

Assignment 1—Hello 2D World & Arbitrary Rotations

Due: Saturday September 12 at 7am

1. Implement a 3-D vector class that stores 3 floats and has the following functionality:
 - a. Constructor from 3 floats
 - b. Read/write access through square brackets operator “[]”
 - c. Addition of two vectors
 - d. Subtraction of two vectors
 - e. Dot product
 - f. Cross product
 - g. Multiplication by scalar
 - h. Division by scalar
 - i. Normalization
 - j. Length computation
 - k. Rotation of “this” point about arbitrary axis
 - l. Rotation of “this” vector about arbitrary direction
 - m. Output to ostream, input from istream
2. Implement a 3x3 matrix class that stores 3 3-D vectors as rows and has the following functionality:
 - a. Constructor from 3 3-D vectors
 - b. Read/write access to rows through square brackets operator “[]”
 - c. Function to get column
 - d. Function to set column
 - e. Multiplication with 3-D vector
 - f. Multiplication with another matrix
 - g. Inversion
 - h. Transposition
 - i. Function to set matrix as rotation about principal axis by theta degrees
 - j. Output to ostream, input from istream
3. Implement a frame buffer class that stores unsigned int pixels and the resolution of the frame buffer and that has the following functionality:
 - a. Constructor from input resolution
 - b. Set all pixels to given color
 - c. Set one pixel to given color
 - d. Drawing of axis aligned rectangle
 - e. Drawing of 2D triangle
 - f. Drawing of circle
 - g. Load / save from tiff file

4. Demonstrate your code.
 - a. Choose a point and an arbitrary axis and rotate the point full circle by 1 degree increments (i.e. 360 steps). Show the point with coordinates (x, y, z) in the framebuffer at location (u, v) , where $u = x$, and $v = y$. Draw the point as a circle with a radius of 5 pixels. Choose the point and the axis such that the point stays on screen at all times, and such that the trajectory of the point is an ellipse (and not a segment or a circle).
 - b. Render a 12s 30Hz 720p video sequence illustrating your rotating point. The video will have one frame for each position, i.e. $12 \times 30 = 360$ frames in total. The video file should be in a popular format. Use the video making software of your choice.
5. Turn in via blackboard one zip archive that contains
 - a. Source code
 - b. Executable
 - c. Video file