CS 350 | Summer 2019

Assignment 3| Spatial Partitioning

Files (submit folder) due

- Week 12th
- By 11:59 PM PDT

Remember: The major point of this assignment is for you to learn how to submit your homework correctly. You should strive to follow all of the directions exactly as specified in the handouts. The syllabus that was handed out during the first day of class also contains important information on how to submit your homework. If you are still unsure of how to do something, you should ask for help, either from myself or from a tutor in the lab or from another student.

Topics

In this assignment, you will build an Adaptive Octree and BSP-tree representation for the environment in your previous assignments.

The requirements for this assignment are:

- **All requirements for Assignment 1 & 2 are fulfilled**
- Create an Adaptive Octree partition corresponding to Powerplant files
- Create an BSP tree corresponding to the Powerplant files
- Display the tree levels with different colors with their contained objects.
- Provide ability to toggle the display of the tree cells

Scene details

(a) You should continue to use the scene setup from the previous assignment

(b) You should be able to load a file from disk and populate the scene. Do not hardcode the values of the filenames in your code.

Adaptive Octree Creation

(a) Create the octree in a top-down fashion in **world space**.

(b) In the **adaptive octree creation**, you will create a subtree if and only if there are triangles within the parent cell. If there exists no geometry in the subtree, you should treat it as an empty cell and terminate the creation algorithm.

(c) If a triangle straddles the boundary of two (or more) cells, then you should split the triangle and assign the portions to appropriate child nodes that it occupies.

(d) Terminating criteria should be – number of triangles in the current cell. This number should be a user defined input. Set default value to 500 for the powerplant model.

BSP Tree Creation

(a) Use top-down criteria for BSP tree creation. Construct the BSP tree in world space using the following guidelines.

(b) **Leaf storing.** This means that every leaf node contains geometry, but internal nodes only contain information about the split plane. Any polygons straddling the split planes must be split into smaller polygons and sent to the appropriate subtree.

(c) Use a heuristic depending on autopartitioning, standard planes (X, Y, or Z aligned), surface area of child nodes, balanced subtree, or your own heuristic, to choose the split planes.

(d) Terminating criteria should be number of triangles (default = 500 triangles)

Optimizing BSP Tree Creation

(a) You could "optimize" the creation of the BSP tree by serializing a pre-computed BSP tree that is then loaded and displayed when the user chooses to display the tree. This will save the grader's time in evaluating individual assignments.

(b) In your README file, provide information about the powerplant model section and its corresponding BSP tree file for the graders to load.

Displaying the trees (see Page 4)

(a) You should provide options to toggle the display for the trees as follows:

   a. Do not display the trees
   b. Display adaptive octree cells with each "level" in different color
   c. Display the BSP tree nodes with each node in different color

(b) See Page 4 for sample screenshots

Note

- You should have the ability to explain and derive every single line of code used in your project
- Provide a README.txt file that explains your implementation and all the applicable key mappings that you have used

Please refer to your course syllabus for submission guidelines.

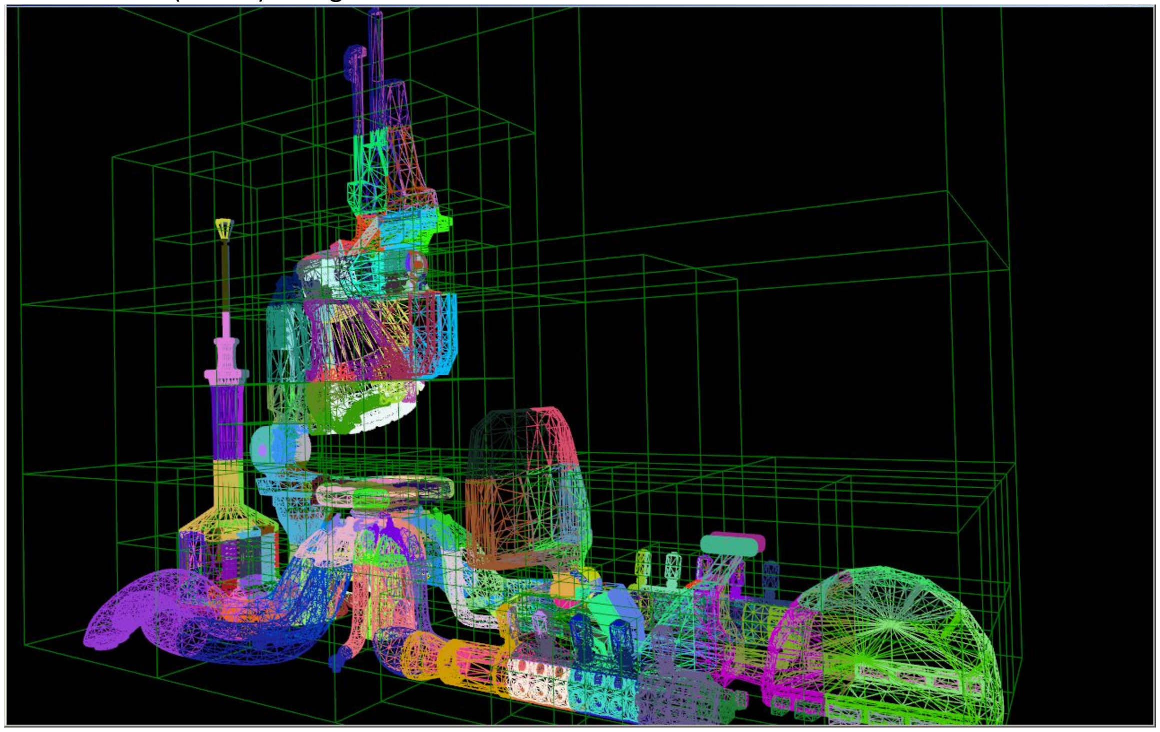
GRADING SHEET (for TAs)

Name of student: ____________________________ Total points obtained : ___________

Implementation Point	Grade	Comments
Assignment 1 & 2 requirements	**10%**	
Assignment 1 & 2 requirements are all completed and working	10	
Scene creation	**10%**	
Scene created using specified objects	10	
Octree	**20%**	
Top down creation of the Octree	10	
Creation of adaptive octree	10	
BSP Tree	**20%**	
Top down creation of the BSP tree	10	
Choosing appropriate split planes	10	
Serialization of tree	**10%**	
BSP Tree serialized and loaded on demand	10	
Interactivity	**20%**	
Toggle Octree display	5	
Show all levels of the octree in different colors	5	
Toggle BSP tree display	5	
Show all nodes of the BSP tree in a different color	5	
Miscellaneous issues	**10%**	
Missing information in README	4	**Automatic zero on rest of the assignment if these criteria are not met.**
Application does not compile	2	
Application cannot be executed	2	
Scene setup incorrect	2	
Total	**100**	

Sample Output (courtesy http://togeskov.net/octbsp.html)

Adaptive Octree – Rendering showing the cells of the Octree. This subdivision terminates when there are 250 (or less) triangles in a cell.



BSP Tree - Each leaf node has its own color. In this example, the creation terminates if the leaf has less than 250 triangles.