## CS-I 84: Computer Graphics

Lecture \#7: BSP and AABBTrees

$$
\begin{aligned}
& \text { Prof. James O'Brien } \\
& \text { University of Californina, Berkeley }
\end{aligned}
$$

```
BSP-Trees
    Binary Space Partition Trees
    - Split space along planes
    - Allows fast queries of some spatial relations
    - Simple construction algorithm
    - Select a plane as sub-tree root
    - Everything on one side to one child
    - Everything on the other side to other child
    - Use random polygon for splitting plane
```


ames




Visibility Traversal

- Variation of in-order-traversal
- Child one
- Sub-tree root
- Child two

Select "child one" based on location of viewpoint

- Child one on same side of sub-tree root as viewpoint


$g: e_{2}: c_{2}: f: e_{1}: a: c_{1}: b: d$


## Your Ray Tracer

RayTrace (image)
For ray in camera
image [pixel] $=$ Trace (ray)
Trace (ray)
t_hit = infinity
For object in scene
t_hit $=$ min(object.intersect(ray), t_hit)
shade at t_hit
possible calls to Trace(new_ray)

## Your Ray Tracer

RayTrace (image)
For ray in camera
image[pixel] = Trace(ray)
Trace (ray)
t_hit = infinity

## For object in scene

t_hit $=\min (o b j e c t$. intersect (ray), t_hit) shade at t_hit
possible calls to Trace(new_ray)

## Your Ray Tracer

RayTrace (image)
For ray in camera
image [pixel] $=$ Trace (ray)

## Trace (ray)

t_hit = infinity
For object in scene
t_hit $=\min \left(o b j e c t . i n t e r s e c t(r a y), ~ t \_h i t\right) ~$
shade at t_hit
possible calls to Trace (new_ray)

## Bounding Shapes

为

Bounding shape completely encloses associated object

- Rays cannot hit object w/o intersecting bounding shape
- Two objects cannot collide if shapes don't overlap

Simplicity -vs- tightness

## Axis-Aligned Bounding Boxes <br>  <br> Axis-aligned bounding box defined by min and $\max x, y, z$

Axis-Aligned Bounding Boxes




Bounding Box of Bounding Boxes



AABB Trees

|  | $A \mathrm{ABB}$ Trees | One of many variations |
| :--- | :--- | :--- |
|  |  |  |
| Transformed Bounding Boxes |  |  |





## Ray Test Against Bound Tree

- RayHitSubTree (\&ray, node)
- If RayHitsBB(ray,node.xfBB)
- ixfRay $=$ Inverse (node. xf)*ray
- If RayHitsBB(ixfRay, node.BB)
- If node is group
- Foreach child in node.children
- RayHitSubTree (ixfRay, child)
- else // node not group
- RayHitGeometry(ixfRay, node.geom)
-ray.collisionInfo.update(ixfRay)


|  | Other Schemes |
| :--- | :--- |
|  |  |
| - Uniform Grid/Octrees |  |
| - Spatial Hierarchies |  |
| $\cdot$ Etc |  |
|  |  |



