CS680 A1 Graduate Introduction to Computer Graphics (Fall 2021) Assignments

Upload Assignment: Programming Assignment 2: 3D Modeling & Transformations

Upload Assignment: Programming Assignment 2: 3D Modeling & Transformations

ASSIGNMENT INFORMATION

Due Date	Points Possible
Tuesday, October 13, 2020 12:00 PM	**100**

1. Overview

This programming assignment is meant to acquaint you with 3D modeling and model transformation. You will implement a creature model and make it move by performing transformations on its joints. You will also need to submit a simple reference image or concept drawing with your code.

Basic Requirements

Before coding, please have a look at file "Component.py"(TOREAD 1): we recommend calling functions in the Component class in your keyboard event functions to make the joint/body of your creature move. Please be patient and have a look at what the Component class provides to make your creature move during running the program.

After reading the code in the "Component.py" file, you should start implementing certain features in certain blocks/files in the skeleton code.

1. TODO 1 in "DisplayableCube.py": please implement your own geometric object classes that later its instances add up to be your creatures. In addition to the cube class we provide you, you could also consider building the geometric object with GLU functions: Cylinder, Cone(no caps), Sphere, Disk Please create another .py file to write your needed geometric object classes other than cube.

2. TODO 2 in "ModelLinkage.py": please implement your own creature class by mimicking the form of the ModelLinkage class. You will need to create a few instances of what'd been defined in TODO 1 and connected them as a whole. Use them to construct a 3D insect, spider, or scorpion model from cylinders and/or ellipsoids. The model need not be anatomically accurate, but each limb should have at least three joints. You can model the body as a single ellipsoid. You

need to figure out how to use hierarchy structure to store components and set them in proper position through chaining transformation. Remember to modularize your design to one class (e.g. like Arm.java in example code), you might need to reuse them in next programming assignment.

3. TODO 3 in "Sketch.py": After building your body and limb classes, you need to create the instances of them in Sketch.py -> __init__() method.

4. TODO 4 in "Sketch.py" or "ModelLinkage.py": Set up the joint behaviors of your creature after finishing TODO 3. Please refer to TOREAD 1 to see what functions are already available to you. Limit the angles at each joint within reasonable ranges so that creature's legs, neck, head and any other parts don't intersect or bend in unnatural ways! The orientation of joint rotations for the left and right limbs should mirror each other. Gimbal lock should be avoided.

5. TODO 5 in "Sketch.py": Set up the keyboard events which could make your creature act. Please also add a multi-select feature to the interface, so you can control several joints at the same time. Also please set up 5 different poses for your creatures as test cases. After that, please check your keyboard events and test cases to see if they work as expected and satisfy our requirements.

6. TODO 6 in "Sketch.py": Implement an eye that always looks at the position of the mouse when no rotation of the camera is performed. An eye should consists of movable pupil and still sclera.

Extra Credits

7. BONUS1 in "DisplayCube.py": Implement the texture mapping that binds texture images to your geometric objects.
8. BONUS2 in "Sketch.py": Use quaternion to implement eyeball rotation.

Programming Style

9. For any modified or newly added source file, you should include a brief explanation about what you did in this file in the file heading and add your name to the author list. Your code should be readable with sufficient comments. You should use consistent variable naming and keep reasonable indentation. In python, we prefer to use reStructuredText format docstring, for more details about this format, please check **here**.

Model Requirements

In addition to roughly following the reference image/drawing you will provide, your model should be suitably complex:

- at least 6 limbs required (tails and wings count)
- each limb must have at least 3 joints
- at least two colors should be used
- at least two shape types must be used

- a pair of opposing limbs must be present (to test mirrored motions)

2. Resources

2.1 Start code

A Python Program skeleton PA2.zip, which includes basic classes, methods, and main pipeline, is provided for you. You are expected to complete the sketch program by completing/modifying Sketch.py. There are comments in the skeleton code that will help guide you in writing your own subroutines. Some of them are noted as "TODO" or "BONUS" which suggest you should complete the corresponding block.

2.2 Environment Setup

Installing the appropriate programming environment should be covered in a lab session. For more step-by-step instructions, please check the environment setup page.

2.3 User Interface

The user interface to the program is provided through mouse buttons and keyboard keys:

- **ENTER/RETURN**: Cycle through components
- **LEFT-ARROW, RIGHT-ARROW**: Iterate through different rotation axis for the currently selected component
- **UP-ARROW**, mouse **SCROLL-UP**: Increase selected component's rotation angle along the selected axis
- **DOWN-ARROW**, mouse **SCROLL-DOWN**: Decrease selected component's rotation angle along the selected axis
- mouse **Left-Drag**: change the viewing angle
- **r**: reset the viewing angle
- **R**: reset everything in the scene

After modifications, your interface may be different from the example program provided here. Don't forget to add a multi-select feature so multiple joints can be changed simultaneously.

2.4 Demo

We prepare a demo video for you. Even though the "creature" in the video doesn't fully satisfy our requirements, this is still good to show how the multi-select works and what will be like if you set the rotation limit correctly.

0:00 / 2:38

3. Submission (due by 11:59 PM, Tuesday, 10/12)

3.1 Source Code

Your program's source files are to be submitted electronically on Gradescope. Please wrap everything in your project folder to a zip file and submit it. The code you submit should conform to the program assignment guidelines.

3.2 Demo

Part of your grade for this programming assignment will be based on your giving a short demo (5 minutes) during the CS480/680 scheduled labs following the assignment due date. You will be expected to talk about how your program works.

3.3 Reference image or drawing

Please submit a reference image or drawing of your design. This can be relatively simple, and is simply to motivate some modeling with intent.

4. Grading

Task
3D Creature model constructed satisfies requirements for shapes, hi
Proper rotation at the joints
Limit rotation so that limbs do not bend in unnatural ways
Your predefined 5 creature poses work properly
The creature design is similar to your reference image/drawing
Eye movement
Eye movement with quaternions
Texture mapping
Programming style (See Assignments > General Guidelines)

5. Code Distribution Policy

You acknowledge this code is only for the course learning purpose. You should never distribute any part of this assignment, especially the completed version, to any publicly accessible website or open repository without our permission. Keep the code in your local computer or private repository is allowed. You should never share, sell, gift or copy the code in any format with any other person without our permission.