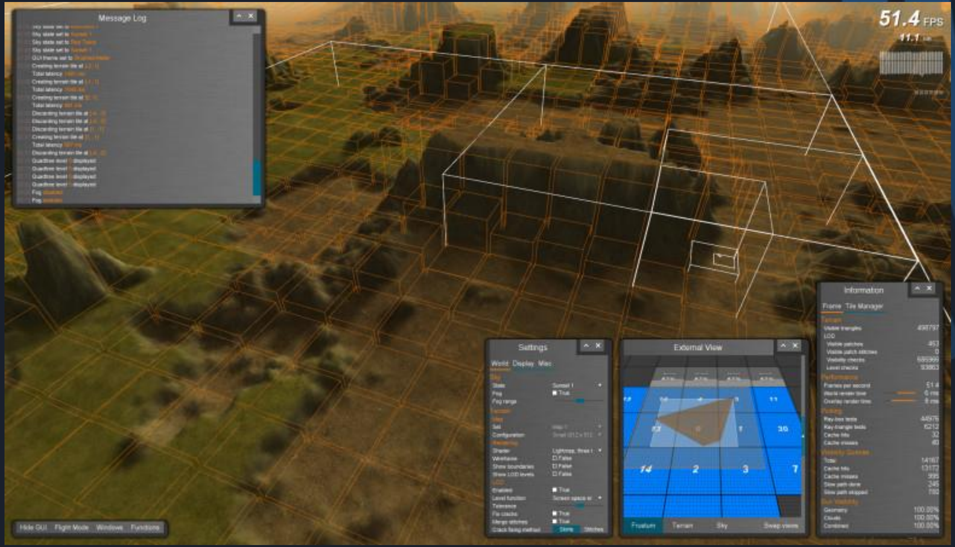
Scene Management

Culling, Bounding Volumes and Scene Graphs

Programming – Computer Graphics

Contents

- Scenes
- Frustum Culling
- Bounding Volumes
- Scene Partitioning
- Scene Graphs
  - Bounding Volume Hierarchies
  - Quad-Trees and Oct-Trees
  - Binary Space Partitioning
- Portals

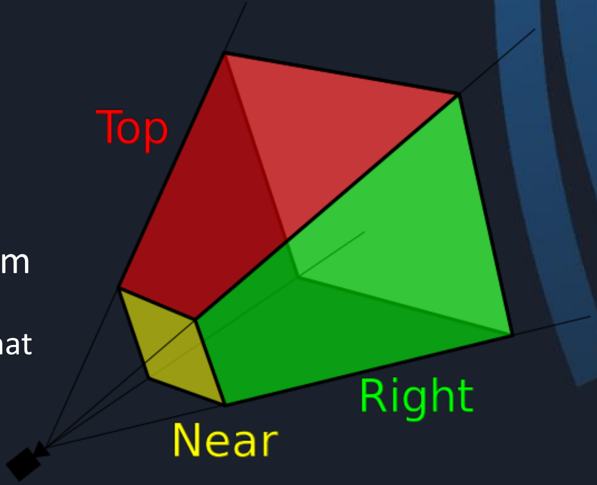


Scenes

- A Scene refers to an environment such as a game level
  - Scenes can become quite large for games and simulations
- Attempting to render all the items that wouldn't be visible can still take time and processing power
  - There is no need to draw things behind you, or update the animations for a mesh in another room
- There are many ways to optimise a scene

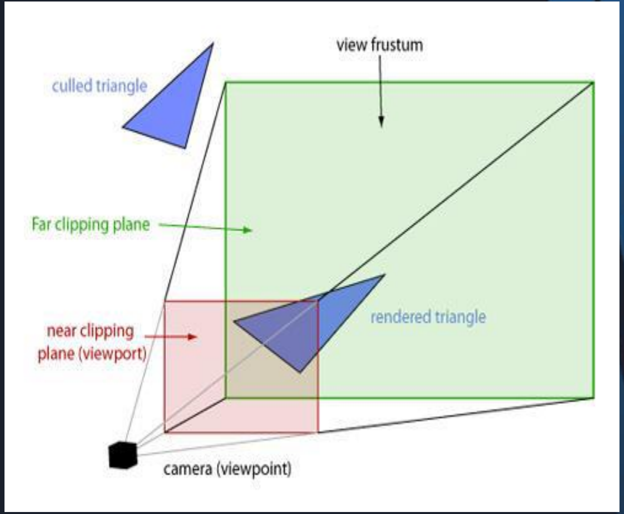
Frustum Culling

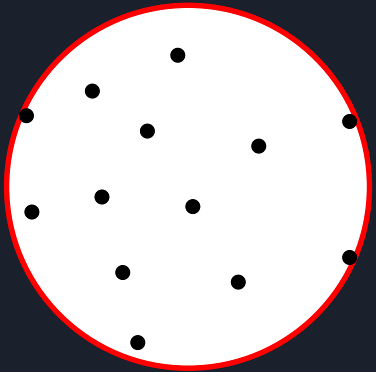
- Frustum Culling is the term given to determining if an item is within view by testing it against the camera's frustum
  - The frustum can be thought of as a pyramid shape with the top cut off
  - The edges of the frustum are treated as 3-D planes with normals facing inwards
- We can compare the geometry of a model against the 6 frustum planes defined by a camera
  - If the model falls behind just 1 plane then it can not be seen by that camera
- The frustum planes can be extracted out of the Camera's projection and view matrices
  - For more information be sure to look up the paper ***Fast Extraction of Viewing Frustum Planes from the World-View-Projection Matrix*** by Gil Gribb and Klaus Hartmann



Bounding Volumes

- Testing all the vertices or triangles of a mesh against a frustum is slow
- Bounding Volumes are shapes that encase all the elements that they bound
  - For example, a Sphere large enough to contain all the vertices in a mesh
- Comparing this bounding shape against a plane then becomes trivial
  - In the case of a Sphere, get the distance from the center of the sphere to the closest point on the plane
    - If the distance is greater than the radius of the sphere and the center was behind the plane then the sphere is fully behind the plane
    - If the distance isn't greater than the radius of the sphere then it overlaps the plane and might still be visible
    - Else it is in front of the plane and is most likely visible





Scene Partitioning

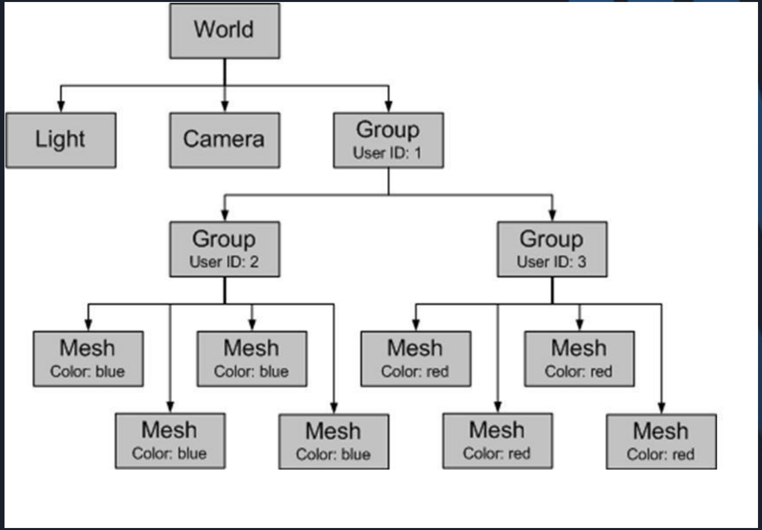
- Frustum Culling helps with determining what is and isn't visible, but more can be done to manage a scene
  - Testing everything against the 6 planes of the frustum can still be slow if there are a lot of items, and we could potentially tell if an item is too far away quite easily
  - An item behind a wall may still be within the frustum

- Scene Partitioning is the term given to dividing up a scene in some manner that allows us to easily cull swathes of data from even needing to be tested against a frustum

Scene Graphs

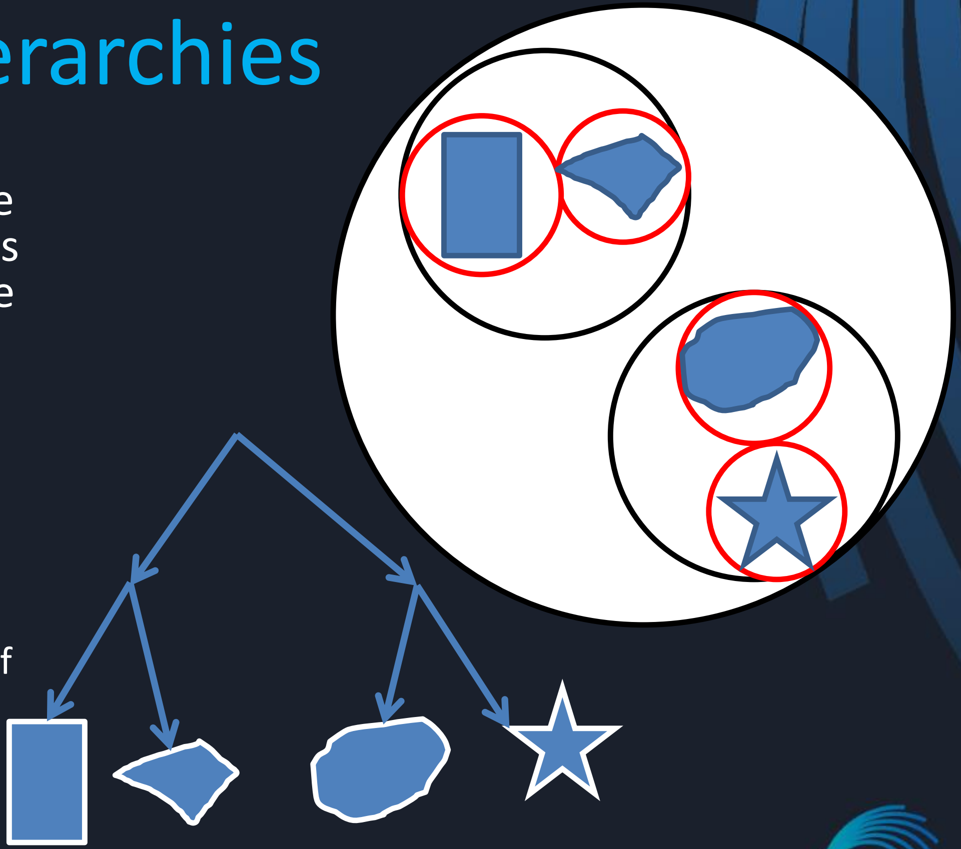
- Scene Graph is the term for structuring a scene within some form of graph structure
  - Typically Directed Acyclic Graphs
  - Tree structures are most common

- We can iterate through the graph
  - If a branch node was culled then it can be assumed its child branches and leaves would also be culled



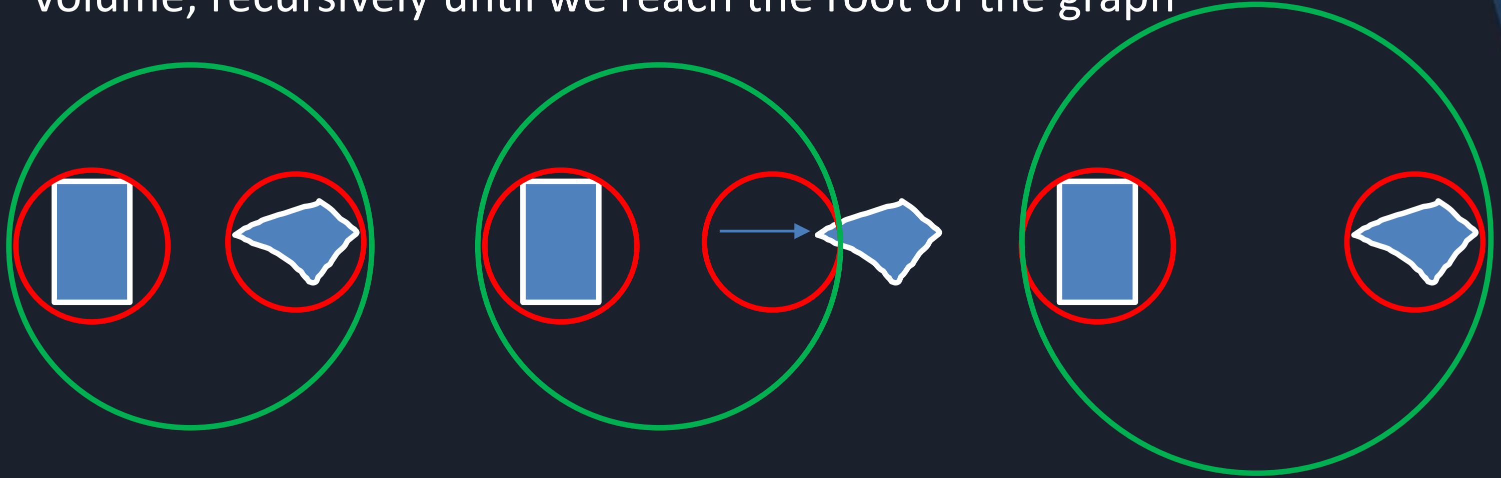
Bounding Volume Hierarchies

- Bounding Volume Hierarchies are tree graph structures where each node has some form of bounding volume shape
  - Sphere, Axis-Aligned Bounding Box, Orientated Bounding Box
- Each node typically has two volumes
  - Local, which contains just the data that makes up the item
  - Global, which contains its Local bounds in addition to the Global bounds of all of its child nodes



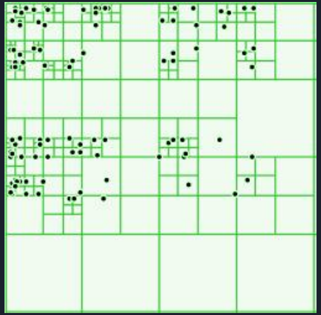
Bounding Volume Hierarchies

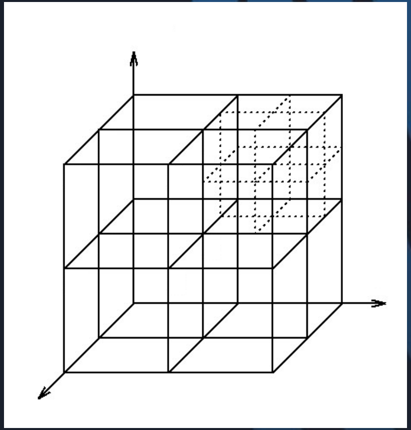
- When an entity moves we would need update its Global bounding volume to contain its Local volume that has moved, plus its child nodes' Global volumes
  - Then we need to inform the parent node to update its Global bounding volume to contain the relocated child Global bounding volume, recursively until we reach the root of the graph

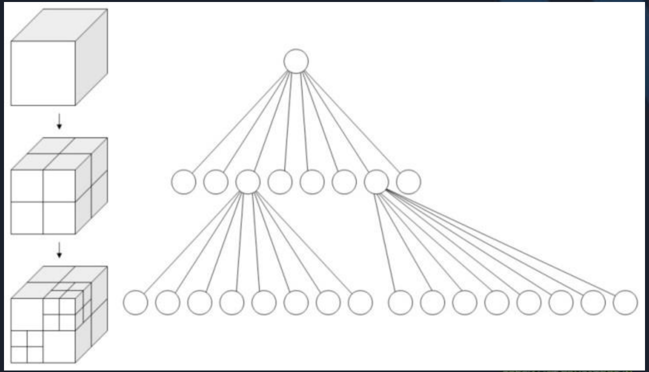


Quad-Trees and Oct-Trees

- Quad-Trees (2-D) and Oct-Trees (3-D) are graphs that divide the scene into quarters
- The scene is sub-divided and items are placed within the nodes that physically contain them
  - Nodes can remain empty if no item is located in the space the node represents
  - A node containing multiple items can be sub-divided further
- Scenes attempt to place a single item within a single node, but in some cases a large item must appear in multiple nodes as it is too big for just one

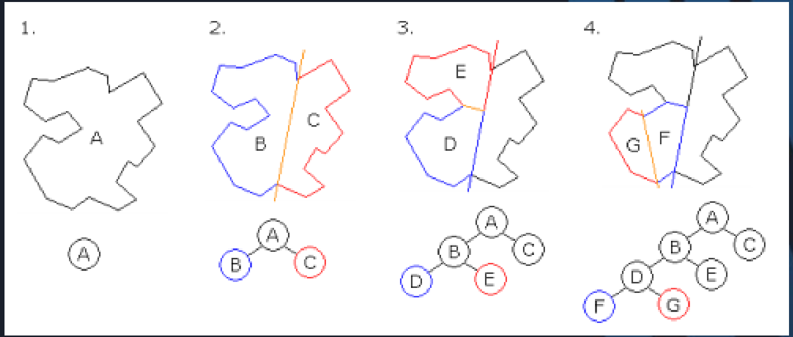


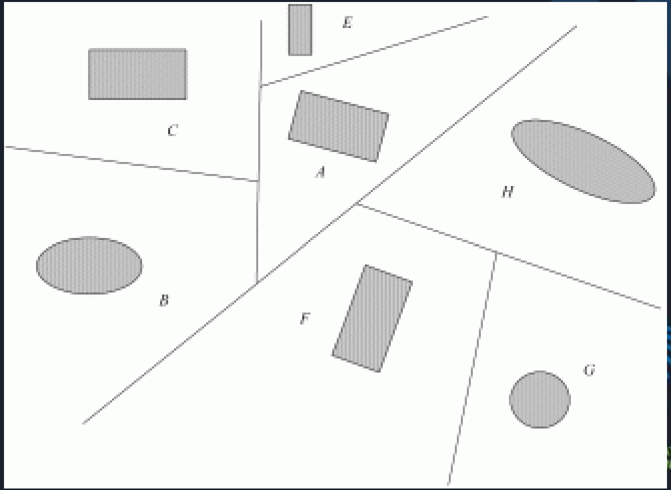




Binary Space Partitioning

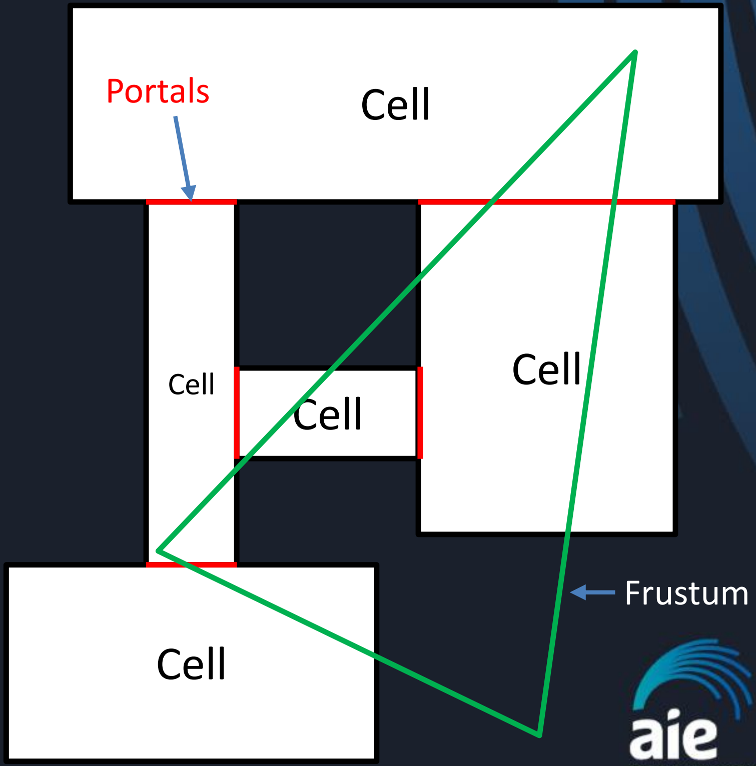
- A BSP tree is a graph where the branches represent a plane that divides the scene in two
  - Items in the front of the plane and items behind the plane
  - If an item crossed the plane then it typically is cut in two
- BSP trees were used heavily in the past when render order mattered due to the lack of a depth buffer
  - Still used in collision systems and some game engines based off the Quake game engines
- Render order could be controlled by determining which side of a plane the camera was on
  - Items behind the plane would first be rendered
  - Items in front of the plane were then rendered





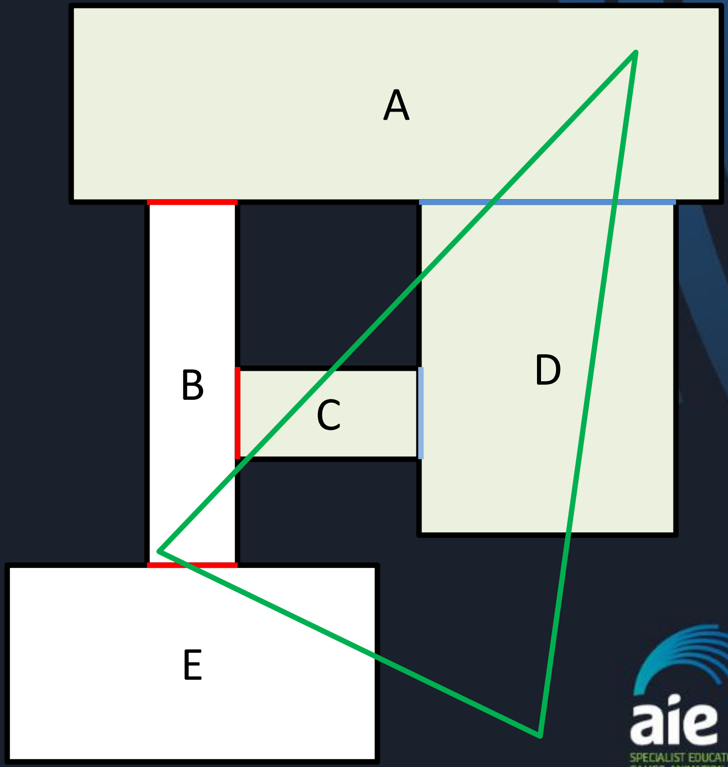
Portals

- Portal Rendering is another form of partitioning
- In Portal rendering a scene is divided up into Cells and Portals
  - A Cell could be thought of as a room
  - A Portal could be thought of as a door into other rooms
  - Commonly a Cyclic Graph
- We first determine which Cell the camera is within and render the contents of the Cell as usual
  - We can arrange Cells to have small Scene Graphs for the items inside the cell
- Then, for each Portal attached to the Cell, we:
  - Determine if we can see a Portal
  - If we can then we render through the Portal into the Cell which it leads to



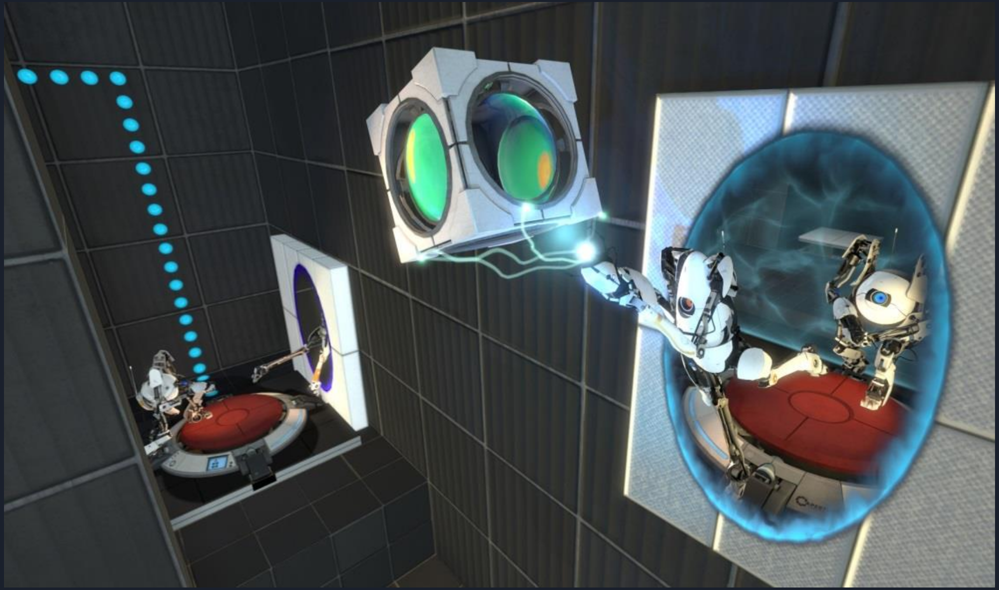
Portals

- In this example the camera starts located in cell A
- We use the frustum to cull visible items within cell A
  - During the culling we determine the camera can see the portal leading to cell D
- We then repeat the process in cell D
  - We use the frustum to cull visible items within cell D and determine we can see the portal leading to cell C
- We repeat with cell C and determine no further portals can be seen
  - Even though cell B and E are within the frustum they aren't rendered as their portals aren't seen



Portals

- Portals don’t have to lead to the adjoining room
  - The Portal could be positioned on a wall, but actually leads to a room with a completely different spatial location



Summary

- Scenes can become quite large
  - Especially with modern day AAA games
- There are many ways to manage a scene
  - Scene Graphs and various culling methods are available
- Almost every game has some form of scene management

Further Reading

- Akenine-Möller, T, Haines, E, 2008, *Real-Time Rendering*, 3rd Edition, A.K. Peters
- Graham, D, McShaffry, M, 2013, *Game Coding Complete*, 4th Edition, Cengage Learning
- Lengyel, E, 2012, *Mathematics for 3D Game Programming and Computer Graphics*, 3rd Edition, Cengage Learning