Unit 2 Graphical User Interface – I

Structure:

2.1 Introduction

In the previous unit you learnt importance of user interface, types of user interface and benefit of good design.In this unit you are going to learn about GUI. A ***GUI is a Graphical User Interface***, usually part of a computer's operating system which is characterized by WIMPs and WYSIWYG. ***WIMPs are Windows, Icons, Menus, and Pointing devices*** (such as a mouse or a trackball). ***WYSIWYG or What You See is What You Get,*** refers to the ability to print out exactly what you see on the screen, this made effective desktop publishing possible.

Objectives

After studying this unit, you should be able to:

- explain history of Graphical User Interface (GUI)
- discuss the popularity of graphics
- list functions of GUI
- describe the concept of direct Manipulation
- explain human factors of a Graphical User Interface.

2.2 History of Graphical User Interface (GUI)

In 1975 the researchers at Xerox PARC (Palo Alto Research Center) moved into their permanent headquarters at 3333 Coyote Hill Road near Stanford University in Palo Alto, California. Jacob Goldman had founded PARC just five years previous, and already at this early date the research team had developed many of the ideas that shaped the future of computing. The Palo Alto Research Center's mission as directed by Xerox management was to create the office of the future. To that end they created many of the technologies we take for granted in the modern office, such as networked personal computers, with E-Mail, word processing, and laser printing, but most significant innovation at PARC was the graphical user interface (GUI), the desktop metaphor that is so prevalent in modern operating systems today. The GUI would make computer graphics an everyday part of the working environment. No longer would the display be simply lines of code and commands, it would be graphical with true representation of typefaces and images.

The bitmapped GUI display would help promote the concept of WYSIWYG (what you see is what you get) allowing people to laser print exactly what they saw on the screen. In workstations of the past, graphics and commands where split between three different screens, a vector device for line graphics, a text display for entering commands, and a video / raster graphics screen coupled with a frame buffer to display the final rendered result. Xerox PARC combined these separate technologies into one raster graphics screen along with an easier way of issuing commands: the popup menus, icons, and desktop metaphor of the graphical user interface. As with many other areas of computer graphics researchers from Utah were going to play a key role in the development of the GUI.

The idea for a graphical user interface (GUI) was first developed by Alan Kay from the University of Utah who went to work at Xerox PARC on the Alto project in 1970. The first GUIs tried at Xerox PARC were very slow to work withand depended too much on the processor to re-draw each bit when moving overlapping windows around. In 1974 a PARC research named Dan Inglis invented a procedure for the movement of whole blocks of bits on the screen called "Bit Blit". This display algorithm allowed overlapping windows to be quickly shuffled around the screen without overtaxing the processor.

The Life of GUI

The Xerox Star was the first commercial personal computer to use the now common desktop metaphor. An early publication (David Smith developer of the Star interface, 1982) about the Star said "Every user's initial view of the

Star is the Desktop, which resembles the tope of an office desk, together with surrounding furniture and equipment. It represents a working environment, where projects and accessible resources reside. On the screen are displayed pictures of familiar office objects, such as documents, folders, file drawers, in-baskets, and out-baskets. These objects are displayed as small pictures or icons". With the Star began 'messy desk' metaphor seen in so many GUIs today.

The GUI and Networking

PARC was initially divided into three units: the computer Science Lab (CSL), the Systems Science Lab (SSL), and the General Science Lab (GSL). The CSL run by Bob Taylor was most responsible for the development of the graphical user interface. Talyor had worked on the ARPAnet, (a distributed network of computers, the predecessor to the Internet) and brought the idea of networked computers to PARC.

By 1979 there were hundreds of Altos networked together with more traffic and 'nodes' than the entire ARPAnet. Xerox PARC even had the world's first computer virus called a 'tapeworm' because it would eat its way through the Ethernet and consume all available resources. Like many of PARC's innovations Networking was ahead of its time, and would not be widely available in personal computers for another decade. The first Macintosh had no network capabilities, and when asked about it, Steve Jobs, threw a floppy disc at a journalist, saying "Here's my network". Alto had the ability to show other computers on the network as icons on its graphical desktop. Much later Macintosh acquired this ability. With the birth of the Internet and the World Wide Web, computers could network to others across the world, but this was done with a separate application that was at first non-graphic (telnet, text-based FTP software). Later applications for accessing the internet became more GUI-like but were not part of the operating system. With the introduction of the first graphical web browser, Mosaic, in 1993 accessing the internet began to look more like a GUI.

Netscape further refined the GUIness of the web browser application. With the release of Windows 95 and NT, Microsoft attempted to blur the distinction of the web browser application and operating system by bundling 95 with Internet Explorer. The windows and interface of 95 could be set so that browsing the local computer looked almost the same as browsing the

Internet. Ether this is a good idea remains to be seen. Microsoft got into legal trouble because the bundling strategy was seen as an attempt to block competition from Netscape and other browsers. Apple now allows users to create an iDisk (a storage space on the Apple webserver) with OS 9 that appears as regular disk icon in the GUI desktop, and the Sherlock (find file) application allows users to search the internet as they would a local disk.

Self Assessment Questions

1. The idea for a graphical user interface (GUI) was first developed by ____________________.
2. __________________ was most responsible for the development of the graphical user interface.

2.3 Popularity of Graphics

A computer user interface is the place where machine and human being communicate with each other. It is how we tell computers what we want and how they present to us the information we request. Today, the most common user interfaces involve a keyboard, viewscreen, and, more and more often, a mouse. A graphical user interface is a computer-user interface that uses graphical screen images as well as typed text, with icons on the screen replacing many of the functions of the keyboard. For example, in a typed text interface, the command to move data from one file to another is typed into the computer as a line of code meaning, Send this letter from file A to file B." In GUI, a user might send this command by placing the cursor on the letter (represented by the image of an envelope), and moving it with a mouse across the screen from file A (represented as a rectangle), to file B (represented as another rectangle). Click the mouse, and the transfer is complete.

Many sighted people find GUI easier to use, because they don't have to remember or to look up special commands for each program function. Less time is spent figuring out how to get the computer to do what you want it to do.GUI can be used by people who are blind and visually impaired, provided they have a reliable screen reader to translate what's on the screen into braille or synthesized speech. The development of screen readers for the older, DOS-based systems is quite advanced, and a number of private companies presently compete for the screen reader market.

The GUI also helped to develop a whole new industry for publishers and designers within Desktop Publishing – revolutionizing (and partly wiping out) the print and typesetting industry. Opposite the fast developing hardware market, the GUI until today did not evolve or change very much, considering what would be possible. Its paradigms like the desktop metaphor, drop-down menus, overlapping windows etc. still are the same. Yet Apple's Operating System Ten, which will be available on the market in approximately half a year's time from now, will be sold with a whole new interface, and called "Aqua".

John Warnock helped to develop Interpress and other printingand page description systems at PARC which allowed the Altoto become the first WYSIWYG computer when coupled withXerox's laser printer. Later Warnock would found AdobeSystems which helped bring about the desktop publishingrevolution of the late 1980s along with Apple computer. Thecombination of the Macintosh, the LaserWriter, and Adobe'spage description software would forever change the world ofpublishing, typesetting, and graphic design. Larry Tesler at PARC, who would later be part of the AppleLISA development team, begin work on Gypsy, the world's firstuser friendly word processing application using pop-up menusand icons in 1974.

Gypsy was later to become the basis forMicrosoft Word when it's co-creator Tim Mott and others atPARC went to work at Microsoft. Gypsy was essentially theworld's first desktop publishing software with advanced features such as drawing and editing graphics within the same application as the word processor. These features have only recently begun to emerge in software packages such as Adobe in Design, and QuarkXPress 4.0.

The IBM PC continued to be more popular with businessesthan the Mac even though it didn't have a GUI. Enter Bill Gatesand Microsoft, which began developing useful applications forth Mac which helped increase sales. Microsoft's early partnership with Apple allowed them access to the Mac OSwhich led to the development of their own GUI, Windows 1.0.Microsoft saw the Mac OS as a threat to their non-graphical operating systems for IBM PCs, MS DOS and knew that they had to develop a GUI to compete.

Windows appeared to copy many of the samemetaphors and icons as the Mac GUI with just the nameschanged, for example the Trash Can in Mac, became theRecycle Bin in Windows (and later the Dumpster in X-Windowsfor SGI Workstations).Windows became more popular notbecause it was better than the Mac OS, but because it was more open, and could run on millions of IBM PCs and Mac Clones. Apple never licensed the Mac OS to clonemakers except for a brief period in the mid-90s. This caused Apple to lose market share, but kept the quality of their product more consistent than with IBM clones. Windows had flaws because it was built on top of the non-GUI MS-DOS, and it would behave differently on each type of PC clone. In 1986 Steve Jobs left Apple after disagreements with the board of directors and founded a new company called NeXT which would develop NeXTStep, a GUI for its UNIX based workstations in 1988. This becomes the first GUI to simulate a three-dimensional screen. Later when Steve Jobs returned to Apple, NeXTStep would merge with the Mac OS to create Mac OS X.

Around the same time as NeXTStep, in the late 80s, other UNIX workstation manufactures wanted a piece of the GUI action. In 1987 X Windows System for UNIX workstations became widely available. Around 1989 several Unix-based GUIs were introduced. These included Open Look, by AT&T and Sun Microsystems, and Motif for the Open Software Foundation by DEC and Hewlett-Packard. Motif’s appearance is based on IBM’s Presentation Manager a rival GUI to MS Windows.

2.4 GUI Functions

The GUI is based on the principle that pointing in menus to a command you want to computer to do is easier than having to remember hundreds of key words like in command line operating systems such as MS DOS or UNIX. The GUI is based on the idea that pointing to something is the most basic human gesture, and the mouse is easier to use than a keyboard. In a graphic user interface a user points at windows, icons, and menus by means of a mouse on a metaphorical ‘desktop’ environment which relates to the user’s known physical office environment. The graphical desktop is a metaphor of an office desk, which files and folders on top of it, making it easier for new users to visualize how the computer works.

The graphical user interface should cover the following main functions:

- To allow access of users to the main system functions
- To facilitate the efficient reception and memorizing of test items, in an accessible form, according to the set purposes.
- To ensure a simple and efficient communication between users and system – which is of the highest importance, as the target users are generally inexperienced computer users; for this reason, a complicate sequence of commands for accessing an usual function (like Help) may lead to a low user acceptance and possibly system errors
- To allow basic functions like printing or exiting, at any time
- To ensure the recording (monitoring) of user actions, which is extremely useful because
    - the user can quickly identify the parts of the content which were not read, especially as the content may be accessed non-sequentially
    - the system administrator (in case of an online system) may use this information for monitoring the system use according to access rights
- To allow the transmission of text, audio-video / multimedia learning items in a simple form
    - for online systems, the connection speeds are crucial for an appropriate usage, and complex / slow loading content may compromise a timely response
    - For offline systems, the used hardware may not offer high performance for playing multimedia content.

Self Assessment Questions

3. The GUI also helped to develop a whole new industry for publishers and designers (True/False)
4. In which year Unix-based GUIs were introduced?

    ____

2.5 Concept of Direct Manipulation

Direct manipulation is a human-computer interaction style that was defined by Ben Shneiderman.The term direct manipulation has the following properties:

1) Continuous representation of the object of interest.
2) Physical actions or labeled button presses instead of complex syntax.

3) Rapid incremental reversible operations whose impact on the object of interest is immediately visible.

Direct manipulation interfaces provide the following features.

- ***Visibility*** of the objects of interest.
- ***Replacement*** of complex command languages with actions to directly manipulate the visible objects (hence the name direct manipulation).
- ***Incremental action*** at the interface, with rapid feedback on all actions.
- ***Syntactic correctness*** of all actions, so that every user action is a legal operation.
- ***Reversibility*** of all actions, so that users are encouraged to explore the product without severe penalty.

The intention of direct manipulation is to allow a user to directly manipulate objects presented to them, using actions that correspond at least loosely to the physical world. Having real-world metaphors for objects and actions can make it easier for a user to learn and use an interface (some might say that the interface is more natural or intuitive), and rapid, incremental feedback allows a user to make fewer errors and complete tasks in less time, because they can see the results of an action before completing the action. An example of direct-manipulation is resizing a graphical shape, such as a rectangle, by dragging its corners or edges with a mouse.

Individuals in academia and Computer scientists doing research on future user interfaces often put as much or even more stress on tactile control and feedback or sonic control and feedback than on the visual feedback given by most GUIs. In these cases the term "graphical user interface" seems inadequate. As a result the term direct manipulation interface has been more widespread in these environments.

The term direct manipulation interface is the term often used in academic or corporate research circles for the GUI or Graphical user interface. It is both a more scientific term and a more descriptive term. The essence of a GUI is not that it is graphical, but that it permits a user to directly manipulate objects presented by the user interface. Blind or vision impaired users can employ GUIs which have been adapted for them by the use of a combination of tactile and sonic devices and software. In that case the word "graphic" seems patently absurd.

Computer scientists doing research on future user interfaces often put as much or even more stress on tactile control and feedback or sonic control and feedback than on the visual feedback given by most GUIs. In these cases the term "graphical user interface" seems inadequate.

Given the recent spread of touchscreens at the beginning of this millennium, and the presence of massive sonic feedback, and even tactile response (joystick feedback actuators) in more and more game platforms and games software, the acronym GUI might seem lacking, in these particular instances.

People will probably go on using the term GUI, regardless of its disconnection with a changing reality. The usefulness of the term "direct manipulation interface" lies in its capacity to remind practitioners (or anyone interested in discussing the user interface) that there is more to the GUI than meets the eye, and to serve as a pointer towards fundamental analysis of the practical essence of the GUI.

Direct manipulation interfaces seem remarkably powerful. Shneiderman has suggested that direct manipulation systems have the following virtues:

1) Novices can learn basic functionality quickly, usually through a demon
2) Experts can work extremely rapidly to carry out a wide range of tasks,
3) Knowledgeable intermittent users can retain operational concepts.
4) Error messages are rarely needed.
5) Users can see immediately if their actions are furthering their goals, and station by a more experienced user. Even defining new functions and features. If not, they can simply change the direction of their activity.

Earlier direct manipulation systems

The concept of direct manipulation actually preceded the first graphical system. The earliest full-screen text editors possessed similar characteristics. Screens of text resembling a piece of paper on one's desk could be created (extension of real world) and then reviewed in their entirety (continuous visibility).Editing or restructuring could be easily accomplished (through rapid incremental actions) and the results immediately seen. Actions could be reversed when necessary. It took the advent of graphical systems to crystallize the direct manipulation concept, however.In practice, direct manipulation of all screen objects and actions may not be feasible because of the following:

- The operation may be difficult to conceptualize in the graphical system.
- The graphics capability of the system may be limited.
- The amount of space available for placing manipulation controls in the window border may be limited.
- It may be difficult for people to learn and remember all the necessary operations and actions.

When this occurs, indirect manipulation is provided. Indirect manipulation substitutes words and text, such as pull-down or pop-up menus, for symbols, and substitutes typing for pointing. Most window systems are a combination of both direct and indirect manipulation. A menu may be accessed by pointing at a menu icon and then selecting it (direct manipulation). The menu itself, however, is a textual list of operations (indirect manipulation). When an operation is selected from the list, by pointing or typing, the system executes it as a command. Which style of interaction-direct manipulation, indirect manipulation, or a combination of both-is best, under what conditions and for whom, remains a question whose answer still eludes us?

Direct manipulation versus WIMP/GUI interfaces

Direct manipulation is closely associated with WIMP and GUI interfaces, as these almost always incorporate direct manipulation to at least some degree. However, direct manipulation should not be confused with these other terms, as it does not imply the use of windows or even graphical output. For example, direct manipulation concepts can be applied to interfaces for blind or vision impaired users, using a combination of tactile and sonic devices and software.

It is also possible to design a WIMP interface that intentionally does not make use of direct manipulation. For example, older versions of windowing interfaces (e.g. Windows 3.1) allowed users to reposition a window by dragging it with the mouse, but would not continually redraw the complete window at intermediate positions during the drag. Instead, for example, a rectangular outline of the window might be drawn during the drag, with the complete window contents being redrawn only once the user had released the mouse button. This was necessary on older computers that lacked the memory and/or CPU power to quickly redraw data behind a window that was being dragged.

Direct manipulation versus WIMP/GUI interfaces

Direct manipulation is closely associated with interfaces that use windows, icons, menus, and a pointing device (WIMP GUI) as these almost always incorporate direct manipulation to at least some degree. However, direct manipulation should not be confused with these other terms, as it does not imply the use of windows or even graphical output. For example, direct manipulation concepts can be applied to interfaces for blind or vision-impaired users, using a combination of tactile and sonic devices and software.

It is also possible to design a WIMP interface that intentionally does not make use of direct manipulation. For example, most versions of windowing interfaces (e.g. Microsoft Windows) allowed users to reposition a window by dragging it with the mouse, but would not continually redraw the complete window at intermediate positions during the drag. Instead, for example, a rectangular outline of the window might be drawn during the drag, with the complete window contents being redrawn only once the user had released the mouse button. This was necessary on older computers that lacked the memory and/or CPU power to quickly redraw data behind a window that was being dragged.

Direct manipulation in point of sale graphic interfaces

The View Touch graphic touchscreen POS (point of sale) GUI developed by Gene Mosher on the Atari ST computer and first installed in restaurants in 1986 is an early example of an application specific GUI that manifests all of the characteristics of direct manipulation. In 1995 the View Touch GUI was developed into an X Window System window manager, extending the usefulness of the direct manipulation interface to users equipped with no other equipment than networked displays relying on the X network display protocol. This application is a practical and useful example of the benefit of the direct manipulation interface. Users are freed from the requirement of making use of keyboards, mice and even local computers themselves while they are simultaneously empowered to work in collaborative fashion with each other in worldwide virtual workgroups by merely interacting with the framework of graphical symbols on the networked touchscreen.

Direct manipulation in computer graphics

Computer graphics are graphics created using computers and, more generally, the representation and manipulation of image data by a computer.

The development of computer graphics has made computers easier to interact with, and better for understanding and interpreting many types of data. Developments in computer graphics have had a profound impact on many types of media and have revolutionized animation, movies and the video game industry

Because of the difficulty of visualizing and manipulating various aspects of computer graphics, including geometry creation and editing, animation, layout of objects and cameras, light placement, and other effects, direct manipulation is an extremely important part of 3D computer graphics. There are standard direct manipulation widgets as well as many unique widgets that are developed either as a better solution to an old problem or as a solution for a new and/or unique problem. The widgets attempt to allow the user to modify an object in any possible direction while also providing easy guides or constraints to allow the user to easily modify an object in the most common directions, while also attempting to be as intuitive as to the function of the widget as possible. The three most ubiquitous transformation widgets are mostly standardized and are:

i) The Translation widget, which usually consists of three arrows aligned with the orthogonal axes centered on the object to be translated. Dragging the center of the widget translates the object directly underneath the mouse pointer in the plane parallel to the camera plane, while dragging any of the three arrows translates the object along the appropriate axis. The axes may be aligned with the world-space axes, the object-space axes, or some other space.

ii) The Rotation widget, which usually consists of three circles aligned with the three orthogonal axes, and one circle aligned with the camera plane. Dragging any of the circles rotates the object around the appropriate axis, while dragging elsewhere will freely rotate the object (virtual trackball rotation).

iii) The scale widget, which usually consists of three short lines aligned with the orthogonal axes terminating in boxes, and one box in the center of the widget. Dragging any of the three axis-aligned boxes effects a non-uniform scale along solely that axis, while dragging the center box effects a uniform scale on all three axes at once.

Depending on the specific common uses of an object, different kinds of widgets may be used. For example, a light in computer graphics is, like any other object, also defined by a transformation (translation and rotation), but it is sometimes positioned and directed simply with its endpoint positions because it may be more intuitive to define the position of the light source and then define the light's target, rather than rotating it around the coordinate axes in order to point it at a known position.

Other widgets may be unique for a particular tool, such as edge controls to change the cone of a spotlight, points and handles to define the position and tangent vector for a spline control point, circles of variable size to define a blur filter width or paintbrush size, IK targets for hands and feet, or color wheels and swatches for quickly choosing colors. Complex widgets may even incorporate some techniques from scientific visualization to efficiently present relevant data (such as vector fields for particle effects or false color images to display vertex maps).

Direct manipulation, as well as user interface design in general, for 3D computer graphics tasks, is still an active area of invention and innovation, as the process of generating CG images is generally not considered to be intuitive or easy in comparison to the difficulty of what the user wants to do, especially for complex tasks. The user interface for word processing, for example, is easy to learn for new users and is sufficient for most word processing tasks, so it is a mostly solved and standardized UI, while the user interfaces for 3D computer graphics are usually either difficult to learn and use and not sufficiently powerful for complex tasks, or sufficiently powerful but extremely difficult to learn and use, so direct manipulation and user interfaces will vary wildly from application to application.

The theory of direct manipulation describes interactive systems where the user physically interacts with their operating system. The fundamental feature of such a system is user control. Instead of typing commands and allowing the operating system to act as a strange intermediary, a direct manipulation system allows the user to feel like she is in control, by allowing her to physically interact with files and directories, and presenting a visual representation of the progress and end point.

Hints of direct manipulation programming environments have been around for quite some time. The first major landmark is Sutherland's Sketchpad, a graphical design program. Sutherland's goal was to devise a program that would make it possible for a person and a computer "to converse rapidly through the medium of line drawings." Sutherland's work is a land-mark not only because of historical priority but because of the ideas that he helped develop: He was one of the first to discuss the power of graphical interfaces, the conception of a display as "sheets of paper," the use of pointing devices, the virtues of constraint representations, and the importance of depicting abstractions graphically.

Scope, Application, and Limitations

Direct manipulation prescribes a general set of rules that have applications across the field of human computer interaction. Because direct manipulation systems present the user with an easy to use, familiar method of interaction, novices particularly have an easier time with these systems. For experts, well designed systems can be sufficiently fast, though often using keyboard inputs will allow an expert to work faster. Thankfully, dual implementation of these systems is generally not that difficult.

Because the system is familiar and easy to use, novices are also likely to learn more quickly. Since direct manipulation shows the progress of steps, errors occur much less often than they do in command type systems. This fact alone gives users confidence to explore and learn more features of the software more quickly. Because of these benefits, direct manipulation is present in designs from word processing to video games.

Example on direct manipulation:

The "trash" on the desktop is an excellent example of direct manipulation within computer systems. Users can see both the trash and the files or folders they want to move to the trash. They then physically select the files and drag them to the trash can. While the user is doing this, all of the selected files move as well, illustrating which items the user has selected. When the mouse is placed over the trash, there is a shading indication that it has been selected. The user then has to release the mouse button to move the files to the trash. If there are many files, a dialogue box will show up illustrating the progress of moving files to the trash. Once the action is completed, the files are no longer visible in their original location. The user can change her mind at any time while dragging the files to the trash. Even after placing them there, they can as easily be taken out of the trash and put back in place.

Applicability to Human computer Interface

Direct manipulation has become the method by which most computer users interact with their machines. Since the introduction of the Mac GUI in List and the early Macintosh machines, and slightly later the Windows interface, users have come to expect a mouse with visual and physical interaction with their operating system and software. Direct manipulation is evident in many other areas as well. Video games are a prime example. With only a small amount of training, users pick up controllers with buttons and knobs that generally map very well to the way the character is moving in physical space.

The extension into virtual reality is an easy one to make. In these situations, users are surrounded by an environment. In environments like the CAVE at Argonne National Laboratories, they can literally move around in that physical space, and visualize with normal movements. Since objects are oriented in 3-dimensional space, actions easily map to natural movements. Users can physically reach out, and using the provided hardware, “grab” an object and move it somewhere else.

Other virtual reality systems detect the 3-dimensional movements of the user and adjust accordingly. Remote surgery provides a good example. A surgeon may wear a head set to detect his position and adjust a camera view in the remote site. Gloves detect his exact position and movements, allowing a robotic counterpart on the remote end to replicate his exact movements. In this case, the surgeon is using direct manipulation as a method for controlling the remote system.

Self Assessment Questions

5. Direct manipulation is a human-computer interaction style that was defined by ____________.
6. Direct manipulation concepts can be applied to interfaces for blind or vision-impaired users, using a combination of tactile and sonic devices and software.(True/False)
7. Video games is an example for direct manipulation. (True/False)

2.6 Human Factors of a Graphical User Interface

In the process of designing the graphical user interface (GUI) many aspects have to be taken into account. The basic goal is to make the interface user friendly while completing the required function. The user is the key element in the process of design of the GUI according to the same nature of the interaction. The suitable attention to its characteristics will determine the success or failure of an application and will provide better elements and requirements for design. The user is limited or qualified by different human factors as psychological, physical, or sociological differences.

GUI builder tools have the same goal of helping programmer build the graphical interface, but they come in a large variety of forms. One important way that they can be classified is by how the designer specifies what the interface should be. Some tools require the programmer to program in a special-purpose language, some provide an application framework to guide the programming, some automatically generate the interface from a high-level model or specification, and others allow the interface to be designed interactively.

Based on the above analysis, four kinds of tools can be distinguished:

- Language Based Tools
- Application Framework
- Model-Based Generation
- Interactive Tools

The application might have the cleanest, fastest functional code written known to man but if the user cannot use it the project will fail. Human performance is a key objective that should be met with an application that will be run by people directly. Everything from the keyboard, mouse, visual display and application performance affects the success rate of your programs. For example, your application might be very easy on the eyes but if the performance is poor the user stains to complete simple objectives. Medium ground must be found balancing the user interface and the application performance. Of course the functionality of the program has to meet the user’s output needs as well but that should be obvious. Every application requires implementation and end-user training. It can be frustrating to find the users do not understand how to do certain tasks repeatedly afterwards. This is either a sign of end-users not accepting change or a poor GUI design among other possible things.

Self Assessment Questions

8. What is the basic goal of GUI?

   ____

9. Model-Based Generation is one of the tool in GUI (True/False)

2.7 Summary

This unit has provided you an overview of graphical user interface, its functions and concept of direct manipulation.

- A GUI is a Graphical User Interface, usually part of a computer's operating system which is characterized by WIMPs and WYSIWYG.
- A graphical user interface is a computer-user interface that uses graphical screen images as well as typed text, with icons on the screen replacing many of the functions of the keyboard.
- Many sighted people find GUI easier to use, because they don't have to remember or to look up special commands for each program function. Less time is spent figuring out how to get the computer to do what you want it to do.GUI can be used by people who are blind and visually impaired, provided they have a reliable screen reader to translate what's on the screen into Braille or synthesized speech.
- The GUI is based on the principle that pointing in menus to a command you want to computer to do is easier than having to remember hundreds of key words like in command line operating systems such as MS DOS or UNIX. The GUI is based on the idea that pointing to something is the most basic human gesture and the mouse is easier to use than a keyboard.
- The theory of direct manipulation describes interactive systems where the user physically interacts with their operating system. The fundamental feature of such a system is user control. Instead of typing commands and allowing the operating system to act as a strange intermediary, a direct manipulation system allows the user to feel like she is in control, by allowing her to physically interact with files and directories, and presenting a visual representation of the progress and end point.

2.8 Terminal Questions

1. List the main functions of GUI
2. Compare Direct manipulation and WIMP/GUI interfaces
3. Explain the use of Direct manipulation in computer graphics

2.9 Answers

Self Assessment Questions

1. Alan Kay
2. Bob Taylor
3. True
4. 1989
5. Ben Shneiderman
6. True
7. True
8. The basic goal is to make the interface user friendly while completing the required function.
9. True

Terminal Questions

1. The main functions are:
   - To allow access of users to the main system functions
   - To facilitate the efficient reception and memorizing of test items, in an accessible form, according to the set purposes. (Refer section 2.4)
2. Direct manipulation is closely associated with interfaces that use windows, icons, menus, and a pointing device (WIMP GUI) as these almost always incorporate direct manipulation to at least some degree. However, direct manipulation should not be confused with these other terms, as it does not imply the use of windows or even graphical output. (Refer section 2.5)
3. Because of the difficulty of visualizing and manipulating various aspects of computer graphics, including geometry creation and editing, animation, layout of objects and cameras, light placement, and other effects, direct manipulation is an extremely important part of 3D computer graphics. (Refer section 2.5)