Cs425 Lecture #5

- Agents/Crowd

Agent-based Simulation

- An agent is
  - responsible for controlling a single entity
  - responsible for **setting goals**, **actions** to achieve those goals, and **react** to external environmental stimuli.
- This method has been used to model
  - flocks of birds, schools of fish, herds of animals,
  - and crowds of people.
- This presentation will focus on modeling **humanoid crowds** in real-time.

Agent-Based Issues

- The primary difficulty in a real-time agent-based simulation is **scalability**
  - Agent-agent interactions can be $O(n^2)$.
    - Sensing: Who is near me?
    - Behaviors / Response to stimuli: What should I do?
    - Very application specific.
  - Animating an articulated human model is expensive

  - **HOW CAN WE SPEED THINGS UP?**

Level of Detail

- **Level of Detail** (LOD)
  - technique used to **approximate** the solution to a complex system.
- LOD is driven by an observer's location.
  - near: simulated with a high degree of fidelity
  - farther: simulated with any number of LODs
- LOD may be applied to
  - sensing, behavior simulation, motion of articulated models, and model geometry

Sensing

- Agents use sensing to gather information about their world/situation
  - Provides input to a **behavior model** which in-turn outputs actions/responses.
- Unlike in the real world, agents (unrealistically)
  - may know the locations of all others.
  - may know the map of the entire environment.
  - may communicate with all others, regardless of location.
- The problem of sensing is in **filtering** all this data

Sensing

- Vision can be implemented in several ways:
  - **Rendering-Based**:
    - images rendered from the agent's POV in false color
    - Object identification
    - Z-buffer for collision avoidance
    - Not scalable to many agents in real-time
  - **Local neighborhood**:
    - Most common.
    - fast and can be tuned for level of detail
    - Example:
      - Database to find nearby agents and obstacles.
        - LOD: Limit search radius. Cap query results.

Behavioral Architectures

- Real-time crowds nearly always require an **interactive** element
  - user⇔agent
  - agent⇔agent
  - agent⇔environment

Behavioral Architectures

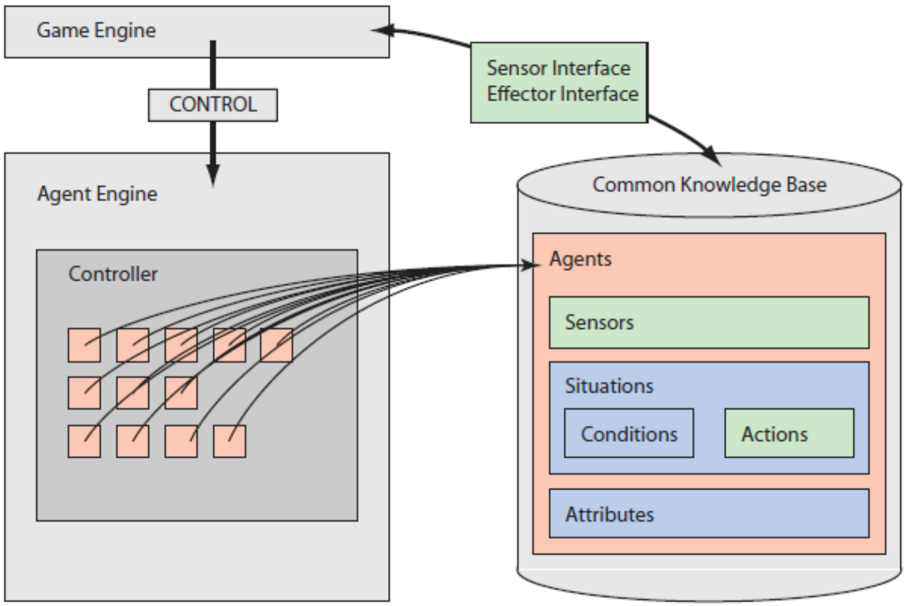
- The typical agent cycle looks like:
  - **Sense**
  - **Think**
  - **Act**
- Can be slow if each agent behaves individually

- Shared sensing, thinking, and acting can be used to reduce computational cost
  - A group generalization may be thought as a LOD.

Behavioral Architectures

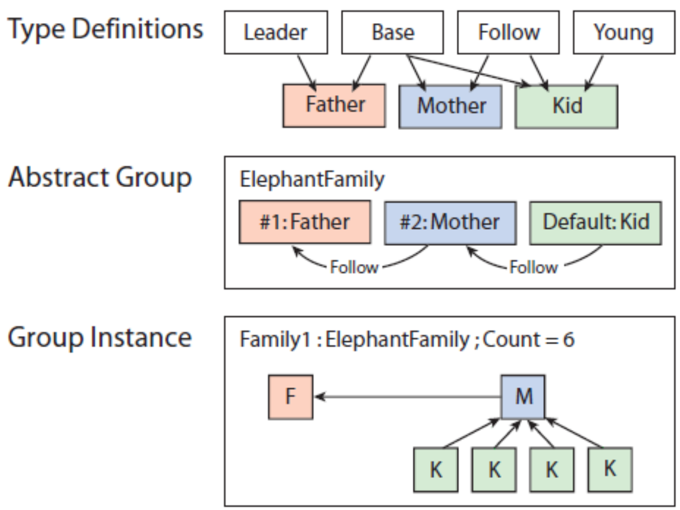
- Shared Knowledge Base
  - All world data is stored in a common knowledge base (KB) shared by all agents.
    - Agent Positional Data
    - Sensors
    - Situations
    - Agent Groups
  - Each agent has KB keys for accessing data.
    - Agent Keys, Group Keys, Sensor Keys

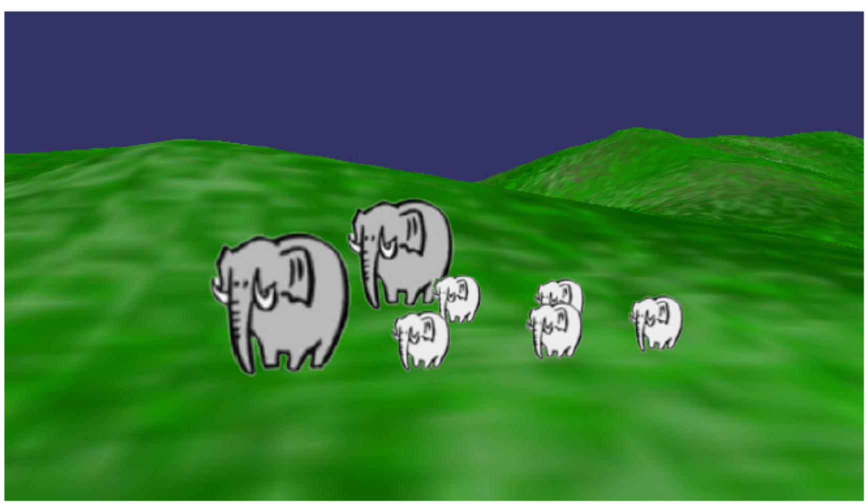
Behavioral Architectures



- Advantages:
  - Agents can quickly access group awareness.
  - Sensors may be shared among agents.
  - Very easy to multithread the “think” phase since it is “read-only”

Behavioral Architectures



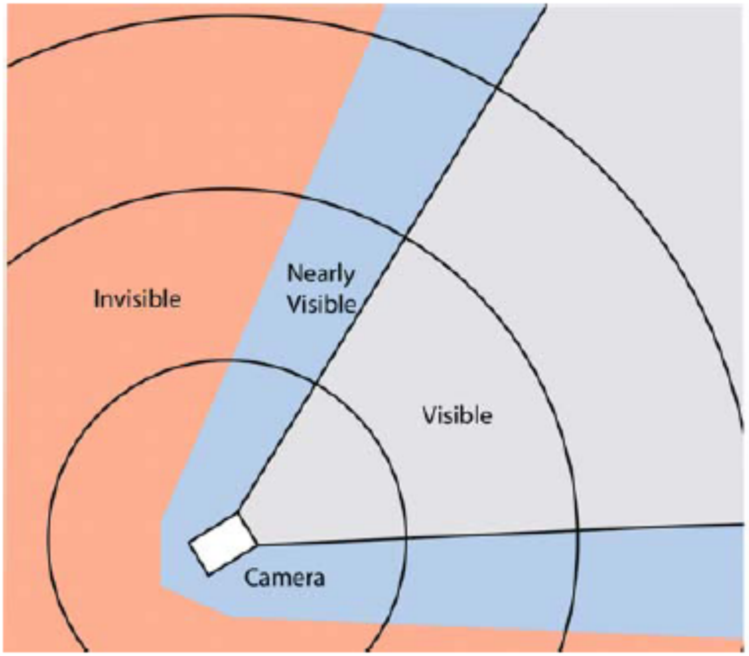


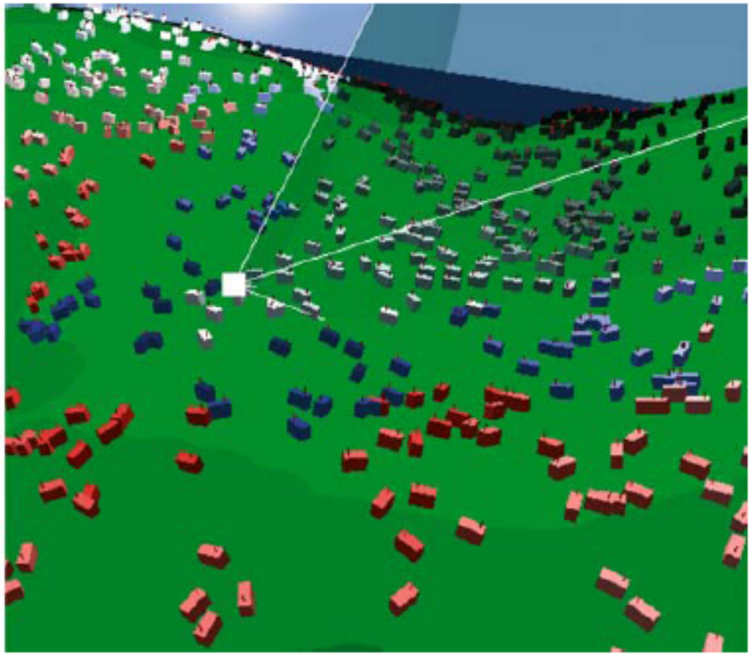
- Group behaviors are realized through hierarchical relations between agents.
  - A typical hierarchy is a leader/follower model
- Heavy “thinking” (such as path planning) is only performed by the **leader**
  - Followers only need to keep up and follow signals raised by their leader

Behavioral Architectures

- Accelerating the Think phase:
  - given a budget (say 10ms) to spend on all agent processing
  - the simulation loop only Thinks for a subset of agents (scheduled using a Round Robin or Priority Queue).
    - LOD may be used to allocate time slice
      - Agents may give up slice to group leader if slice is too small to be useful.
    - Skipped agents continue to perform their previous actions.

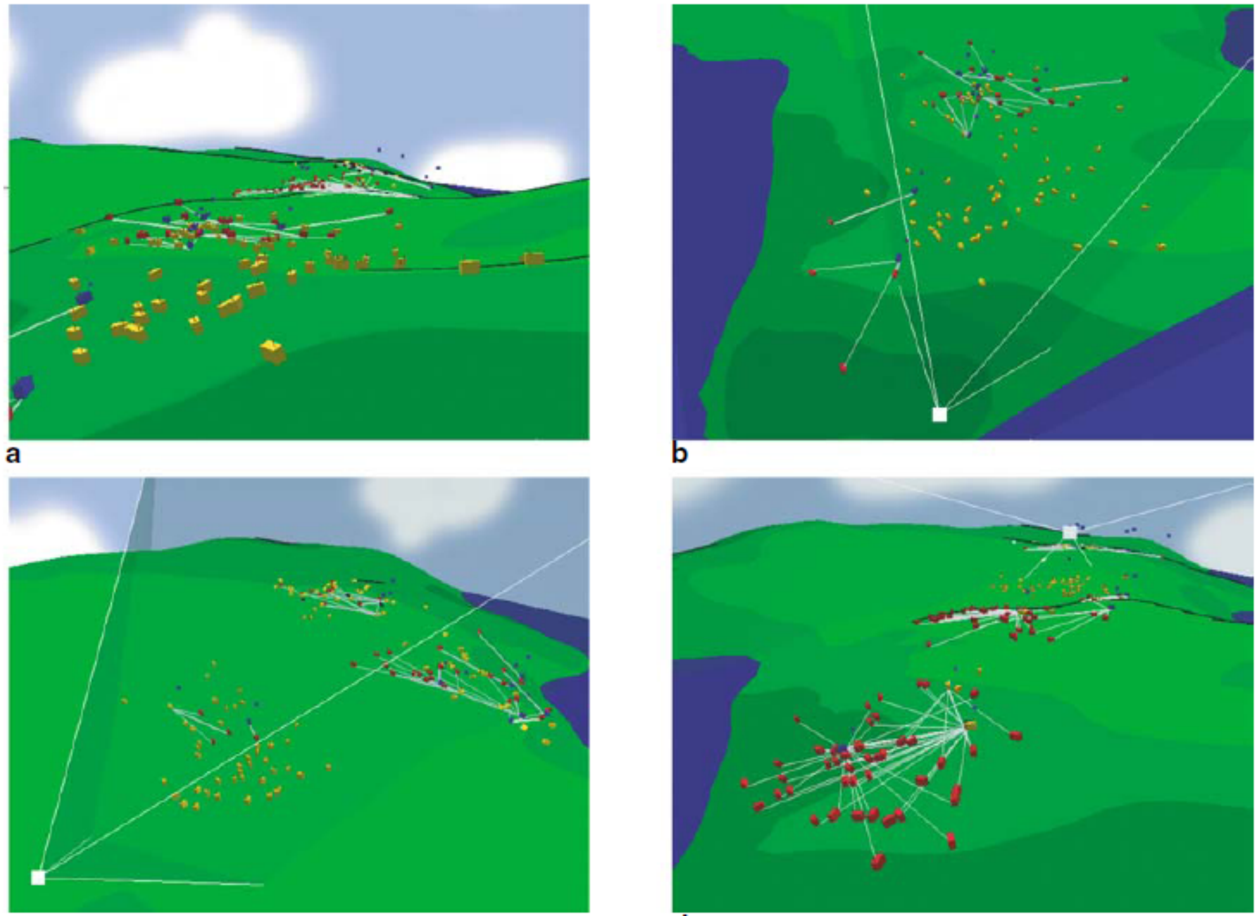
Behavioral Architectures





LOD scheduling priorities are allocated by visibility and distance.

Behavioral Architectures



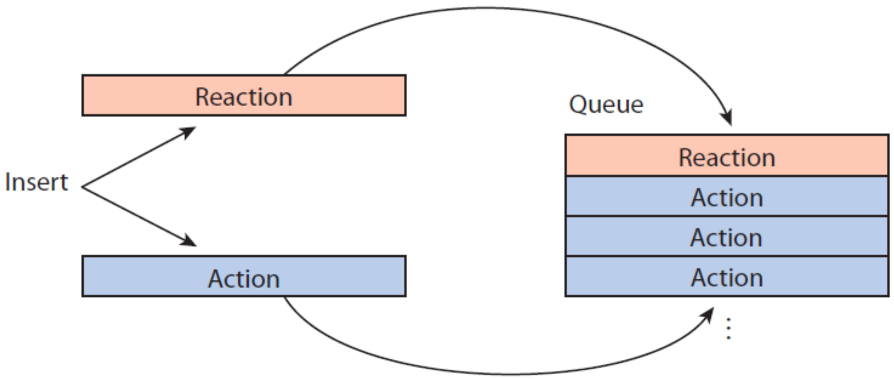
a b c d

Agents in the view and near the camera act independently.

Agents off camera are controlled by group leaders.

Behavioral Architectures

- Accelerating the Act phase:
  - A reduced set of actions can also be executed given time constraints.
  - Actions are prioritized
    - Simulation tries to execute all queued actions, but may have to exit early if the time limit is exceeded.
  - Reactive actions (such as flee a predator) are given priority over less important actions such as follow the leader.



LOD

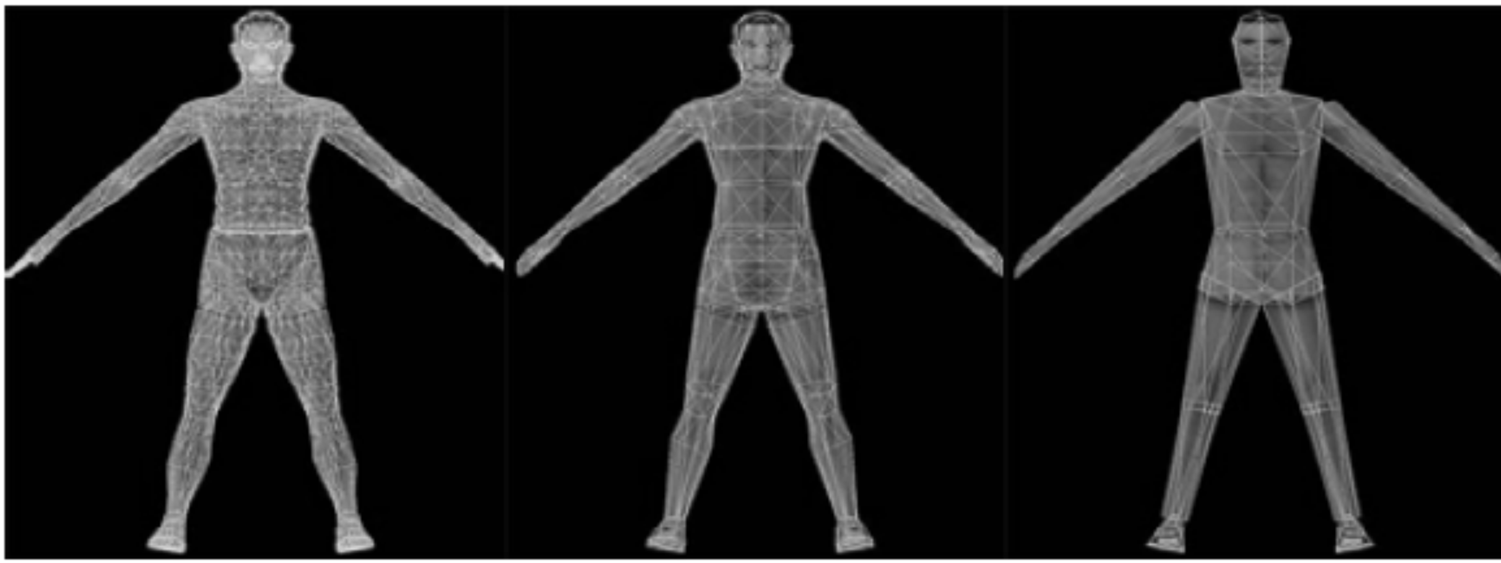
LOD in Rendering

- While not explicitly related to simulation, fast rendering allows for more time to be spent performing simulation.
- There are several techniques that are commonly used to speed up crowd rendering.

Geometric LOD

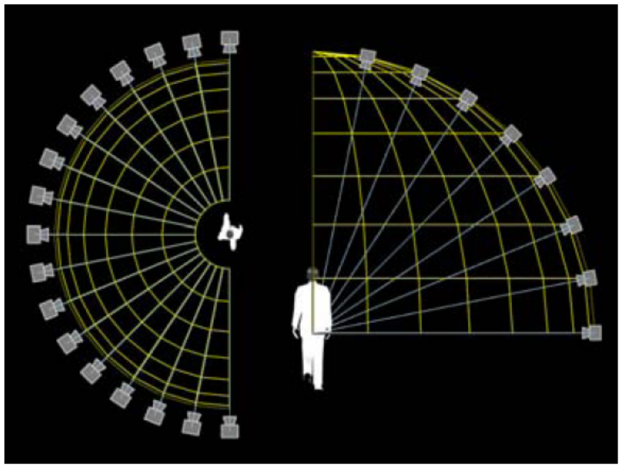
- Geometric & Animation LOD
  - Reduce the number of polygons in the model at different LODs
  - At a certain LOD, stop animating bones of figure and use static-keyframe meshes instead
    - May continue to reduce polygons in the model.
  - At the farthest LOD, use 2D billboards
    - May be a simple sprite animation.
    - Or more complex "polyposter" which is a collection of 2D deformable textured polygons

Geometric LOD



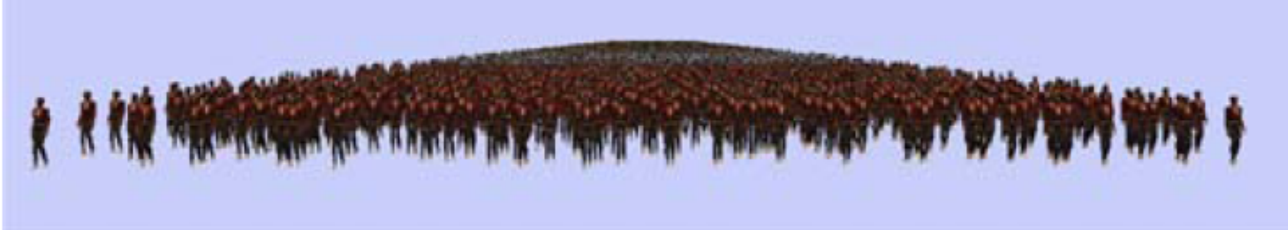
5'000 polygons 1'000 polygons 100 polygons

Geometric LOD



A 2D billboard from 17 views

Billboards can be used to dramatically increase crowd size



10,000 agents: **100** polygons and **9900** billboards

Geometric LOD

Polyposter approximation of a full mesh animation.

Replication

- Replication
  - Commonly used for spectator crowds
  - Transformed agent meshes from an animation can be shared
    - For example, an two agents may share the same model at the same animation frame
  - The model is replicated for both agents
  - Model texturing may be altered for each instance.
    - Ex: Different color of clothes

Clone Attack!

Perception of Crowd Variety