BRAC UNIVERSITY

Inspiring Excellence

Department of Computer Science and Engineering

Course Code: CSE 423	Credits: 1.5
Course Name: Computer Graphics	Semester: Fall 2020

Lab 01

Introducing and Installing Java OpenGL (JOGL) in the Eclipse IDE

I. Topic Overview:

Java binding for OpenGL (JOGL) is an open source library for binding OpenGL graphics in Java. Since 2010, it has been an independent open source project under a BSD license. It is the reference implementation for Java Bindings for OpenGL. JOGL allows access to most OpenGL features available to C language programs through the use of the Java Native Interface (JNI). It offers access to both the standard GL* functions along with the GLU* functions; however the OpenGL Utility Toolkit (GLUT) library is not available for window-system related calls, as Java has its own windowing systems: Abstract Window Toolkit (AWT), Swing, and some extensions. Therefore, JOGL is an excellent choice to demonstrate the basic graphical rendering methodologies and transformations.

Throughout the semester we will use JOGL to implement the basic rendering techniques.

II. Lesson Fit:

a. This lecture provides a basic understanding of JOGL library and its features.
b. Students will be introduced with JOGL and its working mechanism.
c. Students have to add JOGL into their Java IDE i.e. Eclipse
d. Students will be introduced with the basic methods and functionalities of JOGL
e. Students will be taught how to draw dots using JOGL

III. Learning Outcome:

After this lecture, the students will be able to:

a. To integrate OpenGL bindings in their Java applications.
b. Set up the basic Java OpenGL system in Java environment.
c. Implement the fundamental methods of JOGL library.
d. Draw simple dots using JOGL library

IV. Anticipated Challenges and Possible Solutions

a. Students will face difficulty to add JOGL library into their IDE.

Solutions:

1. All the necessary JOGL files are kept under a single folder in TSR
2. A detailed instruction about how to incorporate additional library files into a Java IDE has been prepared as a guidance for students.
3. A hands on demonstration of the whole procedure will definitely make the whole process crystal clear.

V. Acceptance and Evaluation

Students will start setting up their programming environment for OpenGL based graphics application development. Students will learn how to set up the environment and follow accordingly. After successfully integrating OpenGL bindings in their java applications, they have to implement the algorithm for drawing a pixel using the JOGL library methods. Each and every student will be evaluated upon their performance. If a student fails to complete the implementation he/she will have to complete by that night and show it to the class teacher on his/her consultation period. If a student also fails to do that then instead of 10 we will evaluate the student on 5.

1. Lab evaluation marks: **out of 10**
2. Late Lab evaluation marks: **out of 5**

VI. Activity Details

a. **Hour: 1**

Discussion:

1. **Installing JOGL:**

a. Make sure that the Java Development Kit(JDK) is installed in your system.
b. Make sure that a preferred IDE is installed. We will use Eclipse for our demonstration.
c. Copy the necessary JOGL library files to your machine. You will find the files in the folder under the name “jogamp-all-platforms”. The location is as follows: (tsr->Faculty_Initital->CSE423 Computer Graphics Lab->Lab01)
d. Open the Eclipse IDE.
e. Go to window->preferences->java->buildPath->userLibrary
f. Create a new user library name it JOGL (or something else)
g. Add the following jars to it (you can find these jars in jogamp-all-platforms->jars)
    1. gluegen-rt.jar
    2. gluegen-rt-native-windows-amd64.jar
    3. jogl-all.jar
    4. jogl-all-native-windows-amd64.jar
h. If you are on a windows machine use jars like "gluegen-rt-native-windows-amd64.jar" and if you are on a 32 bit machine use jars like "gluegen-rt-native-linux-i586.jar"
i. Frome now on for every project you create goto project->properties->javaBuildPath->libraries->addLibrary->userLibrary-> add the library you just created

b. **Hour: 2**

Discussion:

1. **Drawing Coordinates on the Canvas:**
    1. For (100 times)
        X= randNumb();

Y= randNumb();

Draw pixel (x,y);

Here, randNumb() should generate a random floating point number between -1 and 1

2. Each point is a 2D coordinate (x,y); where 1>=x>=-1 and 1>=y>=-1

c. **Hour: 3**

Discussion:

1. **Drawing points on the Canvas after reading the coordinates from a text file**
    a. Create a txt file "coordinates.txt"
    b. Populate the file with 20 values between -1 and 1
    c. for (EOF)

        X=readNextValue();

        Y= readNextValue();

        DrawPixel(x,y);

    d. Each line of the file is a 2D coordinate (x,y); where 1>=x>=-1 and 1>=y>=-1

VII. Home tasks

a. Draw multiple coordinates in a straight line to actually represent a straight line.

Lab Activity List

Task 1

Installing JDK

Task 2

Installing IDE

Task 3

Installing JOGL

Task 4

Draw points where coordinates are given manually

Task 5

Draw points where coordinates are read from a text file

Reference

1. **Computer Graphics** by A.P.Godse, D.A.Godse.
2. http://jogamp.org/jogl/