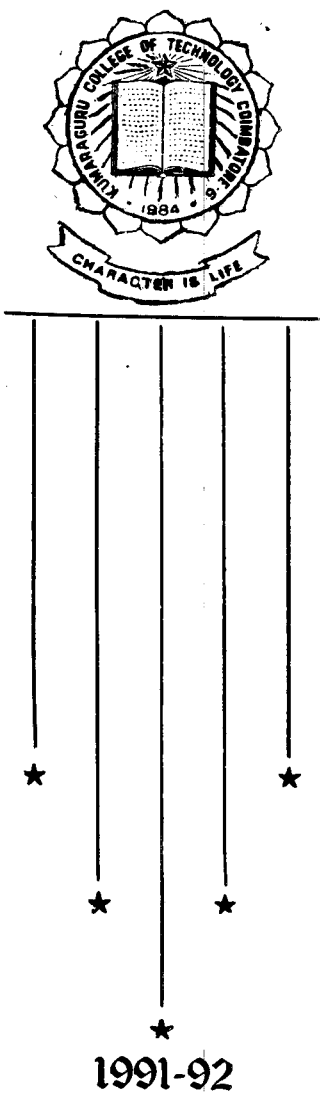
Picture Enhancement Using Autoshade & Animator (EPASA)



Project Report

Submitted by

Linda Susan Mathew
M. Chithara Banu
Susan Abraham

Under the Guidance of

Prof. P. Shanmugam,
B.E.,M.Sc(Engg).M.S.(Hawaii)

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING OF BHARATHIAR UNIVERSITY COIMBATORE

Department of Computer Science and Engineering

Kumaraguru College of Technology

Coimbatore-641 006

Department of Computer Science and Engineering

Kumaraguru College of Technology

Coimbatore-641 006

P-164

Certificate

This is to Certify that the report entitled

Picture Enhancement Using

Autoshade & Animator (EPASA)

has been Submitted by

*Ms.*_____

in partial fulfilment for the award of Bachelor of Engineering in the Computer Science and Engineering branch of the Bharathiar University, Coimbatore-641 046 during the academic year 1991-92.

.....
Guide

.....
Head of the Department

Certified that the candidate was Examined by us in the Project Work Viva - Voce Examination held on ..... *and the University Register Number was* .....

.....
Internal Examiner

.....
External Examiner

ACKNOWLEDGEMENT

We wish to express our sincere thanks to our guide, **Prof. P. Shanmugham B.E, M.Sc (Engg.),M.S.**, Head of the Department of the Computer Science and Engineering, for assigning us this project and for his guidance and encouragement from the initial stage of the project to the final stage of completion.

We are thankful to **Mr. L. Srinivasan**, a colleague of our Head of the Department, without whose help, we wouldn't have been able to obtain some vital software for our project. **Mr. L. Srinivasan** is the head of the AutoCAD Centre in Coimbatore Institute of Technology.

We are also thankful to our principal, **Major T. S. Ramamurthy** and the college management for their kind patronage.

Our warmest gratitude to our class advisor, **Miss R. Pavai**, for motivating us all along the way. And we are grateful to the other members of the faculty of the computer Department and to **Mr. Muthiah** of the Civil Department for helping us in the plan of the building used.

We are deeply grateful to **Miss S. Dhanalakshmi** and **Mr. A. Ramesh**, of CCIT without whose help, our project wouldn't have been completed. We are truly grateful for their support.

SYNOPSIS

AutoCAD is a software package, a product of AutoDESK Inc. It represents the state of the art in computer aided drafting. AutoLISP is an implementation of the LISP programming language, embedded within the ADE 3 package. AutoLISP was created by the developers of AutoCAD to help you modify AutoCAD, save time and boost productivity.

EPASA helps to draw the plan of a house after giving the required dimensions of the house i.e the length, width, wall thickness and height. It even asls for the door and window details. AutoLISP programs are written in order to draw the house, customising AutoCAD and to obtain the house's three-dimensional view.

Here, even AutoSHADE, a rendering program that converts AutoCAD's three dimensional line drawings into realistic pictures is used. It shows the perspecitve, surface shading and specular reflection of the house created by AutoCAD. A curved object is also created by AutoCAD, in order to show the real effect of AutoSHADE.

ANIMATOR, another AutoDESK creation is implemented in this project. It creates animated graphics by using a wide array of techniques and drawing tools. It expresses anyones creativity and imagination.

Video images of objects, photographs and pictures can be captured using video cameras. These images can be saved in a variety of digital formats. Cameras that can be used are a hand-held camera (video), an RGB video camera mounted on a copy stand, and a Hi-8 video camera and a video capture board in your computer with a corresponding monitor. A hand held video camera has the lowest resolution. An RGB camera produces excellent images, but does not work for home movies. A Hi-8 video has excellent resolution, and it makes great home movies.

CONTENTS

FIGURE INDEX

1. ABOUT AutoCAD

1.1 INTORDUCTION

The AutoCAD drafting package is a general purpose Computer Aided drafting application for your computer CAD applications are tremendously powerful tools. The speed and cas with which a drawing can be prepare and modified using a computer offers a phenomenal time saving advantage over "head" preparation. AutoCAD brings this sophisticated techonology, previously available only on large and costly systems, to you, the desktop computer user. There is virtually no limits to the kinds of lines drawings you can prepare using AutoCAD. If a drawing can be created by hand, it can be generated by the AutoCAD. As you draw with AutoCAD, you are creating much more than a drawing. You can place associated objects on layers or group them, forming complex objects that you can operate on as a whole. AutoCAD remembers the locations, sizes and colors of the objects you draw, maintaining them in a database for subsequent retreval, analysis and manipulation. The AutoCAD is available for a variety of desktop computers and engineering workstations, and runs under the PC-DOS, MS-DOS, UNIX, AEGIS, and UMS operating systems. No technical computer knowledge is required to aid AutoCAd effectively; practice and a thorough understanding of its features are the keys to proficiency.

1.2 REQUIRED EQUIPMENT

In addition to a basic computer system (including processor, keyboard, text display screen and disk drives) Auto CAD requires a graphic monitor capable of reasonably high resolution for computers base on INTEL 8086 family of Microprocessors, an 8087, 80287 or 80287 math coprocessor is also required.

1.2.1 DISPLAY MONITOR

On some computers, AutoCAD uses two display Monitors, one for command prompts and text output, and the other for graphics on these systems, the graphics monitor can also display a screen menu along its right edge and a one line prompt area across the bottoms, as in the following illustrations. See Section 1.5 for a description of the screen men on other systems, a single monitor serves for both graphics and tect. Here three lines at the bottom of the screen are reserved for command entry and prompts, and the right edge can contain a screen menu when running on a single screen system, AutoCAD remenmbers a full 24 lines of tect, just like the regular text display. If information two scrolled of the three line display, you can text display when outputting a large amount of informations and returns to the graphic display when drawing anything. A single-screen system is illustrated next. Depending on the capability of your

particular display and on the ADE packages included in your copy of AutoCAD, the top of the graphics area can contain a status line showing the state of various AutoCAD mode switches and the location of the screen crosshairs, and a menu bar from which you can access pull down menus.

1.2.2 OPTIONAL EQUIPMENT

The equipment listed above is sufficient for a basic AutoCAD installation with this equipment. You can learn how to use AutoCAd and make drawings for display on your monitor. Additional equipment is necessary if you want to make "hard copy" print outs of your drawings.

1.2.3 PEN PLOTTERS AND PRINTER PLOTTERS

A pen plotter or a printer plotter can be connected to the system to produce a "hard copy" of a drawing. AutoCAD can support one plotter and one printer plotter on the same system. Some of these devices connect to an RS-232C serial communications part on your computer others connect to a centronics - type parallel output part, and a few require special connections. Also, not all plotters are available on all systems. The models supported on your computer and instructions for installing them for use with AutoCAd can be found in the accompanying AutoCAd installation/performance guide.

1.2.4 POINTING DEVICES

A pointing devices such as a mouse or a digitizing tablet provides the means for instant command and point entry. Keyboard entry is relatively easy, but pointing at the screen and pushing a button is even easier. In addition to locating points and entering commands, you can use a digitizing tablet to trace over existing frawings. Descriptions of the types of devices available appear below.

1.2.4.1 MOUSE

As you move a Mouse around the table top, crosshairs track Mouse Movement on the screen. To select a point or menu items position the corsshairs on it and push the button on the mouse. If the mouse has multiple buttons, they are used to invoke frequently-needed AutoCAD commands.

1.2.4.2 TABLET

Point and Menu item selection using a digitizing tablet are similar to the mouse operation described above. However, you move the tablet's puck or stylus around only on the tablet's surface. The tablet offers two capabilities beyond those of the mouse. You can align the tablet with the coordinate system of an existing paper drawing so that you can

use Autocad to trace over it. You can set aside up to four areas of the tablet for tablet menus.

1.3 CONCEPTS AND TERMINOLOGY

This section presents some terms and concepts you will encounter in this manual and while working with AutoCAD. These items are best understood by working with the program. Read this discussion now, when refer back to it if you have questions while using the program.

1.4 AutoCAD DRAWING

An AutoCAD drawing is a file that describes a graphic image. AutoCAD interprets the objects described in the file and draws them on the screen exactly as you would draw them manually.

1.4.1 COORDINATES

A cartesian coordinate system is used for locating points in the drawing, to position entities, for instance. An X coordinate specifies horizontal location and a Y coordinate specifies vertical location. Thus any point on drawing can be indicated by an X and Y coordinate pair of the form (x,y). The (0,0) point is normally at the lower left corner of the drawing. The figure below shows a cartesian coordinat system.

1.5 DRAWING UNITS

As noted, entities in the drawing are positioned on coordinate points. For example you draw a line by specifying the coordinates of its two end points. The distance between two points is measured in units. Thus, a line drawn between the points (1.1) and (1,2) is one unit in length. There is no need to restrict yourself to integers, however. AutoCAD's drawing database retains at least 14 significant digits of precision for each point, so you can place an object at the point (507.841142,0.0038059) if you like.

A unit can correspond to whatever form of measurement your drawing requires. it can be inches, feet centimeters, angstroms, whatever. Thus, you can draw using "real world" units and eliminate the possibility of scaling errors. When the drawing is complete, you can plot it at whatever scale you like. In fact you can plot a given drawing at several scales, eliminating the need for separate drawings at different scales.

1.5.1 DISPLAY

The term display has two related meaning in this manual usually, displays refers to the portion of the drawing

currently being shown. Occassionally, display means the graphics screen upon which your drawing is shown.

1.5.2 ZOOMING AND PANNING

The display can be zoomed in or out to magnify or shrink the visible image of the drawing. When the display is zoomed out, you can see a large portion of the drawing. Zooming in can "blow up" a small portion of the drawing and show more of its details. You can zoom in to draw intricate parts of your drawing with exacting detail and then "back off" to look at the finished drawing AutoCAD's "Zoom ratio" is about ten trillion to one, more than adequate for most applications.

1.5.3 WINDOW

The graphics screen is used as a window through which you can look at all or part of your drawing, keep in mind that coordinates refer to fixed locations in the drawing, not to the physical location on the display screen. Therefore, the absolute size of a unit remains constant, the points (1,1) and (1,2) are always one unit part, although the apparent distance between points on the screen varies with different zoom levels. When you zoom out, the distance between coordinates appears to be small. A line drawn between (1,1) and (1,2) may be only a quarter inch long measured on the screen. When you zoom in, the distance between coordinates appears larger, so

the one-unit line may appear to be several inches long. In both cases, the absolute distance between the coordinates in constant only the screen display changes.

Panning allows you to view a different portion of the drawing without changing its magnification, a line drawn from (1,1) to (1,2) can thus appear in different locations on the screen.

1.5.4 DRAWING LIMITS AND EXTENTS

The drawing limits are the borders of a rectangle the drawing area, expressed in drawing coordinates. User can select whatever limits make sense for his drawing. If we want to draw a Printed Circuit Board (PCB) of size 8" x 10", if it is possible to have this size of rectangle. User can also change this limits as per his requirement.

1.5.5 THE DRAWING EXTENTS

The drawing extents specify the actual size at the present time. Imagine a rectangle surrounding all the objects in the drawing the smallest such rectangle defines the drawing extents.

1.6 DISPLAY EXTENTS

AutoCAD keeps track of the current screen location by maintaining a set of borders called the display extents. These are the borders of the current display expressed in drawing coordinates. For example, if it is required to display a magnified view of the center of an 8" x 10" PCB, the display extents might be

Lower Left Corner : (4,3)

Upper Left Corner : (6,5)

Zooming and Panning change the display extents. When these occur, the drawing is regenerated or redrawn to show only the portion bounded by the new display extents refer Fig 2.

1.7 RESOLUTION

1.7.1 PHYSICAL RESOLUTION

Physical resolution refers to the amount of detail that can be represented. The resolution of the display device is specified as "dots X by dots Y". Higher resolution means a smother looking display. This affects the work done on that device, not AutoCAD's internal resolution.

While extending the coordinates, it is possible to snap (lock) them to the nearest point on a grid, which need not be visible. The spacing of the grid points is the snap resolution. It is completely independent of the resolution of input or output at any time, or turn it off entirely for "free style "drawing.

1.8 PROGRAM OPERATION

Drawing speed and productivity increas with the knowledge of the program.

1.8.1 THE MAIN MENU

By selecting the appropriate task from this menu, user can create a new drawing , edit an existing drawing, plot a drawing or terminate the AutoCAD session. This menu provides access to all parts of AutoCAD.

1.8.2 INTERACTIVE DRAWING EDITOR

When a user creates a new drawing or edits an existing one, AutoCAD automatically loads the drawing editor. The drawing Editor display drawing and provides commands to create, modify, view and plot drawings.

1.8.3 DATABASE STORAGE AND PORTABILITY

Each command automatically updates all information about the drawing; the size and position of every element, size of the drawing itself, its display characteristics.

1.8.4 POINT AND COMMAND ENTRY

Points are entered using keyboard or a pointing device. From the keyboard, points can be entered by typing absolute, relative coordinates or using cursor control keys.

Commands can be given by typing in or selecting a command from any of the menus described below.

1.8.5 SCREEN MENU

A menu can be displayed on the right edge of the graphics screen while the Drawing Editor is active. This menu lets us to enter a command by simply printing to the command on the screen or by using the keyboard's arrow keys.

1.8.6 TABLET MENU

It is possible to place as many as four menus of AutoCAD commands on the digitizing tablet then enter a command by

simply printing to it with the puck or stylus. Button menu is also available if tablet stylus or mouse has multiple buttons.

1.8.7 PULL DOWN MENU

If the display device has cetain advanced capabilities we can select commands from menus that can be pulled down from a menu bar at the top of the screen.

1.8.8 ICON MENU

Any menu item that is selected can display an icon menu from which we can make further selections. An icon menu displays the choices in pictorial menu. Icon menus require advanced capabilites with display device.

1.8.9 PLOTTING

To check for positioning and dimensioning errors that might not be apparent on screen we can generate 'check-plots' while the drawing is in progress. Plotting can be done on a pen plotter or printer with graphics capability.

1.9 OBJECTS WITHIN A DRAWING

Objects within a drawing may be simple or complex. Computer objects can be built using simple ones construction

of rectangular or circular arrays (pattern) of objects and insertion of entire drawing into the current drawing is possible. Different parts of a drawing can be kept in layers.

1.9.1 ENTITIES

Entities are predefined elements that can be put into a drawing by means of a single command. Some of the entities are lines, arcs, circles, points, text, traces, polylines, 3D times, 3D faces etc.

1.9.2 DRAWING INSERTION

This feature treates an existing AutoCAD drawing as a Block and Merge it into the drawing part and store it. So we can keep a library of symbols and components used often in work.

1.9.3 COLORS AND LINETYPES

AutoCAD allows to assign a color and a line type to each entity. The color code varies from 1 to 255 line type is a specific sequence of alternating line segments and spaces. Standard name have been assigned to the first seven color numbers but the actual colors displayed depends on the display device used.

1.9.4 LAYERS

We can assign various portions of the drawing to different layers and can be defined as amny layers as we want layering allows us to view and plot related aspects of a drawing separately or in any combination.

1.10 AUXILLARY FEATURES

1.10.1 HELP DISPLAY

A Help display is available to remind the command names and the options available for entering prints and others data Help on the format of specific commands is also available. Help is possible about the command with which we are working without exiting from it.

1.10.2 FILE DIRECTORY ACCESS

Listing disk directories and delete, rename or copy files without exiting AutoCAD is possible .

1.10.3 DRAWING INTERCHANGE CAPABILITY

AutoCAD can save drawings in the form of an ASCII text file so that they can be easily processed by user written programs or transfered to a different computer. These drawing

interchange file (D x F TM) can also be created by user written programs for AutoCAD to turn back into drawing files. Translations between AutoCAD and other CAD systems database formats, and special purpose analysis and modifications of AutoCAD drawings can be accomplished by means of this mechanism.

Interchange files can also be read written in the initial graphics exchange standard format.

1.10.4 PATHNAME SUPPORT

AutoCAD makes full use of the tree-structured directories provided by your computers operating system. You can maintain several directories of drawings with a single copy of AutoCAD and its support files. If you prefer you can ignore the tree-structured directories entirely and keep the AutoCAD program files and all your drawings in the current working directory.

1.10.5 EXTENDED AND EXPANDED MEMORY SUPPORT

Certain computers allow installation of additional memory that is not addressable by the MS-DOS or PC-DOS operating system or by most application programs. AutoCAD provides handlers for both the Lotus/Intel/Microsoft expanded memory specification (EMS) and the IBM PC/AT extended memory

interfaces and assess this memory to reduce disk accesses and increase processing speed for large drawings.

1.10.6 DATA/TIME OF CREATION AND REVISION

The date and time of creation and most recent revision are maintained with each drawing and can be displayed upon request. The current date and time, elapsed time in the drawing editor, and a user controlled timer are also provided.

1.11 ADVANCED DRAFTING EXTENSIONS

Three packages od advanced features, ADE-1, ADE-2 and ADE-3 are available as options. The ADE-2 package requires ADE-1 and ADE-3 requires both of the others.

ADE-1

* associative dimensioning is useful for such applications as mechanical engineering and architectural drafting. We can easily add annotations for linear dimensions angular dimensions and circle/arc diameters and radii. AutoCAD can measure the dimension and use the calculated value as the dimension text.

* smooth arcs, or fillets can be drawn to connect two lines, arcs as circles lines and arcs are extended or trimmed as necessary, so that they end precisely on the fillet arc.

* lines, traces aid arcs can be split or broken into two pieces or one end can be cut off. A portion of a circle can be deleted to create an arc.

* on the graphics screen, we can display arcs with any desired spacing of the axis ticks.

* an object can be cross-hatched or pattern filled

* a free hand sketch facility requiring a pointing device, facilitates to draw a series of short connected lines with a specified resolutions quickly and easily.

* a continuous status line displays the condition of various modes and the location of the screen crosshairs. This can be disabled.

ADE-2

* object (geometric) snap makes us to refer points (endpoints, midpoints etc..) of existing objects in the drawing as well as to grid points

* the aspect ratios and rotation of the snap grid, visible grid, and axis ruler lines can be varied

* isometric grid and snap capabilities are provided

* Dynamic specification or dragging allows visually varying monitor before completing the entry.

* we can define and retrieve named views to associate a name with particular display of the drawing and return to that view quickly and easily.

* existing objects can be mirrored about an axis that is specified.

* a slide capability allows saving the current display from the monitor. Demonstration works can be done.

ADE-3

* the Advanced User Interface (AUI) allows entering commands from pull-down menus and icon menus. Several commands display dialogue boxes that show all options and settings at a glance we can change settins by pointing at them.

* 3D level 2 lets is to draw 3D lines and plane sections (3D faces), aid "extrude" 2 D object suppressing the "Hidden" lines is possible.

* film roll file of a 3D drawing for use by the AutoSHADE reading program.

* polylines are composed of connected lines and arc segments.

* AutoLISP is an implementation of the LISP programming language.

* ellipses regular polygons and "doughnut" can be created using simple commands to draw the appropriate polylines.

* spline fitting or curve fitting can be done on existing polylines

* we can explode a polyline, a Block or an associative dimension into simple entities.

* offset curves and parallel lines can be drawn.

* existing objects can be rotated scales, stretched and extended or trimmed to meet other objects.

* adding new commands to AutoCAD is possible. These external commands operate by executing user supplied programs from with AutoCAD.

* using Initial Graphics Exchange (IGES) file format CAD files can be exchanged with other CAD systems.

1.12 CAD PROCEDURE IN DISIGN AutoMATION

The general schematic structure of design and manufacturing process using CAD is shown in Fig No.2

The process consists of 3 phases.

1.12.1 SPECIFICATION

Here the criteria of performance of the design object are laid down which involves information, collection, manufacturing constraints, legal requirements and standards and so on.

1.12.2 GENERAL SYNTHESIS

This phase is the heart of the design process. The computer generates a design depending on the parameters inputted. A new design is created for a small change in the

input for permitting and recombining components or elements in a completely new configuration.

1.12.3 EVALUATION AND ANALYSIS

Here the alternative design which have been generated are tested in turn and compared to see if they must meet the specification. This phase is iterated with generation phase till an optional design is obtained. This optional design obtained is used in the manufacturing of the object.

1.13 ADVANTAGES

a) high productivity
b) improvement in quality (appearance and accuracy)
c) use of library parts and symbols
d) compact storage and easy access
e) graphics grouping in layers
f) ability to select size and scale to suit meets
g) data file creation for design analysis and manufacture

1.14 APPLICATIONS

CAD has a wide range of applications - important among them are

* Architectural drafting
* Mechanical drafting

* Electrical applications in control schematics and connection diagrams etc..
* Robotics
* Aerospace
* Piping and instrumentation diagrams.

CAD systems are now mass produced and continued improvement in each new generation of machinary have decreased the cost of CAD systems. Also software has been extensively developed, enhancing their adaptability to suit individual company. Thus the CAD unit is becoming an integral part of the industry.

2. AutoLISP

2.1 INTRODUCTION

AutoLISP is a special language used to write instructions carried out by AutoCAD. AutoCAD instruction that have been written in AutoLISP are called LISP routines. LISP routines are contained in ASCII files that are called LISP files. LISP files can have any name, that is a valid DOS file name, and they always have file extensions LISP.

AutoLISP is an implementation of LISP programming language which is a powerful tool for optimising AutoCAD's performances. LISP routines enable the AutoCAD user to automat AutoCAD. LISP routines quickly perform calcuations and analysis of data used to generate drawing entities or create new entities by invoking AutoCAD commands directly. The entities created by LISP routines are the same as any created by the user except that with AutoLISP, the process is easier and faster. LISP routines greatly enhance the production of complex drawings as well as eliminating unnecessary menu selections and repetitive keyboard entries.

2.2 AutoLISP - THE FEATURES

AutoLISP is an implementatin of the lisp programming language embedded within the AutoCAD ADE 3 package. AutoLISP

allows users and AutoCAD developers to write Macro programs and functions in a very high level language that is well suited to graphics applications. Lisp is easy to learn and use and is flexible.

* Lisp excels at working with collections of heterogenous objects in various sized groups which is precisely the type of information a CAD system like AutoCAD manipulate.

* A Lisp interpreter is ideally well suited to understand interaction that characterises the design process.

* Lisp is among the easiest language to learn and to master.

* It is the language for research in AI and expert system.

* Also Lisp interpreter is fairly easy to implement.

* Autolisp is enabled using AutoCAd configuration

2.2.1 REGULAR AutoLISP

The requirements of R.Autolisp are,

* a computer supported by AutoCAd with atleast 640 K bytes of memory and hard disk PC-DOS / MS - DOS

* a matching version of AutoCAD with ADE-3

2.2.2 EXTENDED AutoLISP

The requirements of this E. Autolisp are,

* a computer base on the Intel (80286/80386) Micoroprocessor, supported by AutoCAD with at least 640 K bytes of memory and a hard disk.

* at least 512 K bytes of IBM AT-style extended memory not for any other purpose

* PC DOS / MS DOS version

* a matching version of ACAD with ADE-3.

The Extended Autolisp does work on 80286/80386 models in their 'protected' mode with it residing in the extended memory. Ext-Alisp makes available additional DOS memory to ACAD for I/O paging purposes.

Ext.Alisp does not work on PC/XT class computers since the 8088/8086 CPU'S do not allow protected mode. Otherwise there is no difference between Extended Alisp and regular Alisp.

ABOUT EXTENDED LISP ...

It is implemented as a separate program Ext Lisp that must be run before starting AutoCAD. Extended Lisp is a terminated and stay resident (TSR) program that resides partially in extended memory and communicates with AutoCAD by means of the "a cad lx.ovl" overlay file the necessary Extended lisp command can be placed in the autoexec.bat file.

Remlisp command at the DOS prompt removes EXT.Alisp though making the DOS memory occupied by the EXTlisp available memory is as before loading Alisp.

When Autolisp starts up, it requires two large areas of memory for itself. The first called HEAP is the area in which all functions and variables (also called nodes) are stored.

2.3 DATA TYPES IN AutoLISP

Autolisp supports several data type

* lists

* Symbols - symbol names cannot start with a digit and is not case sensitive to Autolisp.

* Strings - strings can be of any length. AutoLISP allows dynamic memory allocation for them.

* Real Numbers - Stored in double precision floating point format providing at least 14 digits (significant) of precision.

* Integers - Autolisp allow 32 bit signed number but when transfered to AutoCAD only 16 bit transfer occurs.

* File descriptors

* AutoCAD entity "names"

* AutoCAD selection sets

* Subrs (built in function)

2.3.1 SPECIAL DATA TYPES

* entity name

* selection set

* AutoLISP - Evaluation of data

* The evaluator takes a line of user input, evaluates it and returns some result. The following is the process of evaluation.

* integers, reals, substrings file pointers and startings evaluate to themselves.

* symbols evaluate to the value of their current binding

* lists are evaluated according to the first element of the list, if it evaluates to

* a list, the list is assumed to be a function definition and the function is evaluated using the values of the remaining list elements as arguments.

* the name of an internal function (subr) the remaining list elements are passed to the subr as the format arguments and evaluated by the subr.

When a Alisp expression is entered against the ACAD command prompt then Alist evaluates the expression and prints its results the Acad command reappears.

2.3.2 LEXICAL NOTATIONS AND CONVENTION

Alisp input be taken in several forms

* read from ACII files

* read from a string variable

* read from keyboard from within ACAD. But conventions have to remain.

* symbol name can consist of any sequesnce of printable characters other than delimiting characters (), " "; etc.. These delimiting characters terminate a symbol or numeric constant.

* Expression can be extended to more than one line

* Literal strings are sequences of characters surrounded by " " within quotes the control characters can be included

  e - escape
  n - newline
  t - tab
  r - return
  nnn - octal code

2.4 ERROR HANDLING

If Alisp encounters an error during evaluation it brings a message of the form

Error - text where is description of error if ERROR fn is defined (non-nel) Alisp executes the fn instead of printing the message if the error fn is not defined Alisp evaluation stops and trace back display is given.

And variables used, the more heap space will be used. The second area called STACK holds function arguments and partial results AutoLISP cannot arguments and partial results. AutoLISP cannot expand its HEAP and STACK space while running under AutoCAD. If enough functions and variables are defined to use up all the HEAP area, then AutoLISP will terminate the execution of the current functions and will display the error message. AutoLISP will not function itil more memory is made available and AutoCAD is executed again.

Default sizes for the HEAP and STACK areas are:

HEAP - 5000 bytes

STACK - 5000 bytes

The amount of memory can be altered and the sum of the two areas cannot exceed 45000 bytes.

3. AutoLISP PROGRAMMING

AutoLisp program is written for the three types of modification. The procedure for developing the progaram is given in figure. First the computer is initialized into the AutoCAD subdirectory. By means of "SHELL' command, we get into the operating system where the program is being input.

The basic parameters of the component is given. The other dimensions are defined as the function of "BASIC" dimensions. The AutoCAD commands are written in the AutoLISP format using those commands the drafting sequence is incorporated in the program.

If the program is lengthy, same part of the input is stored in the virtual memory.

The program is then loaded. The input parameters are obtained from tabulation using "TABLE SEARCH" command. Then the dependent parameters are evaluated. The drafting proceeds module by module and the ouput is obtained.

The block diagram for the AutoLisp procedure is as follows.

Block diagram for AutoLISP PROGRAMMING shown in Fig.3

3.AutoSHADE

3.1 INTRODUCTION

AutoSHADE is a rendering program that converts AutoCad's three dimensional line drawings into realistic pictures that show perspective, surface shading, and surface reflection. Auto shade is a post processor. That is AutoSHADE is used after creating your AutoCAD drawing and exiting from the AutoCAD program. The following AutoCAD entities can be used to create objects for rendering with AutoSHADE.

* 3D faces

* circles, solids, traces, doghnuts, and wide polylines.

* extruded entities

3.2 WHY RENDER A DRAWING ?

Rendering an AutoCAD drawing with AutoSHADE creates a picture that is much more realistic than the corresponding wire frame representation. A rendered drawing allows you to visualise the relative placement of three dimensioal objects much more clearly and easily than wire frame representation. Designers need to see realistic pictures of what their projects will look like when they are completed. The

realistic image generated by AutoSHADE give designers the means to preview designs without having to build physical models or prototypes.

Images created with AutoSHADE can be used to explore possible design alternatives. A set of renderings can be generated to provide concrete examples of the design variations. The images created by AutoShade can also be used as a tool for the verification fo correct design in three dimensional models. Flaws and inaccuracies in a three dimensional model can offen be difficult to find in a wire frame representation. Rendered images can make these previously unnoticed design problems obvious.

3.3 AN OVERVIEW OF AutoSHADE

The drawing is created with AutoCAD before rendering it with AutoShade. The following is a diagram of the AutoShade rendering process which is given in Fig.4.

The first steps of creating the rendering take place while you are working in AutoCAD. The colors you use in the AutoCAD drawing determine the color in the rendering. The AutoCAD drawing defines the x, y and t axis of the objects and their resulting co-ordinates in AutoShade.

Camera and light symbols is placed in the AutoCAD drawing to specify view points and light sources. After placing cameras and lights and scenes, a filmroll file is created using the FILMROLL command. The filmroll file contains the three dimensional objects and the lights and cameras that AutoShade will use to create the rendering. AutoCAD will be exited from and AutoShade is entered.

Upon entering AutoShade, the first step is to load the filmroll file. AutoShade creates its own internal database of the entities created in AutoCAD. Thus, AutoShade is capable of creating any number of renderings with different viewpoint and shading parameters without returning to AutoCAD. AutoShade can display three dimensional drawing in either a wire frame or a shaded representation. the drawing can be displayed from any desired viewpoint and shade it according to the positions and intensities of the light sources you select.

3.4 WORKING OF AutoSHADE

AutoSHADE is a package that needs a three-dimensional object to work on. So this object is created by implementing AutoCAD. AutoSHADE is a tool that makes realistic pictures from models built with AutoCAD. AutoSHADE used lights, cameras and scenes to define the pictures it generates and these are placed in the AutoCAD drawing by a portion of AutoSHADE implemented in AutoLISP.

3.4.1 IMPLEMENTATION OF ASHADE OPTIONS IN AutoCAD

In order to set camera, lights and scene to the 3D object in AutoCAD, the special AutoSHADE additions to AutoCAD, are used by loading the AutoLISP program that implements them. This can be done by either picking ASHADE from the screen menu or by entering the command.

```
COMMAND : (load "ashade")
```

Now, place the camera, lights and scenes in the diagram.

PLACING A CAMERA

In order to place a camera in the diagram the command

```
COMMAND : CAMERA
```

is typed at the command prompt of AutoCAD immediately, the name of the camera will be asked. Any user-defined name is sufficient. Then the target point will be asked for and then the camera location.

After these values are picked on the screen, the camera appears in the drawing at the specified point, aimed at the specified target point.

PLACING A LIGHT

AutoSHADE allows two kinds of light, point and directed sources. A point source will radiate light in all directions while a directed source allows light to shine along a line, like a spotlight.

To place a light in the drawing, the command

COMMAND : LIGHT

should be entered, which will lead to the determination of the name of the LIGHT. The next prompt will ask whether a point or directed source is required. 'D' tells AutoCAD, directed source is required and 'P' tells if the point source is needed. But directed light source will ask for the point, the light source is aimed at. A point light source does not request an aim point. After all this is carried out, the light location will have to be entered.

After this specification is made the light appears in the drawing.

MAKING THE SCENE

In AutoSHADE. cameras and light sources are always grouped into scenes created in AutoCAD, so cameras and lights are selected by choosing a scene.

Here, again, after entering the command

COMMAND : SCENE

the name of the scene will be requested. Now, in one particular scene. there can be many light sources but only one camera. And the camera and lights can be selected only by AutoCAD's object selection cursor. Pointing to the camera or lights, the respective name will be echoed to confirm the selection. After the required camera and light sources are selected, the scene location wil be requested. The scene can be placed anywhere desirable and at the specified point, a clapper icon is drawn and it will name the camera and lights we've selected for the scene. This clapper controls AutoSHADE but does not appear in the pictures AutoSHADE makes.

In order to delete a SCENE, a window or crossing selection which encloses the entire clapper. The scene block is actually composed of several pieces, and if the clapper is just pointed at objects will be left behind. if the clapper

pointed at is to be deleted, it not only clutters AutoCAD drawings, but it also confuses AutoSHADE.

MAKING THE FILMROLL

After creating, cameras, lights and scenes, all that remains is to create a new filmroll file containing the camera, light and scene we've added. The command:

COMMAND : FILMROLL

is entered . The default name for the filmroll file is the same as the drawing name.

With the filmroll made, we're done with AutoCAD, so leave AutoCAD with the END command.

3.4.2 BACK TO AutoSHADE

Filmroll in hand, lets develop the pictures, Fire up AutoShade, choose open from the File Menu and use the select filmroll file dialogue to load the new filmroll muopen. Then choose select scene from the settings menu. The dialogue boxes will show the various scenes created. Pick the scene required and pick O.K to confirm the choice.

CHANGING LENSES

If at all the image seen in the SCENE is not as expected and the camera position has to changed or view is not as expected, then select the settings menu and choose the Camera Position item. The lens can be changed according to the wide angle of the view required. Shorter the focal length, will give a wider view of the scene. Larger the focal length, will give a mere telescopic view.

Focal Length (mm)	Field of view (degrees)	Nomenclature
15	115	Extreme wide angle
17	102	Ultra wide angle
24	84	Extra wide angle
35	63	Wide Angle
50	46	Normal
85	28	Medium telephoto
100	24	Medium telephoto
135	18	Telephoto
200	12	Telephoto
500	5	Long telephoto
1000	2.5	Telescopic

ADJUSTING THE LIGHTS

Suppose the lightings are not right, like either the drawing appears too bright or too dark at certain regions of the drawing, then lights can be adjusted. So settings menu is selected and the lights item is chosen. The set light intensities dialogue appears.

AutoShade sets all light to the default intensity 1, when a scene is chosen. unlike real lights, AutoShade light intensities have meaning only relative to one another. This is because AutoShade's Camera has an unlimited range automatic exposure meter so you can't overexpose or under expose a frame.

SAVING THE PICTURE

AutoSHADE will give a full shaded as well as fast shaded drawings. The fast shaded drawings can be saved on the disk for future reference, if full shading is required.

The display menu is selected and the record item is chosen. This turns on record mode. As long as this is in effect the option of saving all fast and full shaded pictures on the disk is obtained. Select the Display menu again. When the menu rolls down, the record item checked, indicating that record mode is in effect. Choose full shade. With record mode

in effect, a dialogue box titled create rendering file appears. The file name in which the picture is to be save is entered by highlighting the box at the right of file name. Record will be "greyed out" if hardcopy is turned on. Turn Hardcopy off by selecting it, then record is picked. The file name defaults to the name of the filmroll file. Once a redering replay file has been made we can replay it at any time just by choosing the replay item from the display menu. Pick the replay file, select OK and the picture will be rapidly shown on the screen without any of the calculations required by foot shading or full shading. AutoSHADE can be brought up and go directly into REPLAY without loading a filmroll file. REPLAY ALL option will put AutoSHADE on "automatic" and have it show a set of slides over and over.

In order to create a PICTURE.

MAKING A WIRE FRAME PICTURE: This picture is similar to the views displayed by AutoCAD. A wire frame picture is created by selecting the DISPLAY menu, then WIRE FRAME item is chosen. After computations are completed, the wire frame image is drawn on the screen. The WIRE FRAME picture is seen only to see whether the actual view will be obtained, otherwise changes in viewing the scene will have to be made.

MAKING A FAST SHADED PICTURE : A fast shaded picture is created only to determine whether the lightings are correct. The image will be correctly shaded if lightings are right but

skips much of the complex calculation of which the objects in the drawing obscure with other objects.

To make a fast shaded picture, select the DISPLAY menu, then choose the FAST SHADE item from it. Finally, AutoSHADE will display the fast shaded image. There will be errors introduced by the fact that fast shading skips the more complex calculations - they will be fixed when a full shaded image is made.

MAKING A FULL SHADED PICUTRE

After deciding whether the lights and other settings are correct by the fast shaded picture, then a full shaded picture is made.

To make a FULL SHADED PICTURE, select the DISPLAY menu, and choose the FULL SHADE item from it. The mechanism of full shading is the same as fast shading, in addition to the additional time required and the correctness of the output. The errors in the fast shaded image are corrected by the additional calculations performed by full shading.

MOVING THE CAMERA AROUND

After placing the camera in the AutoCAD drawing, and the target point or its position is not correct, change the position of the camera is changed in AutoSHADE itself, instead of entering AutoCAD and changing the camera position. But the

changes in the camera placement in AutoSHADE is not reflected in the AutoCAD drawing or the filmroll.

To see how the camera is moved around in AutoSHADE, select the SETTINGS menu and choose the CAMERA POSITION item from it. The CAMERA SPECIFICATIONS dialogue appears. There is the DEGREES RIGHT box which is the angle of the camera from the X-Axis box which is the angle of the camera from the X-Axis in the X-Y plane with the degrees indicating the camera is looking along the X-Axis. Angles increase as the camera moves counter-clockwise.

The angle in the DEGREES UP BOX is the angle of the camera out of the X-Y plane towards the Z-Axis measured with respect to the X-Y plane. 90° indicates the camera is looking straingt-down and 0° indicates that the camera is parallel to the X-Y plane. The box DISTANCE is the distance of the camera from the target point.

TWIST item in the CAMERA position menu, helps to rotate the camera about the line of sight from the camera to the target. It is specified in degrees. A +ve angle makes a counter-clockwise rotation in the rendered picture and a -ve angle makes a clock-wise rotation.

SAVING IN A SLIDE FILE

The DISPLAY menu, gives the options MAKE SLIDE and MAKE DXB. MAKE SLIDE creates AutoCAD slides of AutoSHADE wire frame images including images with clipping and prospective. It displays a file creation dialogue box requesting the name of the slide file to be created. It produces a regular wire frame picture or your screen and writes the picture to the slide file. It identifies it as an AutoCAD slide file.

MAKE DXB FILES

The MAKE DXB item performs a WIRE FRAME command and it also creates and AutoCAD binary interchange file containing the picture.

The resulting DXB file is composed of line entities in the X-Y plane in a one by one square with the origin (0,0) in the lower left corner. Since DXB files have no provision for passing colors, AutoSHADE places each entiry on a layer with the same name as the color it had in the original drawing. After reading the dxb files into AutoCAD (with the DXBIN) command, the colours can be reconstructed by using the AutoCAD LAYER command to assign the appropriate colors to the layers. This can be done with the AutoLISP program.

One use for the MAKE DXB feature is to create a drawing in AutoCAD that is to be plotted with perspective and clipping. First, create a 3-D drawing in AutoCAD, place the

lights, camera scenes, then write the film roll file. In AutoSHADE read the film role file. Select the desired scene and use the MAKE DXB item to produce the DXB file including the desired perspective and clipping; finally read the DXB file into a new drawing in AutoCAD with the DXBIN command and plot. The hidden wire removal can be done on the resulting drawing.

SHADING MODEL

The dialogue box helps to control the shading calculations used in rendering a drawing. The items in this box are:

Ambient factor : This controls the intensity of the background light between 0 and 1.0 are meaningful, increasing the value of ambient factor uniformly brightens all objects.

Diffuse factor : Controls the coefficient of reflection of all objects. Larger values of diffuse factor increase the amount of light reflected from each surface. The intensity of the diffusely reflected light depends on the value of diffuse factor.

Specular factor : To produce a matte surface set the specular factor to 0. Increase the value

assigned to the specular factor to increase the amount of light from each light source specularly reflected by the objects in the drawing.

Specular Exponent: This controls the width of the beam of light reflected by a shiny surface the more polished the surface, the narrower the beam of light. Settlings in the range from 3 to 20 work best.

Inverse Square : Assigning a non-zero value causes light intensity to drop off as the square of the distance from the light source to the centre of the face, behaving like a point source of light.

Linear Lighting: A non-zero causes the light intensity to decrease linearly with the distance from the light source to the centre of the face.

Inverse constant : Using this factor allows the pictures to look more realistic than the stark lighting generated by inverse square illumination. This factor combined with the linear lighting item allows

approximation of the illumination generated by extended sources such as fluorescent lights.

Background color: This box selects the background color for fast and full shaded pictures. Colors can be selected from a dialogue box or enter any AutoCAD # in the iamge 0 to 255.

Stretch contrast: If stretch contrast is OFF, 0 will represent minimum intensity, 1 represents maximum intensity. The shading model values should be chosen in such a way that the different shading factors (ambient, diffuse, specular) does not exceed 1.0.

If stretch contrast is ON, "automatic exposure" if contrast stretching will adjust it to the range your device can accomodate.

Z shading : When ON, objects closer to the camera appears brighter and those farther from the light source appear darker. If ON, all shading parameters, except stretch contrast, and all light placements and intensities are ignored.

Black & White : If ON, rendering is displayed as a grey scale intended for monochrome devices. But it allows previewing monochrome images on a color screen.

(Color) Components: Red component, Green component and blue component produces color separations. All are initially on, but if we have to make 3 transparencies that can be overlaid to produce a full color picture, we can select each component by itself, render and output to the device 3 times color combinations can be made for additive color (red, green, blue) as well as subtractive color (cyan, magenta and yellow) devices.

B&W operations : If we wish to make the separations as density mashs to control a later combination process, choose this factor and the separations will be rendered in the gray scale.

EXPERT

The items in the EXPERT MENU are used in only more advanced applications. If some bizarre renderings are created changes can be made to the items in the EXPERT menu. But these changes are temporary because they do not change the original values in the AutoCAD drawing or in the filmroll file.

Target X,Y,Z : This contains the target location co-ordinates. If target co-ordinates are changed, automatically DISTANCE item in the CAMERA POSITION will be computed.

Camera X,Y,Z : It contains the X,Y,Z co-ordinates of the cameral location. If you change these co-ordinates, DISTANCE will be recomputed.

Film diagonal: It specifies the diagonal size of the clipping rectangle in mm. It is used in perspective mode to compute the left clip, right clip, top clip and bottom clip settings. The default size is 42 mm, the diagonal size of 35 mm film.

mm/dwg unit : It defines the # of mm per drawing unit. Default value is 25.4. If it is not set correctly, the camera tabt AutoSHADE simulates will not be accurately sized. The film diagonal also should be large compared with the objects in the drawing.

Screen percent : This is the percent of the rendering screen that will be filled in each axis.

Sort round off : This defines the distance by which 2 faces must be separated before they are depth sorted.

Clip round off : This defines the distance by which 2 faces must be separated before they are depth sorted.

Chop round off: This defines the distance by which 2 planes must be separated before they are chopped when INTERSECTION is turned on.

Discard backfaces : Here, back faces will be discarded as the faces are projected to the camera's view point. The default is not to eliminate back faces.

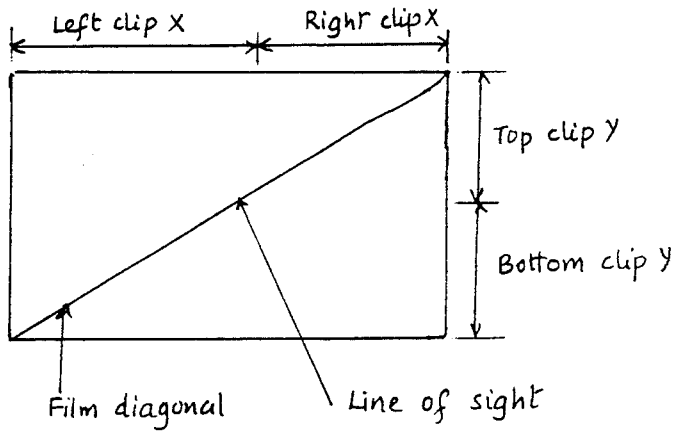
Back norm is neg : It allows us to control which faces are considered back faces.

Intersection : It instructs AutoSHADE to include the addition calculations necessary to check for intersecting faces when making a FULL SHADE picture. Thus, if any intersecting faces are found, they are divided into multiple triangles that do not intersect.

Perspective : When OFF, AutoSHADE will project the wire frame and renderings orthographically, using parallel projection. This is the way AutoCAD displays drawings. When ON,

objects are distorted such that those parts that are close to the cameral appear larger than the parts farther away.

Clipping



Left (Right) Clip: AutoSHADE will trim the wire frame or the rendered object at the specified left clip X or right clip X.

Left (Right) Clip X: These values determine the horizontal size of the viewing window in the view plane. This is not used by AutoSHADE until left clip or right clip is selected..

Top(Bottom) Clip : AutoSHADE will trim the wire frame or the rendered object at the specified top clip Y or Bottom Clip Y.

Top(Bottom) Clip Y: These values determine the vertical size of the viewing window. AutoSHADE uses them only when Top (Bottom) clip is selected.

Front Clip : AutoSHADE trims the picture at the specified from clip Z.

Front Clip Z: When front clip is selected, AutoSHADE trims the rendering at the front clipping

plane, perpendicular to the line of sight from the camera to the target.

Back Clip : AutoSHADE trims the back of the object at the specified Back Clip Z.

Back Clip Z : When back clip is selected, AutoSHADE trims the rendering at the back clipping plane, perpendicular to the line of sight from the cameral to the target. Back clipping is not needed unless you want to section the object.

Clip at camera : Here, AutoSHADE will automatically turn on Front clip and set front clip Z to the DISTANCE so that nothing behind the camera is visible.

STEREO PAIRS

Stereo images can be generated as 2 separate pictures or side by side in the same picture. Stereo pairs can be generated in fast or full shade and directed to either the rendering display or the rendering hardcopy device. When adjacent pairs are being generated, they can be presented either for stereoscope viewing or swapped left for right for crossed eye-viewing.

STATISTICS

It shows the # of faces and triangles the rendering diagram contains. And it provides the extents of the renderings.

FILE MENU

In this menu, there are OPEN item, script, information, Mandelbrot and Quit Items.

OPEN helps to specify a filmroll file to be loaded by AutoSHADE for rendering.

SCRIPT command reads AutoSHADE commands from an ASCII file and executes them. AutoSHADE automatically adds the .SCR extension to the file.

INFORMATION will display the AutoSHADE version #.

MANDELBROT helps to generage the maps of the Mandelbrot set.

QUIT helps to quit AutoSHADE and returns to the operating system prompt.

COMMAND LINE OPTIONS

* To reconfigure : SHADE - R
* To run a script file : SHADE - S <filename>
* To run a script file in batch mode : SHADE -B -S <filename>

* To print a summary of available command-line options :

SHADE - ?

EMERGENCY EXIT

Typing the characters CRASH, while holding down the ALT key helps in existing AutoSHADE.

EQUIPMENT NEEDED TO RUN AutoSHADE

* IBM PC
* 640 K RAM
* 8087/80287/80387 math co-processor
* Hard disk with 20 MB capacity
* Video display and adapter
* Pointing device.

It must be used with AutoCAD version 2.6 and higher.

4.ANIMATOR

4.1 INTRODUCTION

Animator is the most exciting new program for the IBM-Compatible computer market. Animator has many practical uses. Animator is a new graphics program from Autodesk for 8086, 80286 and 80386 based IBM compatibles with a VGA graphics card. It creates animated graphics by using a wide array of techniques and drawing tools that are near-professional quality and that approach some of the capabilities of mainframe computer graphics programs. And becuase of its relatively low cost, Animator enables every one to experiment with computer graphics and have fun expressing his or her own creativity.

Many of the commericals and graphics that you seen on television today are created by means of computers. Animator enables you to create graphic effects similar to the graphics prepared by commercial illustrators.

4.2 WHAT ANIMATOR CAN DO?

Animator has a tremendous amount of flexibility and you can incorporate a number of tools and techniques in any combination of ways to create effects that are limited only by your imagination. But basically, you can use Animator to create the following products:

* Hand drawn animated movies
* Animated or single-frame presentation graphics
* Cut-and-paste graphics and print image processing
* Video image processing and animation.

Animator enables you to crete or import full-screen, 320 x 200 resolution, 256-color VGA images that you can compile into movies, which can be played as fast as 60 frames per second. The entire palette of colors in Animator is 262,144 colors, but a standard VGA graphics card makes for a well-defined screen image even at the apparent low resolution of 320 x 200. The VGA image is more than sufficient for the Animator products that you will learn to create.

Hand-drawn animated movies are created a frame at a time, using Animator capabilities that are generally the same as a screen paint program such as Dr.Halo or EGA Paint. Animator provides what are called blue guids, which enable you to view the location of your hand drawn objects in teh adjoining

frames. This feature assists you in creating professional animation.

Animated or single-frame presentation graphics can aid you in clearly explaining a project, or in presenting graphs, charts, text information, and other graphics to enhance a nusiness or school presentation. You can create title frames to merge into a video presentation about your subject or company, and you can overlay text and other graphics with a videotape of "live" action.

Cut and paste graphics and image processing make it easy for you to do creative layouts from images that are captured from video cameras or scanners. Captured images can be found on many computer bulletin boards and information networks, such as CompuServe. Image processing amounts to taking a picture (or group of pictures), text, and graphic images - much as an advertising consultant would do - and creating images that bring out a point or idea.

Video processing and video animation are sophisticated activities that, at their most advanced level, require expensive equipment to accomplish.

4.3 EQUIPMENTS REQUIRED

* IBM-Compatible, 8086-based personal computer with a minimum 8 MHz clock speed

* PC DOS or MS-DOS 2.0 or later
* 640 K of contiguous main board memory. (All 640K is dedicated to the main memory board of your computer. Any extra memory that you have will be discussed later in the book).
* VGA display board and compatible monitor with 256 colors and 320 x 200 resoultion
* 10M (megabyte) hard disk
* Microsoft-compatible mouse or Summsketch compatible input device.

The additional equipments are

* Animator runs much faster on an 80386-based system such as an IM PS/2 Model 70.
* Animator can use extended memory in the form of a RAM disk.
* Animator reads a file from the hard disk. Some advanced movies can get quite large, so fast hard disk is a nice feature to have.
* A 16-bit VGA graphics card is faster than 8-bit VGA card. In addition, some VGA cards are considerably faster than others due to the type of graphic chips that are used and the amount of memory on the card.
* Animator will use 64K of expanded memory (EMS), if it is available.

4.4 STUDIO LOT

Animator consists of 2 parts

* The cursor, mouse and screen
* Menus and panels

4.4.1. Cursor, Mouse and Screen

The cursor consists of four white lines called crosshairs, with a center that is one pixel in size. The crosshairs actually contain an area that is 3 x 3 pixels but the active area of the cursor is only one pixel with a fine brush size.

At the top of the screen you seen the menu-bar, which contains the words Animator, Flic, Pic, Cel, Trace, Swap and Extra. Notice that if you place the cursor over one of the words, a pull-down menu appears.

At the bottom of the screen is Home panel.

Move the cursor into the black screen and click the right button. The whole screen goes blank. Click the right button again, and the menu bar and Home panel return.

4.4.2. Menus and Panels

All the animator commands and tools are available through the menu bar and Home panel. The menu bar and Home panel lead to other menus and panels, and it makes a difference whether you left-click or right-click many of the items displayed in the panels.

Move the cursor to the Animator option in the manu bar. Select About Animator from the pull-down menu that is displayed.

Animator's menus contain three types of options, which are the following

* Options that cause an action to occur when selected. The About Animator selection is an example of this type of action.
* Options that can be turned on or off, which are also called toggles. An asterisk appears next to an option that has been turned on. If you try to carry out an option that is toggled off, nothing happens.
* Options that lead to more options without causing any command to be executed or any effect to be applied. These options are the options on the Animator pull-down menu between the dashed lines and any menu items followed by an ellipsis (...)

Animator's panels have four types of selections, which are the following:

* Slots. An example of a slot is the one that has the word DRAW in it down at the bottom left of your screen.
* Buttons. A button is a small square that turns white when it is turned on, and returns to the panel color when it is off. An example of a button is the small square

containing the letter K in the lower right corner of your screen.

* **Icons.** An icon is a symbol that can activate another panel or execute a particular command. Examples of icons are the arrows in the lower center of your screen.
* **Slider Bars.** A slider bar enables you to adjust or select options by left-clicking the slider or the slider arrow, or by dragging an icon within the slider bar. Slider bars are of two types: a numeric slider bar, which is horizontal, and a window slider bar, which is vertical and has a selection window to its right.

The main Animator panel is called the Home panel, which has several different types of selections. The Home panel is made up of four main areas along with some miscellaneous functions.

The Home panel provides access to drawing tools, the palette of colors, the frames of the movie and types of ink.

The first section of the Home panel, the Drawing Tools section, contains a series of slots or boxes. The default configuration slots contain a graued slot with the word Home in it, plus slots with Zoom, Undo, Draw, Poly, Spray, Box, Text, and Fill in them. Each slot is a drawing command.

The next section of the Home panel is the Palette and Frame control area immediately to the right of the drawing tools. This area gives you access to any frame or group of frames in the Flic you are working on. You also use this area to select a color to draw with, and it gives you access to the color palette.

In the Palette and Frame Control section, you see a single black box in the upper left part of the section, then a series of seven different color boxes in the upper right part of the section. You use these boxes to pick a color to draw with that may or may not be included in the color bar that you see directly below them. The upper boxes are known as the Mini-palette, and the color bar in the middle is known as the Current Color-cluster.

The color cluster in the middle of the Palette and Frame Control section enables you to select a color from the entire group of colors displayed in the cluster.

The lower part of the Palette and Frame control section contains a set of Frame icons. The far-left arrow (an up arrows) sets the current movie frame to the start of the movie. The next arrow (a left arrow) sets the current frame back one frame. The center icon shows which frame is the current frame.

The next arrow (a right arrow) advances the movie one frame. THe double right arrows play the movie from the first frame to the last. Finally, the down arrow sets the current frame to the last frame of the movie. At this point, all these arrows affect only one frame.

The next section to the right, the Ink Types area, enables you to select an ink type from the six ink types that are displayed. The default configuration displayes the ink types Opaque, Glass, Soften, Vgrad, Scrape and Tile.

The final area, the Mode section, has a little dot in the upper left corner. This dot is an icon that indicates your current brush size, which can be from 1 to 11 pixels wide. The brush size can be switched between two values by left-clicking the brush icon. The slider bar appears.

To the right of the brush icon the Mode section is a box that shows the color you currently are using.

Below the brush icon and the current color box are four buttons. These buttons represent modes that you can use while creating a movie or screen image. The following are the mode buttons.

F. Filled: Drawing tools that can use this option completely fill a shape with ink when this button is turned on.

T.Time: Certain actions take place over a range of frames when this mode is turned on.

M.Mask: This mode enables you to create and use a mask that excludes the masked areas of the screen from the application of ink.

K. Clear Key Color: This mode, when turned on, makes the background color of a frame clear, ie. celluloid. If the button is turned off, the background color becomes opaque.

4.5 Key Players

Animator is divided into three departments

* Art Department
* Production Department
* Film making Department

The Art Department:

The Art department consists of the Drawing Tools, the Color Paletter, and the Ink Types. The Art department's responsibility is to create the images that are used in the movie, place them on the right frame, and to make sure that the artwork is suitable for the desired effect. The Art department has 22 drawing tools, 256 colors (at any one time) and 26 ink types, all of which can be combined in a variety of ways, and many of which have adjustable size, intensity and types of application technique.

The Art department is found within the Home panel. The Art department also must work closely iwth Filmmaking to d

animation, especially when it comes to creating cels, by t most of the work done by th Art department is with drawing tools, the color palette, and ink types. You also can get to the Art deaprtment options via the Animator menu bar selection. Place the cursor on Animator in the menu bar and you see Frames, Palette, Drawing Tools and Ink Types.

The Production Department

The Production department is responsible for manipulating the frames that the Art department creates and provides the special effects that the Art department often needs. The Production department spends a great deal of time working with the Frames plane which is accesible from the Home panel or the Animator menu. The Production department also works extensively with the Optics option on the Animator menu. Put the cursor on Animator in the menu bar and then left-click optics in the Animator menu.

The optics screen is the special effects group. Instead of a different panel, or a pull-down menu there is a whole screen of options to work with. Put the cursor over each of the menu bar options (Presets, Movement and Element). The Production department also is responsible for the text or titling of your movies. The Production department has the capability to display text in a number of fonts and in a variety of ways. Titling is done by using the Text option from the Drawing Tools slots, and then using the Titling option

from the Animator menu to create special effects. Select Titling from the Animator menu. Notice that the Titling panel contains the option New Text, which serves the same purpose as the Text slot in the Drawing Tools section of the Home panle.

The Filmmaking Department

The Filmmaking Department offers a broad base of services, including editing, management of the fil library, and control of animation. Many special image manipulation techniques are available, as are film-merging capabilities such as creating dissolves, fades, wipes and a host of others. The Filmmaking Department is found in the Flic, Pic, Cel, Trace, Swap and Extra Pull-down menus. Many times the Filmmaking department works in unisor wiht the Art and Production departments to crate the desired effect. Certain features of the Filmmaking Department, such as masking, applying ink in special waus, and animation creation may seem to rely more on the Art deaprment than on the Filmmaking Department. THe distinction is that the Art department is responsible for single frame creations, and multiple frame work is managed by the Production department, under the direction of the Filmmaking Department.

On the left side of the Browse Flic screen is a vertical slider bar. To display more flics, you can select the up-arrow at the top of the slider bar. The displayed flics move yp one line for each click. Rapidly left-click twice a

frame that looks interesting to you. This action loads a flic. After the flic has loaded you are returned to the Home panel. TO play the flic, select the double right arrows from the Control section of the Home panel. Of you want to stop the flic, right click.

Objects of Interest

INTRODUCTION

By using AutoLISP programming AutoCAD can be customised. AutoLISP is a direct pipeline to AutoCAD, with it there is access to AutoCAD drawing entities, reference table and to pass files in and out of AutoCAD.

Points, distances and other values can be stored, calculated, compared and used to draw with.

The drawing environment can be controlled through AutoCAD system variable by storing the system variables prompting with drawing status and changing and restored settings.

The system settings can be changed transparently during commands. Entity data can be extracted and accessed, data in the programs can be used and entities modified, all transparently.

OBJECTS OF INTEREST

EPASA helps to create a particular house. The dimensions of the house and the details of the walls, doors, windows, ventilation etc. are to be entered as input in the AutoLISP program. And the program after execution will give as output the three-dimentionsl drawing, plan and elevation of the house.

* Rendered images shown in colour on a good display screen are absolutely impressive.
* After using AutoCAD, the quality of the drawing is increased.
* 3D rendered drawing can show errors and interferences that are not seen in wireframe representations.
* AutoSHADE produces both rendered and wireframe images in perspective.

30 drawings for AutoSHADE

AutoCAD displays 3D images as wireframes. AutoSHADE ignores all entities without surfaces or thicknesses.

It only recognizes -

* Circles, solids, traces, donuts and wide polylines
* 3D faces and 3D polylines, meshes
* Extruded entities.

CONCLUSION

EPASA has been able to create a house in a drawing using AutoCAD and AutoLISP. A three-dimensional view of the house can be drawn by executing the AutoLISP program.

To make the house to appear realistic, AutoSHADE is used in EPASA and different shadings of the house at different views is obtained.

AutoSHADE actually will work well with curved objects, so a vase is drawn and it is shaded and viewed.

By using ANIMATOR, the drawings created are animated in EPASA.

REFERENCES

1. D.RAKER AND H.RICE - "INSIDE AutoCAD" (Release 10) - NEW RIDRES/ MICRO-TECH ASIAN EDITION.
2. DR.TOM W.BERGHAUSER AND DR.PAUL L.SCHLIEVE - "ILLUSTRATED AutoCAD (RELEASE 10)" - BPB PUBLICATIONS.
3. GEORGE O.HEAD - "AutoLISP IN PLAIN ENGLISH" - GALGOTIA PUBLICATIONS PVT. LTD (1990).
4. "AutoSHADE VERSION 1.0 - USER GUIDE (MANUAL)
5. CRAIG SHARP - "USING ANIMATOR" - QUE CORPORATION

APPENDIX A

Autocade Command Reference

Alphabetical list of AutoCAD'S commands are given below with brief descriptions and a summary of options. The commands flagged with a prefix in this list can be transparently.

Aperture

Controls the size of the object snap target box.

Arc

Draws an arc of any size. The default method is by specifiying two end points and a point along the arc, but several other methods are available.

Options

A - included angle

C - centre point

O - starting direction

E - End point

L - Length of chord

R - Radius

RETURN - sets start point and direction as end of last line or arc.

ATTDISP

DESCRIPTION

Controls the visibility of Attribute entiries on a global basis.

OPTIONS

ON : Make all Attributes visible

OFF: Make all Attributes invisble

N : Normal visibility set individually

ATTEDIT

DESCRIPTION

Permits editing of Attributes

ATTEXT

DESCRIPTION

Extracts attribute data from a drawing.

OPTIONS

C : CDF comma delimited format extract

D : DXF format extract

S : SDF format extract

E : Extract selected objects only.

AXIS

DESCRIPTION

Displays a "ruler line" on the graphics monitor.

OPTIONS

ON : Turn axis (ruler line) on

OFF : Turn axis off

S : Lock tick spacing to snap resolution

A(+2) : Set aspect (differing X-Y spacings)

number : Set tick spacing (0 = use snap spacing)

number X : Set spacing to multiple of snap spacing.

BASE

DESCRIPTION

Specifies origin for subsequent insertion into another drawing.

BLIP MODE

DESCRIPTION

Controls display of marker blips for point selection.

OPTIONS

ON : Enable temporary marker blips

OFF: Disable temporary marker blips

BLOCK

DESCRIPTION

Forms a compound object from a group of entities.

OPTIONS

? : List names of defined blocks

BREAK

DESCRIPTION

Erases part of an object, or splits in into two objects.

OPTIONS

F: Re-specify first point.

CHAMFER

DESCRIPTION

Creates a chamfer at the insertion of two lines

OPTIONS

D : Set chamfer distances

P(+3): Chamfer an entire polyline

CHANGE

DESCRIPTION

Alters the location, size, orientation or other properties of selected objects. Especially useful for TEXT entities.

OPTIONS

P	Change common properties of objects
C	Color
E(+3)	Elevation
LA	Layer
LT	Linetype
T(+3)	Thickness

CIRCLE

DESCRIPTION

Draws a circle of a y size. The default method is by centre point and radius but other methods are available

OPTIONS

2P	Specify by 2 end points of diameter
3P	Specify by 3 points on circumference
D	To enter diameter instead of radius
TTR(+2)	Specify by two tangent points and radius

COLOR

DESCRIPTION

Establishes the color for subsequently drawn objects.

OPTIONS

Number	Set entity color number
name	Set entity color to standard color name
BYBLOCK	Set "floating" entity color
BYLAYER	Use layer's color for entities

COPY

DESCRIPTION

Draws a copy of selected objects

OPTIONS

M	Make multiple copies of the selected objects

DBLIST

DESCRIPTION

Lists database information for every entity in the drawing.

DDATTE

DESCRIPTION

Allows Attribute editing by means of a dialogue box.

IDDEMODES

DESCRIPTION

Sets current layer, color, linetype, elevation and thickness via dialogue box.

DDL MODES

Sets drawing aids via dialogue box

DELAY

Delays executioin of the next command for a specified time used with command scripts.

DIM

Invokes dimensioning mode, permitting many dimension notation to be added to a drawing.

DIM1

Allows one dimension notation to be added to a drawing, then returns to normal command mode.

DIST

Finds distance between two points.

DIVIDE

Places workers along a selected object, dividing it into a specified number of equal parts.

DOUGHNUT OR DONUT

Draws rings which specifies inside and outside diameters.

DRAW MODE

Allows control of the dynamic speficiation feature for all appropriate commands.

ON - Honar "DRAG" requests when applicable

OFF - Ignore "DRAG" requests

A - Set "Auto" mode, drag whenever possible

D TEXT

Draws text items dynamically. Here options are same as for TEXT command.

DXBIN

Inserts specially coded binary files into a drawing.

DEFIN

Loads a drawing interchange file

DEFAULT

Writes a drawing interchange file

E - Output selected entities only.

ELEV

Sets elevation and extrusion thickness for subsequantly drawn entities used in 3D visualizations.

ELLIPSE

Draws ellipses using any of several specifications

C - specify centre point rather than first axis end point

R - specify eccentricity via rotation rather than second axis.

I - Draw isometric circle in current ISOPLANE

END

Exists the Drawing Editor after saving the updated drawing.

ERASE

Erases entities from the drawing

EXPLODE

Shatters a block or polyline into tis constituent parts.

EXTEND

Lengthness a line arc or polyline to meet another object/

FILES

Performs disk file utility tasks.

FILL

Controls whether solids, trace and wide polylines are automatically filled on the screen and the plot output.

ON - Solids, traces and wide polylines filled.

OFF - Solids, Traces and wide polylines outlined.

FILLET

Constructs an arc of specified radius between two lines, arcs or circles

P(+3) - fillet an entire polyline

R - Set fillet radius

FILM ROLL

a file for rendering by AutoSHADE

GRAPHSCR

Flips to two graphics display on a single-screen systems used in Command scripts and menus.

GRID

Displays a grid of dots, at desired spacing on the screen.

ON - Turn grid on

OFF - Turn grid off

S - Lock grid spacing to snap resolution

A(+2) - Set grid aspect

Number - Set grid spacing

Number X - Set spacing to multiple snap spacing.

HATCH

Performs cross-hair hatching and pattern filling.

I - Ignore internal structure

N - Normal style : turn hatch lines off and on as internal structure is encountered.

O - Hatch outermost position only.

HELP OR ?

Displays a list of valid commands and data entry options or obtains help for a specific command.

HIDE

Regenerates a 3D visualization with "hidden" lines removed.

IDE

Displays the coordinates of a specified point.

IGESIN

Loads an IGES interchange file.

IGESOUT

Writes an IGES interchange file

INSERT

Inserts a copy of a previously drawn part (object) into the current drawing.

Name	Load file "name" as a Block
Name = f	Create Block "name" from file "f"
* Name	Retain individual part entities.
?	List names of defined Blocks
C	(as reply to X scale prompt)
	specifies scale via two points
	(corner specification of scale)
XYZ (+3)	(as reply to X Scale prompt)
	radius INSERT for X,Y and Z scales.

ISOPLANE

Select the plane of an isometric grid to be the current plane for orthogonal drawing.

L	Left plane
R	Right plane
T	Top plane
Return	Toggle to next plane

LAYER

Creates named drawing layers and assigns color and lnie types properties to those layers

Cc Set specified layers to color "c"

Fa,b(+3) Freeze layers "a" and "b".

Lt Set specified layers to line type "t".

Ma Make "a" the current layer creating it if necessary.

Na,b Create new layers "a" and "b"

ON a,b Turn on layers "a" and "b"

OFF a,b Turn off layers "a" and "b"

Sa Set current layer to existing layer "a"

T a,b(+3) layers "a" and "b"

? List layers and their associated colors and line types.

LIMITS

Changes the drawing boundaries and controls checking of those boundaries.

2 points set lower left/upper right drawing limits

ON Enable limits checking

OFF Disable limits checking

LINE

Draws straight lines of any length

RETURN (as reply to "FROM POINT") start at end of previous line or Arc.

C (as reply to "TO POINT") close polygon

U (as reply to "TO POINT") undo segment

LINE TYPE

Defines linetypes (sequences of alternating line segments and spaces), loads them from libraries, and sets the line type for subsequently drawn objects.

?	List a line type library
C	Create a linetype defintion
L	Load a linetype definition
S	Set current entity linetype

"Set" suboptions:

name	Set entity linetype name
BYBLOCK	Set "floating" entity linetype.
BY LAYER	Use layer's linetype for entities
?	List loaded linetypes

LIST

Lists database information for selected objects

LOAD

Loads a file of user-defined shapes to be used with the SHAPE command

? List the names of loaded shape files

LTSCALE

Sets scale factor to be applied to all linetypes withing the drawing.

MEASURE

Places markers at specified intervals along a selected object.

MENU

Loads a file of Drawing Editor Commands into the menu areas (screen, pull-down, tablet and button).

MINSERT

Inserts multiple copies of a block in a rectangular pattern.

name	Load file "name" and form a rectangular array of the resulting Block.
name-f	Create Block "name" from file "f" and form a rectangular array.
?	List names of defined blocks
C	(as reply to X scale prompt) specifies scale via two points (concenrn specification of scale).
XYZ(+3)	(as reply to X scale prompt) readies MINSERT for X,Y and Z sclaes.

MIRROR

Reflects designated entities about a user specified axis.

MOVE

Moves a designated entities to another location.

MSLIDE

Makes a slide file from the current display.

MULTIPLE

Causes the next command to repeat until cancelled.

OFFSET

Allows the creation of offset curves and parallel lines

number specifies offset distance

T Through : allows specification of a point through which the offset curve is to pass.

OOPS

Restores erased entities

ORTHO

Constrains drawing so that only lines aligned with the grid can be entered.

ON Force lines to horizontal or vertical

OFF Do not constrain lines

OSNAP

Enables points to be preceisely located on reference points of existing objects.

CENT	Center of Arc or circle
END P	Closest endpoint of Arc or line
INSERT	Intersection point of TEXT/BLOCK/Shape
INTER	Intersection of line/Arc/Circle
MDP	Mid point of Arc or Line
NEAR	Nearest point of Arc/Circle/Line/Point
NODE	Node (point)
NONE	None (off)
PERP	Perpendicular to Arc/line/Circle
QUAD	Quadrant point of Arc or Circle
QUICK	Quick Mode (first fuid, not closest)
TANG	Tangent to Arc or Circle
PAN	Moves the display window

PEDIT

Permits editing of polylines

C	Close an open polyline
D	Decurve polyline
E	Edit Vertex (See below for supoptions)
F	Fit curve to polyline
J	Joint to polyline
O	Open a closed polyline

U	Undo one editing operation
W	Set uniform width for polyline
X	Exit PEDIT command

During Vertex editing

B	Set first vertex for Break
G	Go (perform Break or straighten operation)
I	Insert new vertex after one
M	Move current vertex
N	Make next vertex current
P	Make previous vertex current
R	Regenerate the polyline
S	Set first vertex for straighten
T	Set tangent direction for current vertex
W	Set new width for following segment
X	Edit vertex editing, or cancel Break/Straighten

PLINE

Draws connnected line and arc segments with optional width and taper.

H	Set new half-width
U	Undo previous segment
W	Set new line width
RETURN	Exit PLINE command

In line Mode

A	Switch to arc mode
C	Close with straight segment
L	Segment length (continue previous segment)

In arc mode

A	Included angle
CE	Centre point
CL	Close with arc segment
D	starting direction
L	Chord length or switch to line mode
R	Radius
S	Second point of three-point arc

PLOT

Plots a drawing on a pen plotter

POINT

Draws single points

POLYGON

Draws regular polygons with the specified number of side.

E	Specify polygon by showing one edge
C	Circumscribe around circle
I	Inscribe within circle

PRPLOT

Plots a drawing on a printer plotter.

PURGE

Removes unused blocks, text styles, layers or line types from the drawing.

A	Purge all unused named objects
B	Purge ununsed Blocks
LA	Purge unused layers
LT	Purge unused linetypes
SH	Purge unused shape files
ST	Purge unused Text Styles

QTEXT

Enables Text entities to be identified without drawing the text detail

ON	Quick text mode on
OFF	Quick text mode off

QUIT

Exits the Drawing Editor and returns to AutoCAD's main menu discarding any changes to the drawing.

REDEFINE

Restores a built-in command deleted by UNDEFINE

REDO

Reverses the previous command if it was U or UNDO

REDRAW

Refreshes or cleans up the display

REGENAUTO

Controls automatic regeneration performed by other commands.

ON	Allow automatic regens
OFF	Prevent automatic regens.

RENAME

Changes the names associated with text styles, named views, layers, line types and Blocks

B	Rename Block
LA	Rename Layers
LT	Rename Line type
S	Rename Text Style
V	Rename named view

RESUME

Resumes an interrupted command script

ROTATE

Rotates existing objects

R	Rotate with respect to reference angle

RSCRIPT

Restarts a command script from the beginning.

SAVE

Updates the current drawing file without existing the Drawing editor

SCALE

Alters the size of existing objects

R Resize with respect to reference size

SCRIPT

Executes a command script

SELECT

Groups objects into a selection-set for use in subsequent commands.

SETVAR

Allows you to display or change the value of system variables

SH

Allows access to internal PC-DOS/MS-DOS commands.

SHAPE

Draws pre-defined shapes

? Lists available shape names

SHELL

Allows access to other programs while running AutoCAD.

SWITCH

Permits free-hand sketching

C	Connect : restart sketch at end point
E	Erase (backup over) temporary lines
P	Raise/lower sketching pen
Q	Discard temporary lines, remain in sketch
R	Reward temporary lines, remain in sketch
X	Reward temporary lines, exit sketch
	Draw line to current point

SNAP

Specifies a "round off" interval for digitizes point entry so entities can be placed at precise locations easily.

number	set snap alignment resolution
ON	Align designated points
OFF	Do not align designated points
A(+2)	Set aspect (differing X-Y spacings)
R(+2)	Rotate snap grid
S(+2)	Select style, standard or isometric

SOLID

Draws filled-on polygons

STATUS

Displays drawings statistics & modes

STRETCH

Allows you to move a portion of drawing while retaining connections to other parts of the drawing.

STYLE

Creates named text styles, with user-selected combinations of font, mirroring, obliquing and horizontal scaling.

? List currently defined text styles

TABLET

Aligns the digitizing tablet with coordinates of a paper drawing to accurately copy it with AutoCAD.

ON Turn Tablet Mode on

OFF Turn tablet maode off

CAL Calibrate tablet

CFG Configure tablet menus, pointing area

TEXT

Draws text characters of any size, with selected styles

A Align text between two points, with style specified width factor, AutoCAD computes appropriate height.

C Centre text horizontally

F Fit text between two points, with specified height; AutoCAD computes an appropriate width factor

M Center tex

t horizontally and vertically

R Right justify text

S Select texy style

TEXTSCR

Flips to the text display on single-screen systems used in command scripts and menus.

TIME

Displays drawing creation and update times, permits control of an elapsed timer

D Display current timer

ON Start user elapsed timer

OFF Stop user elapsed timer

R Reset user elapsed timer

TRACE

Draws solid lines of specified width

TRIM

Erases the portions of selected entities that cross a specified boundary.

U Reverses the effect of the previous command

UNDEFINE

Deletes the definition of a built-in AutoCAD command.

UNDO

Reverses the effect of multiple commands, and provides control over the "undo" facility.

Number undoes the number most recent commands

A Auto Controls treatment of menu items as undo groups

B Back Undoes back to previojs undo mark

C Control Enables/disables the undo feature

E End Terminates an UNDO group

G Group begins sequence to be treated as one command

M Mark places marker in Undo file (for Back)

UNITS

Selects coordinate and angle display formats and precision.

VIEW

Saves the current graphics display as a Named view, or restors a saved view to the display.

D Delete name view

R Restor named view to screen

S Save current display as named view

W Save specified window as named view

? List named views

VIEWERS

Allows you to control the precision and speed of circle and Arc drawn on the monitor by specifying the number of slides in a circle.

VPOINT

Selects the viewpoint for a 3D visualization

R Select view point via two roation angles

RETURN Select view point via compass & axes tripped x,y,z specifies view point

VSLIDE

Displays a previously -created slide file

file view slide

* file Preload slide, next VSLIDE will view

WBLOCK

Writes selected entities to a disk file

name write specified Block Definition

= Block name same as file name

* Write entire drawing

RETURN Write selected objects

ZOOM

Enlarges or reduces the display of the drawing.

number Multiplier from orginal scale

numberX Multplier from current scale

A	ALL
C	Center
D(+3)	Dynamic PAN/ZOOM
E	Extents ("drawing uses")
L	Lower left corner
P	Precious
W	Window

3D FACE

Draws three-dimensional plane sections.

3D LINE

Draws fully general three dimensional lines

RETURN (as reply to "FORM POINT") start at end of previous line or Arc

C (as reply to "TO POINT") close polygon

U (as reply to "TO POINT") undo segment

```
(defun c:house() (load "thanl"))
(prompt "Please wait\n  LOADING ...")
(load "thanl")


(vmon)
(defun chk () (setvar "lunits" 4))
(defun dtr (a)(* (/ pi 180) a))
(defun lp () (getvar "lastpoint"))
(defun plo (a d) (polar (lp) (dtr a) d))
(defun mderr (s)

(if (/= s "Function cancelled")
    (progn (princ (strcat "  nError: " s))
    (princ)))
 (if (/= s "Console break")
    (progn (princ (strcat "  nError: " s))
    (princ))))
    (chk)
    (setq *error* mderr)
    (setvar "highlight" 0)
            (setvar "cmdecho" 0)
            (setvar "ucsicon" 0)
           (setvar "blipmode" 0)
         (prompt " \n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\
         (textscr)

     (PROMPT " \n                    -------HOUSE--------")
     (PROMPT " \n          ALL DIMESIONS ARE IN Feet AND Inchs ")
     (PROMPT "\n\n            MAXIMUM LENGTH OF THE HOUSE IS 55 Feets ")
     (PROMPT "\n\n            MINIMUM LENGTH OF THE HOUSE IS 35 Feets ")
     (PROMPT "\n\n            MAXIMUM WIDTH OF THE HOUSE IS  30 Feets ")
     (PROMPT "\n\n            MINIMUM WIDTH OF THE HOUSE IS 25 Feets  \n \n")

     (prompt " \n                     PLEASE ENTER.... ")
     (prompt " \n\n          HOUSE LENGTH 420 INCHS TO 660  INCHS")
     (prompt "\n\n           HOUSE WIDTH  300 INCHS TO 360  INCHS")
        (initget (+  2 4))
     (setq len1
        (getdist " \n\n         ENTER THE LENGTH OF THE  BUILDING<540>:"))
     (if (= len1 nil) (setq len1 540))
         (initget (+  2 4))
     (setq len2
        (getdist "\n\n          ENTER THE WIDTH OF THE BUILDING<324>:"))
     (if (= len2 nil) (setq len2 324))
        (initget (+  2 4))
     (setq walthic
        (GETDIST " \n\n         ENTER THE  WALLTHICKNESS <7.5>:"))
     (if (= walthic nil) (setq walthic 7.5))
        (initget (+  2 4))
     (setq walhgt
        (getdist " \n\n         ENTER THE WALLHEIGHT <120>:"))
            (if (= walhgt nil) (setq walhgt 120))
    (while (or (or (< len1 420)
```

```
                (> len1 660)) (or (< len2 300) (> len2 360)))
          (princ "  007")(princ "  007")
          (prompt " \n      PLEASE ENTER.... ")
         (prompt " \n      HOUSE LENGTH 420 INCHS TO 660  INCHS")
          (prompt " \n \n     HOUSE WIDTH  300 INCHS TO 360  INCHS")
          (initget (+  2 4))
 (setq len1
      (getdist " \n \n      ENTER THE LENGTH OF THE  BUILDING<540>:"))
         (if (= len1 nil) (setq len1 540))
          (initget (+  2 4))
 (setq len2
      (getdist " \n \n      ENTER THE WIDTH OF THE BUILDING<324>:"))
         (if (= len2 nil) (setq len2 324))
      )


(setq
      open1 48
      open2 60
     win1ev 24
     toilev 70
      toilen 60
      toiwid 108
     win1hgt 51
     win1br  5
      win11en 96
      win2hgt 51
     win2br  5
      win21en 42
      win3hgt 51
     win3br  5
      win31en 54
      door1hgt 84
     door1br  5
      door11en 42
     door3hgt 84
     door3br  4
      door31en 27
     door2hgt 84
     door2br  5
     door21en 36
     ven1hgt 24
     ven1br  5
      ven11en 30
      lirolen (- (/ len1 2) 30)
      kit1    (+ (* len1 0.22) (* walthic 0.5))
      kitb    (* len2 0.5)
     dinlen  (*  len1 0.31)
     stolen  (abs (* len1 0.13))
     wid     (abs (- (/ len2 2) (* walthic 0.5)))
     bd1     (* len1 0.4)
      dinlen1 (+ (* len1 0.44) (* walthic 0.5))
      bdb     (- kitb (* 0.5 walthic))
     livhgt1 (abs (- (* wid 0.6) walthic))
```

```
        livhgt2 (+ livhgt1 open2)
         livhgt3 (- wid (* walthic 0.5))
         livlen1 (+ (* len1 0.6) (* walthic 0.5))
         toimid  (* len1 0.595)
)

(prompt " \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n")
(prompt "            ------------- CURRENT DIMENSIONS ------------")
(prompt " \n \n                      ALL DIMENSIONS ARE IN INCHS")
(PROMPT " \n \n           LENTH OF THE HOUSE :")
(PRINC LEN1)
(PROMPT " \n \n           WIDTH OF THE HOUSE :")
(PRINC LEN2)
(PROMPT " \n \n           WALTHICKNESS       :")
(PRINC WALTHIC)
(PROMPT " \n \n           WALHEIGHT          :")
(PRINC WALHGT)
(PROMPT " \n \n           ------------------------------------------------")
(PROMPT " \n \n \n \n \n \n")(princ "  007")
   (setq try (getstring "Press RETURN to continue.....,"))

 (prompt " \n \n -------------CURRENT DIMENSIONS-------------------")
 (PROMPT " \n \n            ALL DIMENSIONS ARE IN INCHS              ")
 (PROMPT " \n \n            LENGTH         WIDTH           HEIGHT")
 (PROMPT " \n \n DOOR1       ")
 (PRINC  DOOR1LEN)
 (PROMPT "              ")
 (PRINC  DOOR1BR)
 (PROMPT "              ")
 (PRINC  DOOR1HGT)
 (PROMPT " \n \n DOOR2       ")
 (PRINC  DOOR2LEN)
 (PROMPT "              ")
 (PRINC  DOOR2BR)
 (PROMPT "              ")
 (PRINC  DOOR2HGT)
 (PROMPT " \n \n DOOR3       ")
 (PRINC  DOOR3LEN)
 (PROMPT "              ")
 (PRINC  DOOR3BR)
 (PROMPT "              ")
 (PRINC  DOOR3HGT)
 (PROMPT " \n \n WINDOW1     ")
 (PRINC  WIN1LEN)
 (PROMPT "              ")
 (PRINC  WIN1BR)
 (PROMPT "              ")
 (PRINC  WIN1HGT)
 (PROMPT " \n \n WINDOW2     ")
 (PRINC  WIN2LEN)
 (PROMPT "              ")
 (PRINC  WIN2BR)
 (PROMPT "              ")
 (PRINC  WIN2HGT)
```

```
(PROMPT " \n \n WINDOW3     ")
(PRINC  WIN3LEN)
(PROMPT "                ")
(PRINC  WIN3BR)
(PROMPT "                ")
(PRINC  WIN3HGT)
(PROMPT " \n \n VEN          ")
(PRINC  VEN1LEN)
(PROMPT "                ")
(PRINC  VEN1BR)
(PROMPT "                ")
(PRINC  VEN1HGT)
(prompt " \n ------------------------------------------")
(PROMPT " \n          OPEN PLACE ONE LENGTH :")
(PRINC  OPEN1)
(PROMPT " \n          OPEN PLACE TWO LENGTH :")
(PRINC  OPEN2)
(PROMPT " \n")
(setq try (getstring "                Press RETURN to continue.....,"))
       (INITGET "Y N y n")
(SETQ CHO
      (GETKWORD " \n \n \n YOU WANT TO CHANGE ABOVE DIMENSIONS(Y/N)<N> ")
  (IF (or (= CHO "Y") (= cho "y"))
      (PROGN
(PROMPT " \n \n \n \n")
      (load "than3")
        )
        )
(prompt " \n")
(setq sel nil)
        (while  (= sel nil)
(prompt " \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n \n")
(prompt " \n \n \n          --------ENTER YOUR CHOICE----------")
(PROMPT " \n \n \n                  1.  PLAN VIEW      ")
(PROMPT " \n \n \n                  2.  FRONT VIEW    ")
(PROMPT " \n \n \n                  3.  3 DIMENSION  ")
(PROMPT " \n \n \n          ----------------------------------")
      (INITGET "1 2 3")
(PROMPT " \n \n \n \n \n \n")
(SETQ  SEL (GETKWORD " \n                ENTER YOUR SELECTION :"))
   (IF (= SEL "1") (LOAD "NALP"))
   (IF (= SEL "2") (LOAD "than4"))
   (IF (= SEL "3") (LOAD "than2"))
   (if (= sel nil)
        (progn (princ "  007")(princ "  007")
        (setq try (GETINT " \n INVALID SELECTION....
                          \n PRESS RETURN TO CONTINUE...,"))
                                        )))
            (setvar "highlight" 1)
            (setvar "cmdecho" 1)
            (setvar "ucsicon" 1)
           (setvar "blipmode" 1)
            (graphscr)
```

```
(command "plan" "")
(SETVAR "EXPERT" 128)
(SETVAR "Elevation" 0)
(SETQ
        dinthgt 24
        charhgt 24

         outl1 (list 0 0)
         outl2 (list len1 0)
        outl3 (list len1 len2)
        outl4 (list 0 len2)

         outh1 (list  (car outl1)   (cadr outl1)  walhgt)
         outh2 (list  (car outl2)   (cadr outl2)  walhgt)
         outh3 (list  (car outl3)  (cadr outl3) walhgt)
         outh4 (list  (car outl4)  (cadr outl4) walhgt)


         stohgt1 (list stolen  walthic)
         stohgt2 (list (car stohgt1) (abs (+ wid (* walthic 0.5))))
         sp2     (list (car stohgt2) (- (cadr stohgt2) walthic))
         stohgta (list stolen  (abs (- (cadr sp2) door1len)))
         stohgt3 (list  walthic (cadr stohgt2))
         sp1     (list (car stohgt3) (- (cadr stohgt3) walthic))
         sp3     (list (- (car stohgt1) walthic) (cadr stohgt1))
        spa     (list (car sp3) (cadr stohgta))

          bedca  (list kit1 (- len2 walthic))
         bedcb  (list  (car bedca)  (+ wid (* walthic 0.5)))
         bedc1  (list  (- (car bedca) walthic) (cadr bedca))
         bedc2  (list (car bedc1) (abs (- wid (* walthic 0.5))))

         spkit1  (abs (- (distance stohgt2 bedc2) door2len))
         kitlen1 (list (+ (car stohgt2) spkit1) (cadr stohgt2))
          kitlen2 (list (+ (car sp2) spkit1) (cadr sp2))


        din11   (list dinlen1 walthic)
         din12   (list dinlen1 (abs
                 (- (abs (- wid (* walthic 0.5))) open1)))
         din13   (list (- (car din12) walthic) (cadr din12))
         din14   (list (car din13) walthic)

         liv11   (list liv1en1 walthic)
         liv12   (list liv1en1 livhgt1)
        liv13   (list (- (car liv12) walthic) (cadr liv12))
         liv14   (list (car liv13) (cadr liv11))
         liv15   (list liv1en1 livhgt2)
         liv16   (list liv1en1  livhgt3 )
        liv17   (list (- len1 walthic) (cadr liv16))
        liv18   (list (car liv13) (cadr liv15))
         liv19   (list (car liv13) (+ livhgt3 walthic))
         liv110  (list (- len1 walthic) (cadr liv19))
        dinhgt1 (list (- (car liv19) open1) (cadr liv19))
```

```
        win2pt1 (list 0 0)
        win2pt2 (list win2len 0)
        win2pt3 (list win2len win1br)
        win2pt4 (list (car win2pt1) win2br)
        win2pt5 (list (car win2pt1) (- (cadr win2pt1)
                  (/ (- walthic win2br) 2)))

        )
(setvar "thickness" win2hgt)
(command  "color" "m"
          "pline" win2pt1 win2pt2 win2pt3 win2pt4 win2pt1
                   win1pt5 (plo 0 win2len) (plo 90 walthic)
                     (plo 180 win2len)
                   (plo 270 walthic) ""
           "block" "win2blk" win2pt5 "L" "" )

 (setq
        win3pt1 (list 0 0)
        win3pt2 (list win3len 0)
        win3pt3 (list win3len win3br)
        win3pt4 (list (car win3pt1) win3br)
         win3pt5 (list (car win3pt1) (- (cadr win3pt1)
                  (/ (- walthic win3br) 2)))

         )
(setvar "thickness" win3hgt)
(command  "color" "m"
          "pline" win3pt1 win3pt2 win3pt3 win3pt4 win3pt1
                   win3pt5 (plo 0 win3len)
                     (plo 90 walthic) (plo 180 win3len)
                   (plo 270 walthic) ""
           "block" "win3blk" win3pt5 "L" "" )


  (setq
        door1pt1 (list 0 0)
        door1pt2 (list door11en 0)
        door1pt3 (list door11en door1br)
        door1pt4 (list (car door1pt1) door1br)
        door1pt5 (list (car door1pt1)
                       (cadr door1pt1) door1hgt)

        )
 (setvar "thickness" door1hgt)
 (command  "color" "c"
           "pline" door1pt1 door1pt2 door1pt3 door1pt4 "C"
            "elev"  "" (- walhgt door1hgt)
            "color" "r"
            "pline"  door1pt5 (plo 0 door11en)(plo 90 walthic)
                     (plo 180 door11en) "c"
           "block" "door1blk" door1pt1 "si" "c" door1pt1 door1pt3)



  (setq
        door2pt1 (list 0 0)
        door2pt2 (list door21en 0)
```

```
        door2pt3 (list door2len (* door2br 0.5))
        door2pt4 (list (car door2pt1) (* door2br 0.5))
       door2pt5 (list (car door2pt1)
                      (cadr door2pt1) door2hgt)
        )
(setvar "thickness" door2hgt)
(command  "color" "c"
          "pline" door2pt1 door2pt2 door2pt3 door2pt4 "C"
           "elev"  "" (- walhgt door2hgt)
           "color" "r"
           "pline"  door2pt5 (plo 0 door2len)
                 (plo 90 (* walthic 0.5)(plo 180 door2len) "c"
           "block" "door2blk" door2pt1 "si" "c" door2pt1 door2pt3)

 (setq
        door3pt1 (list 0 0)
        door3pt2 (list door3len 0)
        door3pt3 (list door3len door3br)
        door3pt4 (list (car door3pt1) door3br)
        door3pt5 (list (car door3pt1) (cadr door3pt1) door3hgt)
        )
(setvar "thickness" door3hgt)
(command  "color" "c"
          "pline" door3pt1 door3pt2 door3pt3 door3pt4 "C"
           "elev"  "" (- walhgt door3hgt)
           "color" "R"
           "pline"  door3pt5 (plo 0 door3len)
           (plo 90 (* walthic 0.5))(plo 180 door3len) "c"
          "block" "door3blk" door3pt1 "si" "c" door3pt1 door3pt3)

  (setq
        door4pt1 (list 0 0)
        door4pt2 (list door2len 0)
        door4pt3 (list door2len door2br)
        door4pt4 (list (car door4pt1) door2br)
          door4pt5 (list (car door4pt1) (cadr door4pt1) door2hgt)
        )
(setvar "thickness" door2hgt)
(command  "color" "c"
          "pline" door4pt1 door4pt2 door4pt3 door4pt4 "C"
           "elev"  "" (- walhgt door2hgt)
           "color" "r"
           "pline"  door4pt5 (plo 0 door2len)(plo 90  walthic )
                  (plo 180 door2len) "c"
         "block" "door4blk" door4pt1 "si" "c" door4pt1 door4pt3)

  (setq
        ven1pt1 (list 0 0)
        ven1pt2 (list ven1len 0)
        ven1pt3 (list ven1len ven1br)
        ven1pt4 (list (car ven1pt1) ven1br)
       ven1pt5 (list (car ven1pt1) (- (cadr ven1pt1)
                 (/ (- walthic ven1br) 2)))
        )
(setvar "thickness" ven1hgt)
```

```
   (command  "color" "b"
             "pline" ven1pt1 ven1pt2 ven1pt3 ven1pt4 ven1pt1
                      ven1pt5 (plo 0 ven11en)
                        (plo 90 walthic) (plo 180 ven11en)
                      (plo 270 walthic) ""
             "block" "ven1blk" ven1pt5 "L" "" )

(PROMPT "\n              DINING TABLE...")
(setq dint1en 48
      dinthgt 24
       dintwid 24
       )
       (setvar "thickness" -4)
       (command "color" "c"
              "pline" (list 0 0)
               (plo 0 dintwid)
               (plo 90 dintlen)
               (plo 180 dintwid)
               (plc 270 dintlen) ""
               "block" "din1blk" (1p) "1" "")

(setq dintflen 12
      dintfhgt 24
       dintfwid 12
       )
      (setvar   "thickness" (abs (- dinthgt 4)))
      (command   "pline" (list (+ (car (LP)) 3)
                               (+ (cadr (1p)) 3))
                 (plo 0 (abs (- dintfwid 6)))
                 (plo 90 (abs (- dintflen 6)))
                 (plo 180 (abs (- dintfwid 6)))
                 (plo 270 (abs (- dintflen 6))) ""
               "block" "dintf2blk" (1p) "1" "")

        (PROMPT "\n\n\n              BED1 BLOCK ...")
        (setq  rbed1len  72
               rbed1wid  36
                rbed1hgt  24

                )(setvar "thickness" 0)
                (command "pline" (list 0 0)
                           (list rbed1len 0)
                            (list rbed1len  rbed1wid)
                            (list 0 rbed1wid)
                            "c"
                            "color" "B"
                            "ELEV" RBED1hGT ""
                            "hatch" "grass" "10" "45"  "1" ""
                            "ELEV" "0" "0"
                            "chprop" "p" "" "t" rbed1hgt ""
                             "block" "bed1blk" "0,0" "p" "1" ""
                              )
        (PROMPT "\n\n\n      MAKING BED2 BLOCK  ...")
        (setq  rbed1len  72
```

```
            rbed1wid  36
             rbed1hgt  24

             )(setvar "thickness" rbed1hgt)
             (command "color" "m"
             "pline" (list 0 0)
                        (list rbed11en 0)
                         (list rbed11en  rbed1wid)
                         (list 0 rbed1wid)
                         "c"
                         "color" "r"
                         "ELEV" RBED1hGT "0"
                        "hatch" "grass" "10" "45"   "1" ""
                         "ELEV" "0" "0"
                          "block" "bed2b1k" "0,0" "p" "1" ""
                           )


          (prompt "\n          DRAWING OUTLINES...")
          (setvar "thickness" walhgt)
(setq   stohgta1 (list (car stohgta)
                       (cadr stohgt1) door1hgt))

(command  "color" "g"
           "pline" out11 out12 out13 out14 "C"
          "offset" walthic out11 out12 ""
           "color" "r"
          "line"  stohgt1 stohgta  spa sp3 ""
           "line"  sp1 kit1en2 kit1en1 stohgt3 ""
           "color" "m"
           "elev"  "36" "6"
           "line" (polar  kit1en1 (dtr 180) 12)
                  (plo 90 24)
                  (list (+ (car out11) walthic 24)
                           (cadr (Lp)))
                   (list (car (lp))
                         (- (cadr out14) (+ walthic 24)))
                  (list (car bedc2) (cadr (LP))) ""
            "color" "y"
             "elev" "0"  "0")
(setq walt 6)
(setvar "thickness" walt)
(command

        "pline" outh1 outh2 outh3 outh4 "c"
        "color" "r"
        )
        (setvar "thickness" walhgt)

      (prompt "\n\n\n           KITCHEN OUTLINES...")
     (command
              "line" bedca  bedcb ""
             "line" bedc1  bedc2 ""
        "line" dinhgt1 bedcb ""
```

```
                "line" dinhgt2 bedc2 ""
                "line" dinhgt1 dinhgt2 ""
)

(prompt "\n\n\n       BEDROOM OUTLINES...")
(command
         "line" bed111 bed112 bed114 bed113 ""
         "line" bed211 bed21a bed21b bed213 ""
         "line" bed21c bed212 ""
         "line"  bed212 bed214 ""
         "line"  bed214  bed21d ""
         "line"  bed21d  bed21c ""
         "line" toil1 toil2 ""
         "line" toil4 toil3 ""
         "line" toil4 toil5 ""
         "line" toil2 toil6 ""
         "line" toil7 toil9 ""
         "line" toil8 toil10 ""
         "line" toil7 toil8 "")

(prompt "\n        DINING OUTLINES...")
(command "color" "r")
(command
         "line" din11 din12 ""
         "line" din13 din14 ""
          "line" din13 din12 ""
           )

(prompt "\n\n\n          LIVING ROOM OUTLINES...")
      (command
            "line" liv11 liv12 ""
             "line" liv13 liv14 ""
             "line" liv13 liv12 ""
             "line" liv15 liv16 liv17 ""
             "line" liv18 liv19 liv110 ""
              "line" liv15 liv18 ""
                   "color" "y"
              "elev"  "36" "6"
              "pline" (polar  liv12 (dtr 270) 6)
                      (plo 0  24)
                      (list  (car (lp)) (+ (cadr liv11) 24))
                       (list (abs (- (car out12) 24 walthic))
                                      (cadr (LP)))
                      (list (car (lp)) (+ (cadr (LP)) 60))
                      (list (abs (- (car out12) walthic))
                                    (cadr (lp)))
                      (list (abs (- (car out12) walthic))
                       (abs (- (cadr out12) walthic)) )
                        liv11 "c"
                        )(setvar "thickness" 0)
                        (command "color" "b"
                        ;"hatch" "cork"  "20" "0"  "1" ""

                      "color" "r"
```

```
                    )
          (PROMPT "\n\n\n CALCULATING INSERTION POINTS...")

(setq  winlev 24
       firdoor1  (list  (car liv14)   (cadr liv14) )
       dinwin1   (list (+ (car stohgt1) (* dinlen 0.27))
                  (abs (- (cadr stohgt1) walthic)) winlev)
       livwin2   (list (+ (car liv11)  (* lirolen 0.3))
                 (abs  (-(cadr  liv11) walthic)) winlev)
       livwin1   (list  (car out12)
                  (+ (cadr out12) (* wid 0.2)) winlev)
       bed1win2  (list (+ (car liv110) walthic)
                  (cadr liv110) winlev)  .
        bed1win21 (list (abs (- (car out13) walthic))
                  (cadr out13) winlev)
        bed2win21 (list (car bed213)
                  (+ (cadr bed213) walthic)   winlev)
        bed2win22 (list (+ (car bedc1) walthic)  (cadr bedc1)
                   winlev)
        kitwin3   (list (+ (car out14) (* kit1  0.3))
                        (abs (- (cadr out14) walthic)) winlev)
       kitwin2   (list (car out14) (abs (- (cadr out14)
                          (* wid 0.5))) winlev)
       toiven1   (list (+ (car bed211) (* toilen 0.2))
                          (cadr bed211)  toilev)
        toiven2   (list  (+ (car toil1) (* toilen 0.2))
                         (cadr bed111)  toilev)
        dintin1   (list (+ (car stohgta) (* dinlen 0.4))
                           (abs (- (cadr stohgta) (* wid 0.4)))
                           dinthgt)
        dintin2    (list (+ (car dintin1) 3) (+ (cadr dintin1) 3))
       dintin3    (list (car dintin2)
                  (+ (cadr dintin2) (abs (- dintlen 12))))
       dintin5    (list (+ (car dintin2) (abs (- dintwid 12)))
                          (cadr dintin2))
       dintin4    (list (car dintin5) (cadr dintin3))

        bed1       (list (car liv110) (+ (cadr liv110)  60))
        bed2       (list (+ (car bedc1) 60)  (cadr bedc1))

  )
(command "block" "tab" dintin1 "si" "c" dintin1
(list (+ (car dintin1) dintwid) (+ (cadr dintin1)  dintlen))
          "" "oops")
 (PROMPT "\n\n\n          INSERTING  THE BLOCKS...")
        (command
           "elev" "0" "0"
                "insert" "door1blk" stohgta  "" "" "90"
                      "insert" "door1blk" firdoor1 "" "" "180"
                       "insert" "door4blk"  kitlen2  "" "" ""
                       "insert" "door2blk" bed214 "" "" "270"
                       "insert" "door2blk" bed114  "" "" "270"
                       "insert" "door3blk" bed21c  "" "" "90"
                       "insert" "door3blk" toil7   "" "" "180"
```

```
                    "elev" winlev ""
                    "insert" "win1blk" dinwin1  ""  "" ""
                    "insert" "win1blk" livwin1  ""  "" "90"
                    "insert" "win2blk" livwin2  ""  "" ""
                    "insert" "win2blk" bed1win2 ""  "" "90"
                    "insert" "win2blk" bed1win21 "" "" "180"
                    "insert" "win2blk" bed2win21 "" "" "180"
                    "insert" "win2blk" bed2win22 "" "" ""
                    "insert" "win3blk" kitwin3   "" "" ""
                    "insert" "win2blk" kitwin2   "" "" "270"
                     "elev" "0" "0"
                    "insert" "ven1blk" toiven1   "" "" ""
                    "insert" "ven1blk" toiven2   "" "" ""
                            "elev" dinthgt ""
                  "insert" "din1blk" dintin1  "" "" ""
                    "elev" "0" "0"
                     "insert" "dintf2blk" dintin2  "" "" ""
                     "insert" "dintf2blk" dintin3  "" "" ""
                     "insert" "dintf2blk" dintin4  "" "" ""
                     "insert" "dintf2blk" dintin5  "" "" ""
                  "insert" "bed1blk"  bed1     "" "" "180"
                    "insert" "bed2blk"  bed2     "" "" "-90"

)

(setvar "elevation" dinthgt)
(COMMAND
"insert" "tab" (polar livwin1 (dtr 150) 100) "x" 0.8 0.8 0.8 ""
"insert" "tab" (polar liv12 (dtr 0) 36) "x" 0.8 0.8 0.8 "-90" )
(setvar "elevation"   door1hgt)
(setvar  "thickness"  6)
(command "color"  "y"
        "pline" (setq sun1 (list (+ (car firdoor1) walthic) 0) )
                  (plo 270 48)
                   (polar din11  (dtr 270) (+ walthic 48))
                   (plo 90 24)
                   (setq re (list (- (car dinwin1) walthic)
                                  (cadr (lp))))
                   (plo 90 24) "c"
           "pline" sun1 (plo 270 96)
                   (list (abs (- (car out12) 12))
                      (cadr (lp)))
                       (plo 90  72) (plo 0 32)
                       (plo 90 (+ len2 48))
                       (list  (abs (- (car bed213)
                       (+ win21en walthic))) (cadr (lp)))
                (list (car (lp)) (+ (cadr bed213) walthic))
                       out13 out12  "c"

        "pline"    (list  (+ (car bedc1) win21en walthic)
                   (+ (cadr bedc1) walthic))
                   (plo 90 24)
                    (list (abs (- (car kitwin3) walthic))
                                  (cadr (lp)))
```

```
                    (plo 270 24) "c"
                                  )

(prompt "\n                    STAIRCASE...")
(SETQ STEPLEN (distance dinl1 dinl2)
      stephgt (/ walhgt 9)
      stepl1 (/ steplen stephgt)
      )
      (setvar "thickness" 9)
      (setvar "elevation" 0)
      (command "color" "c"
               "pline" dinl2  (plo 0 doorllen)
                 (plo 270 stepl1) (plo 180 doorllen) "c")
                 (repeat (fix (1- stephgt))
                    (command "copy" "si" "last" "0,0"
                        (list 0 (- 0 stepl1) 9)
                 )
                  )
                  (princ " 007")
(prompt "\n           UPSTAIR ROOM")
    (setvar "elevation" walhgt)
    (setvar  "thickness" (* walhgt 0.8))
(command   "pline"  (list (car dinl4) 0)
                    (list (car livl1) 0)
                    (list  (car livl1) (cadr LIVL9))
                      (list (car DINL4) (cadr LIVL9)) "c"
                                      ""  )
   (setvar "elevation" (+ walhgt (* walhgt 0.8)))
   (setvar "thickness" walHGT)
         (command "color" "b"
          "pline"
                 (list  (car dinl4) 0) (list (car livl1) 0)
               (list  (car livl1) (cadr LIVL9))
                 (list (car DINL4) (cadr LIVL9)) "c"
                 "offset" walthic (1P) (plo 325 12)
                    "" )

    (setvar "thickness" doorlhgt)
    (SETVAR "ELEVATION" 0)
(prompt "\n\n\n                    CARRIAGE...")
(COMMAND "pline" re (plo 270 120)
                    (plo 180 192)
                      (list (car (1p)) walthic)
                      (list 0 walthic) ""
                                      )
(prompt "\n\n\n           TEXT PRINTING...")
(command "color" "b")
(command "style" "linda" "romant"
    "9" "" "" "" "" "" "text" "c" "0,-60" "0" "CARRIAGE"
            "text" "c" "0,-80" "0" "16 '0 " X 10 '0 " "

          "text" "c" (Polar stohgt3 (dtr 45) (* kitl 0.5))
                          "0" "KITCHEN"
```

```
          "text" "c" (Polar bedc2 (dtr 45) 84)
                                "0" "BEDROOM1"

          "text" "c" (Polar bed1L1 (dtr -35) 60)
                             "0" "BEDROOM2"

          "text" "c" (Polar LIVL6 (dtr -35) 60)
                                "0" "LIVING ROOM"

          "text" "c" (Polar DINL3 (dtr 215) 60)
                                "0" "DINING ROOM"

          "text"  (Polar OUTL1 (dtr 35) 60)
                                "90" "STORE ROOM1"

       )
(prompt "\n\n\n          CHANGING THE VIEWPOINT")
(command "vpoint" "1,-1,1")


        (INITGET (+  2 4))
(setq win1hgt   (GETDIST " nENTER THE WINDOW1 HEIGHT  <51>:"))
   (if (= win1hgt nil) (setq win1hgt      51))
        (INITGET (+  2 4))
 (setq  win1br    (GETDIST " nENTER THE WINDOW1 WIDTH  <05>:"))
    (if (= win1BR nil) (setq win1br      05))
        (INITGET (+  2 4))
(setq win11en   (GETDIST " nENTER THE WINDOW1 LENGTH <96>:"))
    (if (= win11en nil) (setq win11en      96))
        (INITGET (+  2 4))
(setq win2hgt   (GETDIST " nENTER THE WINDOW2 HIGHT  <51>:"))
    (if (= win2hgt nil) (setq win2hgt  51))
        (INITGET (+  2 4))
 (setq win2br    (GETDIST " nENTER THE WINDOW2 WIDTH  <05>:"))
    (if (=  win2br nil) (setq  win2br 5))
        (INITGET (+  2 4))
(setq win21en   (GETDIST " nENTER THE WINDOW2 LENGTH <42>:"))
    (if (= win21en nil) (setq win21en 42))
        (INITGET (+  2 4))
(setq win3hgt   (GETDIST " nENTER THE WINDOW3 HEIGHT  <51>:") )
    (if (=   win3hgt nil) (setq  win3hgt 51))
        (INITGET (+  2 4))
 (setq win3br    (GETDIST " nENTER THE WINDOW3 WIDTH  <05>:"))
    (if (=   win3br nil) (setq  win3br 5))
        (INITGET (+  2 4))
 (setq  win31en   (GETDIST " nENTER THE WINDOW3 LENGTH <54>:") )
    (if (=  win31en nil) (setq win31en 54))
        (INITGET (+  2 4))
 (setq door1hgt  (GETDIST " nENTER THE DOOR1 HEIGHT    <84>:") )
    (if (=  door1hgt nil) (setq  door1hgt 84))
        (INITGET (+  2 4))
(setq  door1br   (GETDIST " nENTER THE DOOR1 WIDTH    <05>:"))
   (if (=  door1br nil) (setq door1br 5))
        (INITGET (+  2 4))
```

```
(setq door1len  (GETDIST " nENTER THE DOOR1 LENGTH   <42>:") )
  (if (=  door1len nil) (setq door1len 42))
        (INITGET (+  2 4))
(setq door3hgt  (GETDIST " nENTER THE DOOR3 HEIGHT    <84>:") )
  (if (=  door3hgt nil) (setq door3hgt 84))
        (INITGET (+  2 4))
(setq door3br   (GETDIST " nENTER THE DOOR3 WIDTH    <04>:"))
  (if (=  door3br nil) (setq door3br 4))
        (INITGET (+  2 4))
(setq    door31en  (GETDIST " nENTER THE DOOOR3 LENGTH  <27>:") )
  (if (=  door31en nil) (setq door31en 27))
        (INITGET (+  2 4))
(setq    door2hgt  (GETDIST " nENTER THE DOOR2 HEIGHT    <84>:") )
  (if (=   door2hgt nil) (setq door2hgt 84))
        (INITGET (+  2 4))
(setq    door2br   (GETDIST " nENTER THE DOOR2 WIDTH    <05>:"))
  (if (=  door2br nil) (setq door2br 5))
        (INITGET (+  2 4))
 (setq  door21en  (GETDIST " nENTER THE DOOR2 LENGTH   <36>:"))
  (if (=  door21en nil) (setq door21en 36))
        (INITGET (+  2 4))
(setq  ven1hgt   (GETDIST " nENTER THE VEN1 HEIGHT     <24>:"))
  (if (=  ven1hgt  nil) (setq ven1hgt 24 ))
        (INITGET (+  2 4))
(setq    ven1br    (GETDIST " nENTER THE VEN1 WIDTH     <05>:"))
  (if (= ven1br  nil) (setq ven1br 05))
        (INITGET (+  2 4))
(setq    ven11en  (GETDIST " nENTER THE VEN1 LENGTH    <30>:") )
  (if (=  ven11en nil) (setq ven11en 30))
        (INITGET (+  2 4))
(setq    open1 (GETDIST " nENTER THE OPEN1 LENGTH   <48>:") )
 (if (=  open1 nil) (setq  open1 48))
        (INITGET (+  2 4))
(setq  open2 (GETDIST " nENTER THE OPEN2 LENGTH   <60>:") )
  (if (= open2  nil) (setq open2 60))
        (INITGET (+  2 4))
(setq winlev (GETDIST " nENTER THE WINDOW LEVEL   <24>:") )
  (if (=   winlev  nil) (setq  winlev 24))
        (INITGET (+  2 4))
(setq  toilev (GETDIST " nENTER THE VEN LEVEL      <70>:") )
  (if (=   toilev  nil) (setq   toilev 70))
        (INITGET (+  2 4))
(setq toilen  (GETDIST " nENTER THE TOILET  LENGTH <60>:") )
  (if (=  toilen  nil) (setq  toilen 60 ))
        (INITGET (+  2 4))
(setq  toiwid (GETDIST " nENTER THE TOILET WIDTH   <108>:"))
 (if (= toiwid nil) (setq toiwid 108))


(prompt "\n\n -------------CURRENT DIMENSIONS-------------------")
 (PROMPT "\n\n            ALL DIMENSIONS ARE IN INCHS            ")
(PROMPT "\n\n           LENGTH         WIDTH           HEIGHT")
 (PROMPT "\n\n DOOR1       ")
 (PRINC  DOOR1LEN)
```

```
 (PROMPT "              ")
 (PRINC  DOOR1BR)
 (PROMPT "              ")
 (PRINC  DOOR1HGT)
  (PROMPT "\n\n DOOR2        ")
 (PRINC  DOOR2LEN)
 (PROMPT "              ")
 (PRINC  DOOR2BR)
 (PROMPT "              ")
 (PRINC  DOOR2HGT)
  (PROMPT "\n\n DOOR3        ")
 (PRINC  DOOR3LEN)
 (PROMPT "              ")
 (PRINC  DOOR3BR)
 (PROMPT "              ")
 (PRINC  DOOR3HGT)
  (PROMPT "\n\n WINDOW1      ")
 (PRINC  WIN1LEN)
 (PROMPT "              ")
 (PRINC  WIN1BR)
 (PROMPT "              ")
 (PRINC  WIN1HGT)
         (PROMPT "\n\n WINDOW2      ")
         (PRINC  WIN2LEN)
         (PROMPT "              ")
         (PRINC  WIN2BR)
         (PROMPT "              ")
         (PRINC  WIN2HGT)
          (PROMPT "\n\n WINDOW3     ")
         (PRINC  WIN3LEN)
         (PROMPT "              ")
         (PRINC  WIN3BR)
         (PROMPT "              ")
         (PRINC  WIN3HGT)
          (PROMPT "\n\n VEN         ")
         (PRINC  VEN1LEN)
         (PROMPT "              ")
         (PRINC  VEN1BR)
         (PROMPT "              ")
         (PRINC  VEN1HGT)
  (PROMPT "\n\n          OPEN PLACE ONE LENGTH :")
         (PRINC  OPEN1)
          (PROMPT "\n        OPEN PLACE TWO LENGTH :")
         (PRINC  OPEN2)
        (PROMPT "\n\n")
(setq try (getstring "    PRESS RETURN TO CONTINUE.....,"))


 (command "plan" "")
(SETVAR "EXPERT" 128)
(SETQ
    dinthgt 24
    charhgt 24
```

```
out11 (list 0 0)
out12 (list len1 0)
out13 (list len1 len2)
out14 (list 0 len2)

outh1 (list  (car out11)   (cadr out11)  walhgt)
outh2 (list  (car out12)   (cadr out12)  walhgt)
outh3 (list  (car out13)  (cadr out13) walhgt)
outh4 (list  (car out14)  (cadr out14) walhgt)


stohgt1 (list stolen  walthic)
stohgt2 (list (car stohgt1) (abs (+ wid (* walthic 0.5))))
sp2     (list (car stohgt2) (- (cadr stohgt2) walthic))
stohgta (list stolen  (abs (- (cadr sp2) door1len)))
stohgt3 (list  walthic (cadr stohgt2))
 sp1     (list (car stohgt3) (- (cadr stohgt3) walthic))
sp3     (list (- (car stohgt1) walthic) (cadr stohgt1))
  spa     (list (car sp3) (cadr stohgta))

 bedca  (list kit1 (- len2 walthic))
 bedcb  (list  (car bedca)  (+ wid (* walthic 0.5)))
 bedc1  (list  (- (car bedca) walthic) (cadr bedca))
 bedc2  (list (car bedc1) (abs (- wid (* walthic 0.5))))

 spkit1  (abs (- (distance stohgt2 bedc2) door2len))
 kitlen1 (list (+ (car stohgt2) spkit1) (cadr stohgt2))
kitlen2 (list (+ (car sp2) spkit1) (cadr sp2))


  din11   (list din1len1 walthic)
din12   (list din1len1 (abs (- (abs
              (- wid (* walthic 0.5))) open1)))
din13   (list (- (car din12) walthic) (cadr din12))
        din14   (list (car din13) walthic)

        liv11   (list liv1len1 walthic)
        liv12   (list liv1len1 livhgt1)
       liv13   (list (- (car liv12) walthic) (cadr liv12))
        liv14   (list (car liv13) (cadr liv11))
        liv15   (list liv1len1 livhgt2)
        liv16   (list liv1len1  livhgt3 )
       liv17   (list (- len1 walthic) (cadr liv16))
       liv18   (list (car liv13) (cadr liv15))
        liv19   (list (car liv13) (+ livhgt3 walthic))
        liv110  (list (- len1 walthic) (cadr liv19))
       dinhgt1 (list (- (car liv19) open1) (cadr liv19))
        dinhgt2 (list (car dinhgt1) (cadr liv16))

       bed111  (list (+ toilen toimid)
                     (ABS (- len2 walthic)))
        bed112  (list (car bed111) (+ (+ wid
                     (* walthic 0.5)) door2len))
        bed113  (list (ABS (- (car bed111)
```

```
                    (* walthic 0.5))) (cadr bed111))
        bed114  (list (car bed113)  (cadr bed112))
       bed211   (list (abs (- toimid toilen))
                    (ABS (- len2 walthic)))

        bed212  (list (car bed211) (cadr bed112))
        bed213  (list (abs (- (car bed211)
                      (* walthic 0.5))) (cadr bed211))
       bed214   (list (car bed213) (cadr bed212))
        bed21a  (list (car bed211)
         (abs (- (abs (- (* toiwid 0.5) door3len))
                              (cadr bed211))))
        bed21c  (list (car bed21a)
                      (abs (- (cadr bed21a) door3len)))
        bed21b  (list (abs (- (car bed21a) (* walthic 0.5)))
                              (cadr bed21a))
        bed21d  (list (car bed21b) (cadr bed21c))

       toil1    (list (+ toimid (* walthic 0.5))
                                (cadr bed111))
        toil2    (list (car toil1)(abs (-  (cadr toil1) toiwid)))
        toil3    (list (abs (- toimid (* walthic 0.5)))
                                    (cadr toil1))
        toil4    (list (car toil3) (+ (cadr toil2)
                                    (* walthic 0.5)))
       toil5    (list (car bed212) (cadr toil4))
        toil6    (list (car toil5)  (cadr toil2))
        toil7    (list (+ (+ (car toil4) walthic) door3len)
                              (cadr toil4))
        toil8    (list (+ (car toil2) door3len)
                      (abs (- (cadr toil7) (* walthic 0.5))))
        toil9    (list (car bed114) (cadr toil7))
        toil10   (list (car bed114) (cadr toil8))
    )

(prompt "\n MAKING BLOCK OPERATIONS ...")
    (setq
          win1pt1 (list 0 0)
          win1pt2 (list win11en 0)
          win1pt3 (list win11en win1br)
          win1pt4 (list (car win1pt1) win1br)
         win1pt5 (list (car win1pt1) (- (cadr win1pt1)
                    (/ (- walthic win1br) 2)))
            )
   (command  "color" "m"
             "pline" win1pt1 win1pt2 win1pt3 win1pt4 win1pt1
                      win1pt5 (plo 0 win11en)
                         (plo 90 walthic) (plo 180 win11en)
                       (plo 270 walthic) ""
             "block" "win1blk" win1pt5 "L" "" )



     (setq
```

```
        win2pt1 (list 0 0)
        win2pt2 (list win2len 0)
        win2pt3 (list win2len win1br)
        win2pt4 (list (car win2pt1) win2br)
        win2pt5 (list (car win2pt1) (- (cadr win2pt1)
                 (/ (- walthic win2br) 2)))
         )
(command  "color" "m"
          "pline" win2pt1 win2pt2 win2pt3 win2pt4 win2pt1
                   win1pt5 (plo 0 win2len)
                     (plo 90 walthic) (plo 180 win2len)
                    (plo 270 walthic) ""
          "block" "win2blk" win2pt5 "L" "" )


 (setq
        win3pt1 (list 0 0)
        win3pt2 (list win3len 0)
        win3pt3 (list win3len win3br)
        win3pt4 (list (car win3pt1) win3br)
         win3pt5 (list (car win3pt1) (- (cadr win3pt1)
                 (/ (- walthic win3br) 2)))
         )
(command  "color" "m"
          "pline" win3pt1 win3pt2 win3pt3 win3pt4 win3pt1
                   win3pt5 (plo 0 win3len)
                      (plo 90 walthic) (plo 180 win3len)
                    (plo 270 walthic) ""
          "block" "win3blk" win3pt5 "L" "" )


 (setq
        door1pt1 (list 0 0)
        door1pt2 (list door1len 0)
        door1pt3 (list door1len door1br)
        door1pt4 (list (car door1pt1) door1br)
        door1pt5 (list (car door1pt1)
                       (cadr door1pt1) door1hgt)
        )
(command  "color" "c"
    "pline" door1pt1 door1pt2 door1pt3 door1pt4 "C"
   "color" "r"
   "pline"  door1pt5 (plo 0 door1len)(plo 90 walthic)
   (plo 180 door1len) "c"
   "block" "door1blk" door1pt1 "si" "c" door1pt1 door1pt3)



 (setq
        door2pt1 (list 0 0)
```

```
        door2pt2 (list door21en 0)
        door2pt3 (list door21en (* door2br 0.5))
        door2pt4 (list (car door2pt1) (* door2br 0.5))
       door2pt5 (list (car door2pt1)
                      (cadr door2pt1) door2hgt)
        )
(command  "color" "c"
      "pline" door2pt1 door2pt2 door2pt3 door2pt4 "C"
       "color" "r"
       "pline"  door2pt5 (plo 0 door21en)
         (plo 90 (* walthic 0.5))
         (plo 180 door21en) "c"
     "block" "door2blk" door2pt1 "si" "c" door2pt1 door2pt3)



 (setq
        door3pt1 (list 0 0)
        door3pt2 (list door31en 0)
        door3pt3 (list door31en door3br)
        door3pt4 (list (car door3pt1) door3br)
        door3pt5 (list (car door3pt1)
                       (cadr door3pt1) door3hgt)
        )
(command  "color" "c"
     "pline" door3pt1 door3pt2 door3pt3 door3pt4 "C"
      "color" "R"
      "pline"  door3pt5 (plo 0 door31en)
       (plo 90 (* walthic 0.5))
       (plo 180 door31en) "c"
      "block" "door3blk" door3pt1 "si" "c" door3pt1 door3pt3)


 (setq
        door4pt1 (list 0 0)
        door4pt2 (list door21en 0)
        door4pt3 (list door21en door2br)
        door4pt4 (list (car door4pt1) door2br)
          door4pt5 (list (car door4pt1)
                         (cadr door4pt1) door2hgt)
        )
(command  "color" "c"
  "pline" door4pt1 door4pt2 door4pt3 door4pt4 "C"
  "color" "r"
  "pline"  door4pt5 (plo 0 door21en)(plo 90  walthic )
    (plo 180 door21en) "c"
  "block" "door4blk" door4pt1 "si" "c" door4pt1 door4pt3)


 (setq
        ven1pt1 (list 0 0)
        ven1pt2 (list ven11en 0)
        ven1pt3 (list ven11en ven1br)
```

```
          ven1pt4 (list (car ven1pt1) ven1br)
          ven1pt5 (list (car ven1pt1) (- (cadr ven1pt1)
                    (/ (- walthic ven1br) 2)))
          )
   (command  "color" "b"
      "pline" ven1pt1 ven1pt2 ven1pt3 ven1pt4 ven1pt1
          ven1pt5 (plo 0 ven1len)
                  (plo 90 walthic) (plo 180 ven1len)
                  (plo 270 walthic) ""
            "block" "ven1blk" ven1pt5 "L" "" )


(PROMPT "\n MAKING DINING TABLE...")
(setq dintlen 48
       dinthgt 24
        dintwid 24
        )
        (command "color" "c"
                "pline" (list 0 0)
                 (plo 0 dintwid)
                 (plo 90 dintlen)
                 (plo 180 dintwid)
                 (plo 270 dintlen) ""
                 "block" "din1blk" (1p) "1" "")

(setq dintflen 12
       dintfhgt 24
        dintfwid 12
        )
      (command   "pline" (list (+ (car (LP)) 3)
                               (+ (cadr (1p)) 3))
                 (plo 0 (abs (- dintfwid 6)))
                 (plo 90 (abs (- dintflen 6)))
                 (plo 180 (abs (- dintfwid 6)))
                 (plo 270 (abs (- dintflen 6))) ""
               "block" "dintf2blk" (1p) "1" "")

        (PROMPT "\n MAKING BED1 BLOCK OPERATION ...")
        (setq  rbed1len  72
               rbed1wid  36
                rbed1hgt  24

                )
                (command "pline" (list 0 0)
                           (list rbed1len 0)
                            (list rbed1len  rbed1wid)
                            (list 0 rbed1wid)
                            "c"
                            "color" "B"
                            "hatch" "grass" "10" "45"  "1" ""
                             "block" "bed1blk" "0,0" "p" "1" ""
                              )

        (PROMPT "\n MAKING BED2 BLOCK OPERATION ...")
```

```
        (setq  rbed1len  72
               rbed1wid  36
                rbed1hgt  24
)
                (command "color" "m"
                "pline" (list 0 0)
                           (list rbed1len 0)
                            (list rbed1len  rbed1wid)
                            (list 0 rbed1wid)
                            "c"
                            "color" "r"
                           "hatch" "grass" "10" "45"  "1" ""
                             "block" "bed2blk" "0,0" "p" "1" ""
                              )


              (prompt "\n DRAWING OUTLINES...")
   (setq   stohgta1 (list (car stohgta)
                          (cadr stohgt1) door1hgt))

   (command  "color" "g"
             "pline" out11 out12 out13 out14 "C"
            "offset" walthic out11 out12 ""
             "color" "r"(prompt " ndrawing store room...")
            "line"  stohgt1 stohgta  spa sp3 ""
             "line"  sp1 kitlen2 kitlen1 stohgt3 ""
             "color" "m"
             "line" (polar  kitlen1 (dtr 180) 12)
                    (plo 90 24)
                    (list (+ (car out11) walthic 24)
                             (cadr (Lp)))
                     (list (car (lp))
                       (- (cadr out14) (+ walthic 24)))
                    (list (car bedc2) (cadr (LP))) ""
              "color" "r")
  (setq walt 6)


        (prompt "\nDRAWING KITCHEN OUTLINES...")
       (command
                "line" bedca  bedcb ""
               "line" bedc1  bedc2 ""
          "line" dinhgt1 bedcb ""
                "line" dinhgt2 bedc2 ""
                "line" dinhgt1 dinhgt2 ""
)


(prompt "\nDRAWING BEDROOM OUTLINES...")
(command
         "line" bed111 bed112 bed114 bed113 ""
         "line" bed211 bed21a bed21b bed213 ""
         "line" bed21c bed212 ""
```

```
          "line"  bed212 bed214 ""
          "line"  bed214  bed21d ""
          "line"  bed21d  bed21c ""
          "line" toil1 toil2 ""
          "line" toil4 toil3 ""
          "line" toil4 toil5 ""
          "line" toil2 toil6 ""
          "line" toil7 toil9 ""
          "line" toil8 toil10 ""
          "line" toil7 toil8 "")


(prompt "\nDRAWING DINING OUTLINES...")
(command "color" "r")
(command
          "line" din11 din12 ""
          "line" din13 din14 ""
           "line" din13 din12 ""
            )


(prompt "\nDRAWING LIVING ROOM OUTLINES...")
      (command
            "line" liv11 liv12 ""
             "line" liv13 liv14 ""
             "line" liv13 liv12 ""
             "line" liv15 liv16 liv17 ""
             "line" liv18 liv19 liv110 ""
              "line" liv15 liv18 ""
                   "color" "y"
              "pline" (polar  liv12 (dtr 270) 6)
                      (plo 0  24)
                      (list  (car (lp)) (+ (cadr liv11) 24))
                       (list (abs (- (car out12) 24 walthic))
                                    (cadr (LP)))
                      (list (car (lp)) (+ (cadr (LP)) 60))
                      (list (abs (- (car out12) walthic))
                                   (cadr (lp)))
                      (list (abs (- (car out12) walthic))
                       (abs (- (cadr out12) walthic)) )
                        liv11 "c"
                        )
                        (command "color" "b"
                        "hatch" "cork"  "10" "0"  "1" ""

                      "color" "r"
                      )

           (PROMPT "\n CALCULATING INSERTION POINTS...")

         (setq  winlev 24
               firdoor1  (list  (car liv14)   (cadr liv14) )
               dinwin1   (list (+ (car stohgt1) (* dinlen 0.27))
                               (abs (- (cadr
```

```
                stohgt1) walthic)) winlev)
          livwin2   (list (+ (car livl1)
                             (* lirolen 0.3)) (abs  (-
                    (cadr  livl1) walthic)) winlev)
          livwin1   (list  (car outl2)  (+ (cadr outl2)
                             (* wid 0.2)) winlev)
            bedlwin2  (list (+ (car livl10) walthic)
                               (cadr livl10) winlev)
            bedlwin21 (list (abs (- (car outl3) walthic))
                                    (cadr outl3) winlev)
            bed2win21 (list (car bed213)
                            (+ (cadr bed213) walthic   winlev)
            bed2win22 (list (+ (car bedcl) walthic)  (cadr bedcl)
                        winlev)
            kitwin3   (list (+ (car outl4) (* kitl  0.3))
                      (abs (- (cadr outl4) walthic)) winlev)
           kitwin2   (list (car outl4) (abs
                            (- (cadr outl4) (* wid
                      0.5))) winlev)
          toiven1   (list (+ (car bed211) (* toilen 0.2))
                       (cadr bed211)  toilev)
            toiven2   (list  (+ (car toill) (* toilen 0.2))
                      (cadr bedlll)  toilev)
            dintin1   (list (+ (car stohgta) (* dinlen 0.4))
                        (abs (- (cadr stohgta) (* wid 0.4)))
                         dinthgt)
            dintin2    (list (+ (car dintin1) 3)
                             (+ (cadr dintin1) 3))
          dintin3    (list (car dintin2)
                             (+ (cadr dintin2)
                               (abs (- dintlen
                           12))))
          dintin5    (list (+ (car dintin2)
                             (abs (- dintwid 12)))
                     (cadr dintin2))
          dintin4    (list (car dintin5) (cadr dintin3))

            bed1       (list (car livl10)
                             (+ (cadr livl10)  60))
            bed2       (list (+ (car bedcl) 60)
                                (cadr bedcl))

                          )
      (PROMPT "\nINSERTING  THE BLOCKS...")

(command
              "insert" "door1blk" stohgta  "" "" "90"
             "insert" "door1blk" firdoor1 "" "" "180"
            "insert" "door4blk"  kitlen2  "" "" ""
             "insert" "door2blk" bed214 "" "" "270"
             "insert" "door2blk" bedll4  "" "" "270"
             "insert" "door3blk" bed21c  "" "" "90"
             "insert" "door3blk" toil7   "" "" "180"
             "insert" "win1blk" dinwin1  ""  "" ""
```

```
                "insert" "win1blk" livwin1  ""  "" "90"
                "insert" "win2blk" livwin2  ""  "" ""
                "insert" "win2blk" bed1win2 ""  "" "90"
                "insert" "win2blk" bed1win21 "" "" "180"
                "insert" "win2blk" bed2win21 "" "" "180"
                "insert" "win2blk" bed2win22 "" "" ""
                "insert" "win3blk" kitwin3   "" "" ""
                "insert" "win2blk" kitwin2   "" "" "270"
                "insert" "ven1blk" toiven1   "" "" ""
                "insert" "ven1blk" toiven2   "" "" ""
               "insert" "din1blk" dintin1  "" "" ""
                 "insert" "dintf2blk" dintin2  "" "" ""
                 "insert" "dintf2blk" dintin3  "" "" ""
                 "insert" "dintf2blk" dintin4  "" "" ""
                 "insert" "dintf2blk" dintin5  "" "" ""
                "insert" "bed1blk"  bed1     "" "" "180"
                "insert" "bed2blk"  bed2     "" "" "-90"

)

(command "block" "tab" dintin1 "si" "c" dintin1
(list (+ (car dintin1) dintwid) (+ (cadr dintin1)  dintlen))
"oops")
(COMMAND
"insert" "tab" (polar livwin1 (dtr 150) 100) "x" 0.8 0.8 0.8 ""
"insert" "tab" (polar liv12 (dtr 0) 36) "x" 0.8 0.8 0.8 "-90" )
(command "color"  "y"
    "pline" (setq sun1 (list (+ (car firdoor1) walthic) 0) )
           (plo 270 48)
            (polar din11  (dtr 270) (+ walthic 48))
            (plo 90 24)
    (setq re (list   (- (car dinwin1) walthic) (cadr (lp))))
            (plo 90 24) "c"
                  (prompt " n n1")
           "pline" sun1 (plo 270 96)
            (list (abs (- (car out12) 12))
               (cadr (lp)))
                (plo 90  72) (plo 0 32) (plo 90 (+ len2 48))
                (list  (abs (- (car bed213) (+ win2len walthic)))
                   (cadr (lp)))
        (list (car (lp)) (+ (cadr bed213) walthic))
                        out13 out12  "c"
                        (prompt " n n2")

        "pline"    (list  (+ (car bedc1) win2len walthic)
           (+ (cadr bedc1) walthic))
           (plo 90 24)
            (list (abs (- (car kitwin3) walthic)) (cadr (lp)))
            (plo 270 24) "c"
                                  )

(prompt "\n DRAWING STAIRCASE...")
(SETQ STEPLEN (distance din11 din12)
      stephgt (/ walhgt 9)
```

```
      stepl1 (/ steplen stephgt)
      )
      (command "color" "c"
               "pline" dinl2  (plo 0 doorllen)
          (plo 270 stepl1) (plo 180 doorllen) "c")
          (repeat (fix (1- stephgt))
                  (command "copy" "si" "1" "0,0"
                   (list 0 (- 0 stepl1) 9)
          )
           )
(prompt "\nDRAWING UPSTAIR ROOM")
(command  "pline" (list (car dinl4) 0)
                  (list (car livl1) 0)
                  (list  (car livl1) (cadr LIVL9))
                    (list (car DINL4) (cadr LIVL9)) "c"
                    "offset" walthic (lP) (plo 325 12)
                                     ""  )

(prompt "\n nDRAWING CARRIAGE...")
(COMMAND "pline" re (plo 270 120)
                    (plo 180 192)
                      (list (car (lp)) walthic)
                      (list 0 walthic) ""
                                       )
(prompt "\nDIMENSION TEXT PRINTING...")
(command "color" "b")
(command "style" "linda" "romant"
    "9" "" "" "" "" "" "text" "c" "0,-60" "0" "CARRIAGE"
          "text" "c" "0,-80" "0" "16 '0 " X 10 '0 " "

           "text" "c" (Polar stohgt3 (dtr 45) (* kitl 0.5))
            "0" "KITCHEN"

            "text" "c" (Polar bedc2 (dtr 45) 84)
                    "0" "BEDROOM1"

            "text" "c" (Polar bed1L1 (dtr -35) 60)
                    "0" "BEDROOM2"

            "text" "c" (Polar LIVL6 (dtr -35) 60)
                    "0" "LIVING ROOM"

            "text" "c" (Polar DINL3 (dtr 215) 60)
                    "0" "DINING ROOM"

           "text"  (Polar OUTL1 (dtr 35) 60)
                    "90" "STORE ROOM1"

       )
```

AUTOLISP COMMAND REFERENCE

ACAD.LSP AUTOLISP file loaded each time you enter the drawing editor.

ACAD.PGP File available with AUTOCAD to define programs accessible from within the drawing editor.

Acad freeram Adjustable area of working memory used by AUTOCAD. Should be set to 24

add +Arithimatic operator used in the format (+a 1) means add 1 to a

AEC Autodesk template that incorporates symbols and advanced AutoLISP programs. Available in Architectural and Mechanical versions

and Basic connector, generally used to connect items of comparisions in if statements (if (and(=ba) (>ac) reads if b=a and a is greater than C

angle AutoLISP command that measures the angle of the known points (angle pnt 1 pnt2)

angtos AutoLISP command that converts a variable stored as radians to a string in another format (set of a (angtos ang 1))

Apostrophe Used to introduce a list (set a (3 57)

Argument A value passed from outside a program to a variable within an AutoLISP program

assoc The autoLISP command that searches for a sublist withing any list. With an entity list, assoc uses the entity code number askey (set q c (assoc 40 b)]

Atom A Single element

AUTOEXEC.BAT A batch file exeucted when the computer is first booted

BASIC A high level language often used to program Microcomputers

Breakpoint A stop point placed in a program for the purpose of debugging

Caddr AutoLISP command that produces the third element in the list

Cadr AutoLISP command that produces the first element in the list

Car AutoLISP command that produces the first element in the list

Cdr AutoLISP command that produces the second and remaining element in a list

Command AutoLISP command that lets you use AutoCAD commands the AutoLISP programs [command "Line" Pnt1 pnt 2"]

Cons AutoLISP command that constructs a new list, with the new element put at the begining of the list [set q d (cons (car ch)]

Coprocessor Numeric processor available as 8087, 80287 and 80387. Dramatically enhances the speed

CROSSING To create a window in AutoCAD in order to select entities. If any part of an entity touches the window, it is selected.

DEBUG To correct program logic and syntax errors.

defun The first command in an AutoLISP program ;defines the name of the function or program.

DISTANCE AutoLISP command that measures the distance between two points.(set a(distance pnt1 pnt2)).

DIVIDE Arithematic operator (/a 3) means a divide 3.

DOS Disk operating system ,usually MS-DOS or PCDOS.

dtr In AutoCAD ,an AutoLISP command that converts the value of angles from degrees to radians.

EDIT AutoCAD command for entering EDLIN text editor from inside AUTOCAD, defined in ACAD.P6P file

EDLIN.COM DOS. Text Line editor

Element A single value in a list

Else The third statement in the if-then- else statement [if (=a5) (set 9 b 6) (set 9 b 7)]

Enteget AutoLISP command that secures an entity list (set 9 b (enteget na)]

Entity The smallest element that may be placed in a drawing with a single command. Typical entities are ARC LINE< CIRCLE POINT e.t.c

ENTITY COORDINATE SYSTEM 9ECS) The USER coordinate system in effect at the time the entity was created.

entmad Autolisp command that updates the drawing data base with the new entity list (entmod b1)

entsel AutoLISP command that selects an entity for VIEW and CHANGE. Only one entity many be selected with this command. (setq a (entsel)]

Equal (=) An autolisp command used only to test quality. It isn't an assignment command (if (=a5)

Explanation AutoLISP shorthand command that prints the value of a variable

Extension The second half of a DOS filename. The required extension for all autoLISP files is LSP .LES1. LSP

Findfile Searches for a requested file. If the file is found, then the complete directory path is returned. If the file is not found then nil is returned.

Flatland A system valuable guarantees AutoLISP programs written under Release 9 will work the same in Release 10. If flatland is set to 0 all new 3D functions of AutoLISP Release 10 are disabled. If Flatland is set to 1 all new 3D functions in AutoLISP Release 10 are enabled.

Function An AutoLISP program

Fuzz This Fuzz factor is a real number set to the number of decimal places of precision required for evlauting equality. If the two numbers are equal within this degree of precisiion then the test for equality will pass

getabgle AutoLISP command that lets you find an angle by pointing to two points or by entering the angle from the keyboard (setq a (getangle)]

getcorner AutoLISP command that asks you to pick a second point on the screen with the mouse or cursor. A window is

dragged vissually across the screen from the first point t the second point (setq Pnt 2 (get corner pnt 1 "Pick second point"0]

getdist AutoLISP command that requests input as a real number either through the keyboard or by pointing

gentenv Secures the enviornment setting under which AutoCAD is operating

getkword AutoLISP command used instead of (getsring) to verify only selected inputs. Used with (initgel-1) (initget-1"Y"N) (seta (getkword "Y or "N)

getorient AutoLISP command that lets you find an angle by pointing to two points or by entering the angle from the keyboard. This command maintains east as 0 degrees (setq a (getorient)]

getpoint AutoLISP command that asks you to specify a pointa on the screen [(setq a (get point " pick a point")]

getreal AutoLISP command that request keyboard input as a real number

getstring AutoLISP command that requests keyboard input as a string

getvar AutoLISP command that produces and AutoCAD system variable [(setq a (getvar "osmodel'0]

global A variable whose value remains when the program ends and is usuable by other programs

grpahics AutoLISP command that shifts the monitor to the drawing editor

greater than > Basic comparative [If (>ab) reads if a is greater than b

handcut Secures the entity name of a specific entity HANDLE. Therefore this command indirectly helps you secure the entity list if you know klthe HANDLE

HANDLES A unique permanent reference number that can be assigned to all entities in the enttity database

if AutoLISP branching (decision) command generally in the form of if-then-else where the (if) statement is followed by two or three additional statementsm, the first being the "condition" the second the "then" statement and the third kthe "else" statments [(if (=a5) (setq b 6) (setq b 7)

if not : Autolisp command that tests to see if a function is already loaded before loading the file in which it is contained. (if (not drawline)(load "LesL"))

initget : Autolisp command that initializes various situations depending on the flag setting. One situation is to turn on the z coordinates for 3D (initget 16).

integer : A counting number, plus or minus without fractions or decimals.

Less than < : Basic comparative (if (<ab) reads of a is less than b.

Lisp: High level language used currently in artificial intelligence and used as the basis for AutoLISP.

Lispheap: The area of working memory used by Autolisp variables and functions should be set to 39000.

Lispsotck: The area of working memkory used by Autolisp for recursive opetaions should be set to 5000.

list: A variable with more than one value.

load: Used to load Autolisp program files into memory so that functions can be executed (load "Les1").

Local : A variable whose value is available to one program.

Loop: Commands repeated and controlled numerous times.

Macro: Automated series of keystrokes and commands that may be incorporated into a menu.

Multiply * : Arithmetic operator that's in the following format: (*a3) means multiply a by 3.

Nesting: The process of Autolisp commands acting on other commands. A command that evaluates another command is said to bes nested.

nil: No value assigned to a variable

Node Space: Autolisp heap

or: Basic connector generally used to connect items of comparison in if statements. (if(or(=ba)(>ac) reads if b=a or a is greater than c.

Parenthesis(): Each Autolisp command and list is surrounded by parentheses.

PATH A DOS: Statement that controls the process of searching for a program or file.

PICK: Refers to choosing a point or object on the screen.

pickset: A special type of variable containing the selection set of entities selected.

polar: Autolisp command that derives a point at a given distance and angle from another knwon point (setq pnt2(polar pnt1 ang1 dist1)).

print1: Autolisp print command used primarily to print data to a file.

princ: Autolisp print command

print: Autolisp print command that adds a line feed.

progn: Autolisp command that groups several autolisp statements into one for use as a single then or else statement. (if=ab)(progn(xxx)(xxx)(xxx))).

prompt: Autolisp command that prints to the screen.

Quotation ": Generally encloses strings when assigned to variables or used as constants.

RAM (Random Access Memory): Variable internal memory used for programs and data.

REM : Remarks statements used to denote a comment in BASIC. The ; is used for this purpose in Autolisp.

repeat: Autolisp loop command that repeats a loop x number of times (repeat 5).

Return: Autolisp produces a value when a command issued, often referred to as returning a value.

rtd: A user written function that converts the value of angles from radius to degrees.

rtos: Autolisp command that converts a number or variable to a string, using user-specified units.

semicolon; : Used to denote a non-executable comment line in Autolisp.

Setq: Autolisp's basic assignemnt command (setq a b) assigns the value of b to a.

setvar: Autolisp command that sets AutoCAD system variables. (setvar "OSMODE"1)

slash /: The slash before variable names in the (defin command denotes local variables)

ssget: Autolisp command that selects entities fro VIEW and CHANGE (setq a (ssget)).

sslength: Autolisp command that determines the number of entites in a selection set. (setq n(sslengtha)).

s::startp: If used as the name of the function in the ACAD.LSP file, the function will automatically begin executing when you enter the drawing editor.

ssname: Autolisp command that secures the name of an entity (setq na(ssname aD')).

strcase: Autolisp command that evaluates all characters in a variable as upper case. (setq a(strcase b)).

string: A group of alphanumeric characters generally enclosed in quptes.

subst: Autolisp command that substitutes one entitye sublist for another within an entity list. (Setq b1(subst(40.02500000) cb)).

substract: Arithmetic operator (-a1) means substract 1 from a.

symbols: Same as variables.

system variables: selectable variables that control centrain AutoCAD defaults.

terpri: Autolisp command that issue a carriage return.

Text Editor: Program used to produce and edit text files. EDLIN.COM is an example of a text editor,

Text file: ASCII file without control codes or other special coding sequences used by word processors.

then: The second statement in the if-then-else statement. (if(=a5)(setqb6)(setqb7)).

trans: Lets you translate the coordinates from ine coordinate system to another.

USER COORDINATE SYSTEM: A coordinate system that may be redefined by the user with a new point of origin and/or new positive derivations for X,Y and Z.

Variable: A combination of letters or letters and numbers used to store other values.

Vmon: Vibral paging functions allowing use of extended or expanded memory for AutoLISP functions,

while: One of these basic loop statements in AutoLISP. The loop will continue while the variable or expression is not nil while.

windowing: The process of creating a window in AutoCAD to select entities. Only full entities within the window are selected.

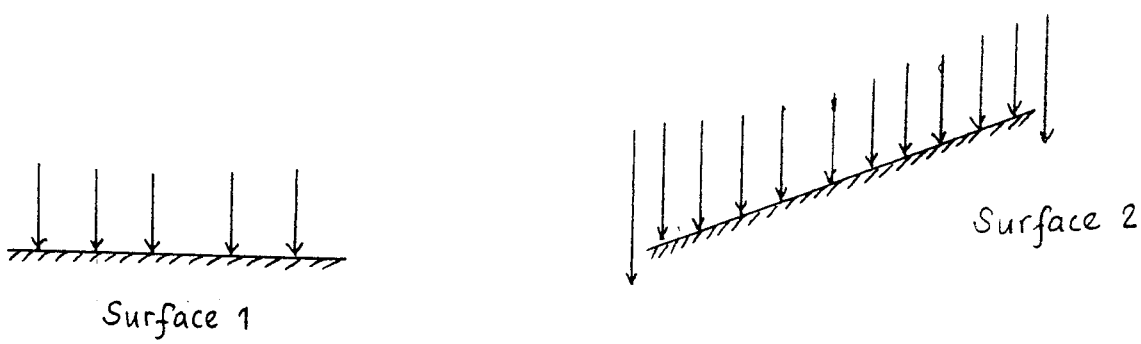
WORLD COORDINATE SYSTEM: A coordinate system with a point of origin where positive X extends to the right and positive Y extends at a 90 degree angle from X toward the top.

3D POLY: An AutoCAD command that produces a 3D polyline.

TECHNICAL NOTES ABOUT SHADING

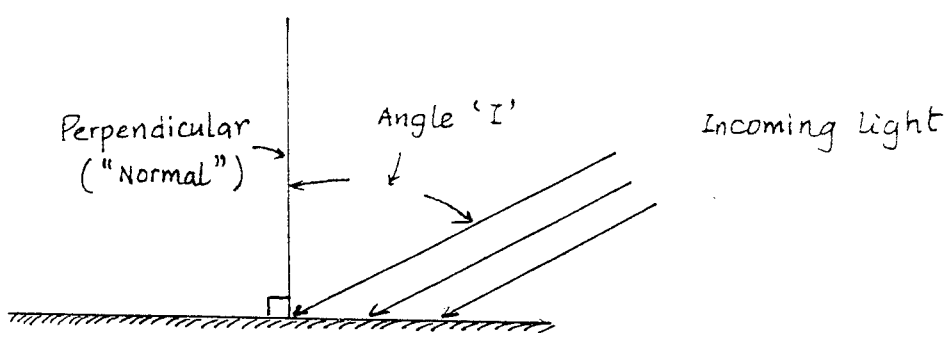
UNDERSTANDING SHADING

All entities in a three-dimensional AUTOCAD drawing consist of one or more polygons. AUTOSHADE creates faceted shadings determining the position of each polygon relative to the source of light. The light has a specific origin, which is the location of light in the drawing.



Surface1 is receiving light directly, surface 2 is receiving it at an angel. Thus surface 1 is receiving more light than surface 2.

AUTOSHADE assigns an exact shade to a surface by determining the angle of the incoming light relative to a line drawn perpendicular to the surface.



The closer angle I is to the perpendicular, the brighter the surface appears.

UNDERSTANDING PERSPECTIVE

AUTOSHADE uses perspective projection, to present visual information in a way that resembles the way we see objects.

In a perspective projection, parallel lines converge at a vanishing point, in a parallel prohection, parallel lines alway appear parallel. The objects and surfaces that are farther away from the eye appears smaller in perspective projection.

THE AUTOSHADE SHADING MODEL

AUTOSHADE produces realistic shading diagrams with no adjustment of the shading parameters such as Z-shading, stretch contrast, ambient factor, specular factor etc. in the shading model dialogue box. But AUTOSHADE permits considerable control over the shading calculation in rendering a drawing.

LIGHT SOURCES IN AUTOSHADE

There are three kinds of light sources supported by AUTOSHADE.

Ambient light: produces a uniform illumination of all surfaces, regardless of their orientation. It is otherwise known as background light.

Point light: radiates light uniformly in all directions.

Direct light: radiates parallel light both along the direction and away from the direction that was given when the light source was defined in AUTOCAD.

We can change the intensities of the lights by using the set light intensities dialogue box.

FORMULA FOR COMPUTING THE SHADING FACTOR

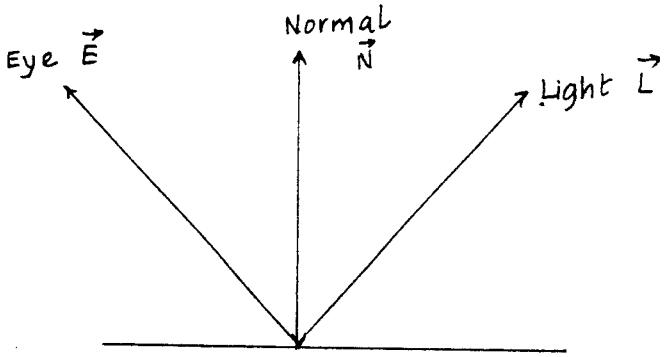
Symbols are assigned to the numbers entered on the shading parameters dialogue box, in order to write the formula, AUTOSHADE uses to compute the shading factor.

A	=	Ambient factor
D	=	Diffuse factor
S	=	Specular factor
X	=	Specular exponent
R	=	Linear lighting
C	=	Inverse constant
n	=	# of lights in the selected scene
I_i	=	intensity of the ith light in the selected scene.

P_f = A partial shade factor computed for a face

$Shade_f$ = final shade factor computed for a face.

For each face in the drawing and each light source, the following are computed.



$\bar{N}$ - unit normal to the face

$\bar{E}$ - unit vector pointing from the centre of the face to the eye or camera position

$\vec{L_i}$ - Vector pointing from the centre of the face to the ith point light source of n.

$\bar{L_i}$ - Unit vector poiting from the centre of the face to the ith point light source of n.

$\bar{T_i}$ - Unit vector pointing from the target to the source of the ith direct light source m.

$\bar{H}$ - $(L+E)/(|L+E|)$ - unit vector pointing half way between the point light vector and the eye vector.

$\bar{J}$ - $(T+E)/(|T+E|)$ - unit vector pointing half way between the direct light vector and the eye vector.

Where the vertical bars indicate the magnitude or length of the vector.

if the camera and the light are on opposite sides of the face and is 1 if the cameral and the light are on the same side of the face.

The intensity of light reflected from each face (the shade)

$$P_f = \frac{1}{n} \frac{I_i(D|L.N|+S|H.N|^X)}{C+R|L|+Q|L||L|} + \frac{1}{m} I_i(D|T.N|+S|J.N|^X)$$

where represents sum of the expression of each of the m direct light sources.

$|L|$ length of vector L

L.N dot product of unit voectors L and Vector N (or) $|L||N|\cos$ (angle between L and N)

If R and Q are zero, AUTOSHADE sets C=1.0, otherwise the value we set is used for C.

When stretch constrast box is checked, final shade factor is computed for each face and then scaled to a # from 0 to 1.0 by the formula.

$$Shade_f = A + (1.0-A)(P_f-\min P)/(\max P-\min P)$$

where, min P minimum value of P_f computed for all faces

max P maximum value of P_f computed for all faces

when stretch contrast is turned off, the final shade factor is computed as follows:

$$Shade_f = A + P_F$$

but $shade_f = 0$ if$[(A+P_f)<0]$ and set to 1

ie. $shade_f = 1.0[(A+P_f)>0]$

FIG 1

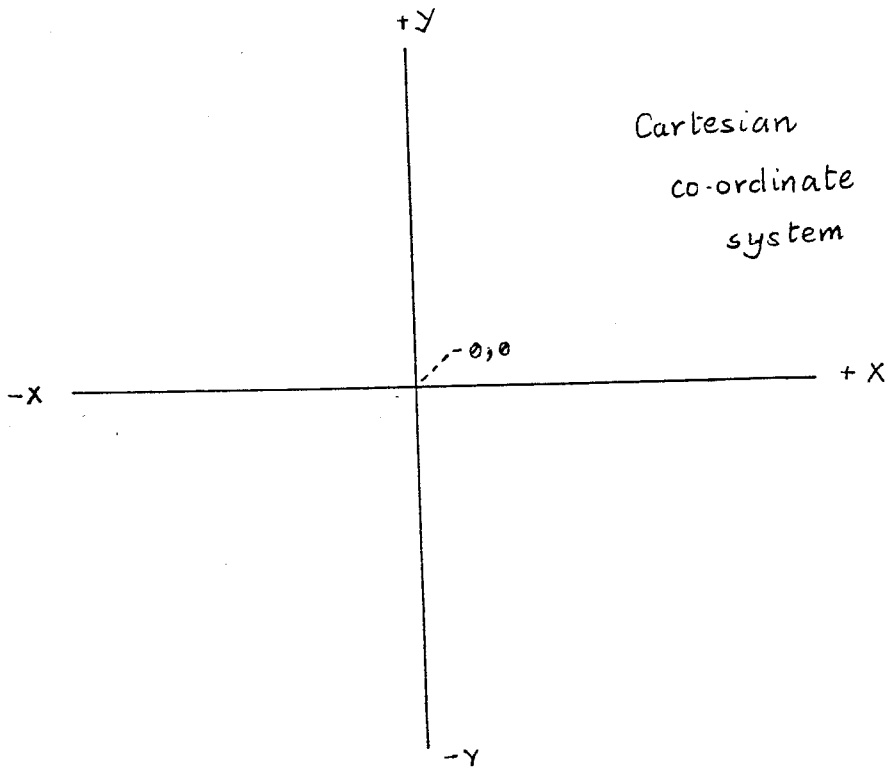
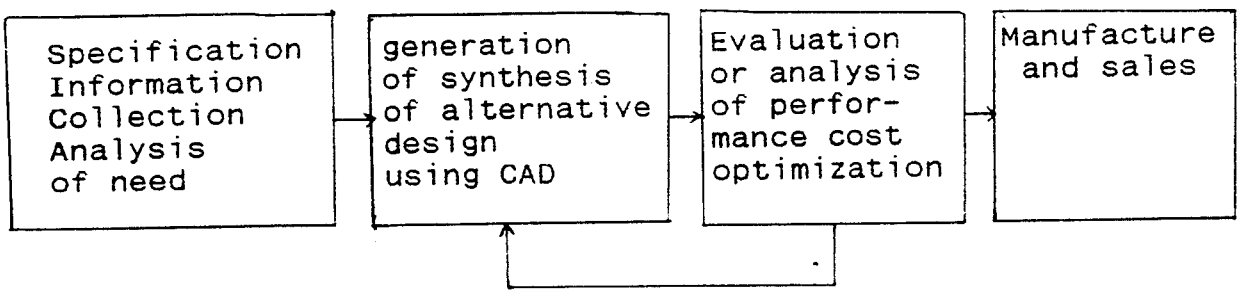
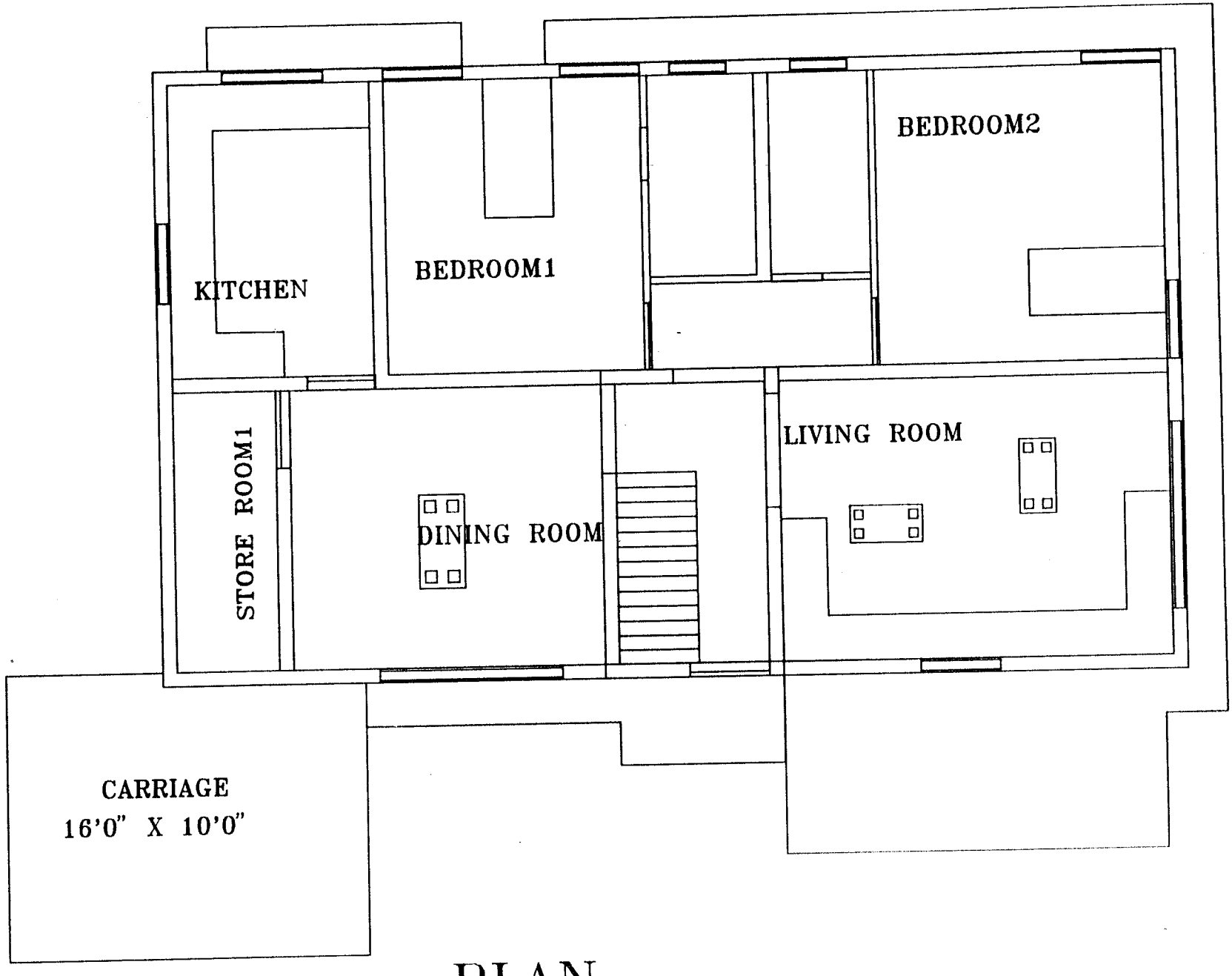


FIG 2


Specification
Information
Collection
Analysis
of need
generation
of synthesis
of alternative
design
using CAD
Evaluation
or analysis
of perfor-
mance cost
optimization
Manufacture
and sales


BEDROOM2
KITCHEN
BEDROOM1
STORE ROOM1
DINING ROOM
LIVING ROOM
CARRIAGE
16'0" X 10'0"

PLAN

ELEVATION