Week 1 Tutorial Solutions

Question 1: Introduction

Go around the class, introduce yourselves and talk about why you're interested in graphics. What's the coolest graphics application you have seen? What would you like to be able to make?

Question 2: java concepts

- What's the difference between an object, a class and an interface?

  An object has state (stored in a chunk of memory allocated for it) and behaviour associated with it and is an instance of a class. A class is like a blue-print for objects. It defines the names and types of fields that an object will have as well as methods for acting on those fields. An interface defines a group of related methods and possibly constants that are not implemented. Both classes and interfaces act like types.

- What does the keyword static mean when applied to a field or method?

  It means the variable or method is shared between all instances of the class. So for a variable, there would be only one copy of the variable, not one for each object. In fact to use static variables and methods you do not even need an object.

- What is an Event Listener? Give some common examples of event listeners you have written in java.

  EventListeners are interfaces defining methods that should be implemented to handle events. Events are typically things like the user moving the mouse or pressing a key etc. We would implement the KeyListener to handle key press events a MouseListener and/or a MouseMotionListener to handle mouse events.

Question 3: Data layout

The Point2DBuffer class represents a buffer of points with a java.nio.FloatBuffer.

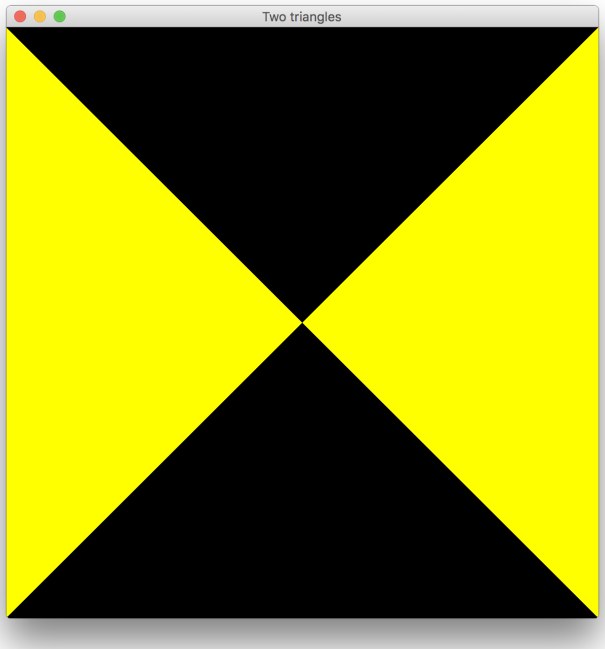
```
public class Point2DBuffer {

    private FloatBuffer floatBuffer;

    private int capacity;

    ...
```

What's the difference between a FloatBuffer and an array of floats (`float[]`)?

An array of floats (`float[]`) is a regular object stored in the Java heap. This means that the Java runtime may move it around or copy it during garbage collection. A FloatBuffer on the other hand is a buffer of floating point values pinned (or fixed) in memory. This means it will not be moved, so is suitable for passing to a third party C library (like OpenGL) that has no knowledge of the Java runtime.

Question 4: Drawing some triangles

The Triangle2D class allows you to create a triangle with a list of 3 points. Write a subclass of Application2D that produces the following image of two black triangles on a yellow background.




```
public class TwoTriangles extends Application2D {

    public TwoTriangles() {
        super("Two triangles", 600, 600);
        setBackground(new Color(1,1,0));
    }

    public static void main(String[] args) {
        TwoTriangles example = new TwoTriangles();
        example.start();
    }

    @Override
    public void display(GL3 gl) {
        super.display(gl);
        List tri1Vertices = new ArrayList();
        tri1Vertices.add(new Point2D(0,0));
        tri1Vertices.add(new Point2D(1,1));
        tri1Vertices.add(new Point2D(-1,1));

        List tri2Vertices = new ArrayList();
        tri2Vertices.add(new Point2D(0,0));
        tri2Vertices.add(new Point2D(-1,-1));
        tri2Vertices.add(new Point2D(1,-1));

        Triangle2D tri1 = new Triangle2D(tri1Vertices);
        Triangle2D tri2 = new Triangle2D(tri2Vertices);
        tri1.draw(gl);
        tri2.draw(gl);
    }

}
```

or, if using the week 2 version of UNSWgraph

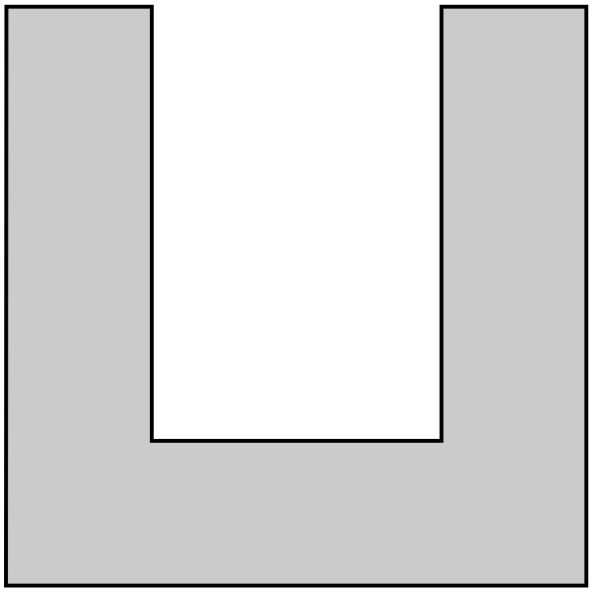
```
public class TwoTriangles extends Application2D {

    public TwoTriangles() {
        super("Two triangles", 600, 600);
        setBackground(new Color(1,1,0));
    }

    public static void main(String[] args) {
        TwoTriangles example = new TwoTriangles();
```

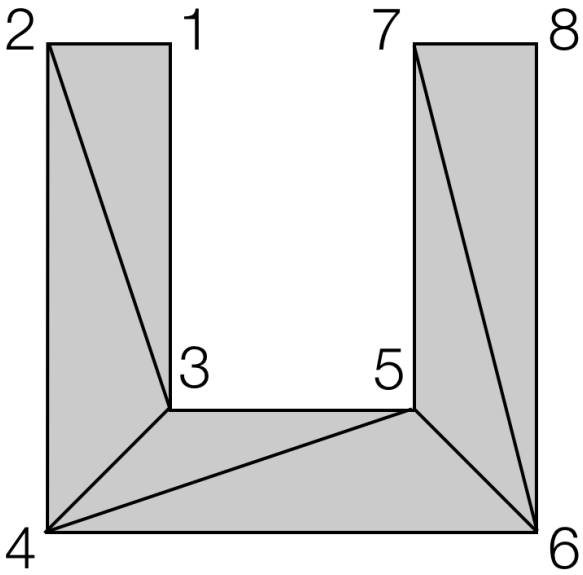
```
        example.start();
    }

    @Override
    public void display(GL3 gl) {
        super.display(gl);
        Triangle2D tri1 = new Triangle2D(0, 0, 1, 1, -1, 1);
        Triangle2D tri2 = new Triangle2D(0, 0, -1, -1, 1, -1);
        tri1.draw(gl);
        tri2.draw(gl);
    }

}
```

Question 4: Triangle strips

In additional to triangle fans, OpenGL also supports triangle strips. Triangle strips are drawn by drawing a triangle between each vertex and the two previous vertices.

How might you represent a 'U' shape like the one below with a triangle strip?





Question 5: lab Discussion

How did you go with the lab? Did you get it finished? Do you have any questions about it?