CS250 A3

Worth: **12.5 points**
Due Date: **03/27/2005**
Due Time: **12 noon**
Hardcopy: **Place your hardcopy version in the CS250 submission box located in the front office. Don't forget to sign the declaration on page 4 of this document.**
Softcopy: **Copy your zipped softcopy version to your CS250A/B network folder.**

Topics

- Bounding Volumes
- Culling
- Triangle Clipping using Sutherland-Hodgeman algorithm modified for "thick" planes.

CS250_A3 Framework

- This framework has been designed and implemented to provide a code base for CS250 students in implementing their assignments.
- It implements a complete software-based graphics pipe. For each assignment, certain features previously implemented by us will be removed and you will be assigned the task of implementing these features. The software renderer is included as a library file in the project: ***GfxLib.lib.*** It will be linked to your contribution to create an executable.
- Please note that this framework has been designed with the single intent of providing an educational tool. It is not suitable for use as a high-performance software renderer for applications such as games.
- To report bugs and for answers to any questions you might have about the framework, contact the CS250 TA: Patrick Laukaitis (plaukait@digipen.edu)
- ***It is required that all CS250 students use this framework for implementing, testing, and submitting CS250 A3.***
- The framework uses the right-handed coordinate frame explained in class: X-axis is oriented to the right; Y-axis is oriented into the paper; Z-axis is oriented vertically up.
- A3 requires that you implement object culling and triangle clipping.
- You can compare your results with the correct A3 implementation using ***CS250_A3 Sample.exe***.
- ***CS250_A3 Sample.exe*** and the project use the following controls:
  - 1 through 5: Switch between displayed models.
  - 'V': Wireframe and fill mode toggle.
  - ESC: Resets view to default mode
  - SPACE BAR: Mouse-Look off and on toggle.

Statement

Do the following:

- First, open the Visual Studio Solution Object containing A3's project: ***CS250_A3 Framework.sln***. The project includes all required libraries. Compile, link, and execute the project. The generated executable ***will not implement*** culling and clipping. To incorporate your culling and clipping code into the project, complete the following steps.
- Your culling and clipping code will be located in ***.\GfxDriver\src\Clipper.cpp***.
- Each object must, at the very least, have a bounding sphere (BS) defined. The BS is generally computed during the process of exporting object geometry from a modeling package. For this assignment, the BS is computed during load time. This requires that you implement the following function.
  - **ComputeBoundingSphere()** to define an object's BS. Use the technique explained in class to compute the smallest BS possible. Otherwise, your picking assignment may not work correctly.
- During each frame, before transferring geometry and render data of an object to the graphics pipe, the object's BS must be checked for trivial acceptance (TA) and trivial rejection (TR) with respect to the frustum. This requires that you implement the following functions:
  - **GetFrustum():** Extract the plane equations of the six surfaces that define the frustum in view frame.
  - **TransformBoundingSphereToViewSpace():** Transform model frame BS of an object to view frame.
  - **Contains():** Construct an object outcode (OC) and check for TR of BS.
  - **Cull():** Check for TA of BS.
- Once TA and TR of BS fail, the object's triangles must be clipped. This requires that you implement the following functions:
  - **ComputeVertexOutcodes()** to compute the vertex outcode (VC) of all vertices of an object in view frame (if both TA and TR of object fail). Use OC to make the process of computing VC's more efficient.
  - **Clip()** to clip object triangles. Note that

- **Miscellaneous Notes:**
  - Use the following enumerations defined in ***.\GfxLib\gfx\src\Device.h*** to identify frustum plane equations, boundary conditions, and compute outcode flags.
    - **gfxClipCode**
    - **gfxClipPlane**
- Provide a ***manual*** page with your hardcopy submission. In your own words, describe what has been implemented, how it was implemented, any modifications made to the assignment specifications, cool features implemented, etc. In addition, the manual must clearly specify any portion of the assignment that was not completed/attempted and/or incomplete. Don't forget to include derivations with proper pictures for derivations of bounding sphere construction, TA & TR of bounding spheres with respect to the view frustum, and the Sutherland-Hodgeman polygon-clipping algorithm and the

modifications for “thick” planes. Note that if we can’t read what you write, we can’t grade your assignment.

Deliverables:

- Softcopy: Zipped folder containing the entire CS250_A3 Framework with object code and executables deleted. We are going to unzip your folder, double-click on ***CS250_A3 Framework.snl*** to open the project and then compile, link, and execute your assignment.
- Hardcopy: Stapled documentation containing all relevant pages as specified in the assignment submission guidelines.

Objectives: CS250 A3

Name: _____

Login Name: _____

SRS Student ID: _____

___1: Correct information was handed in as hardcopy and a README file exists in student directory. If not, penalty is -20 points.

___2: Student program compiles, links and executes. If not, penalty is -100 points.

___3: **Culling is implemented correctly. Worth 50 points**.

___4: **Clipping using "thick" planes is implemented correctly. Note that the use of "thick" planes is mandatory. Worth 50 points.**

___5: A good manual page was submitted. If not, penalty is **–30** points.

DECLARATION:
I have read the statements regarding cheating in both the CS250 course handout and DigiPen student handbook. I affirm with my signature that this is my own solution to A3 and the submitted source code is of my creation and represents my own work.

Signature: