CS 4732:
Computer Animation

Particles, Flocks, Herds, Schools

Robert W. Lindeman
Associate Professor
Interactive Media & Game Development
Department of Computer Science
Worcester Polytechnic Institute
gogo@wpi.edu

Control vs. Automation

- Director's imagination -> infinite
- Time to make movie/game -> finite
- Budget to make movie/game -> finite
- Number of animators -> finite
- Q: What to do?
- A: Automate the animation process
- Other benefits:
  - Movement can be made more ***organic***
  - Mimics some rules of nature
  - Can scale number of elements
    - As long as you scale processing power!

Particle Systems

- Good at modeling "fuzzy" objects
  - Dynamic and fluid
  - Fire, clouds, water
- Stochastic procedural modeling
  - Complex systems can be modeled with little human effort
  - Level of detail can be adjusted
    - fewer particles

Particle Systems: Examples

- Genesis Effect from *Star Trek II*

Basic Model of Particle Systems

- A collection of many minute particles
- For each animation frame:
  - New particles are generated, and assigned a set of properties
  - Old particles die, and are removed
  - Remaining particles change their properties, *e.g.*, position, shape, color
  - The frame is rendered based on this new state
- Creation and attribute manipulation are procedural
  - Can be the result of computations

Changing Particle Properties

- How should the properties of the particles change over time?
  - Where does each particle move to?
  - How does its color change?
- Can be based on *anything*
  - Look at neighboring particles
  - Look at scene objects, like obstacles
  - Look at time
  - Look at distance traveled
  - Look at anything you want!

Basic Algorithm

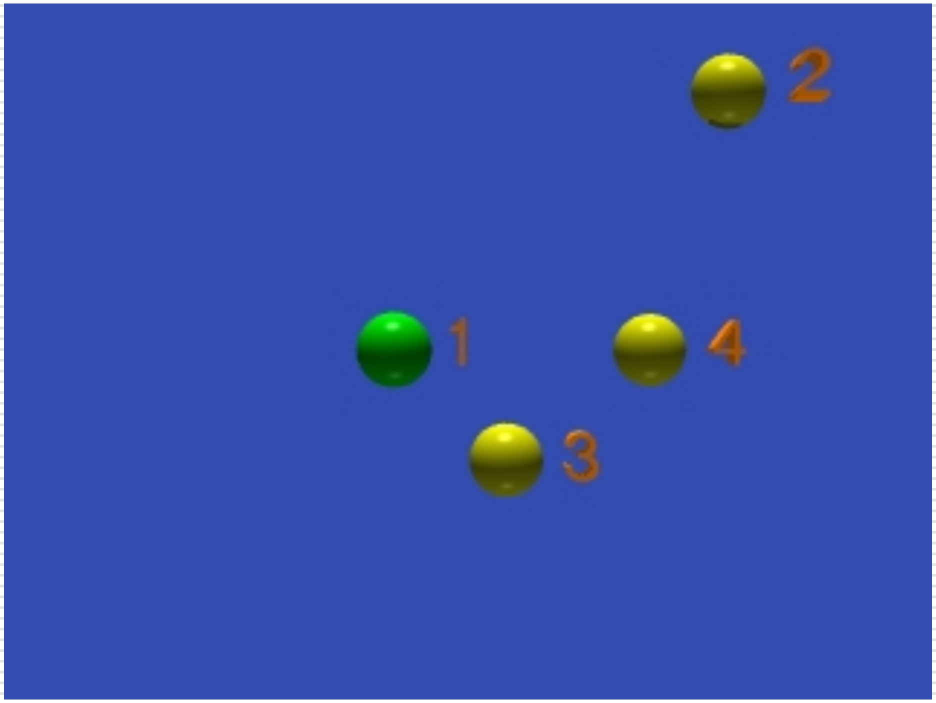
```
Set up particle
While Animation In Progress
  If Particle Not Dead Then
    Add Particle Direction * Speed To Particle Position
    Add Particle Acceleration To Particle Speed
    Modify Particle Speed
    Modify Particle Energy
    If Particles Energy < Threshold Then
      Mark Particle As Dead
    End If
    If Particle Hits Object Then
      Modify Particle Position, Direction, Speed and Energy
    End If
    Display Particle
  End If
End While
```

Example: Movement of Particles

- S_t is the state of all particles at time t
  - At $t=0$: S_0

Images: Greg M. Johnson

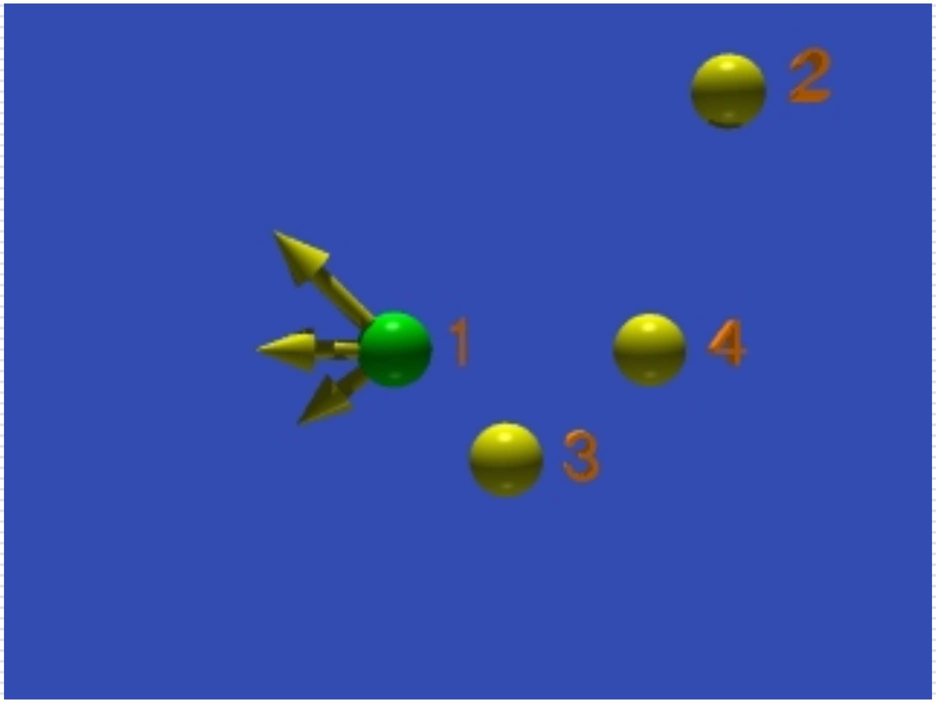
(http://www.geocities.com/pterandon/boids.html)



Example: Movement of Particles

- Compute the influence of all other particles within some range
  - Attraction, repulsion

Images: Greg M. Johnson
(http://www.geocities.com/pterandon/boids.html)

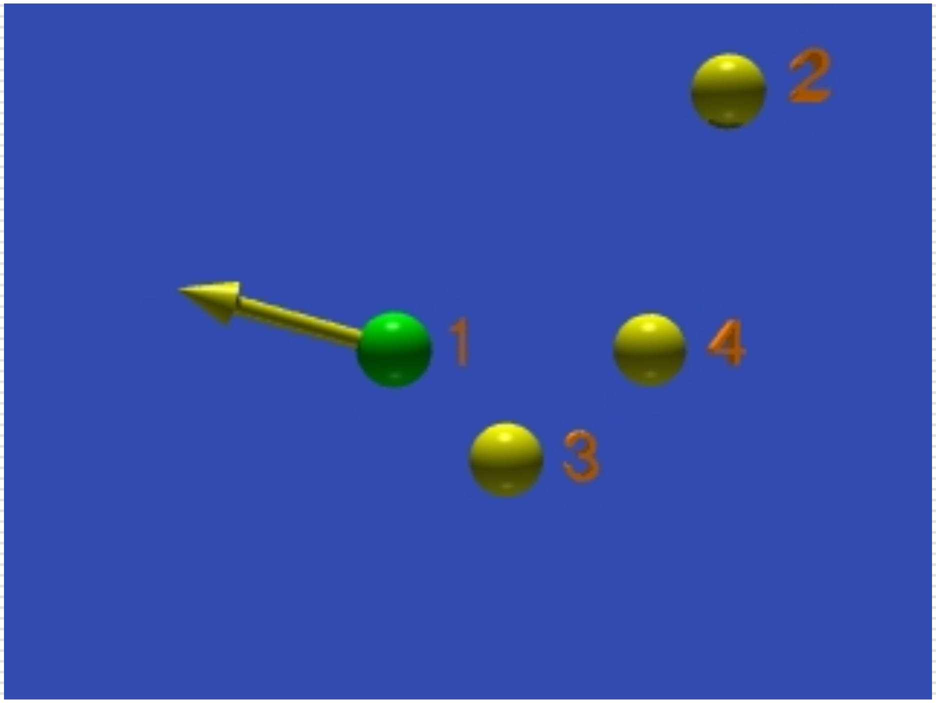


Example: Movement of Particles

- Add all forces together, and use that to update the current position

Images: Greg M. Johnson

(http://www.geocities.com/pterandon/boids.html)

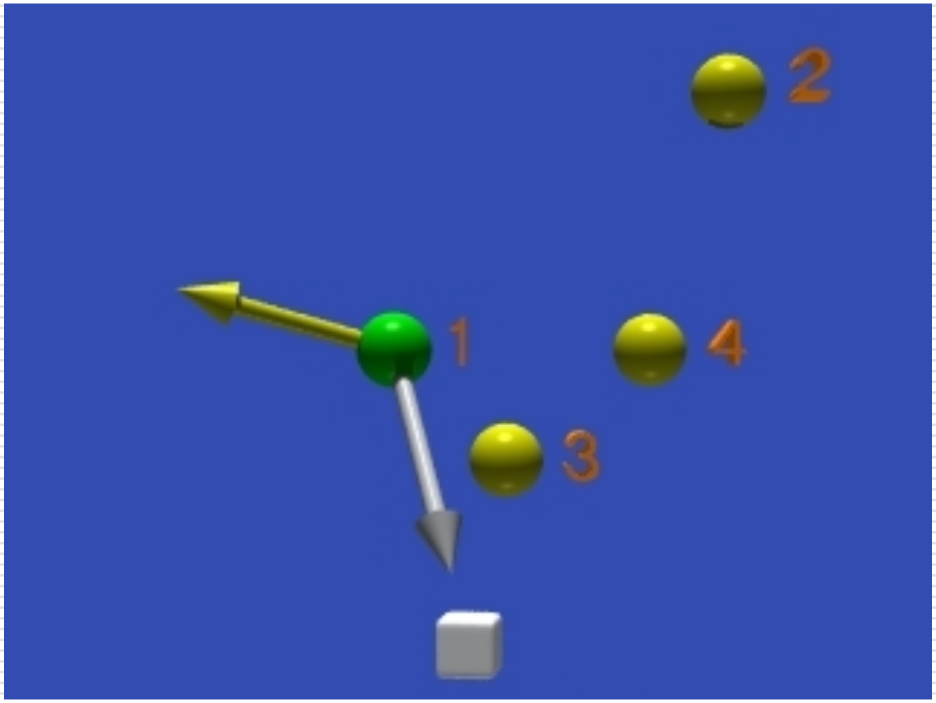


Example: Movement of Particles

- Wait, there might be other forces
  - Whatever the goal is of the scene

Images: Greg M. Johnson

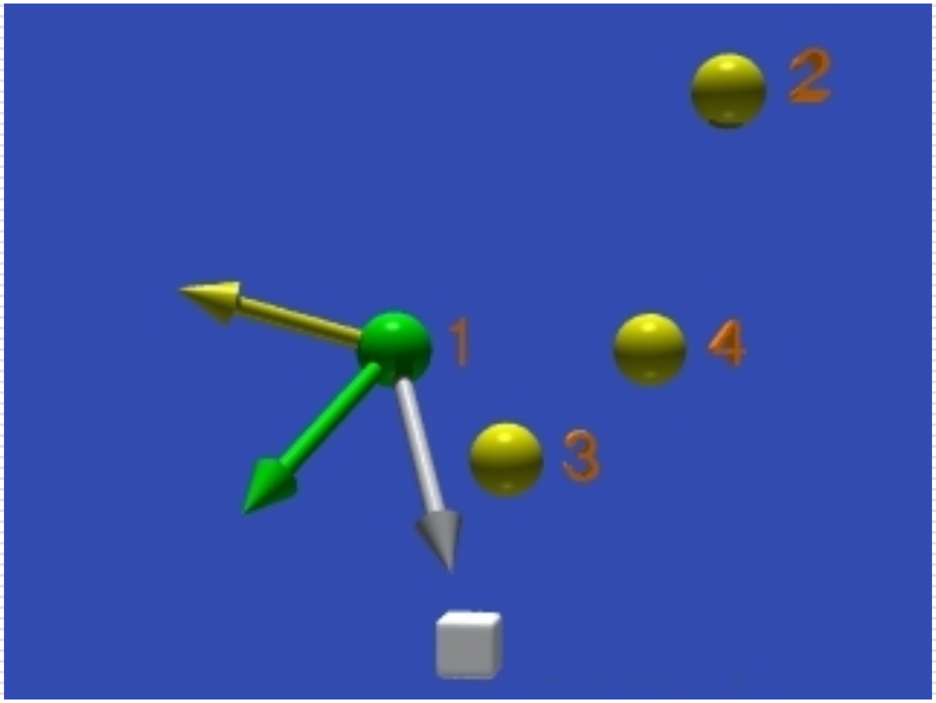
(http://www.geocities.com/pterandon/boids.html)



Example: Movement of Particles

- Again, sum these as the forces on the particle
- Repeat these steps for each particle

Images: Greg M. Johnson
(http://www.geocities.com/pterandon/boids.html)



Particle Systems: More Examples

- Fire
- Explosions

Particle Systems: Final Thoughts

- In many cases, ignore self collisions
  - What does it look like when two fire particles colliding?
- Very general framework!
  - We can make special cases to get specific effects
  - Just change rules, objects, *etc.*
- How would you represent this system in code?

Flocks, Herds, and Schools

- A ***flock*** consists of a group of discrete ***boids*** moving in a visually complex fashion.
- There appears to be some central control, but evidence indicates that the motion is just the aggregate result of individual object motions.
- Problem
  - How do we simulate the motions of a flock in computer animation?

Behavioral Systems

- Special instance of particle systems
- **Flock** is a group of objects that exhibit the general class of polarized (aligned), non-colliding, aggregate motion.
- **Boid** is a simulated bird-like object, *i.e.*, it exhibits this type of behavior. It can be a fish, dinosaur, *etc.*

Flocking Solutions

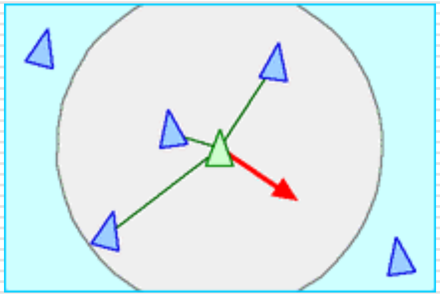
- Well, we could use key-framing for each one
  - We know what we are getting
  - Tough to handle collisions
  - VERY animator-intensive work!
  - Does not scale well
- Instead, allow each object to determine its own behavior

General Approach

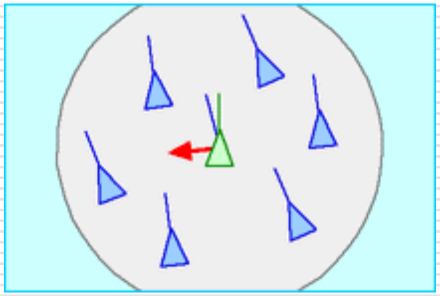
- Each boid maintains
  - An internal state
  - A set of bahaviors
- Fits very nicely into a C++ (Java, etc.) **class**
  - Each boid is an instance of this class
- Three main behavioral rules
  - Separation
  - Alignment
  - Cohesion

Three Rules

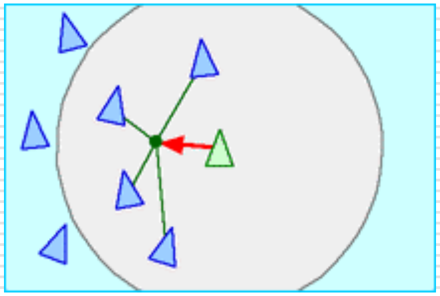
- Separation
  - Steer to avoid crowding local flockmates



- Alignment
  - Steer towards the average heading of local flockmates



- Cohesion
  - Steer to move toward the average position of flockmates

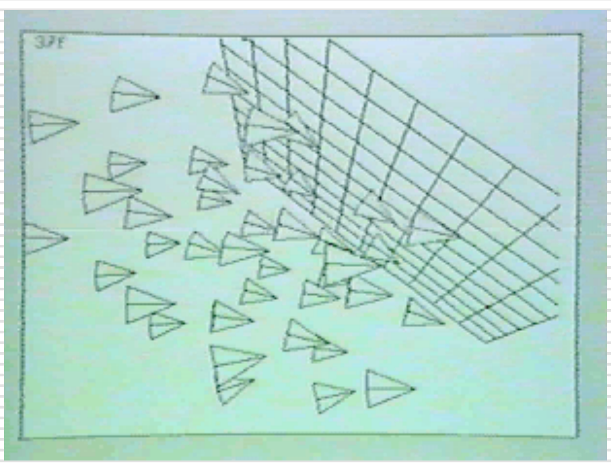


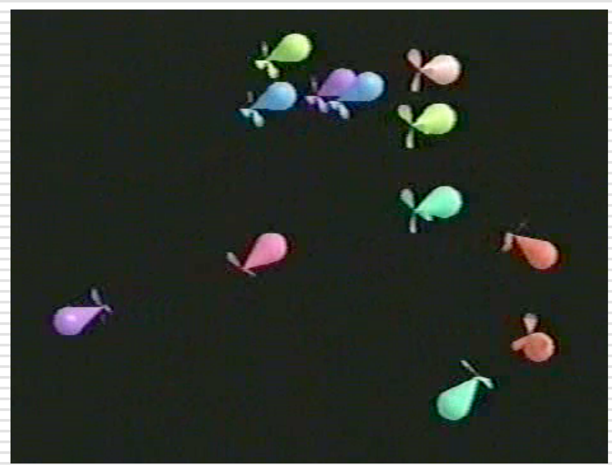
Three Rules, Restated

- Avoid collisions with neighbors and obstacles
- Attempt to match velocity (speed and direction) of neighbors
- Attempt to stay close to neighbors
- These are not orthogonal
  - Collision avoidance helps establish a minimum distance to neighbors
  - Velocity matching maintains it

Boid Brain

- Each boid has access to whole scene
- Each one only considers flockmates in neighborhood
  - Typically defined using a radius
  - Think of fish in murky water, birds in fog



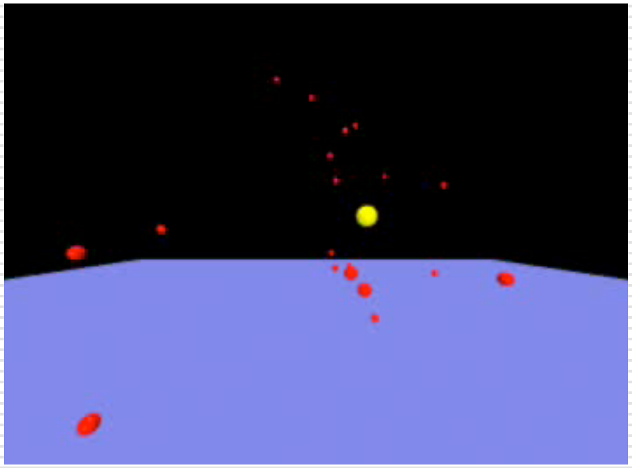


More Rules?

- What else could you do with this?

More Rules: http://www.red3d.com/cwr/steer/

- Seek and flee
  - Food vs. Food?
- Pursue and Evade
- Wander
- Arrival
- Containment
- Wall following
- Path following
- Leader following

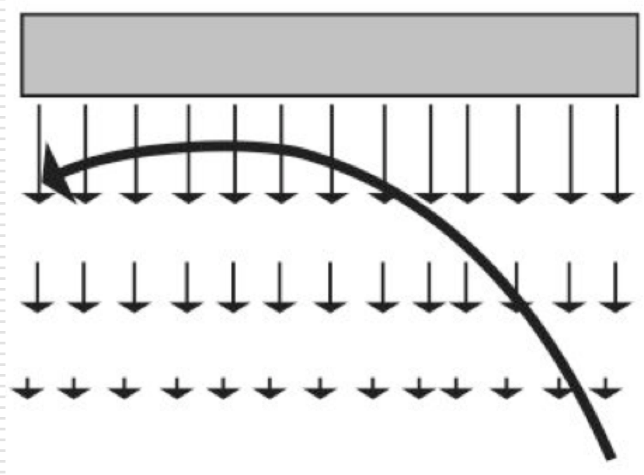


Problems with Behavioral Techniques

- Trade control for automation
  - Difficult to get ***exact*** desired effect
- Solution: Follow the leader
  - How to define leader
- Solution: Use only for background
  - Use something else for foreground characters
- Need to consider ***every*** boid
  - $O(n^2)$ complexity!
  - How can we fix this?

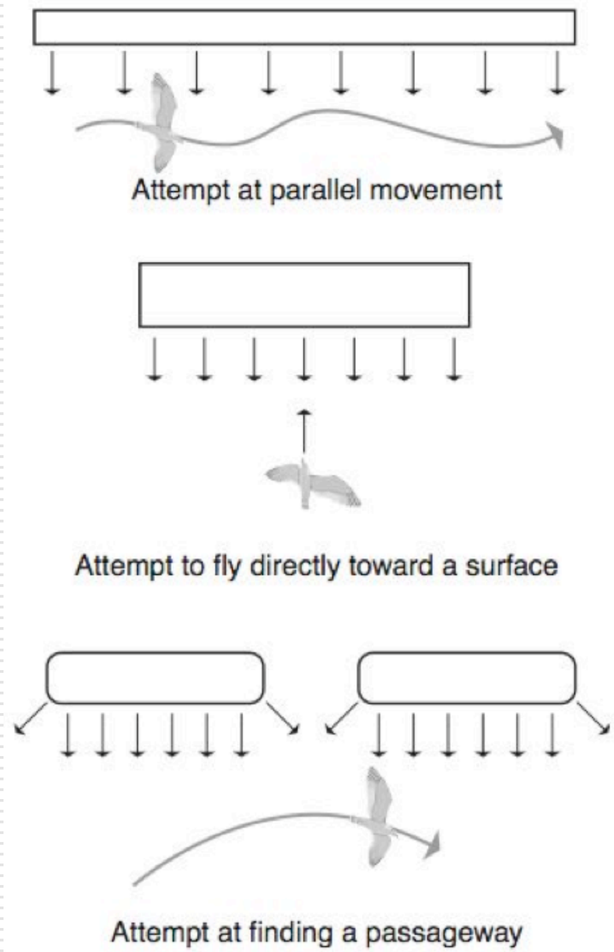
Interacting with the Environment

- We need a way of steering clear of obstacles
  - Just add more force vectors



Problems with Force Fields

- Does not allow boids to get close to objects
- Can lead to stopping
- Tough to move through an opening
- Collision prediction can be used to test if action is needed
  - Ray/sphere intersection test



Knowledge of the Environment

- Boids actually have *perfect* knowledge of the environment
  - Just a database lookup!
- Can led to "super powers"
  - Seeing through walls
- Can use "vision" ("hearing", etc.) to limit accessible knowledge to be local

Vision of a Boid

- Based on “real” vision
  - Limited Field of View (FoV)
  - Visual occlusion
  - Can use only FoV to simplify things
- Can use:
  - Ray casting
  - Simplified Z-buffer
- Once an object is “seen”, access more info from the database
  - E.g., prey vs. predator

Memory of a Boid

- Since vision is fleeting, maybe we need to remember some things
  - Can build a map as you fly, e.g., an occupancy map using an oct-tree
  - Doesn't work for dynamic environments
- Maybe we need something deeper
  - Model more-intelligent behavior
  - An open-ended problem in AI
    - If you think boids are tough, try humans!

Autonomous Behavior

- Modeling cognitive processes
- Must solve the similar problem to simple behavioral motion
  - Balancing various needs and desires
- Cycle for boid decision making
  - Model objects in the local environment
  - Reason about its current state
    - External environment
    - Internal, time-varying urges, desires, emotions
  - Plan a reaction to its current circumstances
  - Carry out actions

Autonomous Behavior (cont.)

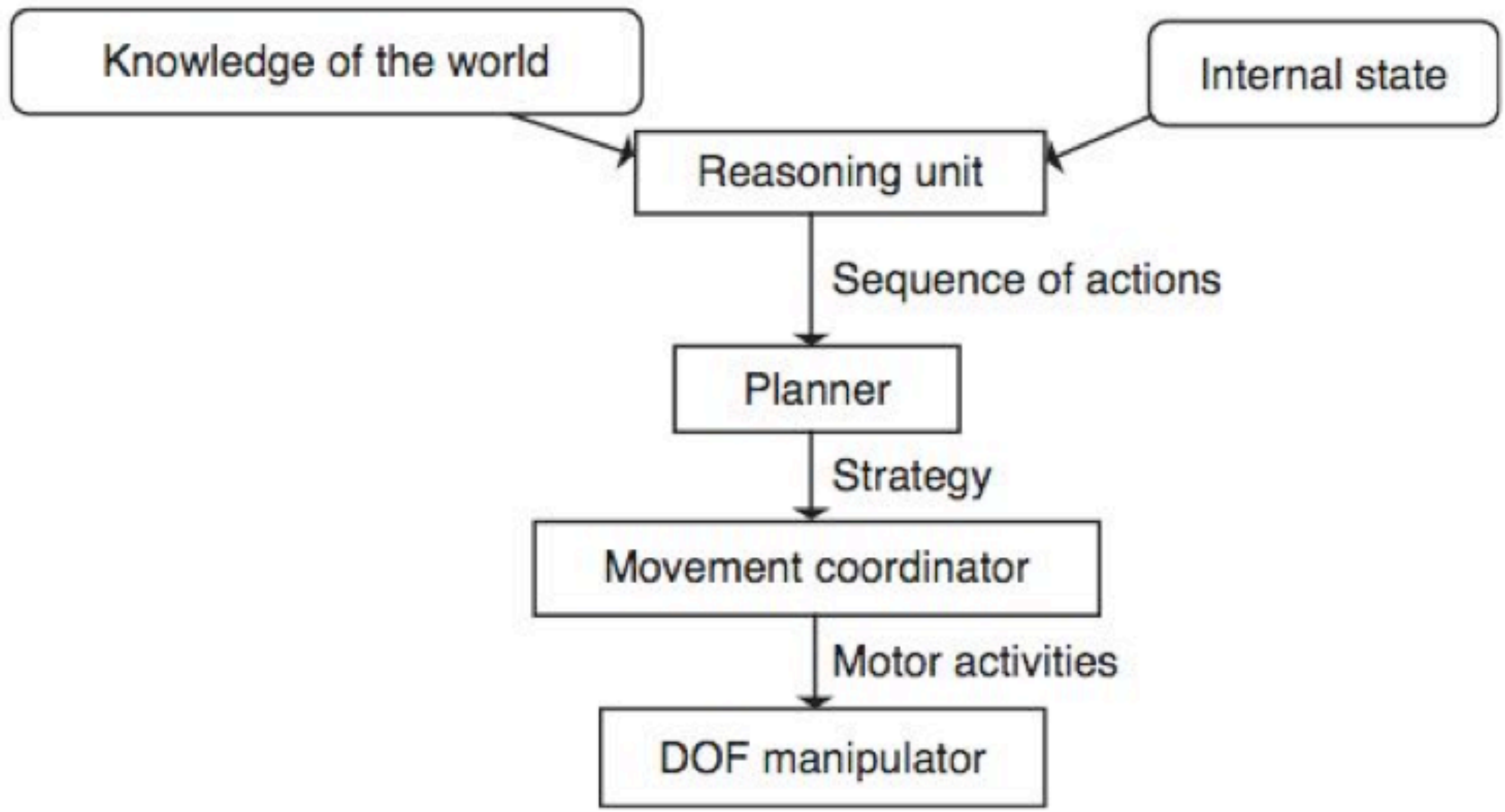
- Can get very complex, very quickly!
  - Need to stop at some point
  - Remember: Good enough is good enough!
- Senses (e.g., vision, touch)
- Perception
- Memory
- Causal knowledge
- Common sense reasoning
- Emotions
- Predispositions

Internal State

- Current feelings can change the weights of different forces
  - Hungry, survival, etc.
- Can divide them into
  - Imperatives
    - Things that *must* get done
  - Desires
    - Things that *should* be done, if possible
  - Idle
    - What to do when I'm not doing anything else

Levels of Behavior

- Can divide things up into levels



Keeping Control

- All of this is about automation
  - What about control?
- Pure automation needs to be tempered with some control
  - At various levels
  - Influence can be used in proportion to animator's desire
    - Action sequences
    - Strategic goals
    - Motor control

Crowd Simulation: Two Main uses

- Visual effects
  - Usually mix
    - live (foreground) action with
    - CG (background) action
    - http://vimeo.com/channels/belalsalem
- Simulation
  - Precision is key
  - Crowds in/out of a stadium or theme park

Crowd Simulation: Differences

- Can be multidirectional
- Can involve psychology
  - Avoidance is primary activity
    - High-density areas:
      - Avoid 5-feet ahead
      - Rotate body, side step
    - Low-density areas:
      - Avoid 100-feet ahead
      - Change paths, move to “open” side, or to the right
      - Pass people by slowing, overtaking, speeding up

Crowd Simulation: Internal Structure

- Subgroups form based on
  - Common urges (going to lunch)
  - Belief systems (political allies)
  - Emotional state (soccer fans)
- Belief system can change
  - Experience, senses, learning, cause/effect

TJ Laughlin's WPI Thesis Work

- People are complex
  - Traits
    - Age, gender, etc.
  - Tendencies
    - Likes dislikes, etc.
  - Mood
    - Changing of weights
  - Social circles
    - Who am I with?
    - Who can see me?

Behavioral Systems: Examples

- Bats and penguins in *Batman Returns*
- All battle scenes in *Lord of the Rings*
- Most battle scenes in *Star Wars*

- Add some stochastic behaviors in order to deter uniformity

- Rob, show Reynolds PlayStation Videos!

References

- W. T. Reeves, "Particle Systems - A Technique for Modeling a Class of Fuzzy Objects", *Computer Graphics*, vol. 17, no. 3, pp 359-376, 1983.
- C. W. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model", *Computer Graphics*, vol. 21, no. 4, pp 25-34, 1987.
- http://www.red3d.com/cwr/boids/
- http://www.red3d.com/cwr/steer/
- http://www.siggraph.org/education/materials/HyperGraph/animation/particle.htm
- http://www.research.scea.com/pscrowd/