Standard Input/Output functions for handling input and output operations.
conio.h: Console Input/Output for console-based input/output operations.
stdlib.h: Standard Library functions for memory allocation and program control.
math.h: Mathematical functions and operations.
graphics.h: Graphics functions for drawing shapes and text.
dos.h: Functions related to system date and time.

2) Tree Function:

void tree(float x, float y, float len, float a, float b): This recursive function generates the tree pattern. It takes parameters such as starting coordinates (x, y), length (len), and branching angles (a and b). It also handles the termination condition when the length of the branch is less than 20.

3)Main Function:

void main(): The main function serves as the entry point of the program. It initializes the graphics environment, sets initial coordinates and angles, and creates multiple trees.

4)Graphics Initialization:

int gd = 0, gm: Variables to hold graphics driver and mode information.
initgraph(&gd, &gm, "C:\\TURBOC3\\BGI"): Initializes the graphics environment with specified driver and mode.

5)Loop for Creating Trees:

The main function contains a loop that generates multiple trees by calling the tree() function with different angles.

6)Drawing and Visualization:

The program uses graphics functions like line(), circle(), and outtextxy() to draw lines, circles, and text respectively, creating the tree pattern.

7)User Interaction:

getch(): Waits for a key press before closing the graphical window.

Brief description of software& hardware used and its programming

Software:

The project utilizes the Turbo C environment for development. Turbo C is an integrated development environment (IDE) and compiler for the C programming language. It was popular during the late 1980s and early 1990s. Though older, it's known for its simplicity and was widely used for DOS-based programming. The specific version used in this project is Turbo C 3.0.

Hardware:

Since Turbo C is a relatively lightweight IDE, it can run on a wide range of hardware configurations. It's compatible with computers of that era, including those with early Intel processors, limited RAM, and VGA display capabilities. The project's graphical output relies on the capabilities of the graphics card of the host machine.

Code:

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<math.h>
#include<graphics.h>
#include<dos.h>
void tree(float x,float y,float len,float a,float b)
{
   float x1,y1;
if(len<20)
{
  delay(50);
  setcolor(14);
  circle(x,y,5);
  setcolor(100);
  circle(x,y,6);
  return;
}
  settextstyle(1,0,1);
  setcolor(14);
  outtextxy(150,5,"CREATED BY PRATHAMESH,GAURAV,TIRUPATI,TANMAY");
  setcolor(10);
  x1=x+len*cos(3.1428*a/180);
  y1=y-len*sin(3.1428*a/180);
  line(x,y,x1,y1);
  tree(x1,y1,len*0.75,a-b,b);
  tree(x1,y1,len*0.75,a-b,-b);
}
void main()
{
   int gd=0,gm,i,rd;
   float x=320,y=550,t=140;
   clrscr();
   initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
   for(i=0;i<=3;i++)
   {
      tree(x+i,y,t,90,30);
      tree(x+i,y,t,90,-30);
   }
   getch();
   closegraph();
}
```

Results and Conclusion

Results:

The tree project successfully generates captivating recursive tree patterns using the Turbo C graphics library. Upon execution, the program creates multiple trees, each with distinct branching angles, resulting in visually intricate and appealing tree-like structures. The graphics environment provided by Turbo C facilitates the clear visualization of the generated patterns.

Conclusion:

This project demonstrates the effectiveness of using recursive algorithms in generating complex visual patterns. By leveraging the power of C programming and the Turbo C graphics library, the program efficiently creates a series of trees with varying parameters. The utilization of recursion allows for the creation of branching patterns that closely resemble natural trees. Additionally, the inclusion of credits showcases a thoughtful touch, acknowledging the creators of the project.

However, it's worth noting that the program is designed for use in a specific environment (Turbo C on DOS), which may limit its compatibility with modern systems. As a retro-styled project, it serves as a valuable educational tool for understanding the principles of recursion and graphics manipulation in a historical context.

References

1)Books Preferred:

Let us C by Yashwant kanetkar
Computer Graphics [Class Modules]

2) Website Resource

https://www.youtube.com
https://github.com
https://whimsical.com