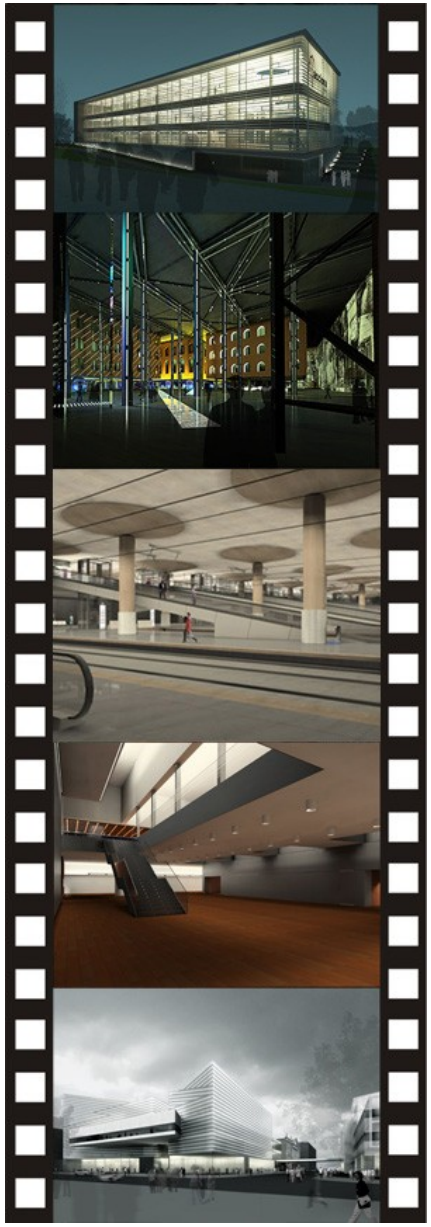
WGLRAD - A windows opengl - raytrace viewer for Radiance

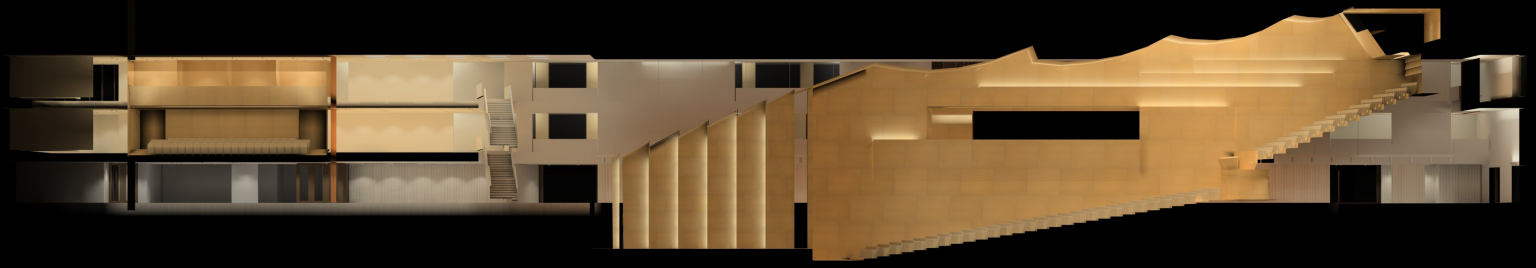
8th International RADIANCE Workshop

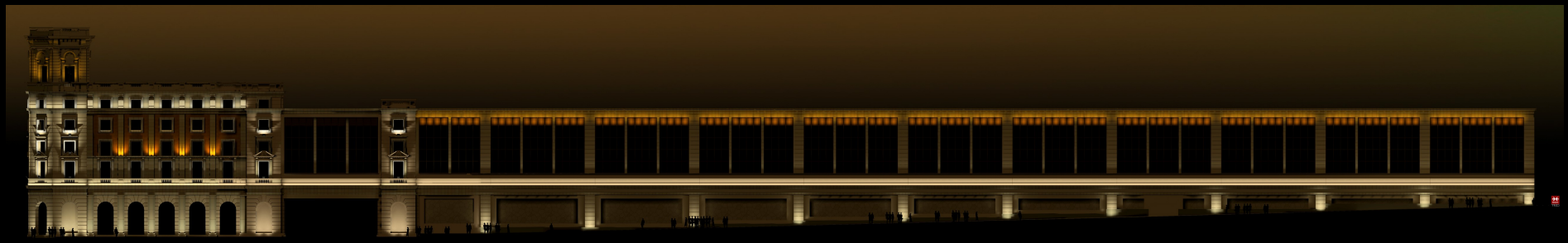
Ignacio Munárriz
Architectural Innovation Studio
www.aisarquitectura.com
info@aisarquitectura.com
Pamplona, Spain
04.07.2009



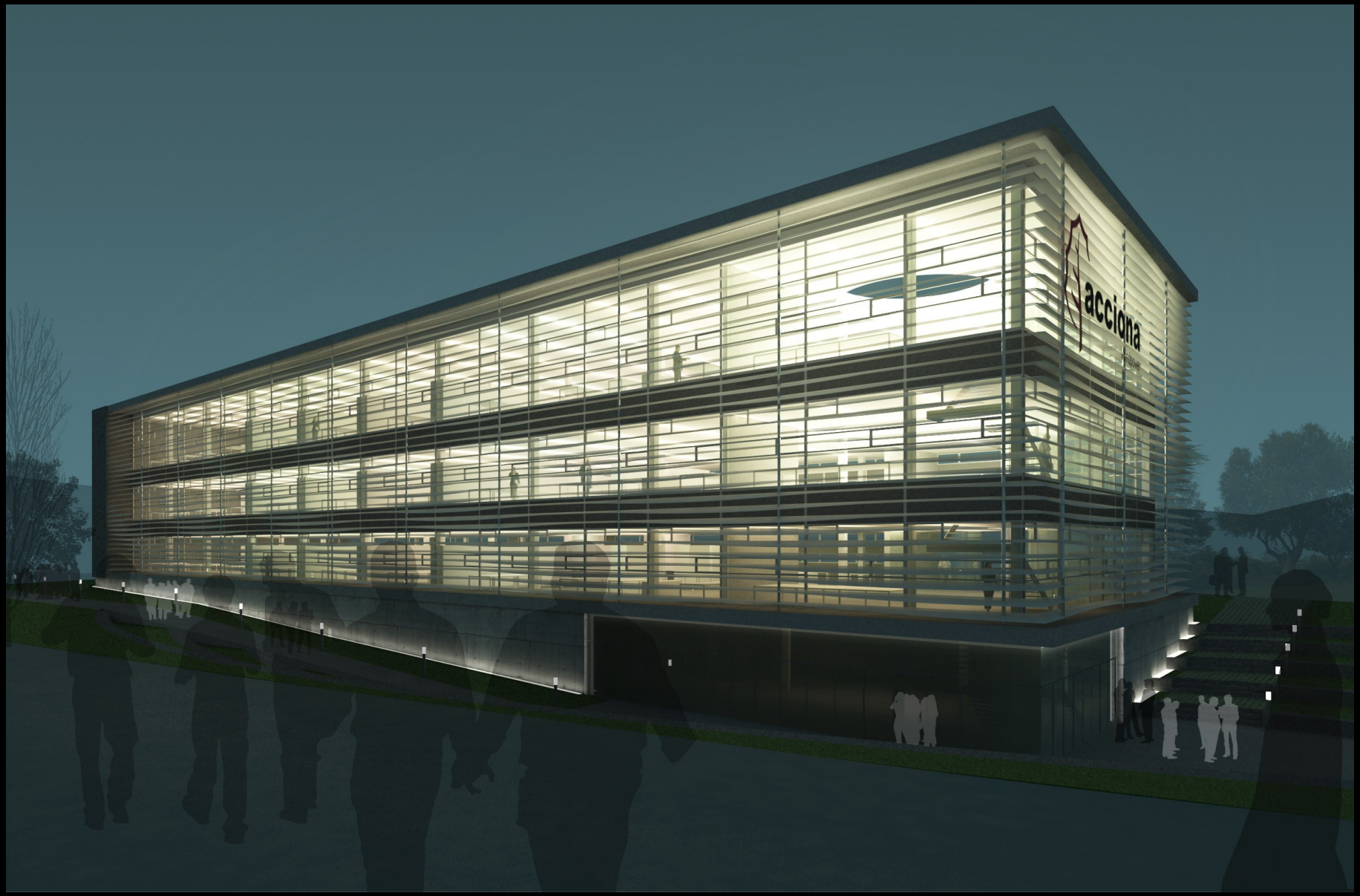
Former work

- Simulation commercial packages
- Architectural images
- Using photometric data
- Not special equipment to measure materials
- Recent work at university and later



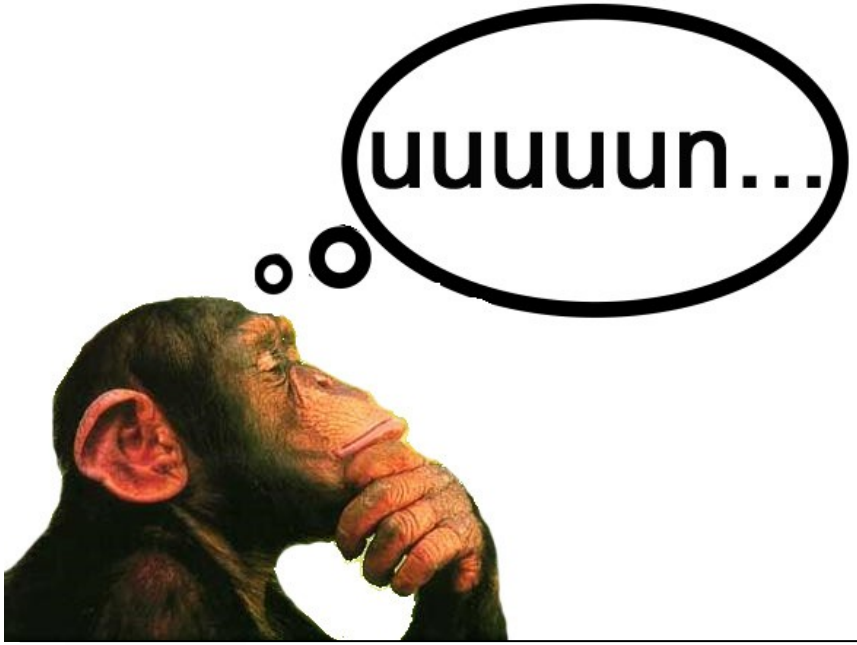



modelo
Terminales fibra óptica VFG 1605-1615
Bañadores de pared QBR600 20x20
techo palcos
sofitos palcos
antepechos
Barra de cristal-luz RSL130 techo
Proyectores PAR56 techo
Fluorescencia lineal de relleno conchas RENNES 996
Downlights MBN 210 75/150W


acciona

Goals

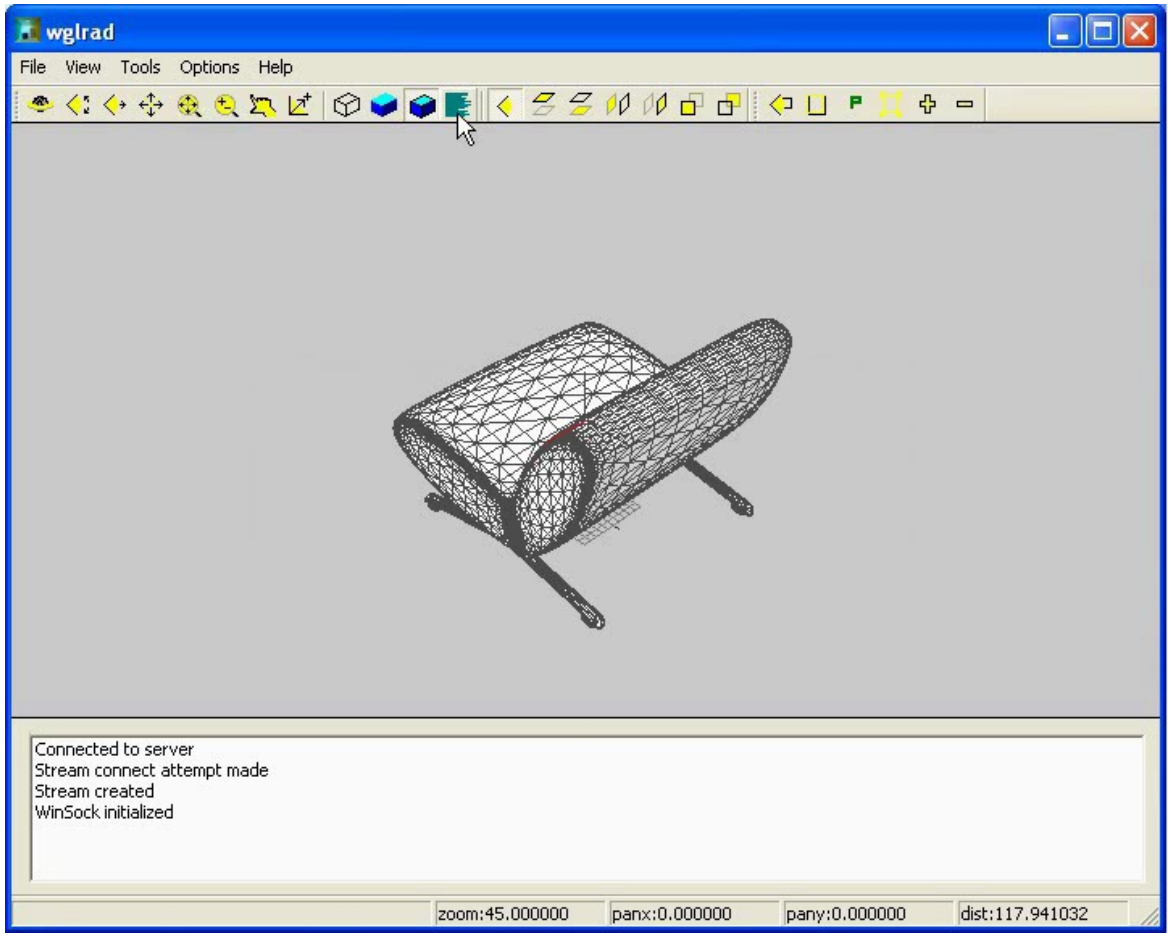
More control & flexibility in software
More accurate software
Light & Material simulation
Similar performance



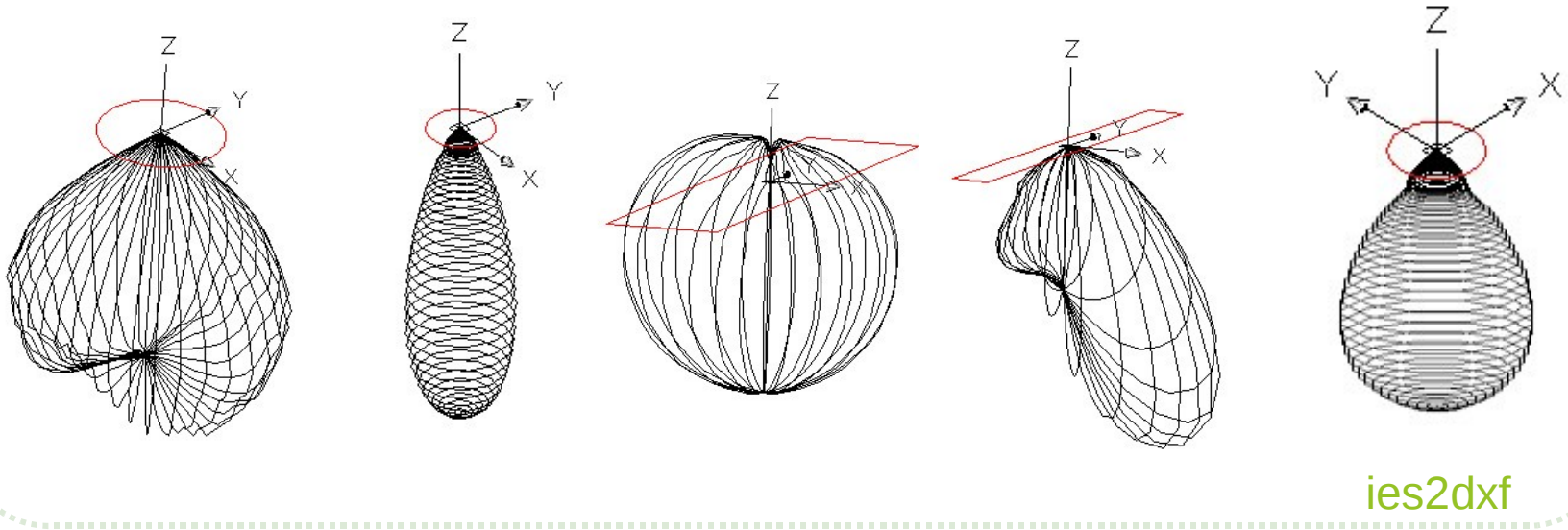
Inconvenients

Amateur programming
Modifying code
Very complete packages in market
Radiance command line oriented

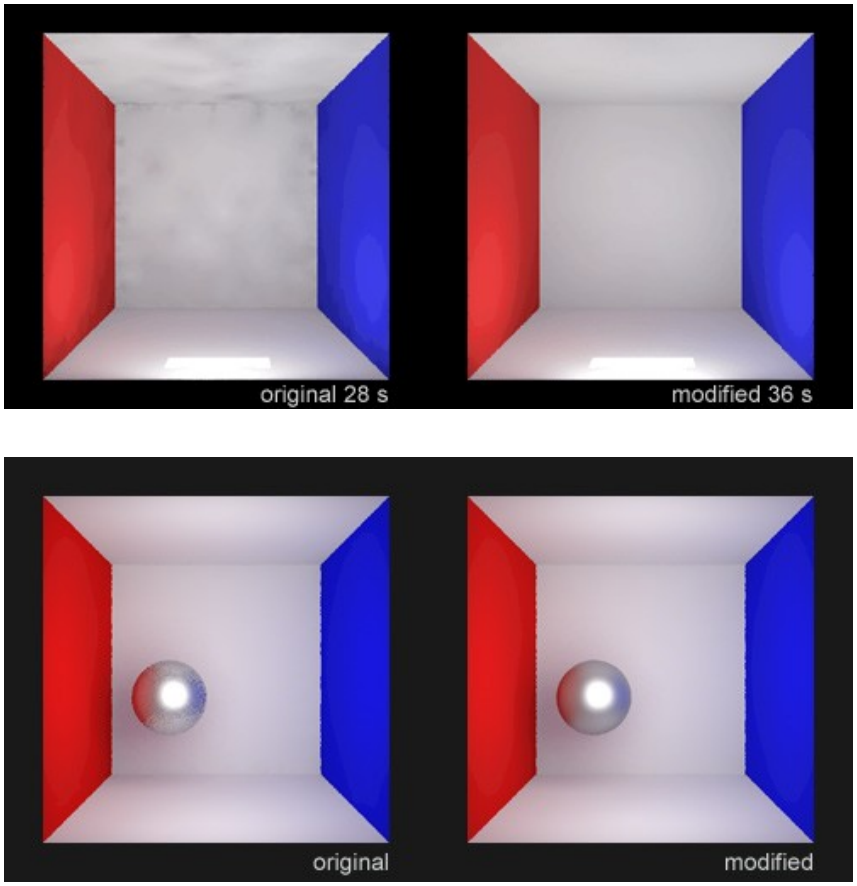
Several utilities



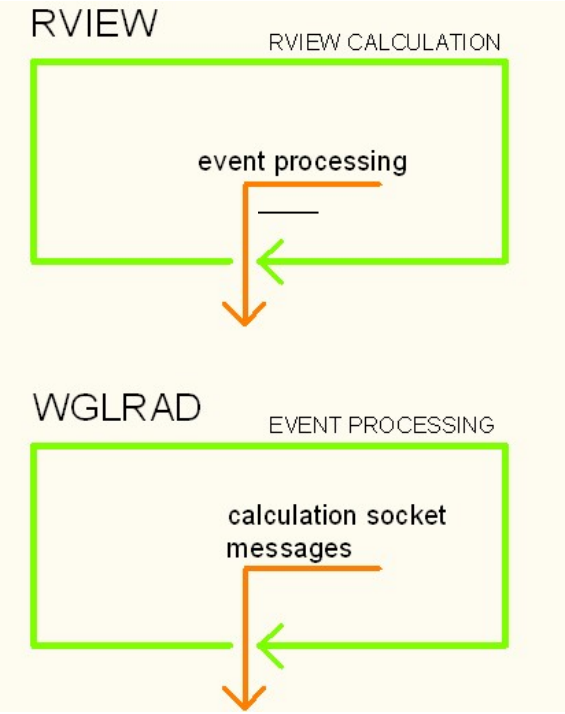
Several utilities



Several utilities



event-driven



Rview recursive function that check events

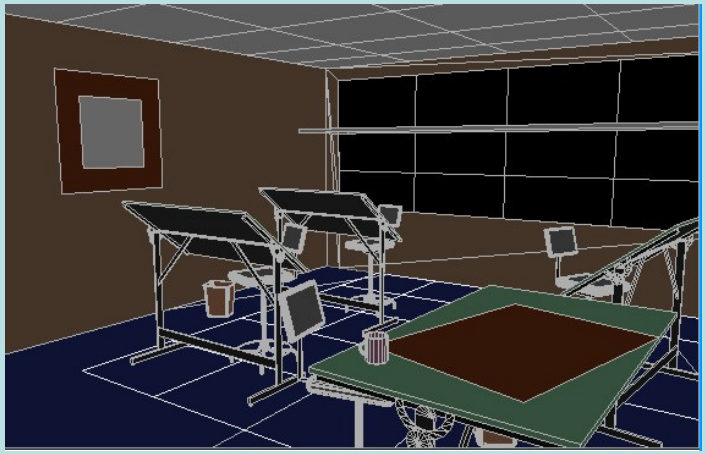
Wglrad event-driven message based

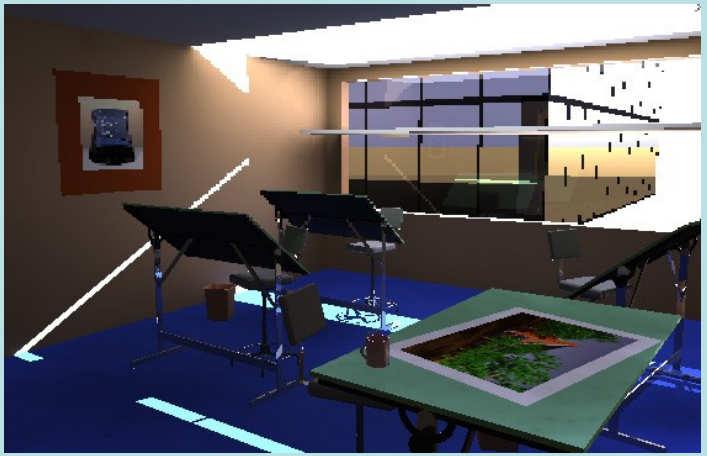
External process calculation to avoid gui collapse

Messages → ray packets through sockets

Allow remote and cluster calculation

Rview & glrad mix





1 Opengl converted by glrad routines

or

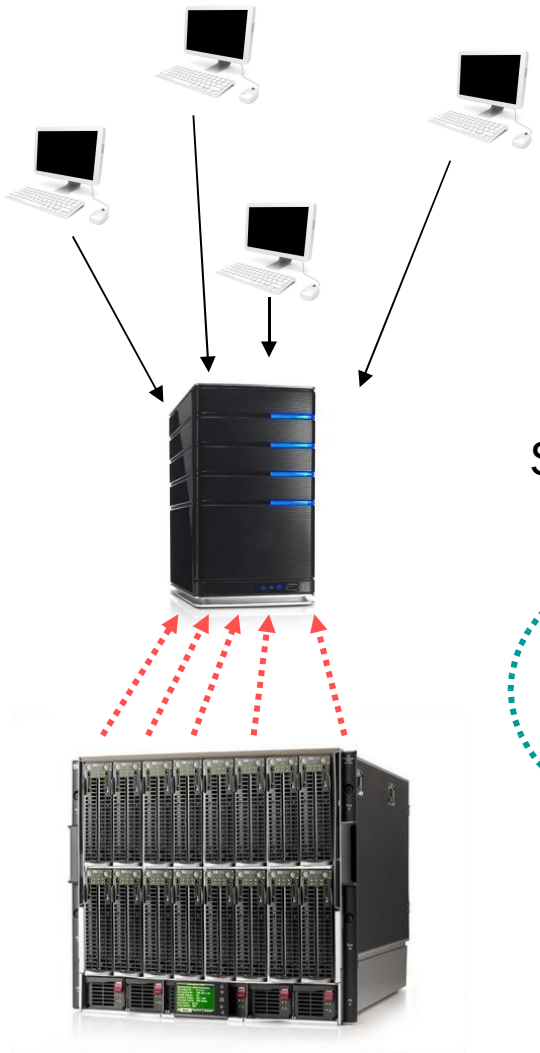
2 Opengl Sended progressively as mesh

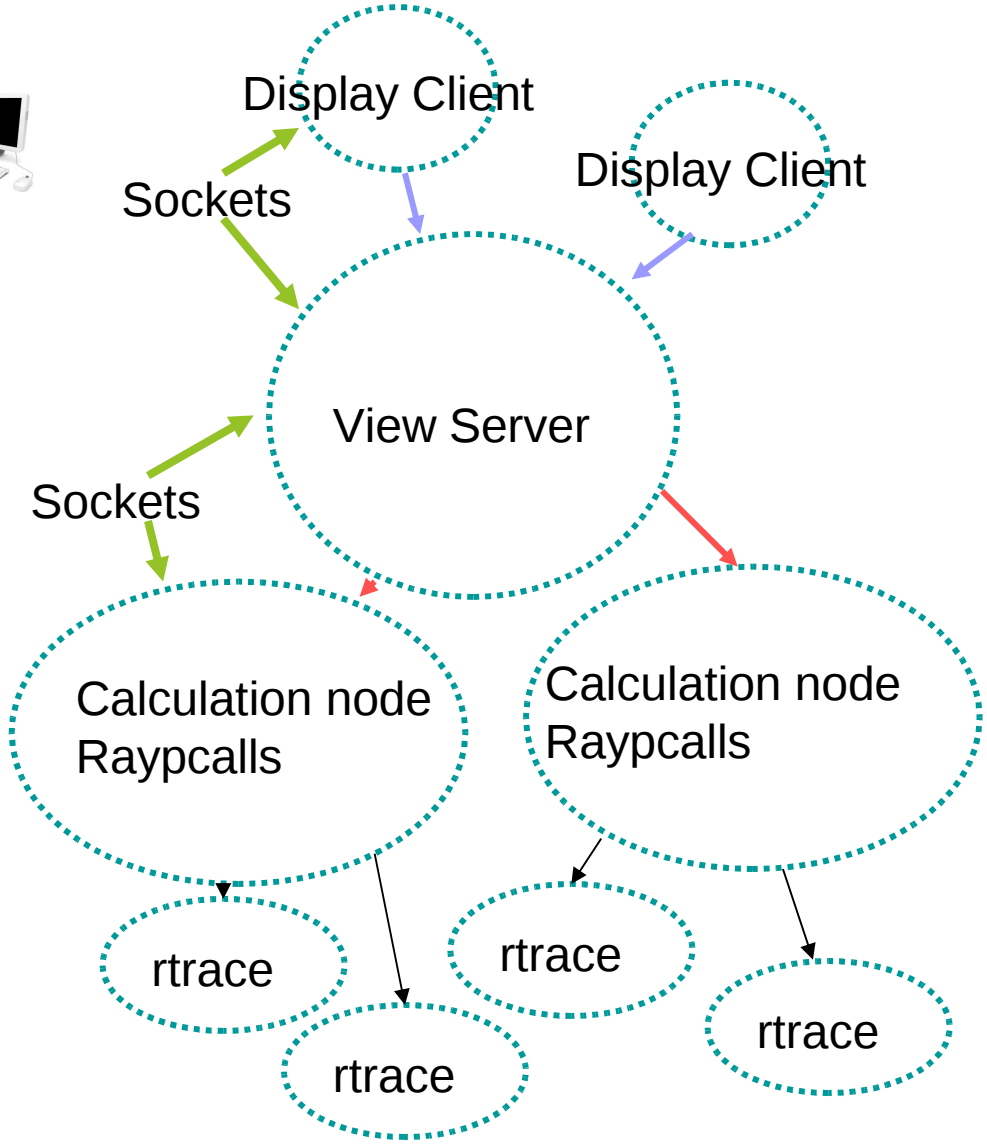
Ordered by area

Limit geometry by graphics hardware

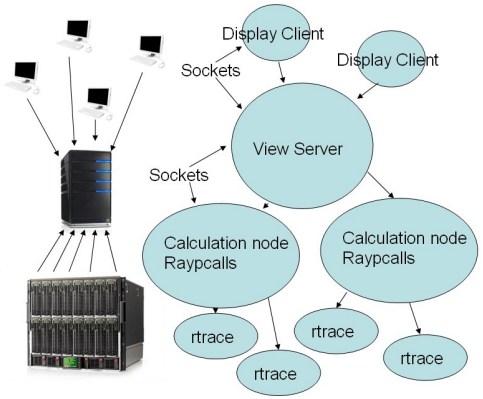
Light mapped

COMPONENTS




Display Client
Display Client
Sockets
View Server
Sockets
Calculation node
Raypcalls
Calculation node
Raypcalls
rtrace
rtrace
rtrace
rtrace

Process

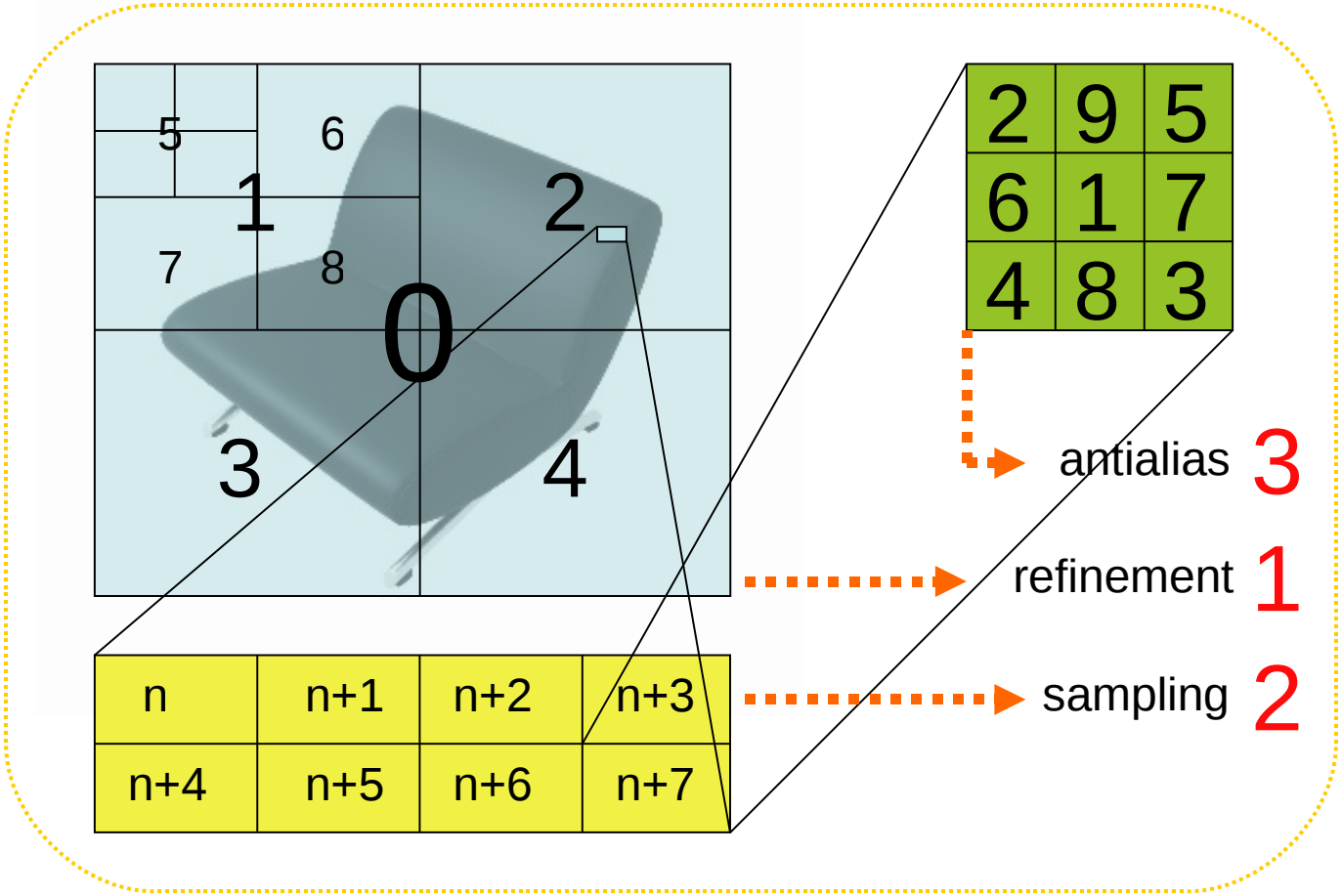


1.Client connects and ask for a project
2. Server returns opengl model
3. Server divides total calculation projects between display clients
4. Clients sends a view
5. Server sends packets to calculation nodes to raytrace
6. Server sends back the calculated rays to display clients compressing information

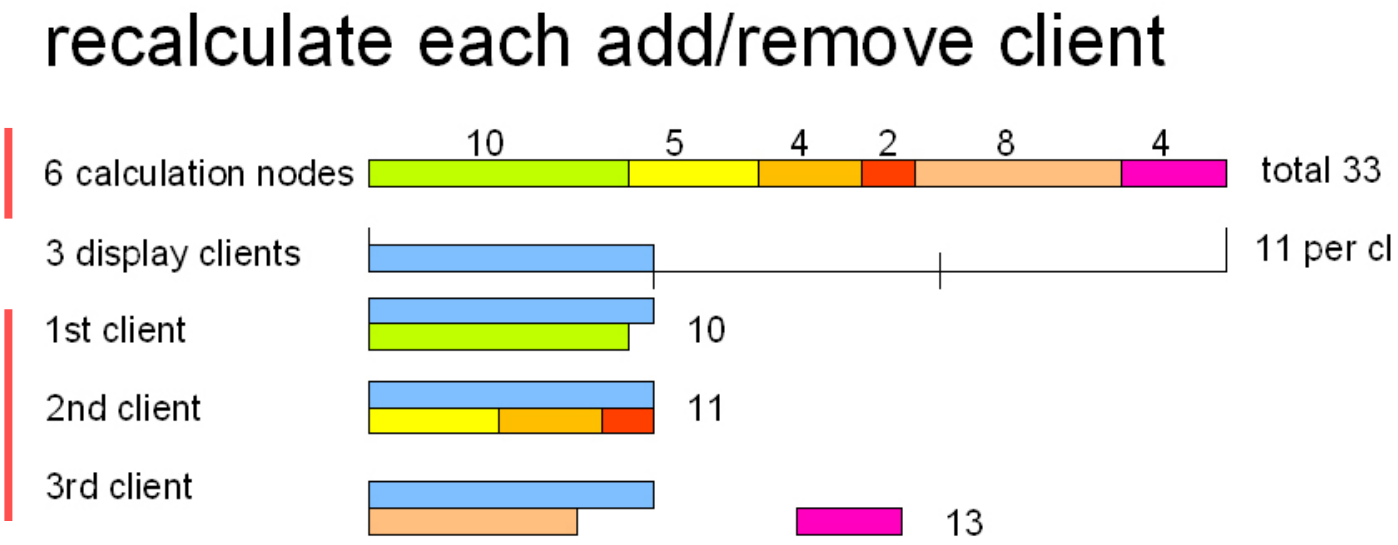
Opengl & raytrace

Integrate glrad
Achieve interaction
Opengl transmitted at the beginning | through streaming

Recursive vs Ordered

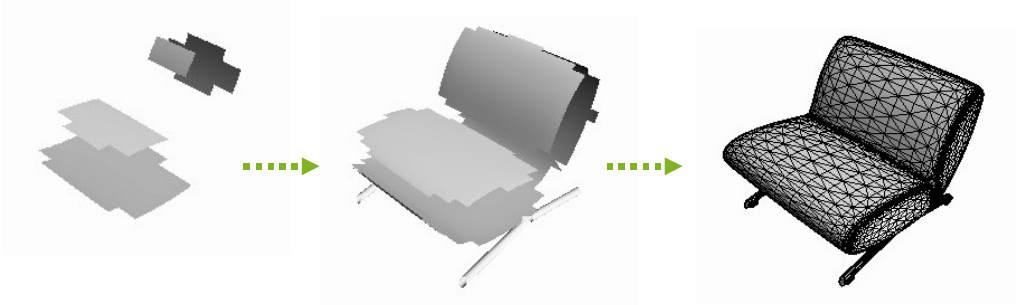


Redistrubute calculation



Divides available calculation power between current display clients

Data format



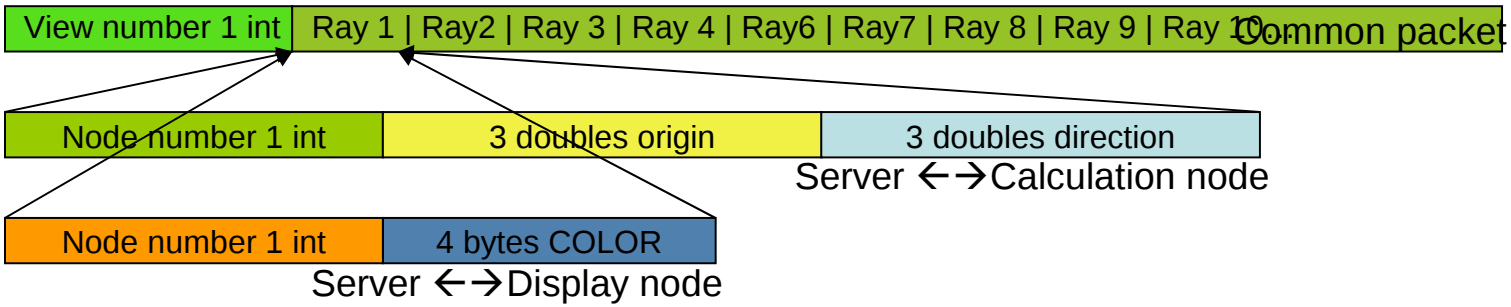
1 **Opengl** information

- mesh as **file**
- mesh **streaming**

> wglrad send the three vertex for each triangle doesnt use vertex indexes
> triangles(uv coordinates, vertex, normal) ordered by area

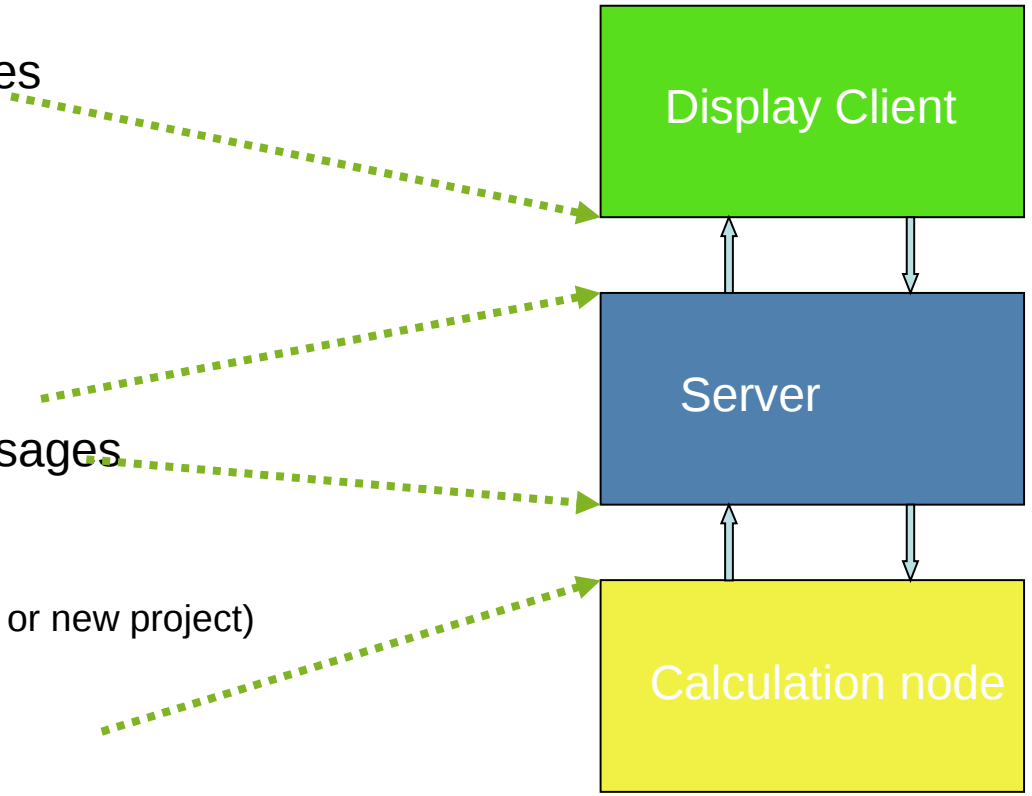
2 **Raytraced** information

- packets of rays containing each packet the view number and the rays
- rays **not calculated** → node an integer to define the node and the origin and direction as 6 doubles
- rays **calculated** → radiance color(4 bytes) and an integer for the node

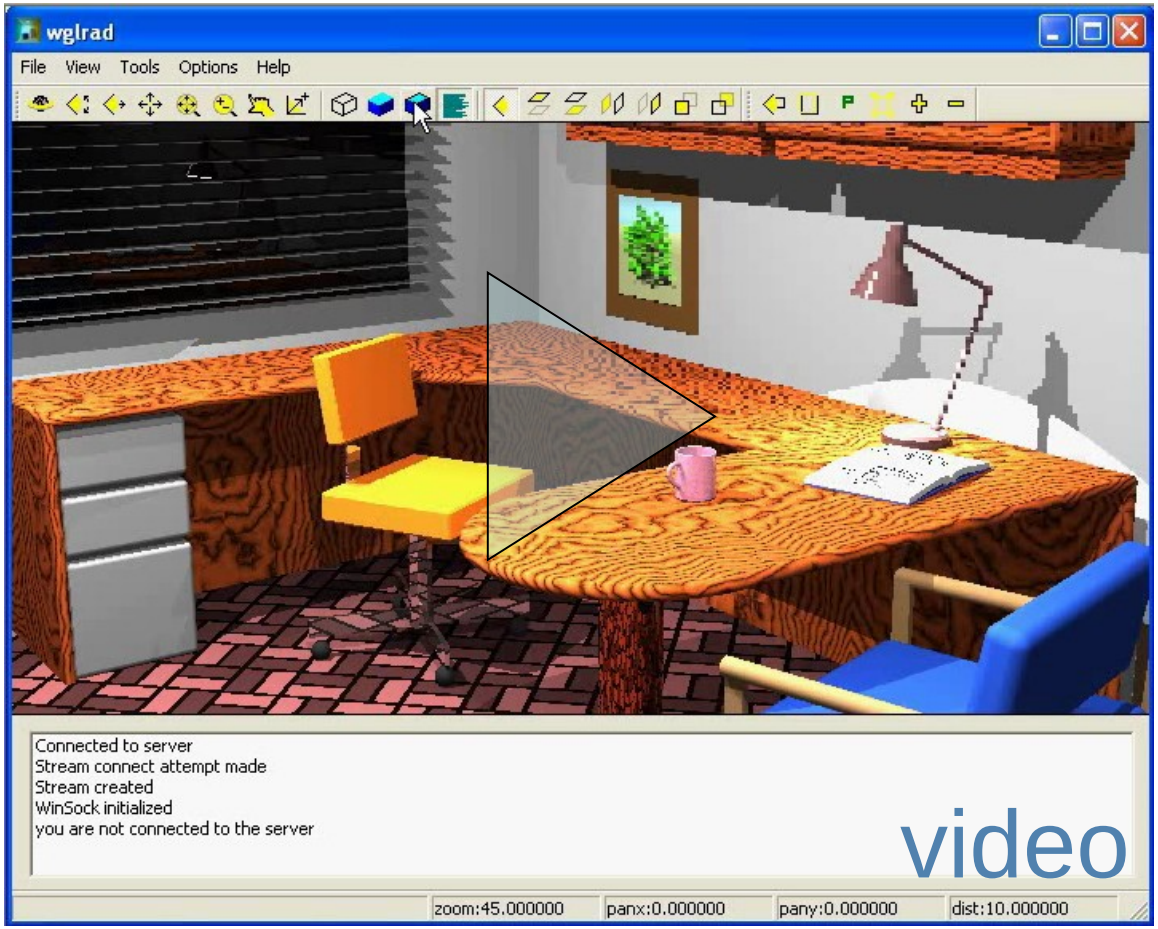


Message types

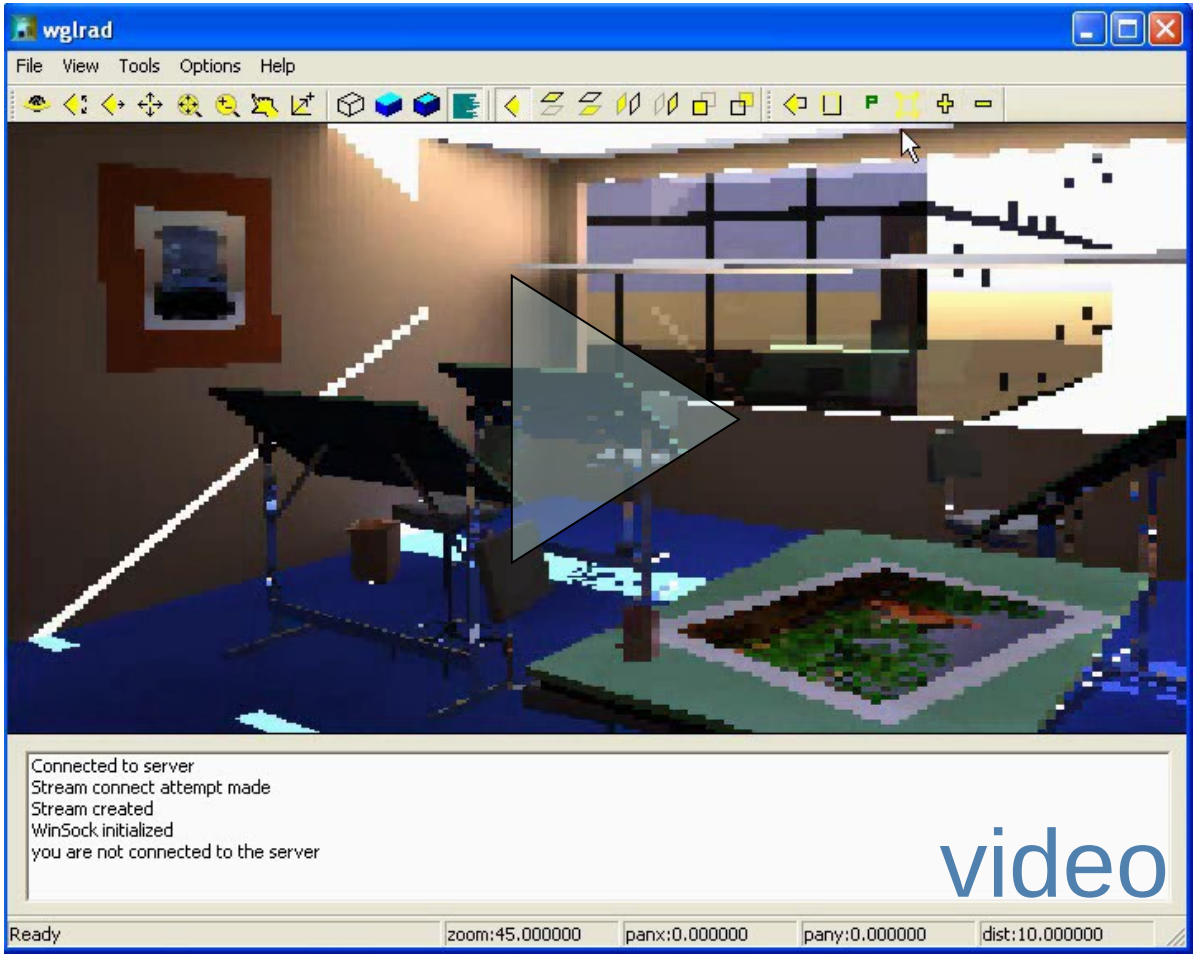
- Display nodes messages
  - Project
  - View
  - Parameters change
  - Opengl messages
  - Ray packets
- Calculation nodes messages
  - Project identification
  - Ray packets
  - Kill calculation(new view or new project)
  - Flushing
  - Parameters



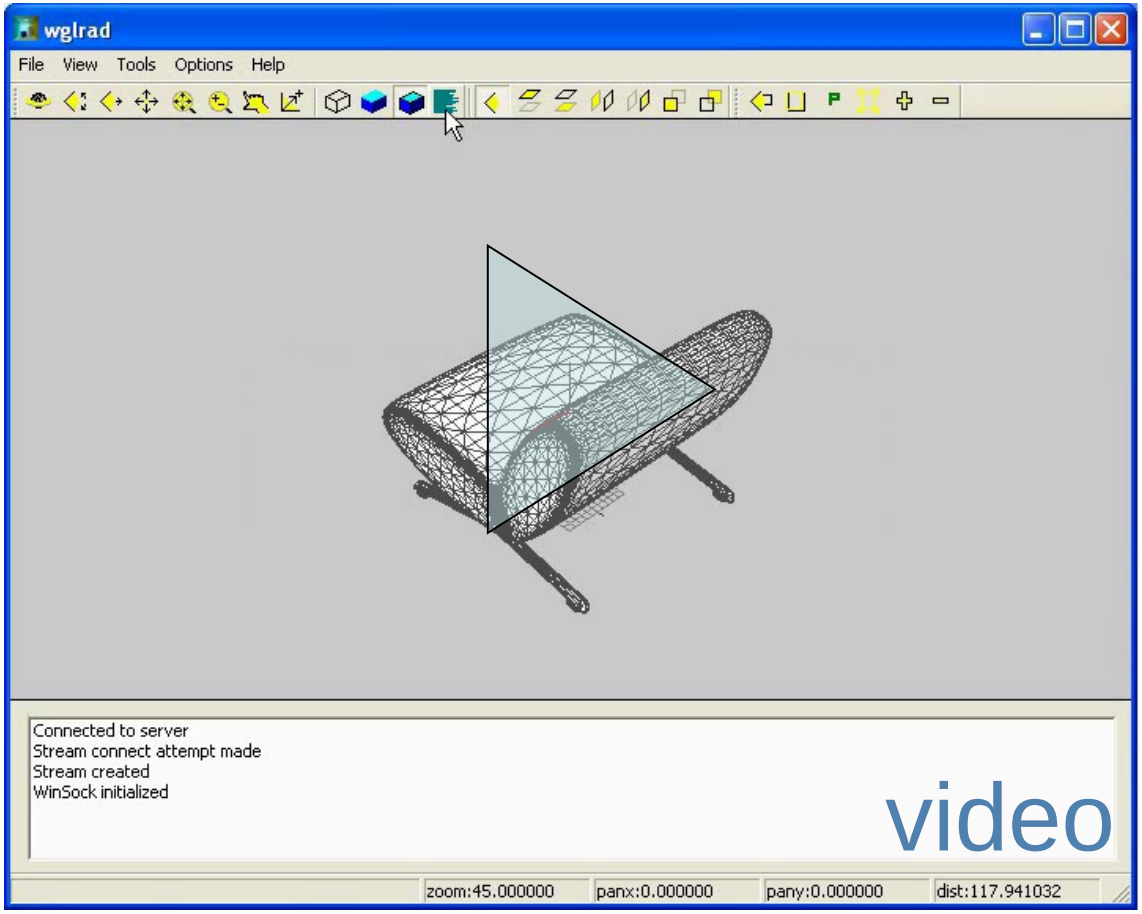
Wglrad in action 1



Wglrad in action 2



Load model with name



http embedded

Control Files

⟹ 1 Server

- one line per project
- name of the project and route to the geometry mesh file

⟹ 2 Calculation Nodes

- one line per project
- name of the project, number of processors, radiance rtrace parameters, route to the oct file

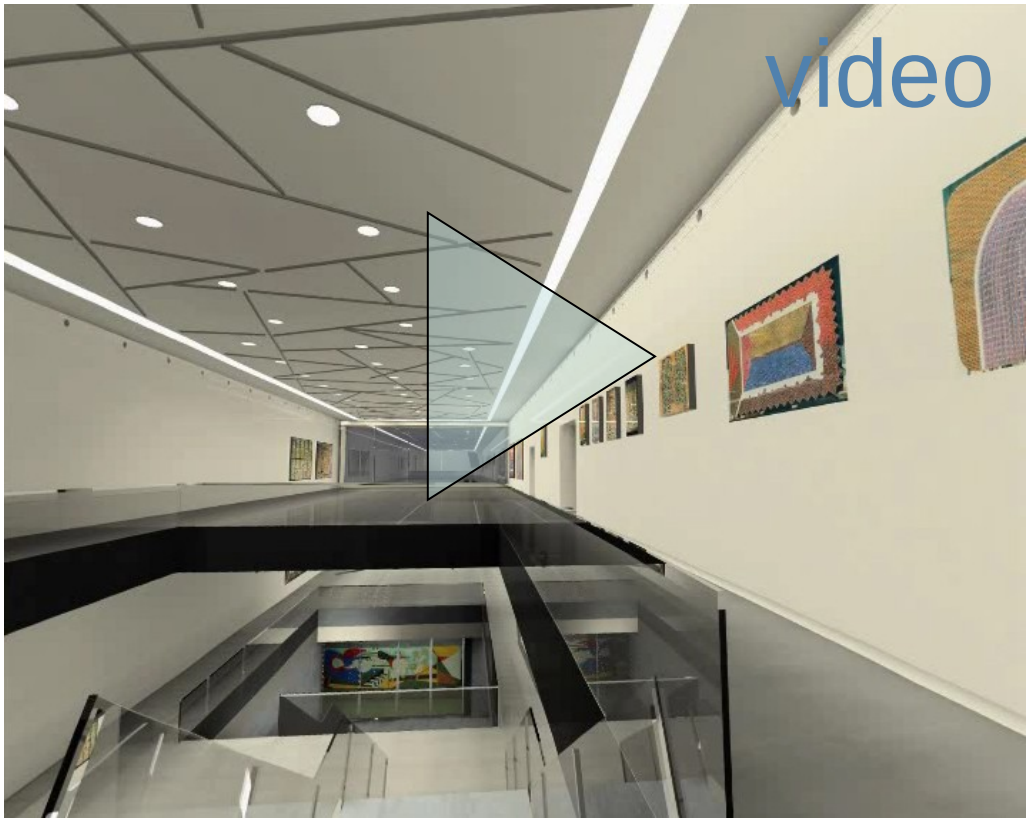
⟹ 3 Applet

- parameters: project name, ip server, gamma, exposition, interior | exterior, english | spanish

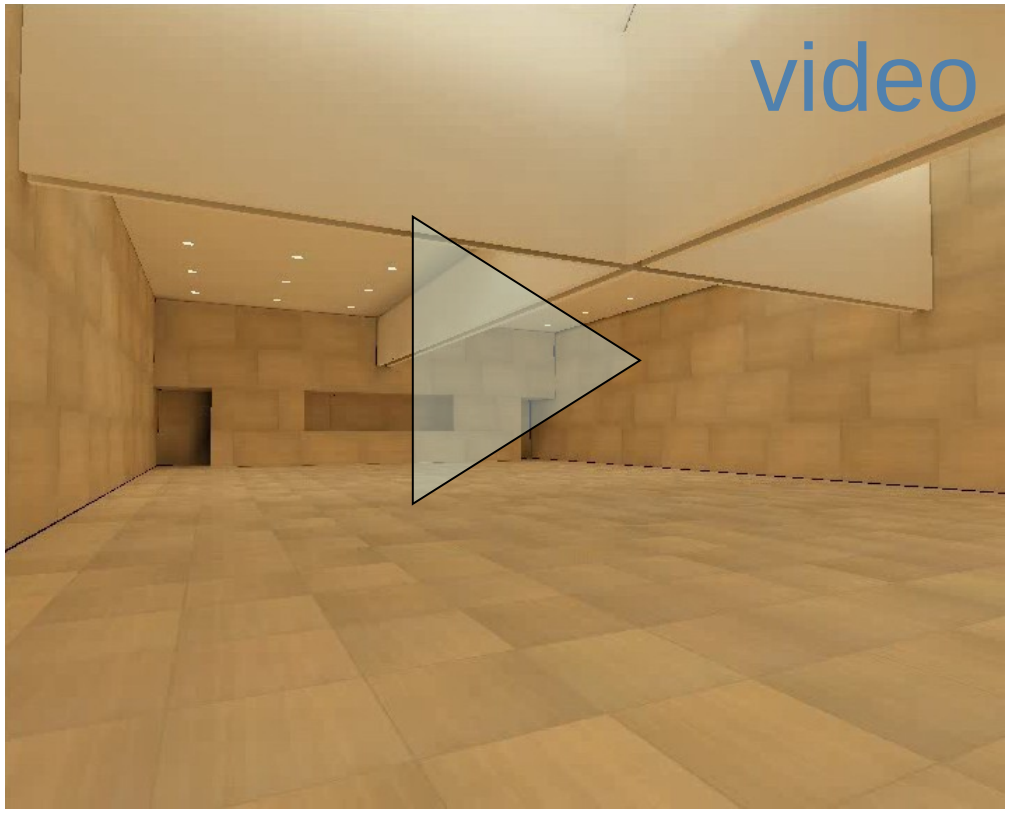
Available Interactive tools

Cylindrical – Spherical panoramas
Opengl mapping (Lightmapping with Radiance)
Rholo – rview
wglrad

Panoramic



Opengl mapped



Interactive viewer

Thank you

Any Questions?