2D Graphical Arrangement Tool incorporating L-System

by

Nur Shazana bt. Ghazali

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
(Information Communication Technology)

JANUARY 2006

Supervisor: Mr. Mohd. Nordin Zakaria

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

2D Graphical Arrangement Tool incorporating L-System

by

Nur Shazana bt. Ghazali (3975)

A project dissertation submitted to the
Information Communication Technology Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION COMMUNICATION TECHNOLOGY)

Approved by,

(Mr. Mohd Nordin b Zakaria)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
January 2006

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

NUR SHAZANA GHAZALI

ABSTRACT

A lot of software for arrangement tools and drawing tools had been developed nowadays. Drawing tools have made drawing much easier. Everybody can draw what ever they want to draw by using the drawing tools. Although there are many drawing tools in the market nowadays but none of them have integrated the L-System into it. L-System is a model consists of an initial string called the axiom and a set of production rules which are iteratively applied to that string. The string is examined for single characters which have associated production rules. When a rule is found; the string which corresponds to that character (as defined by the production rule) is substituted into the string. This project is basically a drawing tool that integrates the element of L-System into it. In this project, it still focused on the L-System but the difference is it applies in a way where user just can play around with the models. By using the tools provided, user can see the usefulness of L-System. The objectives of this project are to create user interface that integrate 2D L-System with the functions of arrangement tools, to understand on L-System, to understand on Open GL User Interface (GLUI) and to show the usefulness of 2D L-System. The project is developed using the Open GL and C++ language and the engine for the L-System tree is using the LParser.

ACKNOWLEDGEMENTS

First and foremost, the author would like to praise God the Almighty for His blessing for the author to complete this "2D Drawing Tools" Project in the time frame.

The author would like to express the most gratitude to Mr. Nordin, Supervisor upon the completion of this project. Without her generous support and guidance, the author may not be able to do this project properly. A very special thank you expressed to Mr. Nordin for the time which have been allocated to assist the author upon completing the project.

The author also would like to express the appreciation to Mohd Hakimi Jauhari Jamalus, a very good friend of the author, in helping the author related to the project throughout this semester. Without his help, the author may not be able to achieve the objectives of this project.

Last but not least, the author also would like to express this thankfulness to all the lecturers for all the knowledge and information given. All the knowledge and information is used as guidance towards completing this project. This appreciation also goes to family and all colleagues who assist the author directly or indirectly during completing this project. Thank you.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Background of Study

Drawing is one of many ways to making an image. It is the process of making marks on a surface by applying pressure from or moving a tool on the surface. These marks may represent what the artist sees when drawing, a remembered or imagined scene or abstraction, or, in the case of automatic drawing, may have much to do with the automatic motion of the artist's hand across the paper [13]. People who can draw can be considered as gifted because not all people can draw. People draw because for some reason such as they want to express their opinion and feelings, to explore the imagination and identity and also as the communication tools. Drawing also been used to develop ideas and concepts. Drawing can be done not by hand only but because of the technology, lots of software has been developed to ease the drawing process. For example, drawing tools that have developed and used by many people are Adobe Photoshop, open canvas and Microsoft Paint. By using the drawing tools, people will save more time and cost and the result are more accurate. For example when using the Microsoft Paint, people can easily uploading and editing the image and do to the image whatever they want it to do. For this project I will developed a 2D drawing and arrangement tool that integrate with 2D L-System. An L-system or Lindenmayer system is a formal grammar (a set of rules and symbols) most famously used to model the growth processes of plant development, though able to model the morphology of a variety of organisms. L-systems can also be used to generate self-similar fractals such as iterated function systems [14]. By using the 2D L-System it will produce a 2D tree.

In this project, I will develop a 2D graphical arrangement tool where user can play around with the figures such as rearrange the object. My project will integrate some of L-System, polygon and texturing using the Open GL and visual C++. I create the figures using polygon, lighting and some of texturing. In this project also I used Open GL user interface (GLUI) where user can interact and choose what the models that to be put in the interface.

The difference between my project and project is it focused on the integration of L-System where the main figure for the project is the tree that used L-System grammar. The background and other model are own created where there will be several adding models such as house, mountains and sun.

1.1.1 2D drawing tools

A lot of software for drawing tools had been developed nowadays. Drawing tools have made drawing much easier. Everybody can draw what ever they want to draw by using the drawing tools. Below are some of the examples of drawing tools that have in the market nowadays.

Microsoft Paint

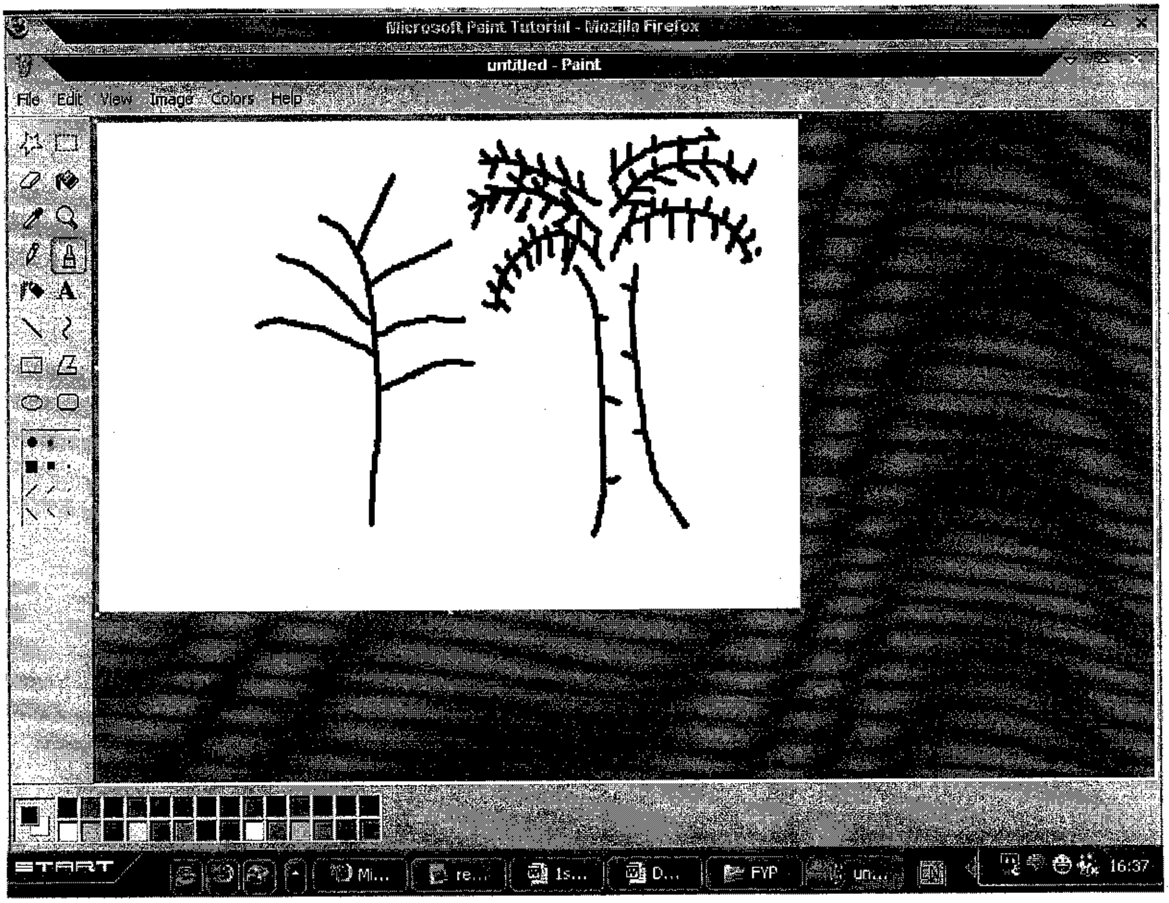


Figure 1.1.1 (a):Example of using Microsoft Paint

By using the Microsoft Paint, user can choose what tool that they want to choose. For example above, user choose to use brush for the drawing.

Adobe Photoshop

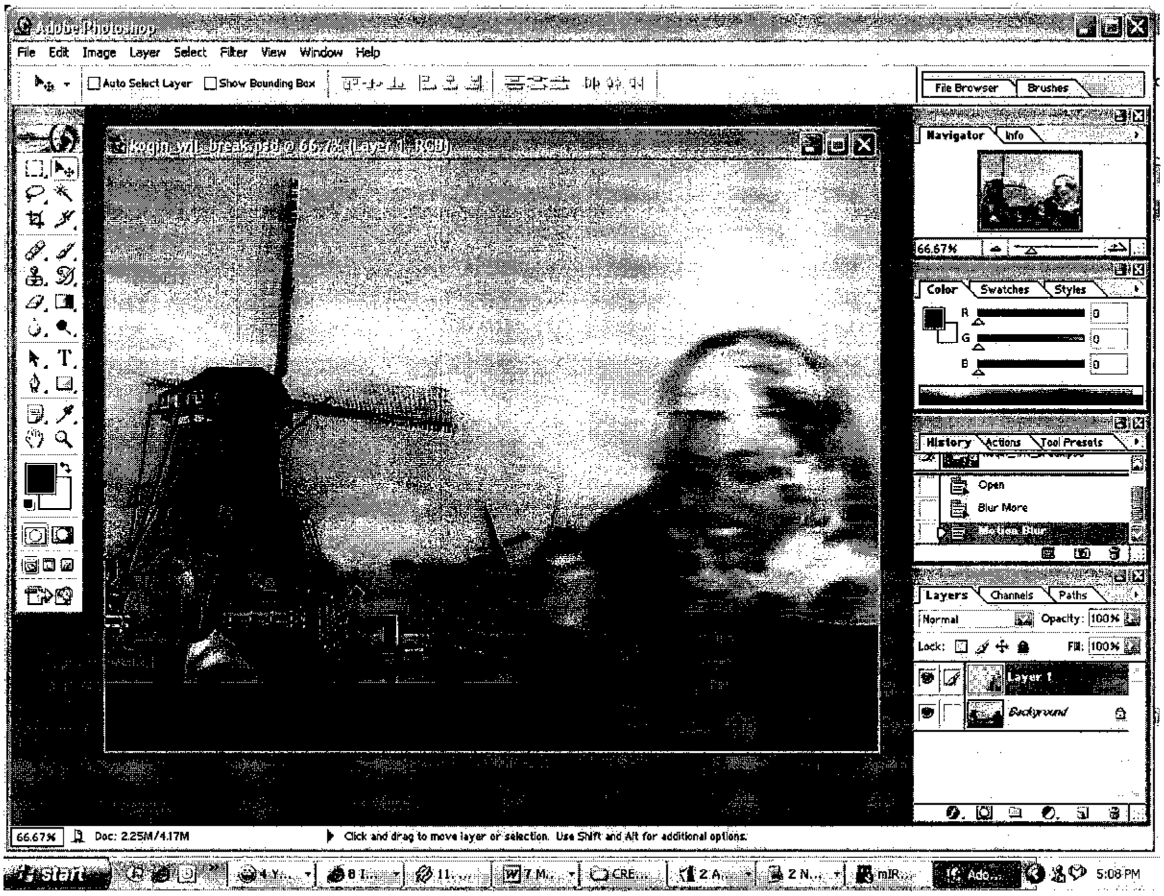


Figure 1.1.1 (b) Example of using Adobe Photoshop

For the example above, user used Adobe Photoshop to edit the image. User put some effect of blurring into the image and also edits the image where user put background from other picture and combines it together.

1.1.2 L-System [Lindenmayer, 1960's]

L-System is developed by Aristid Lindenmayer in the late 1960's. L-systems are rule-based productions based on Chomsky string replacing grammars where the productions are applied in parallel. The grammar used in L-systems is typically interpreted as Turtle Graphics commands. The resulting images are fractal-like expanding on the concept that simple rules when recursively applied can create complex structures (ala

Chaos/Complexity theory). It is systematically extended to provide greater flexibility and more accurate plant models. Below are some examples of 2D L-System:

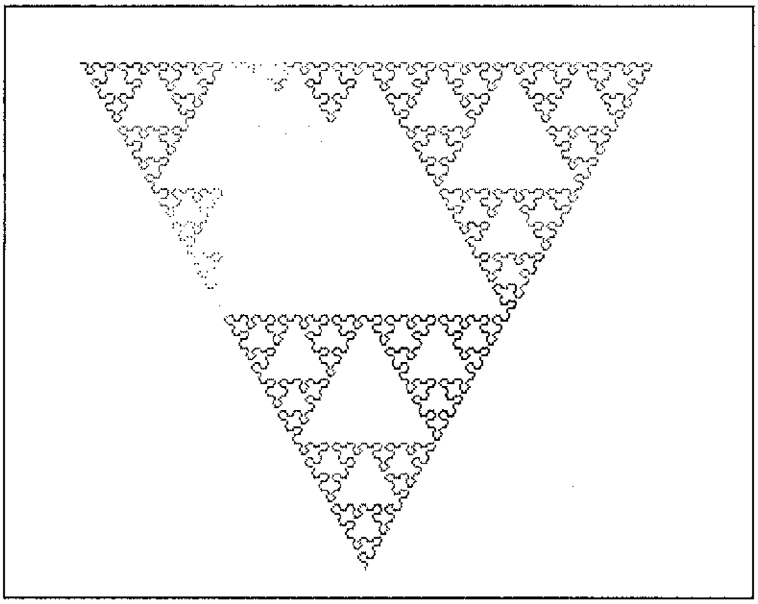


Figure 1.1.2 (a): Arrowhead curve (`X -> YF+XF+Y, Y -> XF-YF-X`**, 60°)**

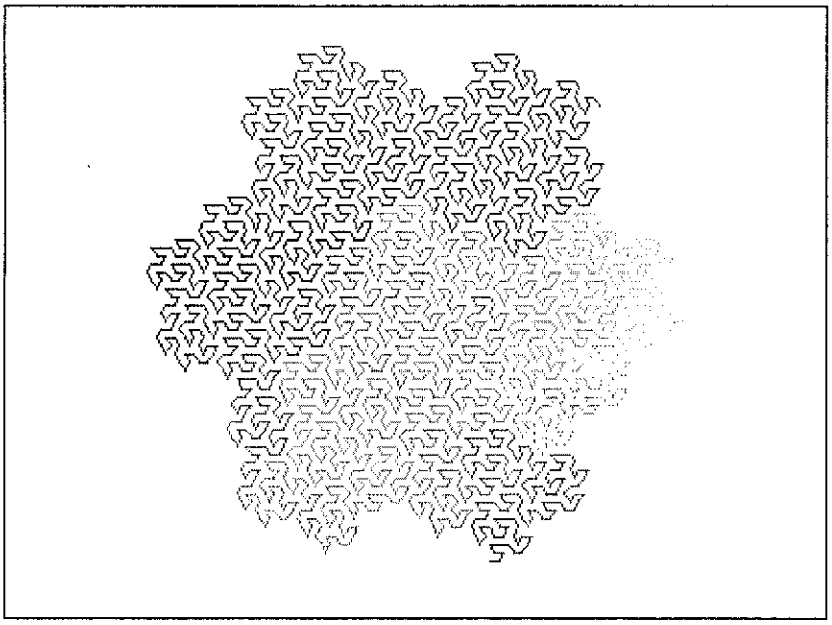


Figure 1.1.2 (b): Hilbert curve (`L -> +RF-LFL-FR+, R -> -LF+RFR+FL-`**)**

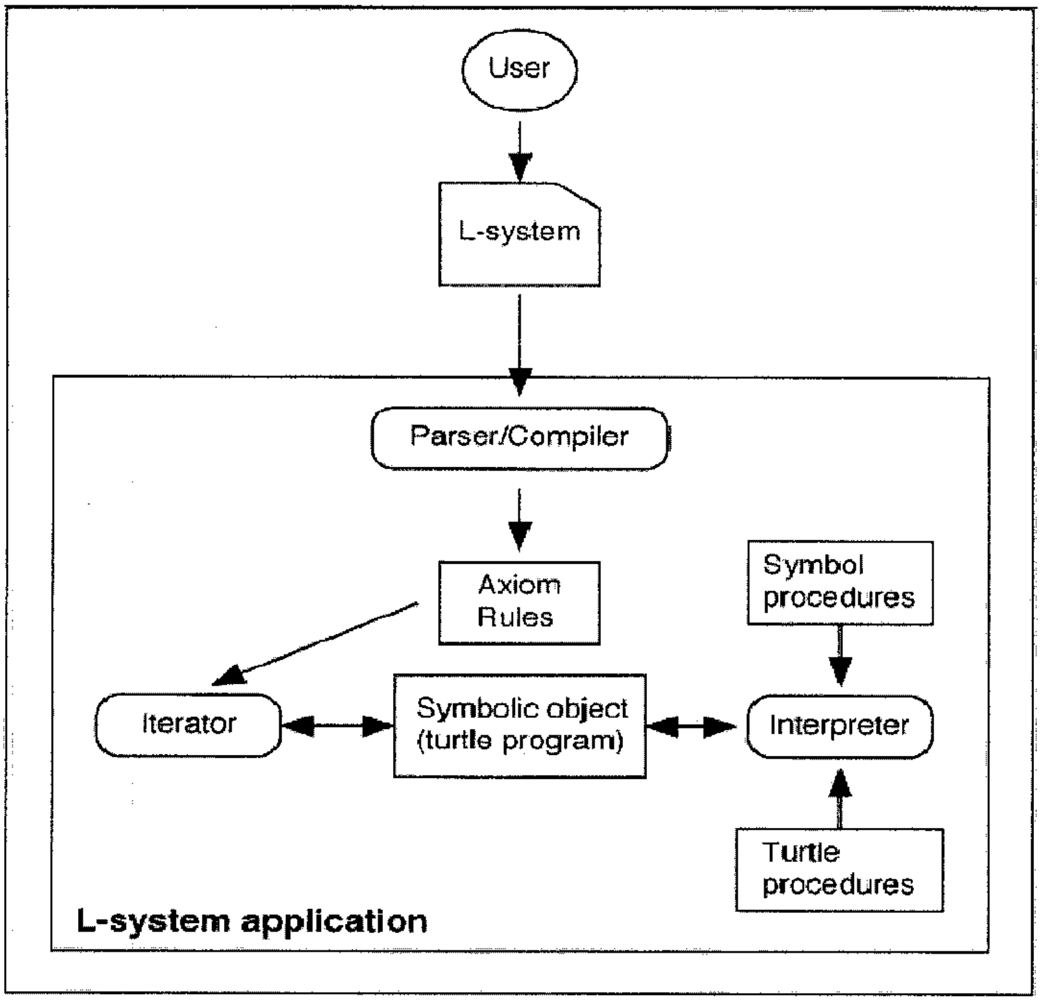


Figure 1.1.2 (c): L-System Architecture

1.1.3 L-Parser

I used the L-Parser as the engine of the L-System tree. L-Parser is the implementation of L-System parser (called **Lparser**) that also allows mutations to take place during the growing of the form. These mutations can change the form slightly or quite dramatically. This allows user to create a form once and then create a whole series of forms all clearly descendant from the original Lsystem.

1.1.4 Open GL user interface (GLUI)

GLUI is used to make the system more user-friendly and easy to be used. Usually, when using the Open GL, the output of the program will run on the blank interface and if we want to change anything, we will change it at the coding. GLUI is a GLUT-based C++ user interface library which provides controls such as buttons, checkboxes, radio buttons, spinners, and list boxes to OpenGL applications. It is window-system independent, relying on GLUT to handle all system-dependent issues, such as window and mouse management.

1.2 Problem Statement

1.2.1 Problem Identification

Although there are a lot of drawings tools nowadays, none of them have integrate the L-System into a drawing tool. Most of the drawings tools are much likely the same where user can edit, zoom, colored and many more. There are no drawings tools that can interact with user such as when user clicks at the image none will happen unless user chooses to zoom it or edit it or color it and also even though it has proven that L-System is part of many official standards worldwide, but there is still area in the implementation can be improved and vary it. Lots of implementation of L-System has been done but by varying it can make people easy to understand what is L-System is all about By implementing it in different way such as drawing tool, people can easily understand what L-System can do. Most of the software 2D drawing tool are focusing on the images that been import.

1.2.2 Significant of the project

In this project, I want to implement the L-System in more user-friendly and fun. In order to make it more interactive and interesting, one way is by developing a project like a drawing tools but in the other hand also integrate the L-System. The expected end of this project is a 2D drawing tool that uses the L-System as one of the models. The idea of this prototype is to give an understanding and knowledge about what the L-System is basically and to create a 2D nature environment. With the development of this project, user who used this software would not have to draw the grammar tree but use the L-System that have been integrated in this project.

1.3 Objectives

1.3.1 To create user interface that integrate 2D L-System with the functions of arrangement tools

As the project is to provide arrangement tools, the author has to learnt more about the function and coding of Open GL and language of C++. In this part, reading and studying of the Open GL Red and Super Bible is important. This is because the knowledge will be used to develop the project.

1.3.2 To understand on L-System

In order to understand the L-System, the author needs to be well-versed of all grammar and algorithm of L-System. This is an essential part as next the author needs to convert the coding into C++ language and can be run using the Open GL and GLUI.

Up to this extend, the author has understand the theoretical facts about L-System.; appropriate method on implementing the L-System, the algorithm and the grammar.

1.3.3 To understand on Open GL User Interface (GLUI)

As the project is about implementing the environment and L-System into the interface, the author needs to know and develop a skill of GLUI in order to create an appropriate and efficient interface that are more user-friendly and easy to understand. As stated in GLUI user guide by Paul Rademacher [10]; many applications can be built using only the standard GLUT input methods - the keyboard, mouse, and pop-up menus. However, as the number of features and options increases, these methods tend to be greatly overworked. It is not uncommon to find glut applications where almost every key on the keyboard is assigned to some function, and where the pop-up menus are large and cumbersome. The GLUI User Interface Library addresses this problem by providing standard user interface elements such as buttons and checkboxes. The GLUI library is written entirely over GLUT, and contains no system-dependent code.

1.3.4 To show the usefulness of the 2D L-System

As we know, 3D drawing is more realism compared to the 2D. Although there is a lot of 3D implementation, there are always advantages of using the 2D. By using the 2D, we can easily understand and see the structure of the models because it only involves the x-axis and y-axis. By using the 2D modeling, the structure of the tree will look clearer. By implementing the 2D we can always integrate it to 3D models once we fully understand the 2D structure.

1.4 Report Organization

The report is organized as follows where in Chapter 2, we will look at the literature review which include about 2D L-System, Open GL User Interface (GLUI), C++ language and 2D drawing tools. In Chapter 3, we will look at the project planning which is about requirement analysis, tasks and domain analysis, design of the environment and interaction, design of user support and navigation analysis and evaluation. In chapter 4, I will discuss on the results of the project. Finally, in Chapter 5 I will conclude about the project and also put some recommendations that can be done for future enhancement.

CHAPTER 2: LITERATURE REVIEW AND THEORY

The significance of this project is to develop a drawing tool that integrate the 2D-Lsystem and put in the user interface and can be play around with it. The idea of 2D drawing tools is to show what can be done using the 2D L-System that can be more interesting to learn and understand. The first challenge that the author has to cope is understand the L-System and implement it into the OpenGL user interface. Studies have to be made on the algorithms of 2D L-System and methods used in the GLUI. The next thing that needs to be considered is to link the L-System file into the GLUI. In order for the author to integrate the 2D L-System into the GLUI and provides a drawing tool in the program, sufficient related work pertaining this area needs to be reviewed. Important points of related works are discussed in this section.

2.1 2D L-System

L-systems (also called Lindenmayer systems or parallel string-rewrite systems) are a compact way to describe iterative graphics using a turtle analogy. An L-system is created by starting with an axiom, such as a line segment, and one or more production rules, which are statements such as "replace every line segment with a left turn, a line segment, a right turn, another segment..." When this system is iterated several times, the result is often a complicated fractal curve. [1]

Throughout the research, I learnt that much explanation has been given about the L-System. One that I have found is the basic functionality of an L-System model consists of an initial string called the axiom and a set of production rules which are iteratively applied to that string. The string is examined for single characters which have associated production rules, when a rule is found; the string which corresponds to that character (as defined by the production rule) is substituted into the string. Each successive iteration produces a longer and more complicated string of characters. [2]

L-systems (Lindenmayer, 1968; Prusinkiewicz, 1999) provide a basically modular approach to modeling, enabling plants and canopies to be described as a collection of modules. In a functional structural approach, L-systems are combined with environmental models. The virtue is that each next step in the development of the 3D structure can be made dependent of external driving forces, acting on the structure as a whole, but also (and more interestingly) on each individual module. Such an approach is particularly suited for modeling tillering in wheat as influenced by light conditions at the level of the individual organs. For this to be correctly simulated, it is necessary to establish accurate values for the developmental kinetics and geometry of the cultivar of interest. [6]

In a 2d L-system the final string, after all iterations have been applied, is used as a set of drawing commands to produce a rendering of the plant structure. The drawing commands encoded in the string are used to operate on a cursor, also known as a turtle (from the LOGO programming language), that can be rotated by a given angle and moved along its current heading. For each segment a line is drawn connecting the previous cursor position with the new cursor position.

2D drawing commands:

F: Draw a segment

f : Move cursor without drawing

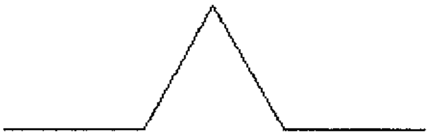
+: Rotate cursor clockwise by given

- : Rotate cursor counter-clockwise by given angle

When programming L-systems, one typically represents the axiom as a sequence of characters, such as F, and the production rules as replacement rules of the form F -> F+F--F+F. Then, carry out the string replacements (in parallel) as many times as desired. [2]

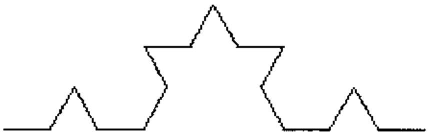
We start with the axiom F, and replace every occurrence of F with F+F--F+F, resulting in the string and the result will look like:

F+F--F+F.



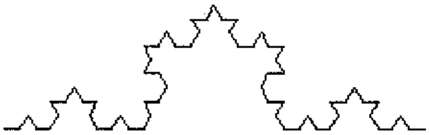
We then replace every occurrence of F in the result with F+F--F+F, resulting in the string

F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F.



Carrying out the replacement one more time results in the following:

F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F-- F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F.



[1]

Some other definition that I found about L-System is An L-system consists of a set of textual rules called *productions* that describe the development of plant branches, leaves, flowers, and other components. In a *generation phase* these productions are applied in a sequence of *derivation steps* to the initial string, called the *axiom*. The state of the L-system model after any number of steps is encoded in a string of symbols, called the *L-string*. In a subsequent *interpretation phase* the L-string is converted to a geometric representation of a plant. L-strings encode form using *turtle geometry* [1]. A turtle, starting at a specified location and orientation in world-space, interprets an Lstring as series of position- and orientation-changing instructions. The position of the turtle is represented by a vector ***p***; its orientation is given by three vectors ***h***, ***l***, and ***u***, indicating in the turtle's local frame of reference which directions are forward (or *heading*), *left*, and *up*. As the L-string is scanned from left to right, these four state vectors change according to the instructions encoded in the string. [9]

2.2 Open GL User Interface (GLUI)

The OpenGL Utility Toolkit (GLUT) is a popular user interface library for OpenGL applications. It provides a simple interface for handling windows, a mouse, keyboard, and other input devices. It has facilities for nested pop-up menus, and includes utility functions for bitmap and stroke fonts, as well as for drawing primitive graphics objects like spheres, tori, and teapots. Its greatest attraction is its *window system independence*, which (coupled with OpenGL's own window system independence) provides a very attractive environment for developing cross-platform graphics applications. Many applications can be built using only the standard GLUT input methods - the keyboard, mouse, and pop-up menus. However, as the number of features and options increases, these methods tend to be greatly overworked. It is not uncommon to find glut applications where almost every key on the keyboard is assigned to some function, and where the pop-up menus are large and cumbersome. The GLUI User Interface Library addresses this problem by providing standard user interface elements such as buttons and checkboxes. The GLUI library is written entirely over GLUT, and contains *no system-dependent code*. A GLUI program will therefore behave the same on SGIs, Windows machines, Macs, or any other system to which GLUT has been ported. Furthermore, GLUI has been designed for programming simplicity, allowing user interface elements to be added with one line of code each. [10]

GLUI can associate *live variables* with most types of controls. These are regular C variables that are automatically updated whenever the user interacts with a GLUI control. For example, a checkbox may have an associated integer variable, to be automatically toggled between one and zero whenever the user checks or uncheck the control. A editable text control may maintain an entire character array as a live variable, such that anything the user types into the text box is automatically copied into the application's character array. This eliminates the need for the programmer to explicitly query each control's state to determine their current value or contents. In addition, a

GLUI window can send a *GLUT redisplay message* to another window (i.e., a main graphics window) whenever a value in the interface is changed. This will cause that other

window to redraw, automatically using the new values of any live variables. For example, a GLUI window can have a spinner to manipulate the radius of an on-screen object.
When the user changes the spinner's value, a live variable (say, float radius) is automatically updated, and the main graphics window is sent a redisplay message. The graphics window then redraws itself, using the current (that is, the updated) value of radius - unaware that it was changed since the last frame. Live variables help make the GLUI interface transparent to the rest of the application. [10] Below, the author has put some coding and snapshot just to show the output of GLUI.

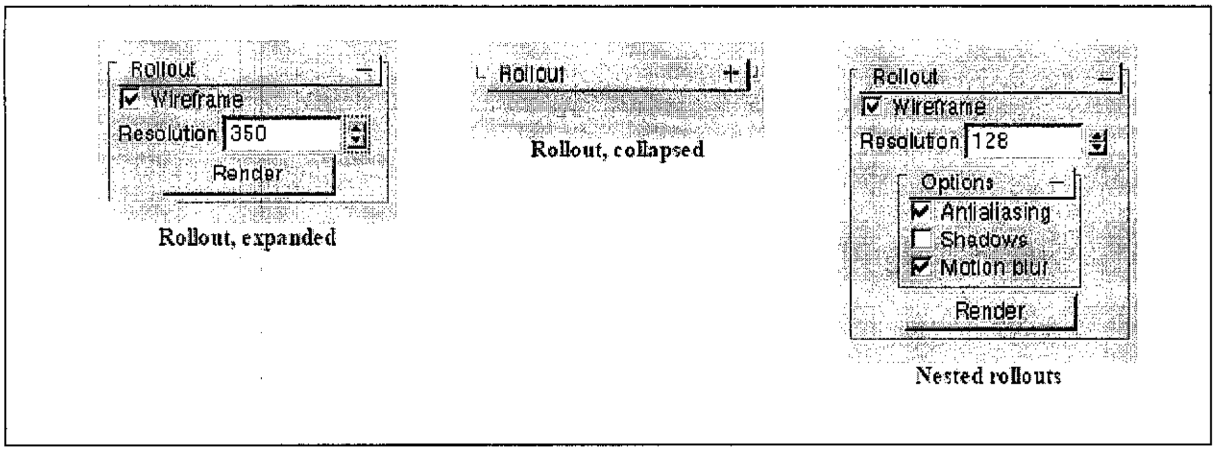
Examples:

add_rollout, add_rollout_to_panel

Adds a new rollout to a GLUI window, optionally nested within another rollout or panel.

Usage

```
GLUI_Rollout *GLUI::add_rollout( char *name, int open = true );
GLUI_Rollout *GLUI::add_rollout_to_panel( GLUI_Panel *panel, char *name,
int open = true );
name - Label to display in the panel. If string is empty, no label is
displayed
open - If true, rollout will initially be open. If false, rollout will
initially be collapsed.
panel - Panel (or rollout) to place column in.
Returns: A pointer to a new Rollout control.
```



Rollout, expanded

Rollout, collapsed

Nested rollouts

Figure 2.2(a): Example of roll out using GLUI

Columns

```
glui->add_checkbox("foo");
glui->add_checkbox("bar");
GLUI_Panel *panel = glui->add_panel( "Panel" );
glui->add_checkbox_to_panel(panel, "Hello");
glui->add_column_to_panel(panel, true);
glui->add_checkbox_to_panel(panel, "World!");
```

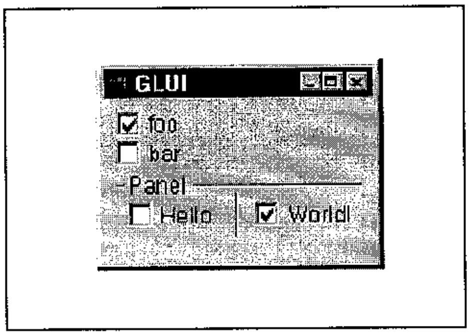


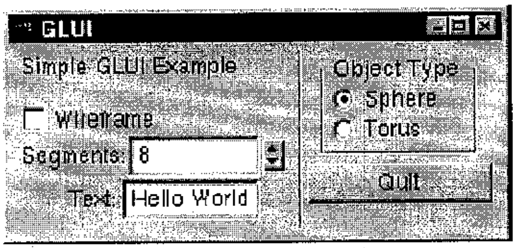
Figure 2.2(b: Example of columns using GLUI

Full coding in creating simple user interface:

```
/** Now create a GLUI user interface window and add controls **/
GLUI *glui = GLUI_Master.create_glui( "GLUI", 0 );
glui->add_statictext( "Simple GLUI Example" );
glui->add_separator();
glui->add_checkbox( "Wireframe", &wireframe, 1, control_cb );
GLUI_Spinner *segment_spinner =
glui->add_spinner( "Segments:",GLUI_SPINNER_INT, &segments );
segment_spinner->set_int_limits( 3, 60, GLUI_LIMIT_WRAP );
GLUI_EditText *edittext =
glui->add_edittext( "Text:", GLUI_EDITTEXT_TEXT, text );
glui->add_column(true); /** Begin new column - 'true' indicates **/
/* * a vertical bar should be drawn **/

GLUI_Panel *obj_panel = glui->add_panel ( "Object Type" );
GLUI_RadioGroup *group1 =
glui->add_radiogroup_to_panel(obj_panel,&obj,3,control_cb);
glui->add_radiobutton_to_group( group1, "Sphere" );
glui->add_radiobutton_to_group( group1, "Torus" );
glui->add_button( "Quit", 0,(GLUI_Update_CB)exit )
```

```
/** Tell GLUI window which other window to recognize as the main gfx
window **/
glui->set_main_gfx_window( main_window );
/** Register the Idle callback with GLUI (instead of with GLUT) **/
GLUI_Master.set_glutIdleFunc( myGlutIdle );
/** Now call the regular GLUT main loop **/
glutMainLoop();
}
```



□ Wireframe
Segments: 8
Text: Hello World
● Sphere
○ Torus

Figure 2.2(c): Open GL user interface example

2.3 C++ Language

The language that been used in this project is C++. This is because C++ is fully integrated editor, compiler, and debugger and has the ability to create complex software system Open GL user interface (GLUI). The C++ programming language provides a model of memory and computation that closely matches that of most computers. In addition, it provides powerful and flexible mechanisms for abstraction; that is, language constructs that allow the programmer to introduce and use new types of objects that match the concepts of an application. Thus, C++ supports styles of programming that rely on fairly direct manipulation of hardware resources to deliver a high degree of efficiency plus higher-level styles of programming that rely on user-defined types to provide a model of data and computation that is closer to a human's view of the task being performed by a computer. These higher-level styles of programming are often called data abstraction, object-oriented programming, and generic programming. [11]

2.4 2D Drawing Tools

Drawing is a means of making an image, using any of a wide variety of tools and techniques. It generally involves making marks on a surface by applying pressure from a tool, or moving a tool across a surface. Common tools are graphite pencils, pen and ink, inked brushes, wax color pencils, crayons, charcoals, pastels, and markers []. There is many computers software that provides 2D drawing tools. Some of the 2D drawing tools are:

InkScape

Inkscape is a Free Software vector graphics editor. Its goal is to become a powerful graphic tool while being fully compliant with the XML, SVG and CSS standards. It is a cross-platform application that runs on Microsoft Windows, Mac OS X and Unix-like operating systems; the primary development platform is Linux. Some features that have in InkScape are object creation, object manipulation, fill and stroke, text support and rendering. One of the priorities of the Inkscape project is interface consistency and usability. This includes GNOME HIG compliance, universal keyboard accessibility, and convenient on-canvas editing [17]. Below are some of the snapshots using the InkScape:

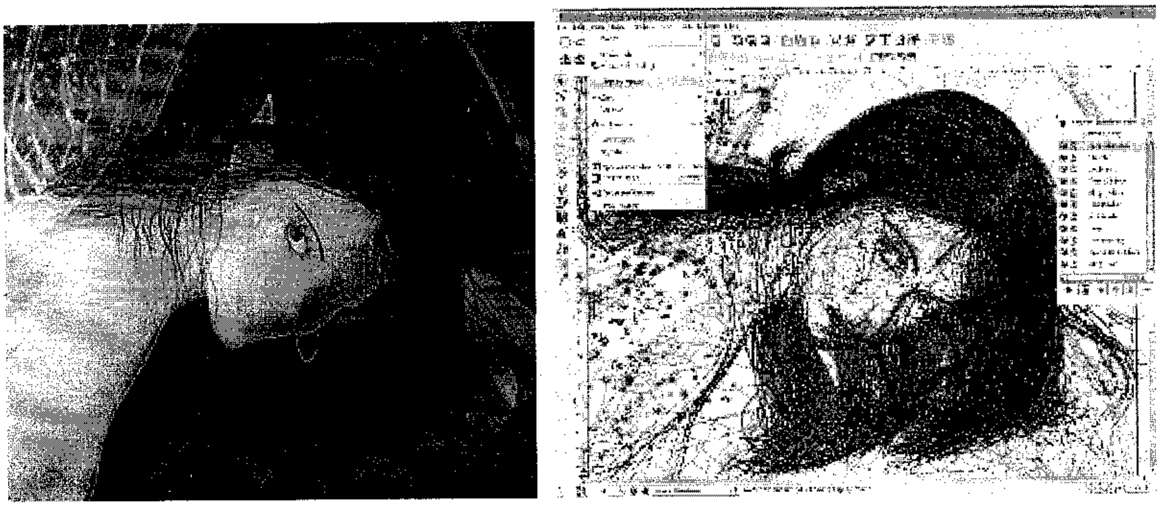


Figure 2.4 (a): Example of using Inkscape

When working on complex projects with thousands of objects, it can be quite a chore to efficiently manage the contents of the document. One difficulty may be finding those pesky objects that share the same color as the document background. Another more common difficulty presented is managing layers in an intuitive way. As shown here in this screenshot by ScislaC, the **Outline** display mode reveals hundreds of objects layered behind foreground objects throughout this piece Gaze [16].

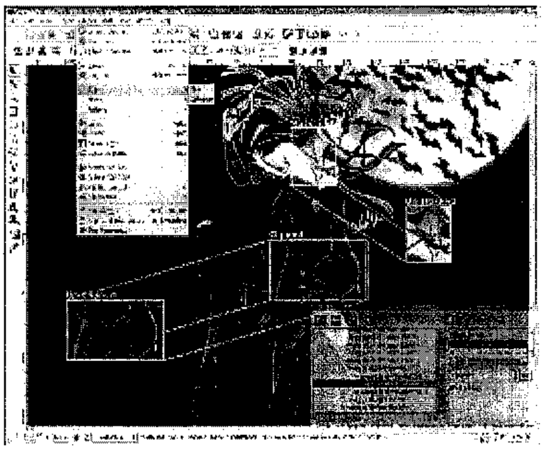


Figure 2.4 (b): Example of using Inkscape

One of the benefits of working with vectors is the ability to non-desructively modify the appearance of objects. In the new release, it added support for two new features to do just that. As shown in this screenshot by ScislaC, it have **Clipped** and **Masked** objects. The bat that is selected is a complete shape that normally would be overlapping with her other leg; however, it has an SVG clipPath applied to it to constrain the visibility to within the intended boundaries. On the neck and chest area also can see shadows that were masked to make them semi-transparent for smooth blending. [16]

Adobe Illustrator

Adobe Illustrator CS2 soft ware gives a powerful new creative options, easier access to tools, and the versatility to quickly produce flexible graphics for print, video, the Web, and mobile devices. Features that includes in Illustrator are Live Trace which make quickly and accurately convert photos, scans, or other bitmap images to editable and scalable vector paths. Live Paint is used to apply color to any area of user artwork and use overlapping paths to create new shapes with the Live Paint tool, which intuitively colors artwork and automatically detects and corrects gaps. Colorized grayscale is used to assign a spot color to a linked, embedded, or opened grayscale image, or even apply a spot color to a drop shadow and be confident that artwork will separate properly when printing. Gradient Mesh is to simulate airbrush and watercolor effects by adding multiple colors and precisely controlling the transitions between colors. Symbolism tools can be used to enhance user artwork with additional design elements by spraying, scaling, colorizing, and adjusting the transparency of repeated symbols while keeping file sizes small. With a flexible brushes user can add flair to paths with customized brushes. User can also choose from calligraphic, scatter, art, and pattern brushes. User can apply live effects, including the hand-drawn Scribble Effect, without affecting the underlying artwork, so user can edit at any time without starting over. [18]
Below are some of the examples using the Microsoft Illustrator:

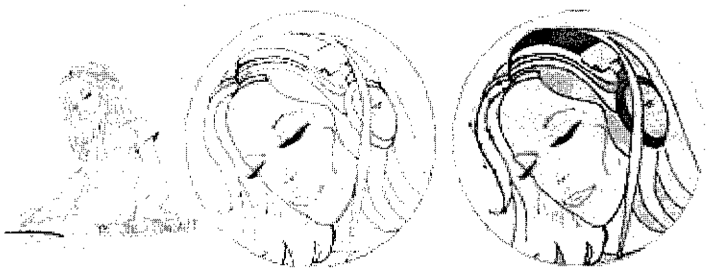


Figure 2.4 (c): Example of using Adobe Illustrator

Live Trace: Quickly and accurately convert photos, scans, or other bitmap images to editable and scalable vector paths with the Live Trace feature.

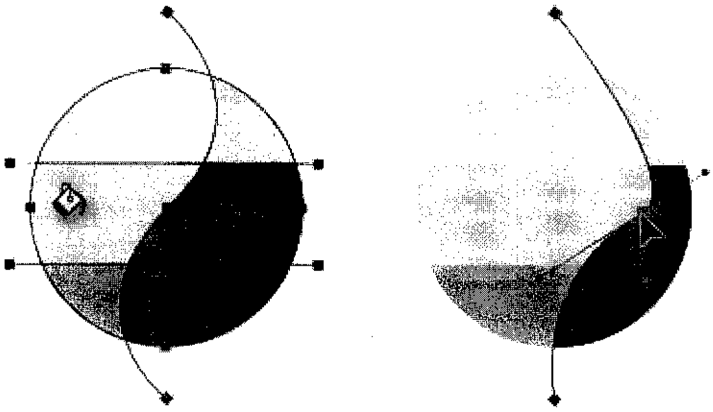


Figure 2.4 (d): Example of using Adobe Illustrator

Live Paint: Apply color to any area of your artwork and use overlapping paths to create new shapes with the Live Paint feature, which intuitively colors artwork and automatically detects and corrects gaps.

Microsoft Paint

Microsoft Paint is an inexpensive painting program that can be used to teach students the basics of painting software. Many sophisticated graphics software applications (such as Paint Shop Pro or Photoshop) use the same basic principles that can be learned using MS Paint. Using any painting program involves learning how to use the menu bar and the tool bar. MS Paint also includes a color palette that allows the user to choose both foreground and background colors with which to work. MS Paint also includes a great Help system that will have user mastering the virtual brush-strokes in minutes. This compendium of information was produced from screenshots of many of the MS Paint Help Files. [15]. The program opens and saves files as Windows bitmap (24-bit, 256 color, 16 color, and monochrome, all with the .bmp extension), JPEG, GIF, PNG, and TIFF. Older versions cannot open or edit PNG files, and can only open GIF, JPEG, and TIFF files with a graphics filter for the specific file type. The program comes with the following options in its Tool Box: Free-Form Select, Select, Eraser/Color Eraser, Fill With Color, Pick Color, Magnifier, Pencil, Brush, Airbrush, Text, Line, Curve, Rectangle, Polygon, Ellipse, and Rounded Rectangle. MS Paint does not have the ability to automatically create color gradients. [17] Below are some screenshots of using Microsoft Paint:

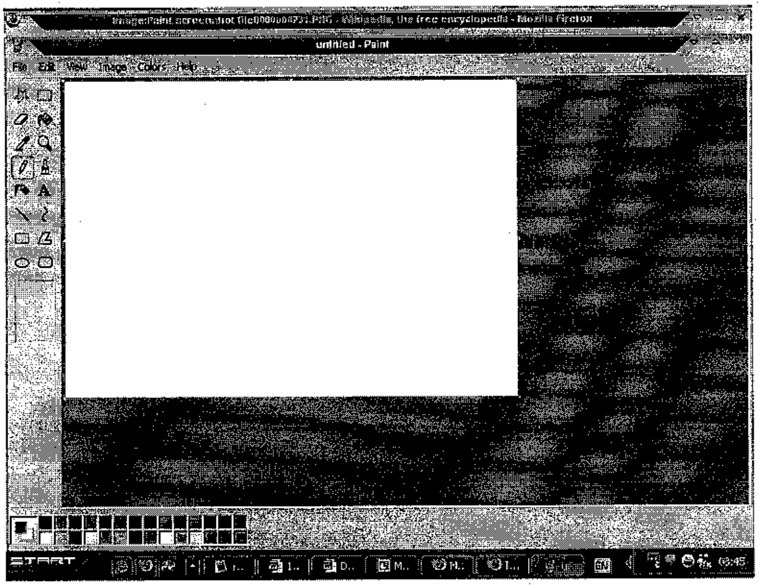
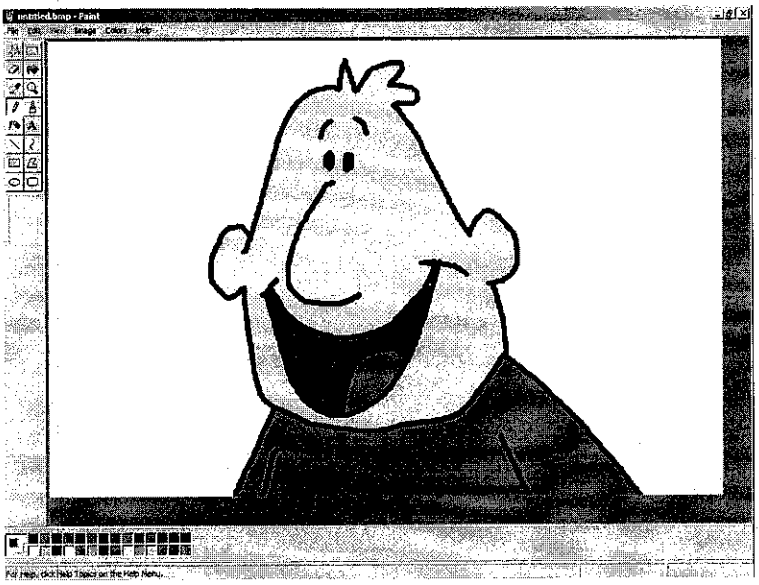


Figure 2.4 (e): The Microsoft Pain main toolbars allowing selection of tools and colors



For Help, click Help Topics on the Help Menu.

Figure 2.4(f): Screenshot of Microsoft Paint

Adobe Photoshop

Adobe Photoshop CS2 software, the professional image-editing standard and leader of the Photoshop digital imaging line, delivers more of what user craves. Groundbreaking creative tools help user achieve extraordinary results. Unprecedented adaptability lets user custom-fit Photoshop to the way user work. Using Photoshop, user can saves time because of Photoshop include a multitasking which take advantage of compact floating mode in Adobe Bridge to process images while simultaneously working in Photoshop CS2 or other CS2 applications. Photoshop also put a viewing option where can be easily browse user Adobe Creative Suite and other graphics files in Adobe Bridge and also present user images with features like Slideshow and Filmstrip mode and also scale thumbnails to any size using a slider; and view and edit metadata. This also include the automated batch processing which automatically process batches of images in Adobe Bridge to rename, convert format, adjust exposure, create a Web gallery or Adobe PDF Presentation, and more.[18]

CHAPTER 3: PROJECT PLANNING

There are five major stages involved in designing the 2D drawing tools. The stages that need to be undergone in order to complete the project are as followed:

- Requirement analysis
- Tasks and Domain Analysis
- Design of the environment and interaction
- Design of user support and navigation analysis
- Evaluation

3.1 Requirement Analysis

The requirement analysis that been made for this project has identified that things that required to understand are 2D L-System, 2D drawing tools and Open GL User Interface. The main objective of the project is to provide a drawing tool that integrates 2D L-System. The features that should be included in the project are mouse function, keyboard navigation and background that can be chooses by user. The function of the project will require creating a drawing tool that can be used easily and user-friendly. The method of the development is totally on the Open GL and C++ language.

3.2 Tasks and domains analysis

The tasks that have been done throughout the project development are research and development. A lot of readings and surfing need to be done in order to fulfill the requirement needed for the project. The studies have help to give better understanding on the how to do the project. Throughout the analysis, a few methods have been analyzed and some of the methods can be implementing in the project. The projects adapts the usage of the LParser as the engine for the creating the 2D L-System. In this project, it allows user to navigate within the application whereby they can move the objects and using the mouse and keyboard function. Whereas, when user use the mouse click function at the tree, there will be a flower as the output of the clicking. Thorough analysis need to be done in doing the mouse function.

3.3 Design of the environment and interaction

This stage focuses on the development of the project itself. Earlier I have studied a few materials that have significant information which can be applied to the application. In this design phase, I will adapt and integrate all of the knowledge that I have gained in order to come out with the virtual environment.

The tasks that need to be executed in this stage are as followed:

- Create background for the environment

  For this task, I have chosen the size of the window and also set the background colour. For the environment, user can choose what to be put in the environment. For this project, I included the sun, mountain and house models.

- Coding development

  As for the coding, I used the Open GL User Interface to create the interface. The tree has created using the LParser and the main thing that I did is to link the source code of the L-System into the interface. After the link function has success, the next thing that has been done is putting on the keyboard function. User can move the tree to where they want the tree to be. Once it is success, I moved to developing the background for the environment which is creating the mountain, house and sun. Both of the models are create using the Open GL polygons and I added lighting in the models to make it more interesting. The last part for the coding development is the mouse function. Mouse function will interact when user click at the tree. Once the mouse is click at the tree, there will be an output of flower.

- Equipment for interaction

  The project does not highlight on the usage of virtual equipment such as HMD, trackers or other equipment because basically the program will only used keyboard and mouse as the major interaction for the project. With the key from the keyboard, the user will be able to execute the functions such as moving forward and backward, zoom in and out and also to rotate around the environment.

Below is the diagram for design and user support navigation analysis:

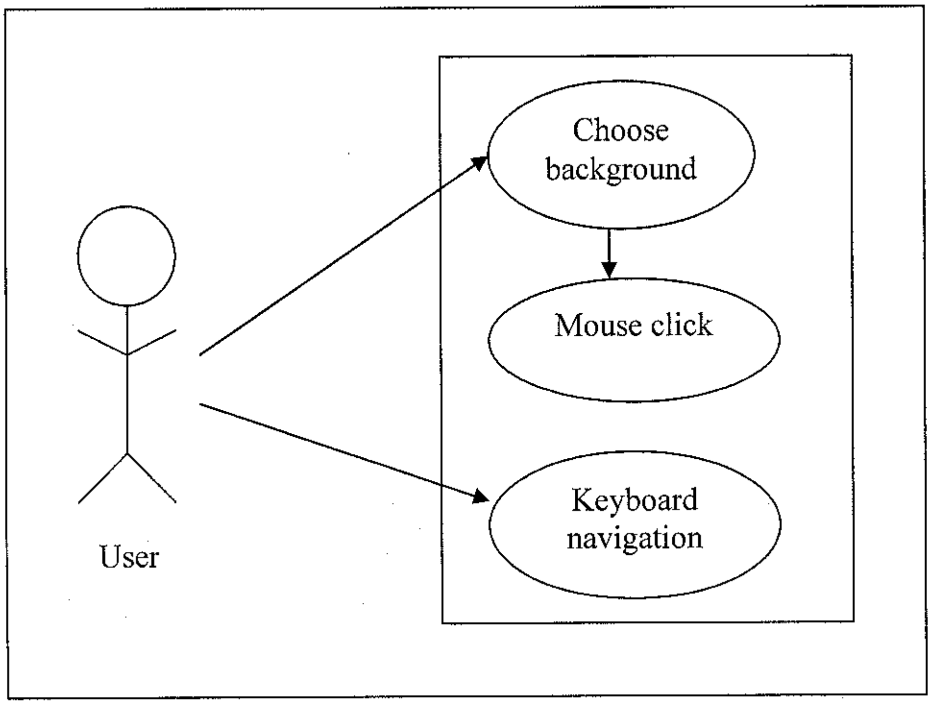


Figure 3.3.1: Use Case Diagram

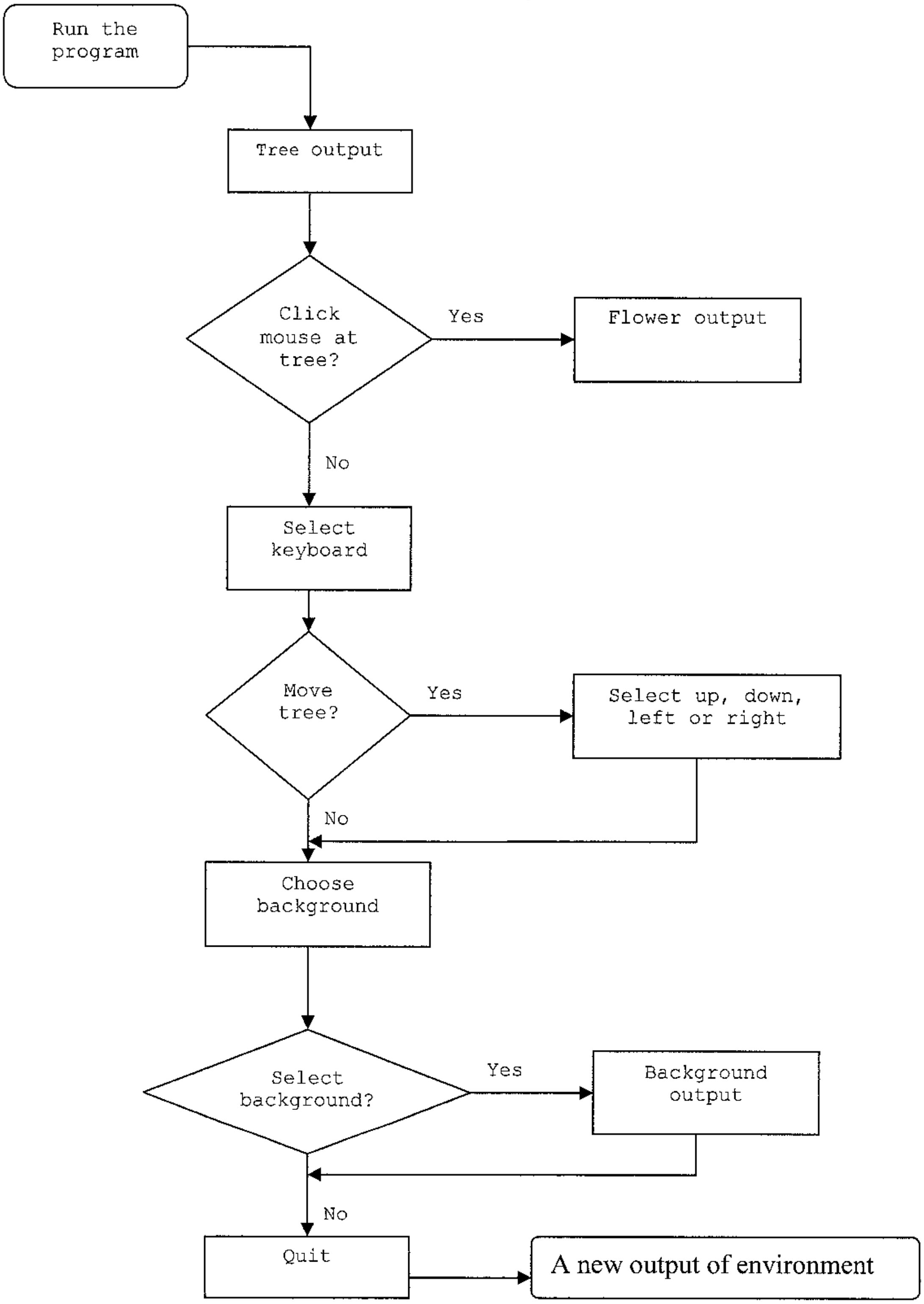


Figure 3.3.2: Flow Chart of 2D Drawing tool

3.4 Design of User Support and Navigation Analysis

The system will only provide the user with the capability of playing with the models. There will be some navigation in the system as the user will be able to select the objects in the environment. The system provides only the basic on mouse and keyboard function. Besides that, other additional function is to allow the user to zoom in and out of the environment and also to rotate around the scene. User can also play around with the placement of the models. They can rearrange the models using keyboard.

3.5 Evaluation

The evaluation stage is the same as testing phase in the software development life cycle. At this point of stage, the system will be tested in order to find any defects and to make sure that the system working properly according to the requirement specified. Ongoing test will be done throughout the development of the program in order to detect error or problems with the program as soon as possible.

3.6 Tools and Devices

3.6.1 Development Tools

Programming language that been used in developing this project is C++. This is because the author has more experienced in writing programs using this language. Apart from the author's experiences of writing programs using this language, the major reason of choosing C++ to write the program is because the author found that L-System is much easier to be used in C++.

Furthermore, C++ is compatible in using the Open GL as the platform. C++ is easier to understand and as mentioned before, I used the LParser as the engine of the L-System.

The LParser is written using the C++. The application that is used to edit the program in C++ is Microsoft Visual.Net.

3.6.2 Libraries

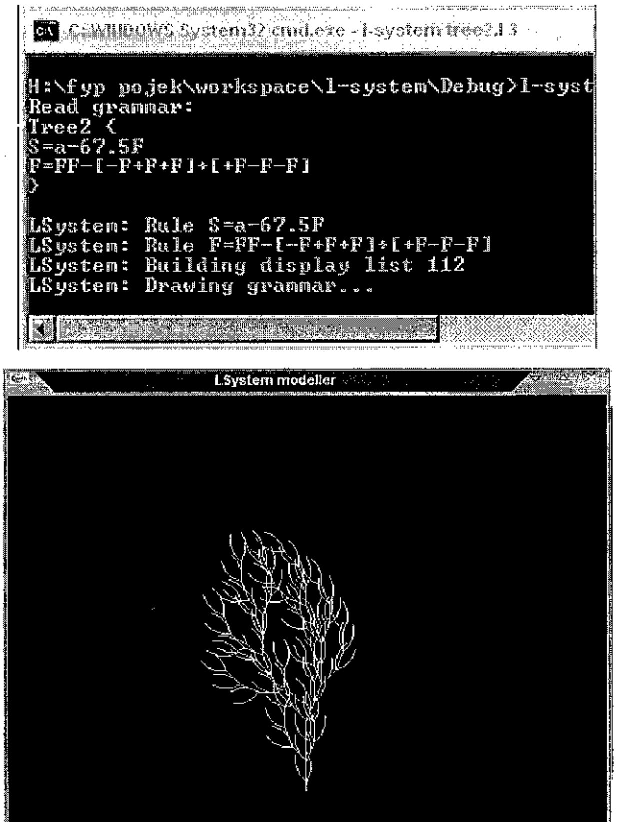
GLUI is a GLUT-based C++ user interface library which provides controls such as buttons, checkboxes, radio buttons, spinners, and listboxes to OpenGL applications. It is window-system independent, relying on GLUT to handle all system-dependent issues, such as window and mouse management. Some of the features of the GLUI User Interface Library include complete integration with GLUT toolkit, simple creation of a new user interface window with a single line of code, support for multiple user interface windows and standard user interface controls.

CHAPTER 4: RESULTS AND DISCUSSION

4.1 Project results

4.1.1 Results from the LParser

Figure 4.4.1 shows the result using the LParser engine. Using the LParser engine, user needs to use command prompt to run the program. It will take such a long time just to get the output. User will need to find where the directory that L-System been saved in before debugging the result. From this problem, it gave the idea of integrating it into a user interface and adding the functions of drawing tools that can make the project more interesting.



```
H:\fyp pojek\workspace\l-system\Debug>l-syst
Read grammar:
Tree2 {
S=a-67.5F
F=FF-[-F+F+F]+[+F-F-F]
}

LSystem: Rule S=a-67.5F
LSystem: Rule F=FF-[-F+F+F]+[+F-F-F]
LSystem: Building display list 112
LSystem: Drawing grammar...
```

Figure 4.1.1: L-System using the LParser

4.1.2 Results from the 2D drawing tools

Figure 4.1.2(a) shows the output of the L-System in GLUI. Beside the interface is the function interface where user can choose the background models. By integrating the 2D L-System into the user interface and some functions, user can understand the basic things about L-System. The figure below shows, once user run the project, a tree will automatic appears at the screen. This is because; the project is basically the integration of L-System into a user interface. Throughout the project, the author has faced a problem in integrating the tree into a user interface. This is because, since it's been used the LParser coding, there are some changes that need to be done due to integration to user interface. At some point, the author cannot find the problem why the trees do not appear in the screen. After some changes at the coding where the author disable the function lighting, then the trees appears.

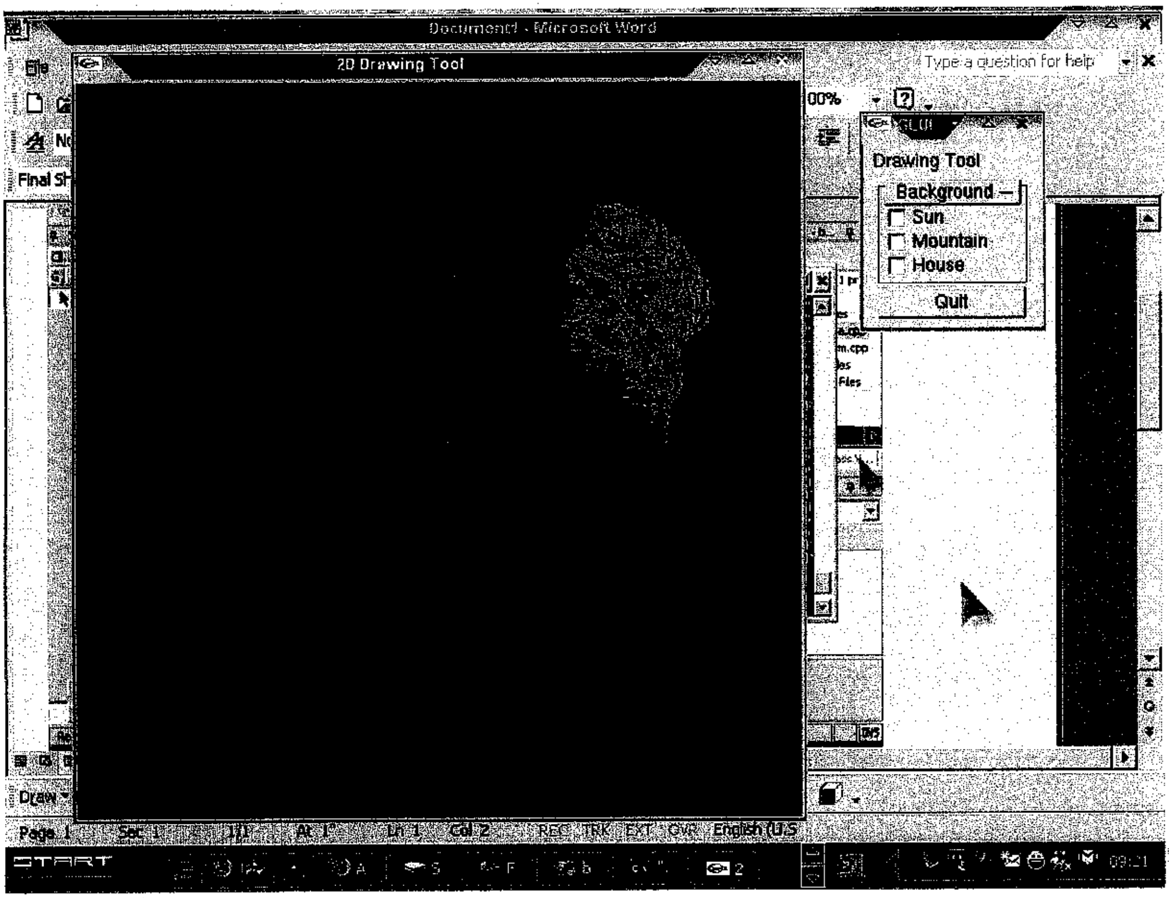


Figure 4.1.2 (a): L-System tree in GLUI

Figure 4.1.2 (b) shows that by using the keyboard function, user can rearrange the coordinate of the objects. In this project, user can choose whether to use keyboard or mouse as the navigation tool. In the figure below, user use keyboard to move the tree to the left. User can also moved it up, down or to the right. In the text area, user can write a text that they want to be put in the interface.

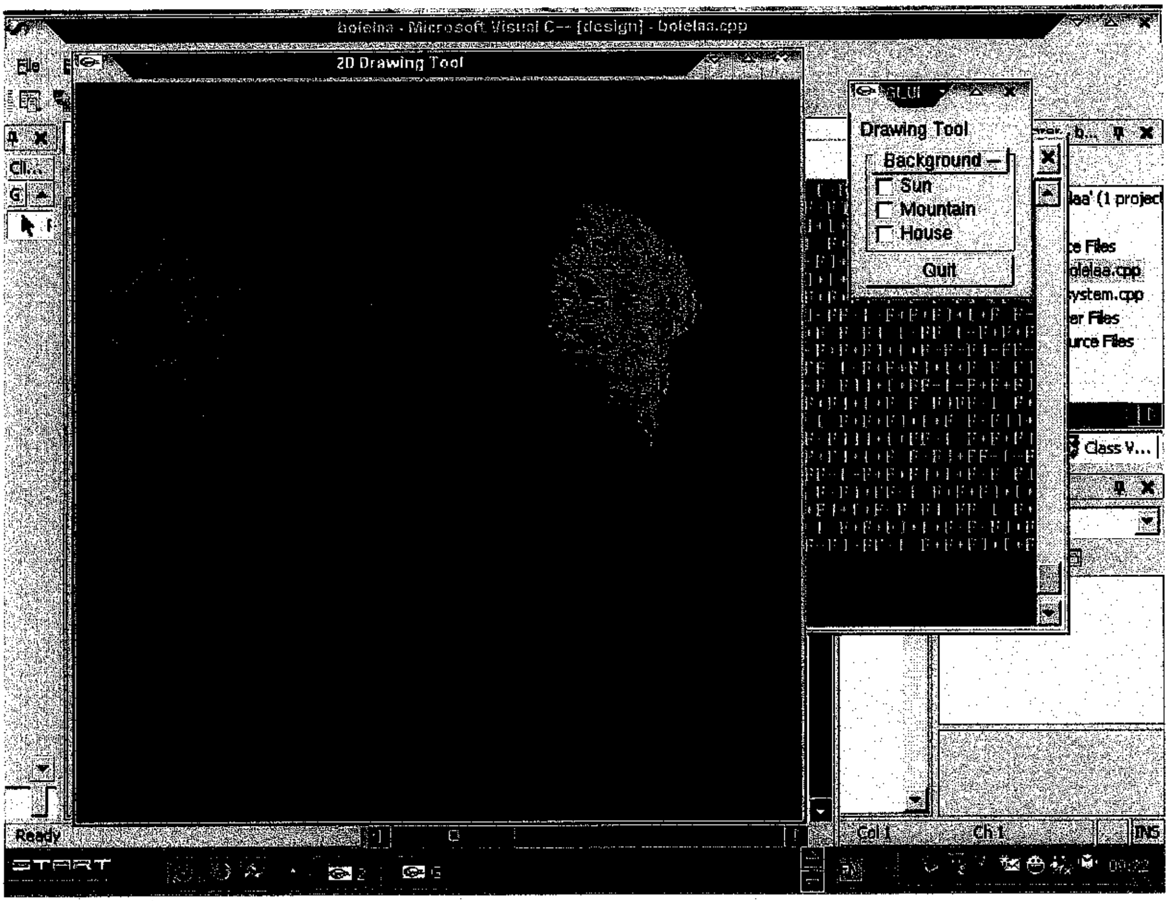
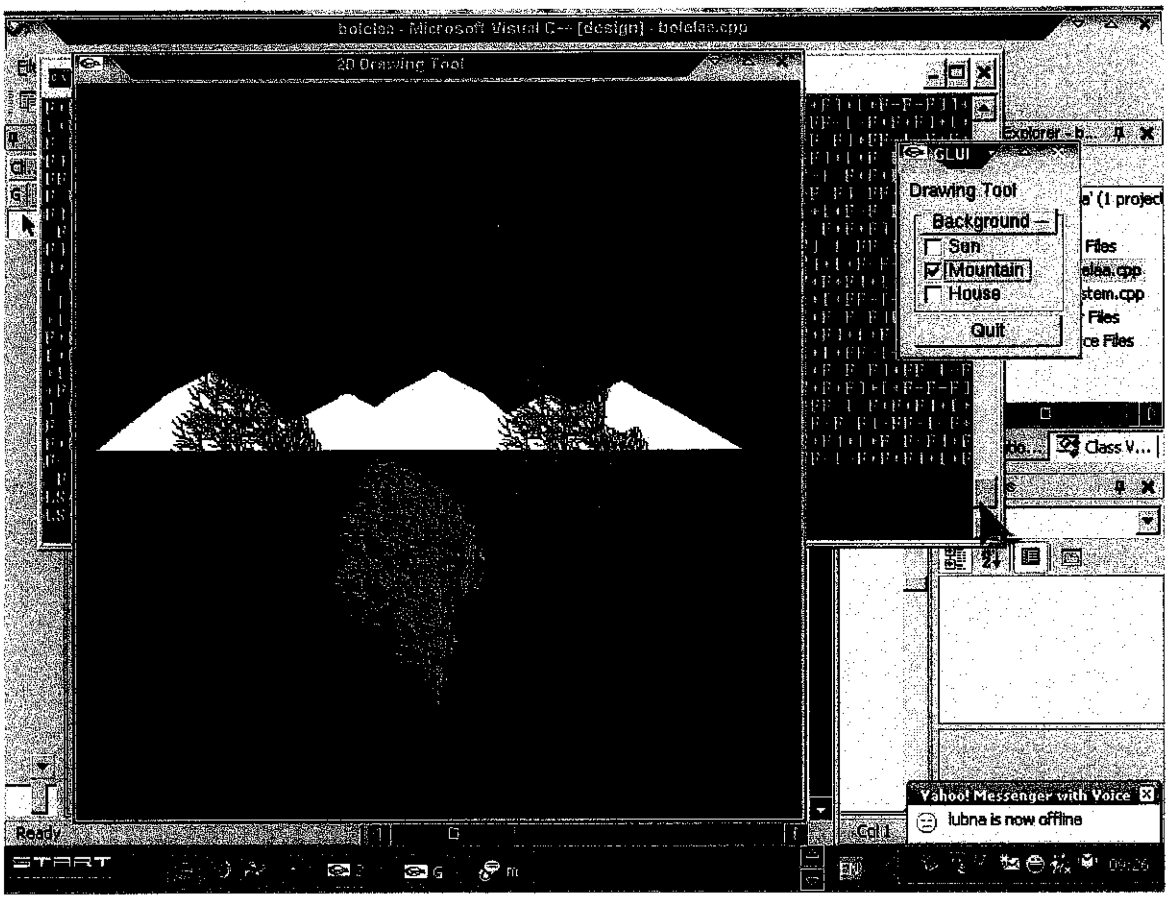


Figure 4.1.2 (b): Tree that been moved

Figure 4.1.2(c) shows that when user clicks at the mountain choice, the mountain will appear at the screen. The function goes at other choices also where when user clicks at house and sun, the models will appear and user can play and navigate the models using the keyboard or mouse. In the figure below, it shows the mountain output as user clicks at the mountain. As can be seen, the mountain is created using the 2D polygon.


boleiaa - Microsoft Visual C++ [design] - boleiaa.cpp
2D Drawing Tool
GLUI
Quit
Ready
START
Yahoo! Messenger with Voice
lubna is now offline

Drawing Tool

Background

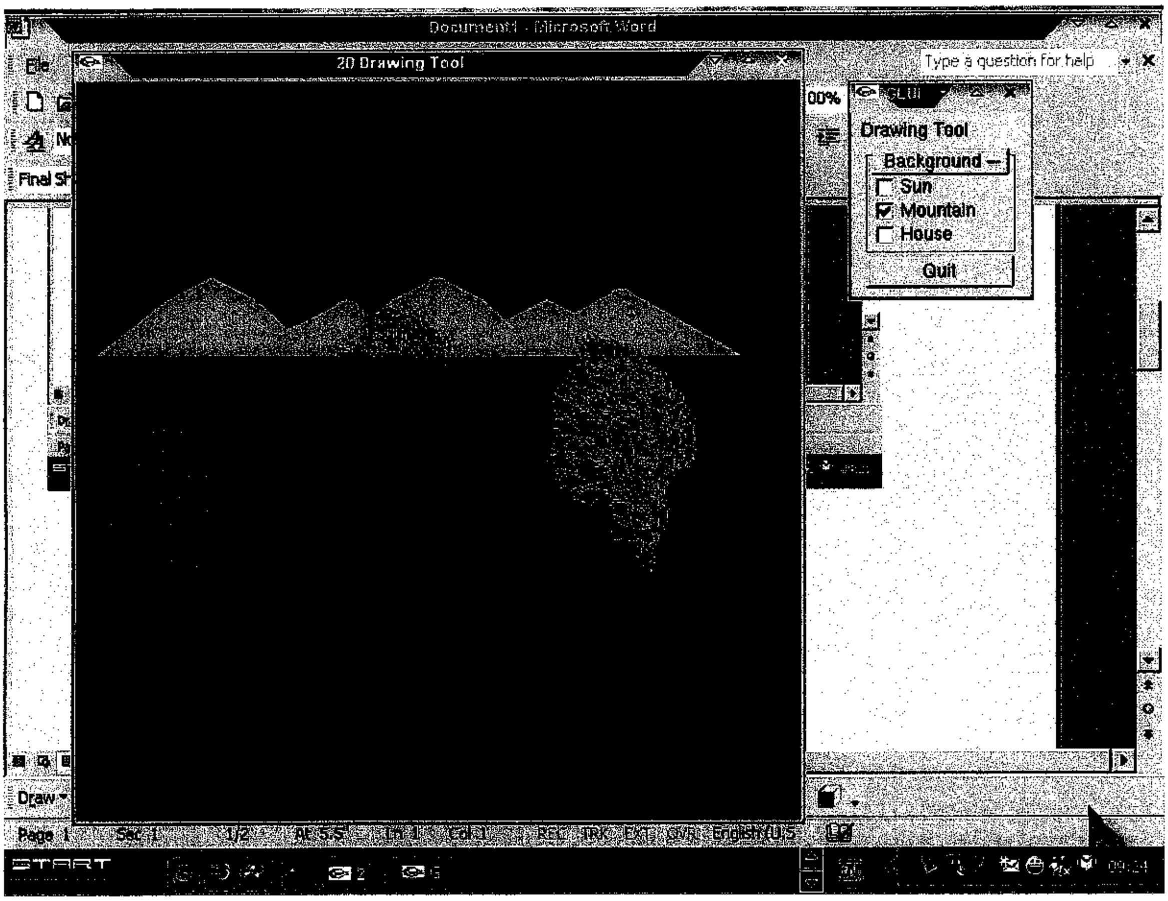
☐ Sun

☑ Mountain

☐ House

Figure 4.1.2 (c): User click at Mountain

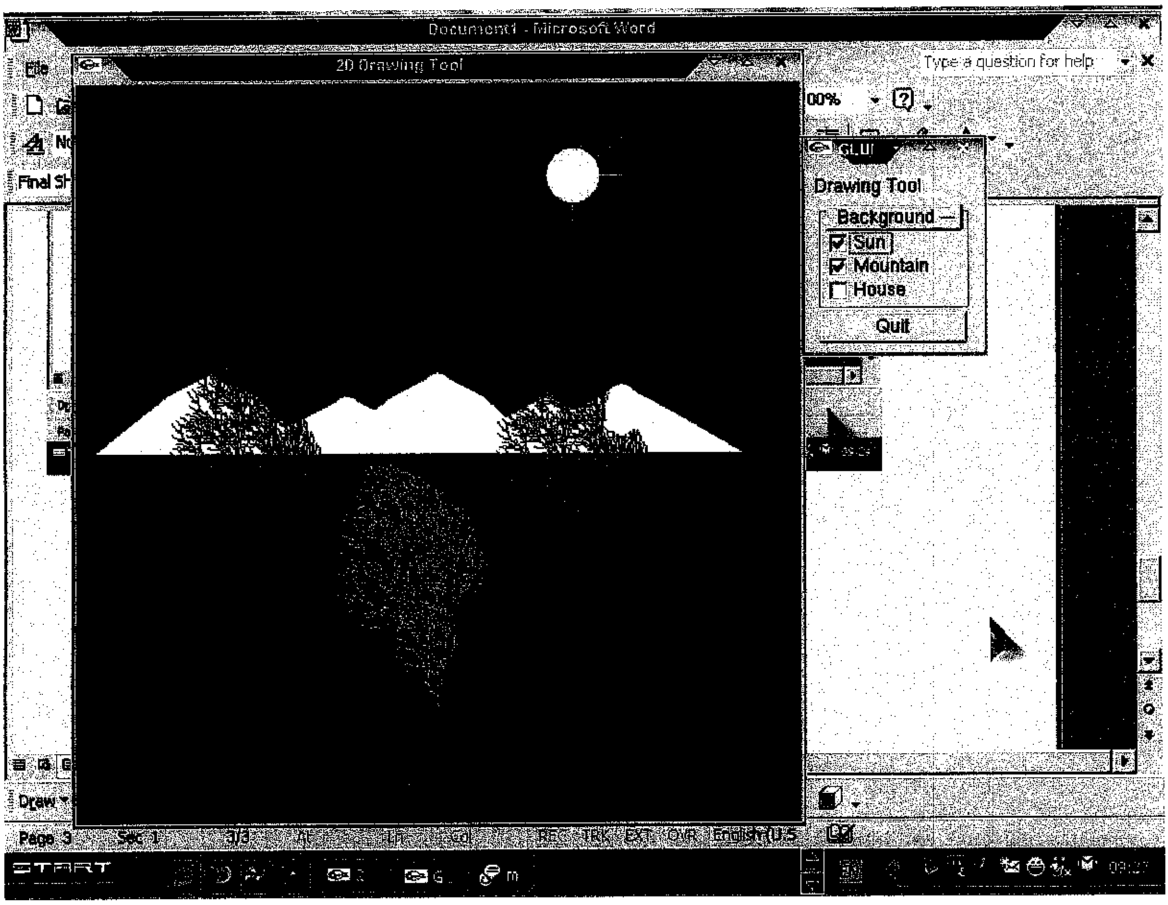
Figure 4.12 (d) shows that when user clicks at the mountain using mouse, user can select colors for the mountain. There are 6 choices of color that user can choose of. User can also moving up the mountain by pressing "U" at the keyboard, "D" for moving down, "L" for moving to the left and "R" for moving to the right.



☐ Sun
☑ Mountain
☐ House

Figure 4.12 (d): User click at the mountain to select color

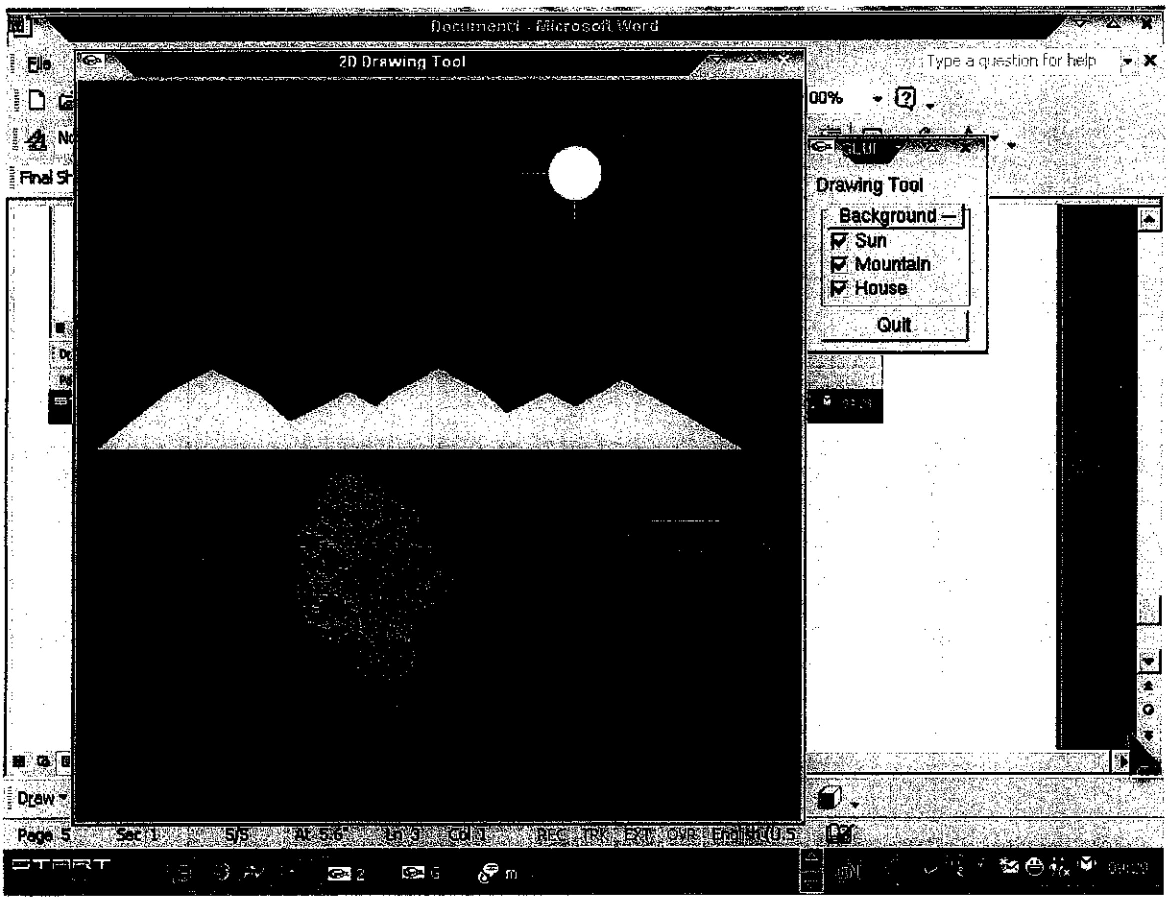
Figure 4.12(e) shows that when user clicks at "Sun", the output for of sun will appear at the screen. The function is the same like to the mountain function where user can also move the sun. In order to select which background that needs to be move, user can press "b" at keyboard. This will allow user to select which background that need to be moved.



☑ Sun
☑ Mountain
□ House

Figure 4.12(e): User click at mountain and sun

Figure 4.12 (f) shows that when user clicks at the backgrounds, the backgrounds appears in the coordinate that been specific but if user wants to change it, user can press “b” to select the background that want to be moved.



☑ Sun
☑ Mountain
☑ House

Figure 4.12 (f): User click at the backgrounds

CHAPTER 5: CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

Many methods have been developed just to ease the user in drawing and arrange models. The most popular 2D drawing tools are Microsoft Paint and Adobe Photoshop. For any Windows user, Microsoft Paint has installed directly into the system. User won't need to download or buy any outside software because it is built in with the operating system. Adobe Photoshop is much complicated to use but because of the complicatedness the features is more interesting to learn and use. Adobe Photoshop gives more function compared to Microsoft Paint. Some features that are interesting are the intelligent image editing which is Revolutionary Vanishing Point. When user used it, user can achieve amazing results in a fraction of the time with the groundbreaking Vanishing Point, which lets user clone, paint, and paste elements that automatically match the perspective of the surrounding image area. Other specialty of Photoshop is spot healing brush where it is effortlessly retouch photos including 16-bit images in a single click with the advanced power of the new Photoshop CS2 Spot Healing Brush. In drawing tools, function of arranging model is usually have in it.

Among all the drawing tools and arrangement tools, none of them has integrated the L-System. L-System is a suitable system to create a nature environment. This is because it is systematically extended to provide greater flexibility and more accurate plant models. By integrating the L-System in this drawing tools project, people can understand the basic things about L-System. Using the 2D L-System that been integrated into user interface, user would not have to draw the tree. Drawing a tree at some point is harder than drawing a human face. This is because tree branch has different recursive ways. There are a lot of method that been used in tree modeling such as hand modeled plants, directional billboards and Billboard" texture maps. L-System is used rather than other tree modeling is because the resulting images are fractal-like expanding on the concept that simple rules when recursively applied can create complex structures. Besides, L-

System is well-known as one of the best method in creating plant environment. Basically the objectives of this project is to create user interface that integrate 2D L-System with the functions of arranging models, to understand on 2D L-System, to understand on Open GL User Interface (GLUI) and to show the usefulness of the 2D L-System.

Under the project planning, the author can conclude that the five major stages have been fulfilled. The stages are requirement analysis, tasks and domains analysis, design of the environment and interaction, design of user support and navigation analysis and evaluation. Under the design of the environment and interaction, the author has put use-case diagram and a flow chart to show the flow of how the project runs. Under the topic of project planning, the author has put tools and devices that will be used in the project. For the development tools, the author used C++ language ad Open GL as the platform. For the libraries, the author used Open GL User interface (GLUI). GLUI is a GLUT-based C++ user interface library which provides controls such as button, checkboxes and list boxes to Open GL applications.

Other models that been developed for this project are sun, house and mountain. All the models are developed using the polygons. The polygons have been made using basic polygon. Generally, it can be concluded the objectives of this project is to integrate the 2D L-System into arrangement tools.

5.2 Recommendations

The 2D graphical arrangement tool is an interesting project but due to limitation of time to accomplish the project, certain application did not manage to be put into the project. These recommendations made for future improvement of this project if this project is to be continued by other students.

1. Instead of limiting it to the 2D application, it is more interesting if the project can be integrated to 3D application. This is because by integrating to the 3D, the function can be much wider and interesting.
2. The background of the project also can be added some more such as cloud, birds or other nature environment to make it more fun and interesting to be used. If more of L-System tree and flowers can be implement in one scene also can made the project more interesting because we can see the different of each structure for each tree.
3. Another recommendation that can be made is to be add a function that user can create their own tree using the grammar of L-System. The project can be more useful if user can directly create their own tree by incorporating the L-System commands.

REFERENCES

1. Robert M. Dickau - 2D L-Systems, http://mathforum.org/advanced/robertd/lsys2d.html
2. Director Online Article Polygon Gardens, http://director-online.com/listArticles
3. Statistical data directed evolution of L-system models for botanical trees
4. L-System Plant Geometry Generator , http://www.nbb.cornell.edu/neurobio/land/OldStudentProjects/cs490-94to95/hwchen/
5. Kenrick J. Mock, Journal of Wildwood: The Evolution of L-System Plants for Virtual Environments
6. J.B. Evers, J. Vos, C. Fournier, B. Andrieu, M. Chelle, P.C. Struik, Journal of A 3D approach for modelling tillering in wheat (Triticum aestivum L.)
7. http://www.vterrain.orgPlantsModelling
8. Joanna L. Power, A.J. Bernheim Brush, Przemyslaw Prusinkiewicz, David H. Salesin, : Journal of Interactive Arrangement of Botanical L-SystemModels
9. Jon McCormack: Journal of Interactive Evolution of L-System Grammars for Computer Graphics Modeling
10. GLUI user manual by Paul Rademacher
11. An Overview of the C++ Programming Language by Bjarne Stroustrup, AT&T Laboratories Florham Park, NJ07932-0971, USA
12. http://www.research.att.com/~bs/crc.pdf
13. http://www.google.com.my/search?hl=en&lr=&defl=en&q=define:drawing
14. http://www.google.com.my/search?hl=en&lr=&defl=en&q=define:L-system
15. http://www.fayette.k12.il.us/99/paint/paint.htm

16. http://inkscape.org

17. www.wikipedia.org

18. http://www.adobe.com/

19. http://www.devpapers.com/article/233

20. http://www.acm.org/crossroads/xrds8-2/plantsim.html

21. http://www.fotoclipart.com/pictures/pics

22. http://forums.narutofan.com/showthread.php?t=57582

23. http://www.absolutearts.com/artsnews/2004/09/15/32366.html