

DISTRIBUTED RENDERING OF 3D IMAGES

Team 19 – iWizards

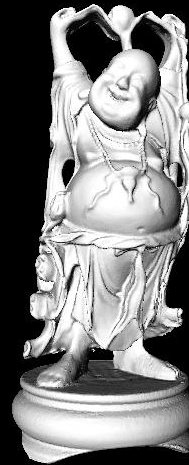
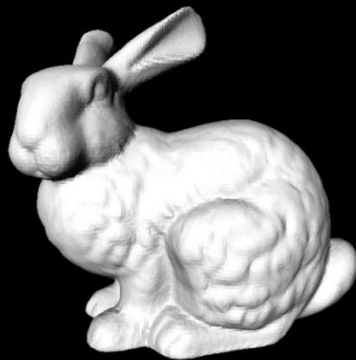
Archit Jain – 200801014

Kaustav Kundu - 200801039

This presentation is in partial fulfillment of the requirement of the Cloud Computing Project

PROBLEM STATEMENT

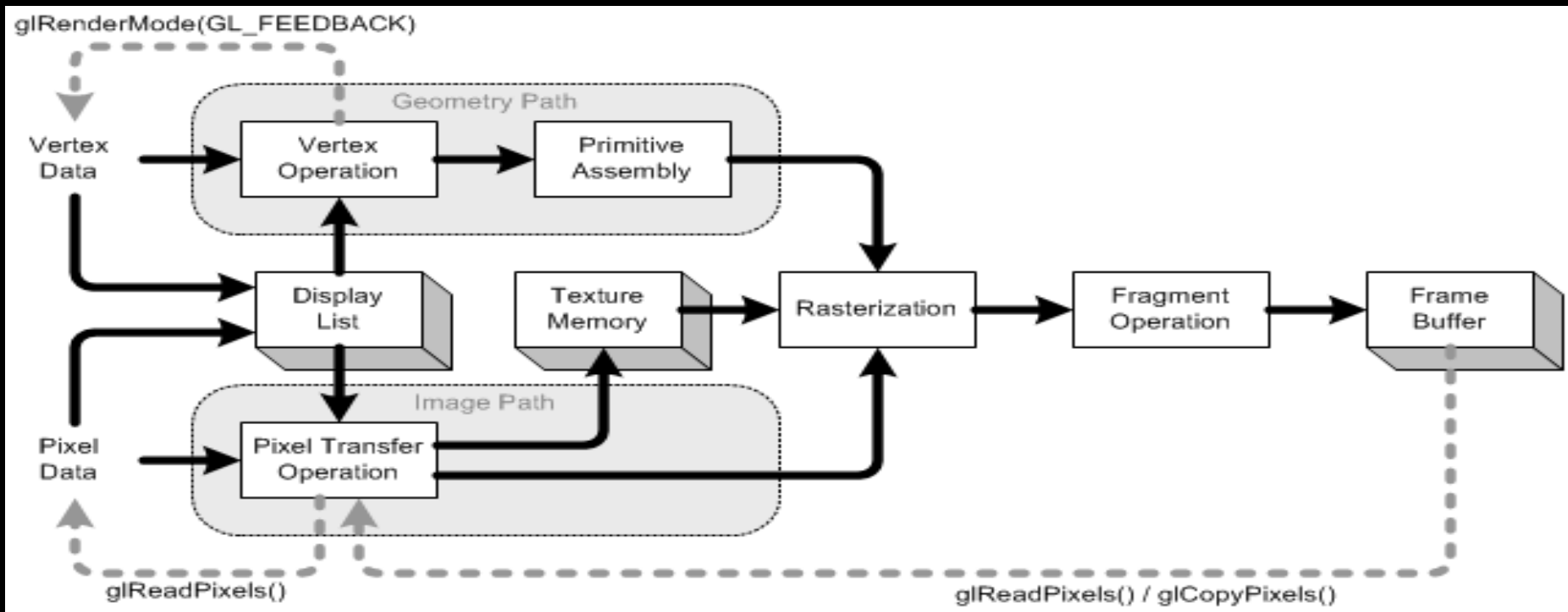
Our project aims at rendering a 3D image, model or any scene onto a 2D screen.



The advantages of this method are :

- Faster rendering process
- Each machine uses lesser amount of resources
- Improving performance of computer graphics creation software

Where is it Distributed?



Up to the **Primitive Assembly Stage**, the same PLY file is read and clipped to different frustums. Once this is done, then in the following stages, distributed rendering helps us.

The computations in these steps get reduced by distributing the rendering of the 3D images

- **Rasterization** – Includes scan conversion and Visibility Determination.
- **Fragment Operation** – Includes fog calculations, scissor test, alpha test, stencil test, depth test and texel generation.
- **Frame Buffer**

ARCHITECTURE

Input file with specific format

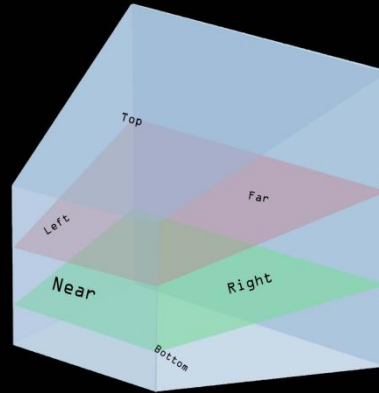
MAP TASK

Output of Map Task:

<Frustum ID, {left, right, bottom, top, near, far}>

Each Line of the input file will have two parts:

1. Name of the PLY file
2. Parameters describing the Frustum, lookAt and window dimensions.



1. Map Task - It divides the entire frustum into different parts.
2. Reduce Task – It renders the image corresponding to each distinct divided frustum.

REDUCE TASK

REDUCE TASK

REDUCE TASK

REDUCE TASK

Files containing information of the rendered images in each of the distributed frustums. (One file per reduce task)

The frustum IDs assigned to each of the distributed frustums is unique.

METHODS IMPLEMENTED

DISPLAY (main function)

MAP CLASS

Functions Implemented :

map : Frustum parameters are divided and sent to different reduce functions

REDUCE CLASS

Functions Implemented :

reduce : Initiates the distributed rendering using JOGL

init : initializes the parameters required for rendering in JOGL

display : renders the scene at each time step

update : updating the values used to represent coordinates of the scene

render : renders the scene when called from display

readPixels : reads the FrameBuffer contents rendered by JOGL and stores the absolute coordinates in HDF5

dispose : releases the resources used by JOGL

reshape : responds to the change when the dimensions of the window display

METHODS IMPLEMENTED (continued)

KEY LISTENER EXAMPLE CLASS

Functions Implemented :

keyPressed : responds to change when keys are pressed

keyReleased : responds to change when keys are released

keyTyped : responds to change when keys are typed

STORAGE CLASS

Functions Implemented (all static functions):

getFrustum and **setFrustum** : returns and sets the frustum parameters respectively

getLookAt and **setLookAt** : returns and sets the lookAt parameters respectively

getWindowDimension and **setWindowDimension** : returns and sets the window parameters respectively

getPrimitiveList and **setPrimitiveList** : returns and sets the list of primitives respectively

getPlyFileName and **setPlyFileName** : returns and sets the name of the input ply file respectively

getPlyOutFileName and **setPlyOutFileName** : returns and sets the name of the output ply file respectively

getIdentifier and **setIdentifier**: returns and sets the identifier to identify the frustum respectively

METHODS IMPLEMENTED (continued)

PLY FILE READER CLASS

Functions Implemented :

plyFileReader : constructor

initPly : initializes the parameters

plyOpen : opens the ply file and returns false if file not present.

plyReadHeader: reads the header of the ply file and stores the number of vertices and primitives (faces) present.

plyRead: reads the information contained by each of the vertices and faces.

getPrimitiveLength : returns the length of the list of primitives

getPrimitives : returns the list of primitives

plyClose : closes the file and associated resources used

FACE CLASS

Functions Implemented :

Face : constructor of the class

getVertices: returns the list of vertices

VERTEX CLASS

Functions Implemented :

Vertex : constructor of the class

CLASS DIAGRAM

DISPLAY

MAP

REDUCE (implements
GLEventListener)

KEY LISTENER EXAMPLE
(implements
KeyListener)

PLY FILE READER

plyFile – name of the ply File
noOfvertices – number of vertices
noOfFaces – no of Faces
vertexList[] – list of vertices for one face
faceList[] – list of faces for the entire image

STORAGE

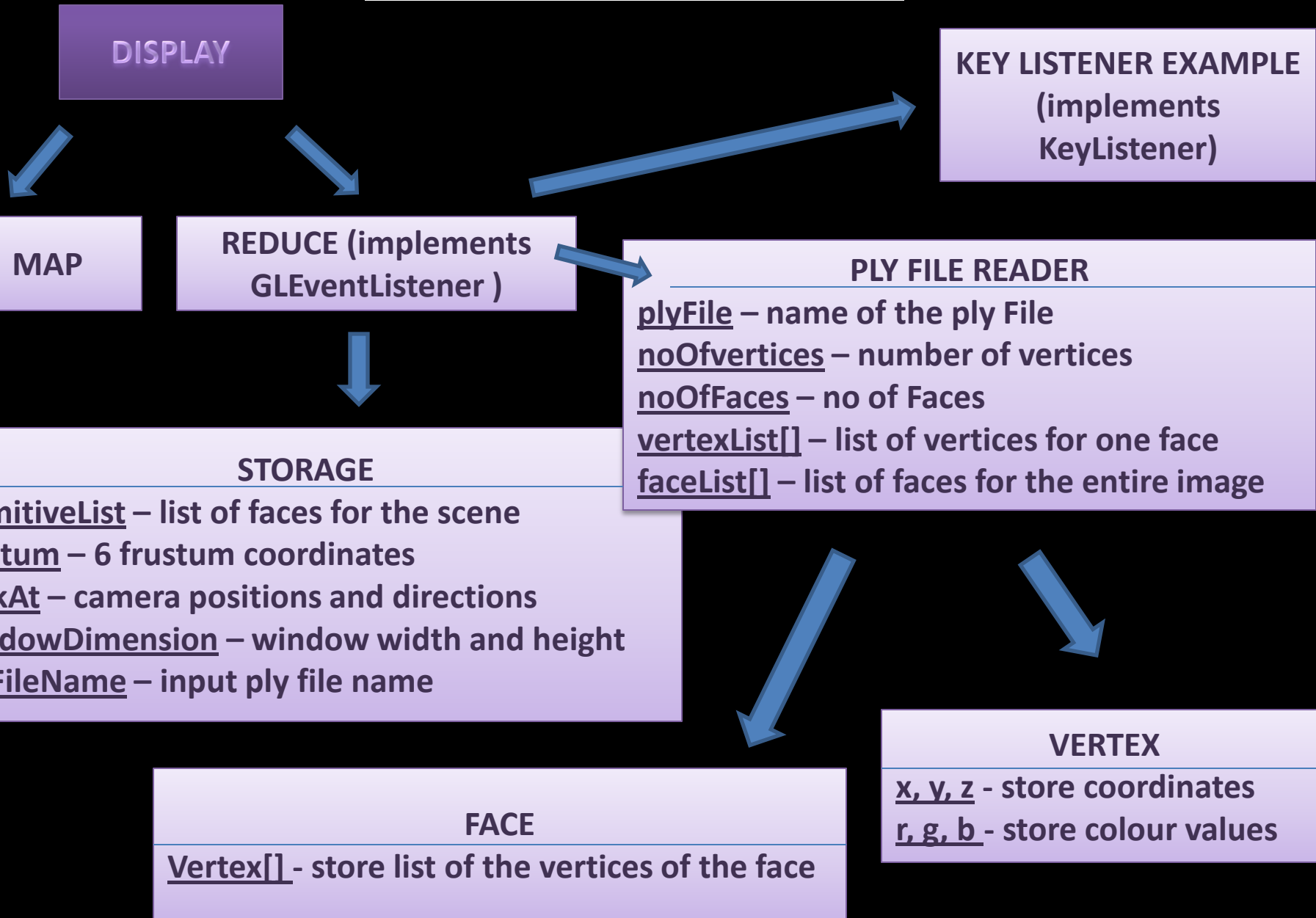
primitiveList – list of faces for the scene
frustum – 6 frustum coordinates
lookAt – camera positions and directions
windowDimension – window width and height
plyFileName – input ply file name

VERTEX

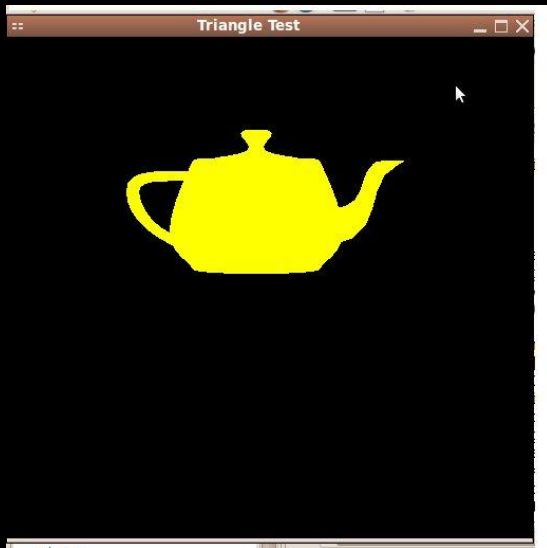
x, y, z - store coordinates
r, g, b - store colour values

FACE

Vertex[] - store list of the vertices of the face



PRELIMINARY RESULTS



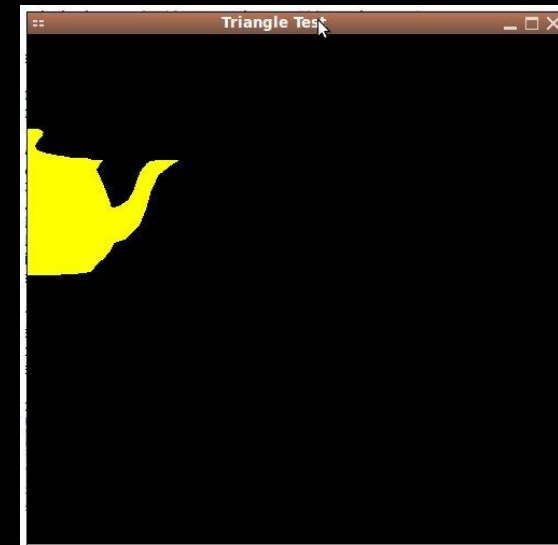
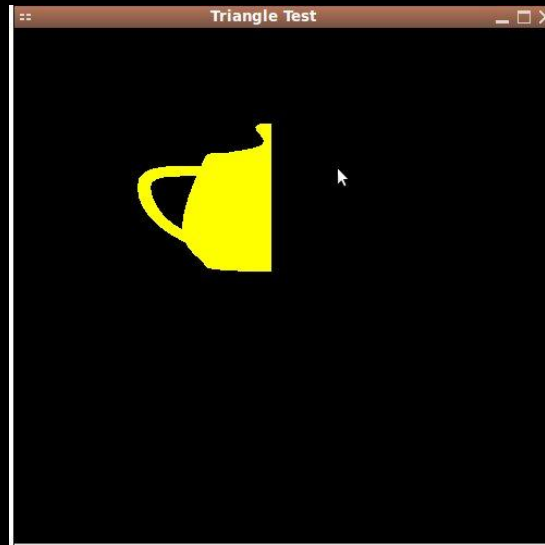
ORIGINAL
IMAGE

The Input File contains :

- a) 1177 vertices
- b) 2256 faces

This was divided into frustums.

2 output images from
different reduce tasks



Thank You !!!