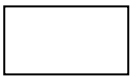
OpenGL is a vast topic involving many of the underlying concept of Computer Graphics - in ITGM 315 we will take a cursory look at OpenGL and use it as a tool to implement a visual interface with the final assignment that you will develop in this class. Most of the games you develop will be text-based. Please refer to the examples in the Materials/OpenGL folder in the dropbox. Some of the commands below are used in these examples.

Beginning with OpenGL:

#include <gl/glut.h>	already includes #include <windows.h> #include <gl/gl.h> #include <gl/glu.h>
glut.Init(&argc, argv);	glut initialization
glutCreateWindow(“The Window name “);	gives the display window a name
glutDisplayFunc(someDisplay);	specifies what the display window is to contain someDisplay uses OpenGL functions glutDisplayFunc assigns the picture to the display window
glutMainLoop();	display windows that have been created are activated. It displays the initial graphics and puts the program into an infinite loop that checks for devices such as mouse or keyboard
glutInitWindowPosition(50, 100);	place the window that is to be opened in a position, say 50 units in x and 100 units in y relative to the top left corner
glutInitWindowSize(400, 300);	initialize the size of the window say to 400 by 300
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB)	buffering, then logical OR symbol, then choice of color modes
glClearColor(1.0, 1.0, 1.0, 0.0); glClear(GL_COLOR_BUFFER_BIT);	sets the background rgba color displays the background color This would set the display to have a white bkgd

glColor3f(1.0, 1.0, 0.0);	to set object colors
glMatrixMode(GL_PROJECTION) gluOrtho2D(0.0, 200.0, 0.0, 150.0);	need to tell OpenGL how we want to "project" our picture. ie. gluOrtho2D(0.0, 200.0, 0.0, 150.0); (0,0)  (200,150)

glBegin(someType)
glEnd();
	these are used for various drawing such as
	glBegin(GL_LINES);
	glVertex2i(180, 15);
	glVertex2i(10, 145);
	glEnd();
	
	In addition to lines there are GL_POLYGON, GL_TRIANGLES, GL_TRIANGLE_STRIP, GL_TRIANGLE_FAN, GL_QUADS
	
	there are also shorthand functions, for example
	glRecti (x1, y1, x2, y2);

glutWireCube(size);
glutSolidCube(size);
glutWireSphere(radius, slices, stacks);
glutSolidSphere(radius, slices, stacks);
glPointSize(size);
glLineWidth(width);
	useful glut geometry calls

glFlush();	forces all buffers to be flushed
glloadIdentity();	load the identity matrix
glutReshapeFunc(void(*func)(int w, int h))	indicates what action should be taken when the window is resized
glutKeyBoardFunc, glutMouseFunc, glutMotionFunc	keyboard and mouse button link to invoke function, and call back when mouse is moved while button pressed
many more, but this will get us started ….	