CIS 636 Interactive Computer Graphics
CIS 736 Computer Graphics
Spring 2011

Homework 1 of 10 (Problem Set)
Warm-up: Scan Conversion, Clipping, and Matrix Transformations

Assigned: Mon 31 Jan 2011
Due: Mon 07 Feb 2011 (before midnight)

The purpose of this homework is to give you practice in applying the mathematical concepts covered in lecture.

This homework is worth a total of 20 points (2%).
Unless otherwise stated, each problem is for both 636 and 736.

Upload an electronic copy of the assignment in PDF form (converted from your word processor, or scanned) to your K-State Online (KSOL) drop box before the due date and time.

References

- VanDam notes, Brown: http://www.cs.brown.edu/courses/cs123/lectures.htm
- Midpoint line algorithm: http://en.wikipedia.org/wiki/Bresenham%27s_line_algorithm
- Midpoint circle algorithm: http://en.wikipedia.org/wiki/Midpoint_circle_algorithm
- Viewing tutorial, I. Wachsmuth, Universität Bielefeld: http://snurl.com/1z0a5 (16, 21 – 23, 41 – 46)
- Angel, E. (2009). *Interactive Computer Graphics: A Top-Down Approach using OpenGL, 5th edition*. Boston, MA: Addison-Wesley.
- Rigid bodies: http://en.wikipedia.org/wiki/Rigid_body
- Linear transformations: http://en.wikipedia.org/wiki/Linear_transformation
- Affine transformations: http://en.wikipedia.org/wiki/Affine_transformation

1. **(30% for 636, 20% for 736) Review: matrix algebra and analytic geometry warm-up.**
   a) Exercise 2.4, p. 24 Eberly 2e. Verify that the determinant of the matrix in Equation (2.22) is 1. Verify that the determinant of the matrix in Equation (2.23) is -1.
   b) Why is multiplication of n×n rotation matrices is **not** commutative for n > 2? (You may use a counterexample, but explain it in your own words.)
   c) Is scaling commutative? Why or why not?
   d) Is scaling a rigid body transformation? Why or why not?
   e) (Problem 5.1, p. 226 Foley *et al.,* 1991) Prove that we can transform a line by transforming its endpoints and then constructing a new line between the transformed endpoints.

2. **(20%) Bresenham's algorithm, *aka* the Midpoint Line Algorithm.** Read Section 2.5, p. 77 – 92, especially the mathematical explanation on p. 78 – 79 and the source code on p. 80 – 81. Trace through some concrete examples as needed, and consult Van Dam's notes on line drawing. (Scan Conversion I: PowerPoint http://bit.ly/7U01dW, PDF http://bit.ly/5OE13l) and on drawing circles and ellipses (Scan Conversion II: PowerPoint http://bit.ly/4PP8vK, PDF http://bit.ly/7Xejnj). Read Van Dam's description of the baseline

"incremental" algorithm (slides 5-10 in Scan Conversion I) and Bresenham's midpoint line algorithm (slides 11-15). Explain the following, **in your own words**:

a) What is costly about the incremental algorithm besides the call to the math library function `Round`? What are the main differences between the incremental and midpoint algorithm (slide 18)? (Hint: see http://bit.ly/8aeUXy)
b) A comment at the bottom of slide 17 reads: "Bresenham's line algorithm is same but doesn't generalize as nicely to circles and ellipses". Look at Scan Conversion II, especially slides 13, 14, and 16, then explain what this means. (Hint: Understand what "`ax`" and "`ay`" mean on p. 80 of the book. Again, compare http://bit.ly/8aeUXy and slide 18 to Bresenham's original algorithm, http://bit.ly/5Nq2aQ.)
c) How many symmetric cases are actually handled by `DrawLine`? That is, what is the size of each wedge of the Cartesian plane (in degrees) handled by the case logic? Consider the possible values of x_0, x_1, y_0, and y_1. How many sign combinations must Bresenham's algorithm handle? What range of and dx and dy values does this represent? What about m = dy / dx? (As an outside reference, see p. 350-351 of Angel 5^e.)
d) What is being pre-computed as a *forward difference* in the midpoint algorithm? Explain in your own words how it is used to calculate which pixel to draw.

3. **(30% for 636, 20% for 736) Definitions.** Write down definitions for the following terms in your own words. Consult the textbook; you may use outside sources, but cite *all* your sources, including the textbook, by page.

- **D**, **U**, **R** – Section 2.1, p. 8 – 10; Section 2.3.1, p. 43 - 48
- Affine transformation (distinguish from **linear** transformation) – Section 2.2.1 & Section 2.2.2
- World space (*aka* world coordinates) – Section 2.3.3, p. 48 – 50; Section 2.3.7, p. 58
- View/camera/eye (*aka* view/camera/eye coordinates) – Section 2.3.4
- Rasterization – Section 2.5

4. **(20%) 3-D Modelview Transformation.** 736 only. Explain the difference between model and world coordinates, as defined in Section 2.3, and explain the purpose of the modelview transformation (MVT) in your own words.

5. **(20%) Clipping.** See Figure 3.2, p. 157. How many sides can the polygon that results from clipping a triangle by a rectangle have? Discuss all possible cases including trivial and boundary cases.

Extra credit (5% each):

- **EC1.** Consider matrix transformations used in animation. Describe, in your own words, a use of the *shear transformation*.
- **EC2.** p. 79 of Eberly 2^e reads "The special cases of horizontal, vertical, and diagonal lines can be factored out if desired." Explain how this would be done, exactly what code would be moved, and why.

Class Participation (required):

Select a term project topic by Fri 18 Feb 2010. Post a brief discussion of the topic you are considering to the Term Project thread in the Discussions message board before you finalize your choice, and ask any questions you like. State your choice by e-mail to CIS736TA-L@listserv.ksu.edu.