CSE 167

Discussion 01 ft. Yining
10/02/2017

Announcements

- Private posts are for emergencies or a question that explicitly shows a part of your code
- Office hour schedule is up on Piazza
- Project 1 is due next Friday

Contents

- Starter code overview
- Linear algebra
  - Homogeneous coordinate
  - MVP matrices
  - Matrix multiplication
- glm functions

Starter code overview: OBJObject

- What is an “object”?
    - A “thing” that you would like to place in your scene
- Where to create an object?
    - Hopefully, the same place where cube was created
- What functions need to be in OBJObject?
    - parse(): for parsing .obj files
        - Make sure you are NOT parsing vertex normals as vertices!!!
        - The number of vertices specified in .obj file should be the same as the number of parsed vertex lines
    - some_function(): for spinning the object
    - Matrix operation helper functions: for translation, scaling, orbiting, and resetting
        - Pay attention to the matrix multiplication order!!!

```
####
#
# OBJ File Generated by Meshlab
#
####
# Object bunny_n.obj
#
# Vertices: 34835
# Faces: 69666
#
####
vn -1.345425 -4.896516 -1.808985
v 0.296502 -0.907931 0.450151 0.752941 0.752941 0.752941
vn -2.353551 -5.669126 -0.381383
v 0.315114 -0.913622 0.435867 0.752941 0.752941 0.752941
vn -3.331857 -4.821201 -1.723500
v 0.324517 -0.920404 0.443869 0.752941 0.752941 0.752941
```

Starter code overview: Window

- Window::initialize_objects()
    - DON'T parse an object every time you switch, parse them all at once in the beginning
    - bear.obj alone has 866,394 vertices, if you parse it every time you switch back to bear, it's going to take a lot of time...
- Window::idle_callback()
    - You need to spin bunny/dragon/bear constantly; that means you need to do something here
- Window::display_callback()
    - Here, you will call the draw() function
- Window::key_callback()
    - You specify what to do when a certain key is pressed

Linear algebra: homogenous coordinate

- (xz, yz, z) is called a set of homogenous coordinates of (x, y)
  - Note that since z is nonzero, (xz, yz, z) can also be written as (x, y, 1)
- What does this mean geometrically?
  - Chalkboard time
- Why do we need this?
  - Given (x, y, z) we can extend this to a homogenous coordinate (x, y, z, w) with w = 1
  - This means a 3D point can be represented as a 4D vector
  - Then we can multiply 4 X 4 matrices and 4 X 1 vectors
  - So what?

Linear algebra: homogenous coordinate

Let R_y be a rotation matrix with respect to the y-axis and $\theta = 90$:

$$R_y = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

And let T be a translation vector:

$$T = \begin{bmatrix} 0 \\ 2 \\ 3 \end{bmatrix}$$

To rotate a point $A(-1, 0, 0)$ by 90 degrees and then translate by T:

$$A' = R_y \cdot A + T$$

In other words,

$$A' = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 4 \end{bmatrix}$$

So this is a combination of matrix multiplication AND matrix addition.

Linear algebra: homogenous coordinate

But what if you started off with 4×4 matrices and 4×1 vector in the first place?

$$R_y = \begin{bmatrix} cos\theta & 0 & sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -sin\theta & 0 & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

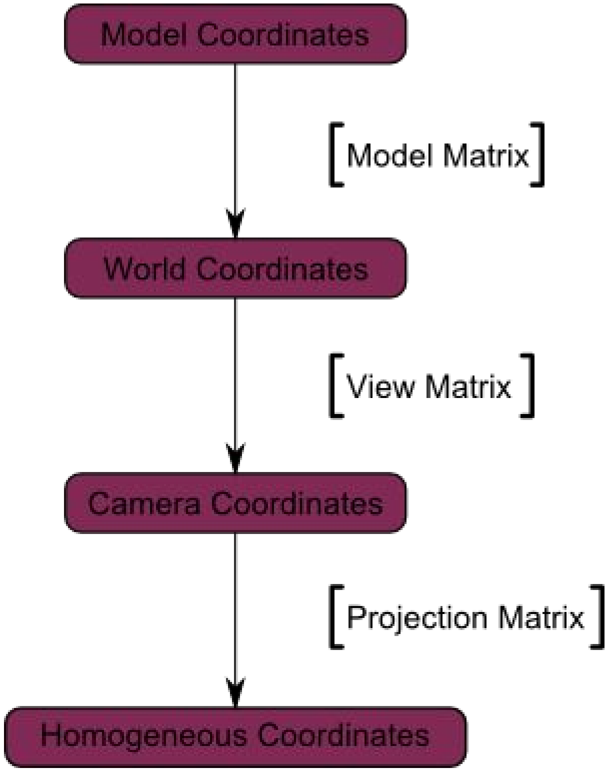
And let $A(-1, 0, 0, 1)$. Then,

$$A' = T \cdot R_y \cdot A$$

In other words, using homogenous coordinate and 4×4 transformation matrices, we can transform a point by just a series of matrix multiplications.

Linear algebra: MVP matrices

- M: place the object
- V: place the camera
- P: set up the camera
- Chalkboard time



http://www.opengl-tutorial.org/beginners-tutorials/tutorial-3-matrices/

Linear algebra: matrix multiplication

- In what order do we multiply?
  - Let's say I have a transformation matrix M that was the result of the previous example
  - If I want to rotate an object with respect to the world's y-axis, which one is right?
    - M = R * M?
    - M = M * R?
  - If I want to rotate an object with respect to its own y-axis again, which one is right?
    - M = R * M?
    - M = M * R?
- If you understood this part, you now know what M is actually the "toWorld" matrix in the starter code

glm functions

- glm::translate()
- glm::rotate()
- glm::scale()

- glm::lookAt()

- glm::perspective()