Ray Casting

Hao Li

Applied Geometry Group
ETH Zurich
Ray casting is not always ray tracing

*Image courtesy of Tron (1982)*
Ray casting is not always ray tracing

Image courtesy of Wolfenstein3D (1992)
Ray casting is not rasterization
Exercise 1
Tasks

• Understanding the framework
Tasks

• Understanding the framework

• Ray generation
Tasks

• Understanding the framework
• Ray generation
• Ray-surface intersection
Tasks

- Understanding the framework
- Ray generation
- Ray-surface intersection
  - plane, sphere, triangles
Tasks

- Understanding the framework
- Ray generation
- Ray-surface intersection
  - plane, sphere, triangles
- Constant shading
Tasks

• Understanding the framework
• Ray generation
• Ray-surface intersection
  • plane, sphere, triangles
• Constant shading
Constant shading can be cool

Image courtesy of Apple Inc.
Optional Tasks

Progressive sampling
Optional Tasks

Quadrics (quadratic surface)

Image courtesy of DGP Toronto
Optional Tasks

\[ ax^2 + by^2 + cz^2 + 2fyz + 2gzx + 2hxy + 2px + 2qy + 2rz + d = 0 \]

Quadrics
The Framework
Ray Tracing Framework
Ray Tracing Framework

Data flow:

Load Scene → Load Renderer → Render Scene in GLUT viewer

SceneDescription.xml contains:
- 1 camera
- n elements
- m light sources

via SceneParser.cpp
Ray Tracing Framework

Data flow:

1. Load Scene
2. Load Renderer
3. Render Scene in GLUT viewer

via ConfigParser.cpp

Config.xml contains:
- Sampler
Ray Tracing Framework

Data flow

Load Scene

Load Renderer

Render Scene in GLUT viewer

via Renderer::render()

Image
Rendering Pipeline

Renderer
- Sampler
- Shader

Scene
- Camera
- List of Elements
- List of Light Sources
Rendering Pipeline

Renderer
- Sampler
- Shader

Scene
- Camera
- List of Elements
- List of Light Sources

Sample
Rendering Pipeline

Renderer
- Sampler
- Shader

Scene
- Camera
- List of Elements
- List of Light Sources

Sample
Primary Ray
Rendering Pipeline

Renderer

- Sampler
- Shader

Scene

- Camera
- List of Elements
- List of Light Sources

Sample

Primary Ray

Ray
Rendering Pipeline

Renderer

Sampler

Shader

Scene

Camera

List of Elements

List of Light Sources

Sample

Primary Ray

Ray

Intersection
Rendered Pipeline

**Renderer**
- Sampler
- Shader

**Scene**
- Camera
- List of Elements
- List of Light Sources

Flow of Data:
- Sample
- Primary Ray
- Intersection
- Light

Relationships:
- Primary Ray from Shader to Scene
- Intersection from Scene to Shader
- Light from Scene to Shader
- Sample from Shader to Scene
Rendering Pipeline

Renderer
- Sampler
- Shader

Scene
- Camera
- List of Elements
- List of Light Sources

Sample Color
- Sample
- Primary Ray
- Ray
- Intersection
- Light
First Build and Run
Orientation
Orientation

- `main()` is in RayTracer.cpp
• **main()** is in RayTracer.cpp
• Rendering function is **Renderer::render()**
• main() is in RayTracer.cpp
• Rendering function is Renderer::render()
• Utility classes (vector, ray, etc...) in utils/
Orientation

- `main()` is in `RayTracer.cpp`
- Rendering function is `Renderer::render()`
- Utility classes (vector, ray, etc...) in `utils/
- Geometric elements in `sceneelements/geometry/`
Orientation

- main() is in RayTracer.cpp
- Rendering function is Renderer::render()
- Utility classes (vector, ray, etc...) in utils/
- Geometric elements in sceneelements/geometry/
- Sampler, shader and other classes in rendererelements/
What to hand in?
What to hand in?

• Visual Studio 2005 project with source code (www.ides.ethz.ch for MSDNAA license)
What to hand in?

- Visual Studio 2005 project with source code (www.ides.ethz.ch for MSDNAA license)
- Custom scene description files
What to hand in?

- Visual Studio 2005 project with source code (www.ides.ethz.ch for MSDNAA license)
- Custom scene description files
- Html readme file (follow template)
What to hand in?

• Visual Studio 2005 project with source code (www.ides.ethz.ch for MSDNAA license)

• Custom scene description files

• Html readme file (follow template)

introcg@inf.ethz.ch
Please, only submit VS 2005 projects
hao@inf.ethz.ch