

CS 432 – Interactive Computer Graphics

Assignment 2 – Transformations, Animation and Interaction

The goals of this assignment are

1. Allow for Animation
2. Allow for Interaction

Requirements

For each programming assignment you should submit a **zip file** containing the code (and MSVS/Xcode project/solution files) and a **report** stating: what you did; how you did it; any particular features you want to draw attention to; any problems with the program that you know about.

In addition:

1. Your code must be original. You may discuss approaches with your colleagues but not how to code it.
2. You must use GLSL (shaders), vertex buffer objects (VBOs), and vertex array objects (VAOs) and **may not** use the old-style approach (glVertex*, glBegin, etc...)
3. Make sure your program complies and runs on either a Windows 7/8/10 machine with VC++, freeglut and glew or on a Mac with OSX and Xcode

What to Submit

For your submission, zip together the following.

- A README file with instructions on how to compile and run your program(s). This should also include the environment in which you are developing and running (OS version, Developing Environment version, Shader Version) and any issues/features you want to draw attention to.
- Your source files
- Your file(s) containing rules for compiling and running your program(s) (i.e. project files, make files, etc...)
- **A short screencast video that you talk over (5 minutes max)**

Description

In this assignment you will be building your first animated and interactive program. The idea is as follows:

- You start with an empty “canvas”
- When you left click on the window, a red square is generated centered at that location.
- When you right click on the window, a blue triangle is generated centered at that location.
- Hitting space bar toggles an animation:
 - Objects (squares and triangles) rotate around their own origins.
 - Objects brightness range from 0 to 1.0 over time
- Hitting ‘q’ or ‘Q’ quits the program.

Now that we are having several objects in a scene, several of which are instances of the same type of object, we want to start structuring our code well. As such:

- Each object (triangle or square) should be exactly that, an instance of a C++ Object.
- You may want all drawable objects (for this assignment this will be squares and triangles) to be derived from a Drawable class
 - Then your main code could just contain a vector of pointers to homogenous Drawable objects
 - Of course then you'll want to think about things like virtual methods...
- Make sure when your program exits, that it calls the objects' destructors, which in turn delete the VBOs and VAOs.
- **The BBlearn starter code should help you get, well, started, with structuring your code base this way.**

Additional Notes:

- Your window should be 500x500 pixels.
- Each object should have at least its own instance of
 - Shader program
 - Vertex Array Object (VAO)
 - Vertex Buffer Object (VBO)
- And each class should have at least
 - Constructor
 - Method through which to draw the object.

Grading Scheme (100pts)+5e.c.:

1. Your code is organized into objects: (20pts)
2. Your code allows for dynamic adding of objects via left/right clicking: (20pts)
3. Your code adds objects to the correct locations: (10pts)
4. Your code does rotation correctly: (15pts)
5. Your code correctly toggles animation via the keyboard. (5pts)
6. Your code correctly varies the brightness of the object's color based on the timer. (5pt)
7. Your code correctly destroys objects in order to avoid memory leaks. (5pt)
8. All computations are done in homogenous coordinates. (10pts)
9. Transformations are done in the shader program via passing in a model matrix. (10pts)
10. Extra Credit: If the shift button is held down when a mouse button is clicked then the object is drawn multi-colored. (5pts)

