

CSC 476 – Lab2 – Due Tuesday 2/7/12 - To be demoed in lab

View frustum culling (VFC): Simple game including time based movement, textures and view frustum culling

This lab builds off of your first lab. You will work with your previously created world with moving game objects (moving using time based motion), which includes collision detection. Please increase the size of the world and the number of game objects significantly (try at least 50 game objects). Also include in your world, some simple obstacles scattered throughout (for example some hierarchically modeled trees or just do something simple like boxes at minimum). Implement view frustum culling to remove any game objects not in the current view frustum. Though not required you are encouraged to use any textures to improve the look of your world.

Learning Objectives

- Learn about **view frustum culling**
- **Programming Design and Implementation Requirements**
 1. Generate a simple game with time based motion with many game characters (please feel free to use your copy of lab 1).
 2. Make sure all objects are included in some kind of bounding structure, for example bounding spheres or AABB. If you are ready to build a bounding hierarchy you are encouraged to do so, but that is not required at this time.
 3. Implement view frustum culling. You should cull out any objects that are not in the view volume.
 4. Your display needs to include a standard view into your world, AND a top down view of what is being rendered to demonstrate that the view frustum culling is culling out game objects not in the frustum. The top down view should be a small viewport, rendered from above to show that indeed objects are culled that are out of view of the main “camera” window.

Please note that I will provide an out-dated example base code for this assignment which includes sub-window code, where the sub-window draws a top down view of your scene and a 2D figure of the view volume, which illustrates what objects are being drawn in your frame. This code will not work as is and needs to be rewritten to have the top down view just be an additional viewport into your main window (not a sub-window)– see the Nehe tutorial:

<http://nehe.gamedev.net/data/lessons/lesson.asp?lesson=42>

You need to do the re-write because sub-windows use different OpenGL rendering contexts, which means that the data in your VBOs is not shared across the windows. In addition, I encourage you to re-write this code to not use the OpenGL matrix stack, but to use glm matrix calls instead.