attribute: A property that affects details about how a primitive is drawn, provided in memory or in a file, etc.

primitive: Things to draw, like a point (or dot, or marker), line segment, circle, polygon, character from a font or string of them, an image already in memory or in a file, etc.

**Boldface words** are technical vocabulary whose specific meanings and usage in our subject are parts of the course.

**Primatives and attributes**

To display a 2-D line segment on the screen, the program calls OpenGL functions that can be written in C/C++.

---

For Project I.

Read: HB ch. 1 and 2; Study selected sections there and within ch. 3 and 4.

Introduction to Project I.
to put a window on your desktop to draw the graph. A

terminal window used to start it.

Stop it by deleting the resulting graphics window or typing Control-C.

Run it using the command:

Getting started on LINUX:

\verbatim
\text{ec -T glut -Ttxt -Txf -Tm -e profit.c}
\endverbatim

Compile it using the command:

2. Fix or enhance the program (except for the first time through).

Fix profit.c

1. Type the program or HB sec. 2-9 into a file, give it a name, like

Get your started on LINUX:

\verbatim
\text{ec -T glut -Ttxt -Txf -Tm -e profit.c}
\endverbatim

A \text{X} server is absolutely necessary for

\verbatim
\text{ec -T glut -Ttxt -Txf -Tm -e profit.c}
\endverbatim

They need

\verbatim
\text{ec -T glut -Ttxt -Txf -Tm -e profit.c}
\endverbatim

\verbatim
\text{ec -T glut -Ttxt -Txf -Tm -e profit.c}
\endverbatim

\verbatim
\text{ec -T glut -Ttxt -Txf -Tm -e profit.c}
\endverbatim

\verbatim
\text{ec -T glut -Ttxt -Txf -Tm -e profit.c}
\endverbatim

\verbatim
\text{ec -T glut -Ttxt -Txf -Tm -e profit.c}
\endverbatim

\verbatim
\text{ec -T glut -Ttxt -Txf -Tm -e profit.c}
\endverbatim

\verbatim
\text{ec -T glut -Ttxt -Txf -Tm -e profit.c}
\endverbatim

\verbatim
\text{ec -T glut -Ttxt -Txf -Tm -e profit.c}
\endverbatim

\verbatim
\text{ec -T glut -Ttxt -Txf -Tm -e profit.c}
\endverbatim

\verbatim
\text{ec -T glut -Ttxt -Txf -Tm -e profit.c}
\endverbatim

\verbatim
\text{ec -T glut -Ttxt -Txf -Tm -e profit.c}
\endverbatim

\verbatim
\text{ec -T glut -Ttxt -Txf -Tm -e profit.c}
\endverbatim

\verbatim
\text{ec -T glut -Ttxt -Txf -Tm -e profit.c}
\endverbatim

\verbatim
\text{ec -T glut -Ttxt -Txf -Tm -e profit.c}
\endverbatim

\verbatim
\text{ec -T glut -Ttxt -Txf -Tm -e profit.c}
\endverbatim

\verbatim
\text{ec -T glut -Ttxt -Txf -Tm -e profit.c}
\end{verbatim}
ZERO points. It is trivial for YOU to test if it compiles, and it gives you
and think them. One grading rule we have is „If it doesn’t compile, it gets
Projects will be graded on INTSUNIX using the above command to compile
repeat steps 2-4.
then each individual addition.
which should work for our graphics programs with no special efforts.
In one of the user rooms, there are also a few Sun Ultra workstations
and password to log into INTSUNIX in the SSH window.

SSH choice in the „XWin-32“ submenu. Use your VLabany NETID

(q) Repeat all the above mean instructions except, in the end, activate the
to the „start X server”

XWin32“ submenu. From the submenu, activate the „Start Programs“ menu, go to

(a) Starting from the „Start” button, activate the „Programs” menu, go to
absolutely required steps:

(TC-3 or TC-4) Log on to a Windows PC, and do the next two
If you don’t know a better way, go to VLabany’s public user rooms

University at Albany Computer Science Dept.
the Cygwin software, you can have a UNIX-like environment with gcc that
from the Internet and install the OpenGL and GLUT libraries. Finally, with
you can also use MS Windows if (1) have a C compiler and (2) download
application software with your own computer.
and run graphics applications on Linux, and (2) compile and run graphics
With a Linux installation, you can both (1) connect to Linux and run
for gcc, XFree86 DEVELOPMENT, and GLUT.
Current distributions provide g++ and OpenGL if you install the packages
science students to install, learn about, and use in their own computers.
UNIX variants, particularly Linux, but also BSD, are helpful for computer
you are welcome (and encouraged) to develop projects on your own, more
without any changes at all.
the unexpected syntax or other failures, THEN turn in the program file
space, or adding or editing a comment, then you RECOMPILE the file and
expect that if you make any change WHATSOEVER, even a single
expect you to do this if you want any credit for the work you hand in.
the chance to use the error messages to find and fix the syntax errors. We
Linux or other Unix-like environment in your PC.

But if you can, it is more efficient and instructive to run your own native PC's.

That’s why we can use LTSSNIX the same way as from the TCG user room runs on top of Windows. Also, Cygwin-X provides a free X-server with
required books and other important policies and information.

for details about the textbooks you might NOT have to buy all the

See the syllabus (distributed on paper in Lect. 2 and posted on the Web)
You can get on the Web the free personal learning edition of Maya. Maya, which is current technology in computer generated movie studios, does the basics of the more fully featured but expensive and proprietary free/open source 3-d modeling, rendering, and animation application. It will cover and require you do 1-2 tutorials on basic usage of Blender, an

But it will help, I hope.

XSi/Illustrator, AutoCad or any other graphics application software.

Becoming a skilled user of Blender/Maya, GIMP/Photoshop, of computer graphics in today's world.

What people do with computer graphics (but do teach this to yourself)

Creative Graphic Design, i.e., an art course.

What this course is NOT:

University at Albany Computer Science Dept.
which are affected by the point's attributes.

However, the word point is used in Graphics Library specifications for a

This idea definitely has a point!

students say it is drawn the same size no matter how far away from it you

Some "point" is a mathematical abstraction. It has zero volume or size. Some

University at Albany, Computer Science Dept.
void terminate ( void )
{
    glOrtho2D ( 0.0, 200.0, 0.0, 150.0 );
    glMatrixMode ( GL_PROJECTION );
    glLoadIdentity ( 1.0, 0.0, 1.0, 0.0, 0.0, 0.0 );
}

#include <GL/gl.h>

/*
Its command window.

Stop it by deteting its window or Control-C in
a.out

Run it with the command:
    gcc -Telut -Igl -Igl -Ixt -Ixt -Iw -lm -g main.c
Compile/link on VANDERY’S ITS Unix by the command:
    *Our First OpenGL Program main.c
*/
displayFunc(Ti
usegment);

int();

createWindow("An example OpenGL Program");

initWindowSize(400, 500);

initWindowPosition(50, 100);

initDisplayMode(GLUT_SINGLE | GLUT_RGB);

initWindow(0, 0, 1, 1);

}

}
function named T نفسهment and STORES this address in a (function
executed, the GLUT library function receives the ADDRESS of the
Here is what happens: When glutDisplayFunc (T نفسهment) ;
is
the line.

in T Himselfment ( ) . HB purposely omitted the ( ) after T Himselfment in
in T Himselfment ( ) . A possibly empty argument list, as
name of the function FOLLOWED BY a possibly empty argument list, is
function, NOT a function call. (A function call parameter is coded by the
parameter T Himselfment to glutDisplayFunc ( ) ;
In HB first program, when the line
GUI element,
invites you to write the code that will be run WHEN the user activates the
elements (GUI) elements. Graphically. Then, for each GUI element, it
Visual Basic Programming, the Visual Studio invites you to build graphical
The „control structure” is new to most of you. It is event driven. In

on pages 76-80 in HB.

We discussed this program line-by-line...you can read a similar discussion
University at Albany Computer Science Dept.
The function `getEventLoop()` never returns. It runs a loop that repeatedly checks if any event has occurred. The operation to check for an event of that kind of event, determines what kind it is, and then calls any call-back functions registered whenever an event occurs, the window system situation is called an event. Whenever an event occurs, the window system previously displayed window gets "uncovered", or "de-iconoed". This the window system detects when a window is first displayed, and when the "display" event.

Registering a callback for `getDispatchFunc()` is called a call-back function, and the operation coded by `mousemove()` is called `mousemoveArg`.

Assembly language, the C1333 pre-requirement of this course, allows real-time access to stored addresses as the argument in a function call instruction. Then, the primitives must be drawn in the display window, the library software will then do the graphical display. Some time in the future, when the graphics system detects that graphical

TIME.

EVENT DISPATCHFUNC() returns. `mousemove()` is NOT CALLED AT THIS TIME.
call) So `count()` never returns.
and makes the program exit immediately by calling the `exit()` system
When the initial window is destroyed, the `count()` detects this

Kind.
loop classifiers what Kind it is and calls any callbacks registered for that
event will wait if there was no event yet. When an event is directed, the