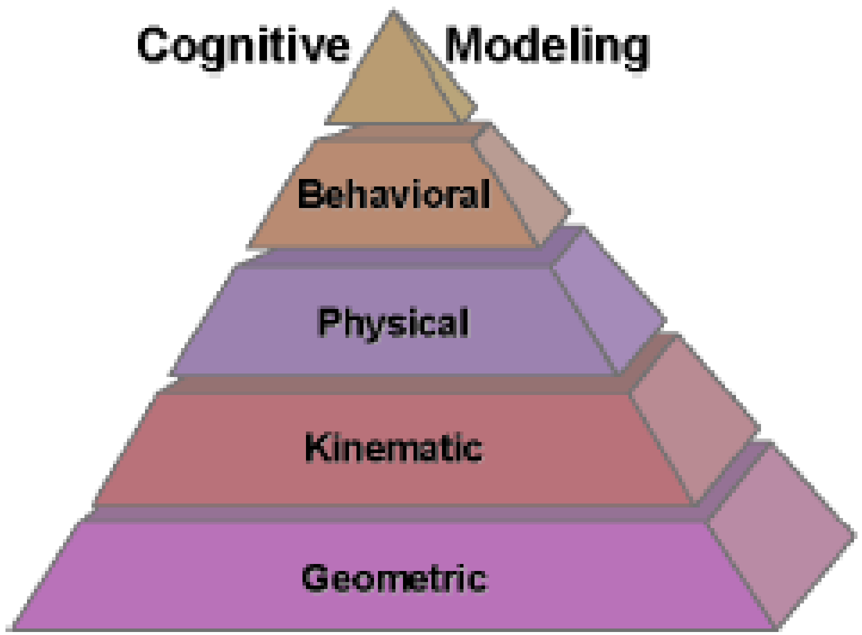
Behavioural Animation

- *slides from Christopher Peters*

CG Modelling Methods

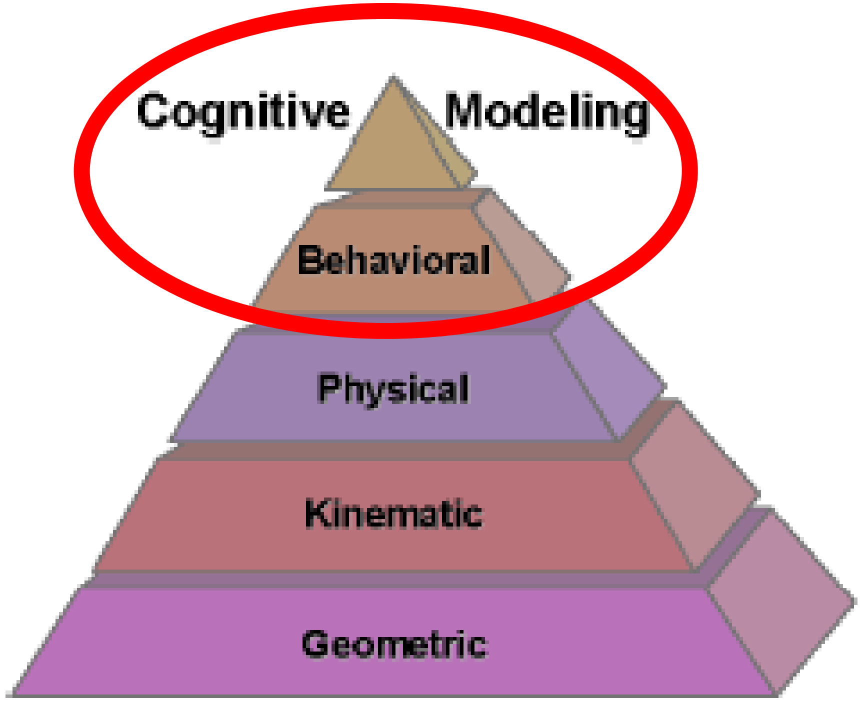
- Geometric
  - Transform vertices, edges, faces (ugh!)
- Kinematic
  - Forward, Inverse Kinematics
- Physically-based
  - Particles, rigid body dynamics, mass-spring systems, hair, cloth
- Behavioural
  - Self-animating characters, reactive, 'senses'
- Cognitive
  - In-depth knowledge representation, reasoning and planning



The CG Modelling Hierarchy

CG Modelling Methods

- Geometric
  - Transform vertices, edges, faces (ugh!)
- Kinematic
  - Forward, Inverse Kinematics
- Physically-based
  - Particles, rigid bodies, deformable solids
- Behavioural
  - Self-animating characters, reactive, ‘senses’
- Cognitive
  - In-depth knowledge representation, reasoning and planning



The CG Modelling Hierarchy

© Funge et al., 1999

Behavioural Animation

- What is it ?
  - A methodology for controlling the animation of agents
  - So what's an agent ?
    - Large research domain - we deal here with real-time *embodied* 3D agents
    - They have a graphical appearance and like to be animated
- High-level animation control method
  - Do not specify control manually at a low-level
    - E.g. rotate arm 45 degrees around x-axis
  - Instead, give each agent repertoire of actions to choose from
    - Actions link to lower-level animations
  - *Agent* decides, to some degree, what actions to perform based on its *internal state* and the *external situation*
    - According to predefined rules and programs
- Intersection between A.I. and graphics research

Simple example

- Greek Hoplite

```
…
If (enemy is in spear range)
        if (enemy == cavalry)
            if (morale == high)
                    attack with spear
            else retreat
        else
            close to short sword range
```

Simple example

- Greek Hoplite

```
…
If (enemy is in spear range)
        if (enemy == cavalry)
            if (morale == high)
                attack with spear
            else retreat
        else
            close to short sword range
```

External situation

Internal state

Link to play an animation

- This is referred to as *action selection*
    - More sophisticated than this in practice…

In Movies

- MASSIVE
  - Multiple Agent Simulation System In Virtual Environment
  - Agents decide on reaction based on the situation, using a ‘brain’
  - Reactions activate key-framed or motion captured animation clips

Lord of the Rings, © *New Line Cinema*

Interactive Entertainment

Rome Total War, © *The Creative Assembly*

- Total War Series
- Challenging:
  - Real-time, interactive
  - Networking issues to be considered

Behavioural Animation

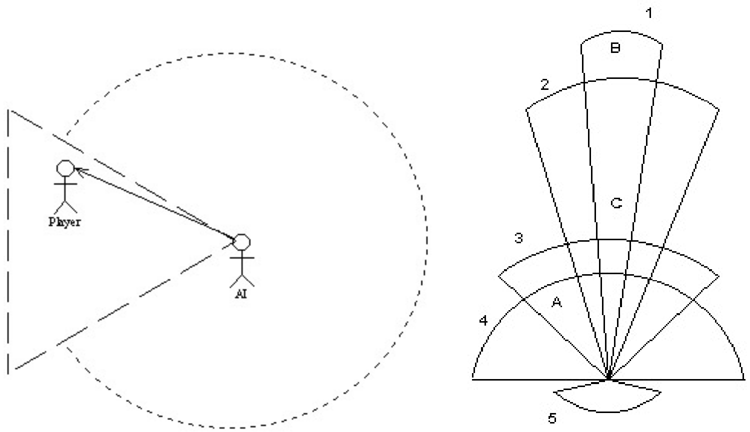
- Important Factors:
  - **Degree of Autonomy**
    - Agent has selection of potential actions it can make in certain situations and decides *itself* (according to its program) which to select given the current situation
    - Decisions may take place at multiple different control levels
  - **Reactive Behaviour**
    - Decision-making mechanisms are often lightweight …
    - …although term 'behavioural animation' sometimes used as umbrella concept encapsulating agents that are cognitive to different degrees, capable of more sophisticated reasoning
  - **Sensing**
    - Agent capable of sensing the external environment
    - Varying degrees of sensor sophistication
      - E.g. Ray casting → synthetic vision

Autonomy

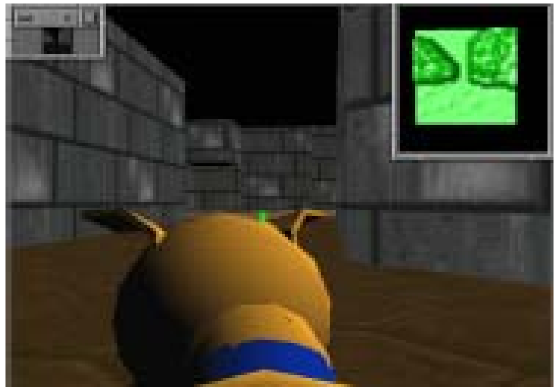
- Directors chair
    - E.g. ‘Go here and fight them’
  - Director specifies the decision making rules
    - Animator is becoming an A.I. programmer !
    - More in-depth A.I. considered in cs7et03 – here, we do consider so much reasoning, learning etc
  - Agents and groups can be autonomous to varying degrees
    - MASSIVE: agents not fully autonomous
      - Scenario set-up done manually, ‘brain’ then manually edited to match general desired outcome of scenario
    - Full autonomy: agents endowed with internal motivations and generate their own intentions, goals and tasks
      - Think of an A.I. player in Total War series

Sensing

- Why ?
  - Environment database contains all objects and their states
  - Agents can have *unrestricted* access to this data
    - Agents may be perceived to have unrealistic abilities
    - Behaviours are more similar since all agents have same input
  - Provide agents with simplified senses
  - Examples:
    - **Volume tests** –anything falling within the volume is ‘sensed’
    - **Ray-casting** – shoot one or more rays out and check for collisions
    - **Synthetic vision** – render the scene from perspective of agent
      - None attempt to tackle machine vision problem
- Active vision – orient senses



© Leonard 2003



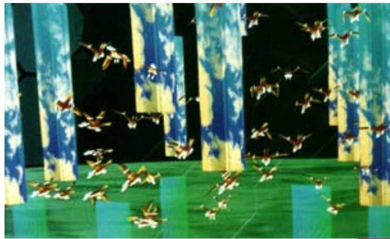
© Blumberg 1997

Issues

- Advantages:
  - Emergent behaviour
    - Simple set of rules can lead to the *appearance* of complex behaviours
  - Useful for animating large crowds
    - Do not have to manually adjust the animation of each agent individually…
    - …but important to provide differentiation in the behaviour of agents so they do not all act in the exact same way
- Disadvantages:
  - Hard to achieve specific desired effects
    - Agents animated only through indirect means
    - Reacting to each other and to environment
    - Many variables to consider so outcomes can be hard to predict

Key Research from Graphics

- Boids
- Artificial fish
- ALIVE: Silas T. Dog
- Improv
- Cognitive modelling



© Reynolds, 1987

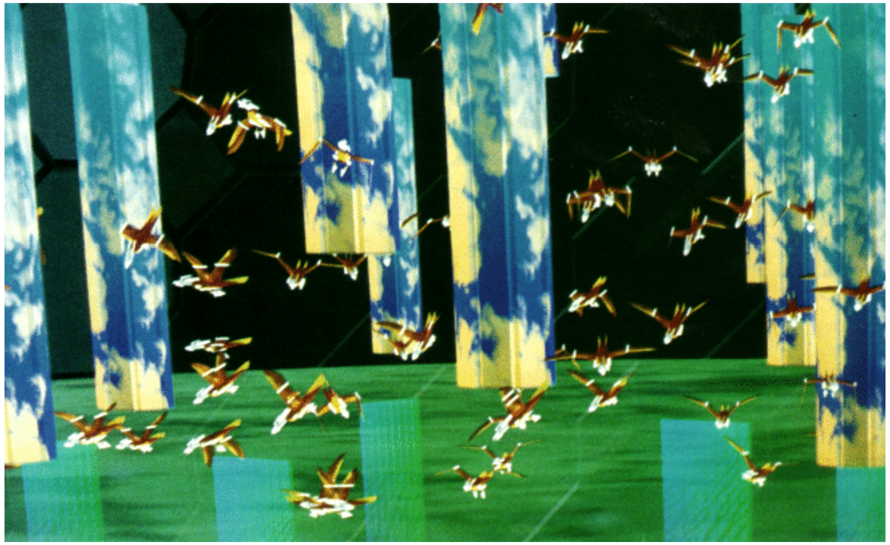
© Blumberg and Galyean, 1995

© Tu and Terzopoulos, 1994

© Funge et al. 1999

© Perlin and Goldberg, 1996

“Boids”

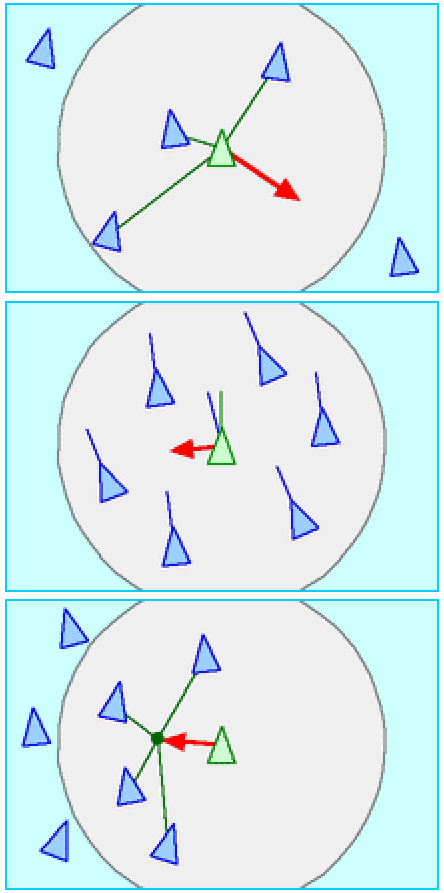


© Reynolds, 1987

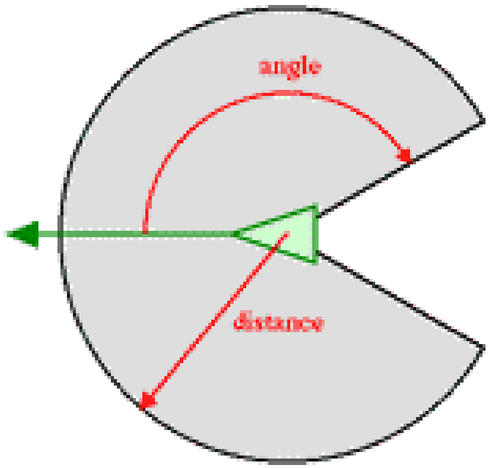
- Bird-oids
- Reynolds, SIGGRAPH 1987
- Useful for modelling animal behaviour
  - Flocks, herds and so on

The core model

- Each boid maneuvers based on the positions and velocities of nearby flock-mates
- Core model contains three types of steering behaviours
  1. Separation: avoid crowding
  2. Alignment: steer towards average heading
  3. Cohesion: move towards average position
  - Each boid ‘senses’ only within local spherical neighbourhood of itself
  - Boids also avoid static and mobile objects



Sensing



© Reynolds 1999

- Each boid has direct access to scene database
- Only reacts to flock-mates within certain volume
- Volume defined by distance and angle
  - *Distance* from center of boid
  - *Angle* from direction of travel
- Defines characters perceptual field-of-view

Why it looks good

- The flock appears to be a single entity, staying together and moving in the same direction
  - Each boid, however, does not do the exact same thing
- The type of behaviour generated is complex, neither *chaotic* nor *ordered*
  - Unpredictability over moderate time-scales
    - In three minutes time, cannot predict which direction the flock will be moving in
  - Predictable at very short-term time-scales
    - Easy to predict exact next movement of each boid
- At edge of chaos
  - Important for perception of life-like behaviour

Time complexity

- Original model was $O(n^2)$
  - Each boid checked to see if each other boid was a nearby flock-mate
- Use of spatial data structure reduces complexity to $O(n)$
  - E.g. see the *Approximate Nearest Neighbour Library*
  - http://www.cs.umd.edu/~mount/ANN/
- Allows fast querying of nearby flock-mates for given boid

Method

- Separation
  - Compute repulsive force for all nearby flock-mates
    - Rf(MyPos,OtherPos) = Normalise(Offset(MyPos – OtherPos) / r^2
  - Repulsive forces are summed together to produce overall steering force
- Cohesion
  - Compute average position (centroid) of all nearby flock-mates
  - Apply steering force towards this position
- Alignment
  - Find average unit forward vector of all nearby flock-mates
  - Steering force is difference between this vector and current vector

Putting it together

- Sum up all three steering force vectors to obtain combined steering vector
- Better control by:
  - Normalise and weigh each vector before summing
- Flocking behaviour speicifed by 9 parameters:
  - $Separation_w$, $Separation_d$, $Separation_a$
  - $Cohesion_w$, $Cohesion_d$, $Cohesion_a$
  - $Alignment_w$, $Alignment_d$, $Alignment_a$
  - Where w = weighting, d = distance, a = angle
  - *d* and *a* relate to the sensory volume, or 'neighbourhood' around each boid

Moving beyond "boids"

- Boids behaviour repertoire extended
  - General steering behaviour applicable to more than animal flocking

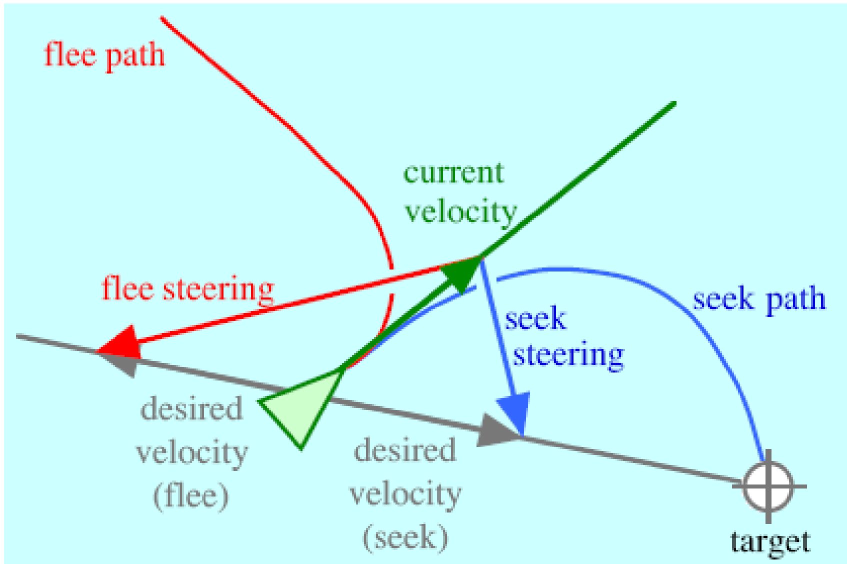
- Core model:
  - *Separation*
  - *Cohesion*
  - *Alignment*
- New behaviours:
  - *Seek*
  - *Flee*
  - *Pursuit and Evasion*
  - *Arrival*

- And …
  - *Obstacle avoidance*
  - *Wander*
  - *Path following, wall following and containment*
  - *Flow-field following*
  - *Unaligned collision avoidance*
  - *Leader following*
  - *Interpose*

Seek and Flee

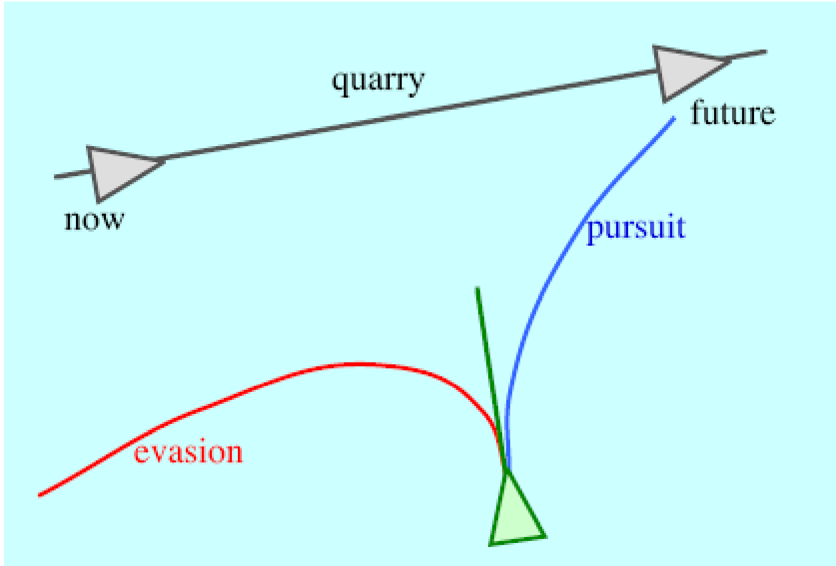
- Steer character towards specific *static* global target position
- Desired velocity is vector from character to target
  - (Normalise (MyPos – OtherPos)) - currentVelocity

- *Flee* is inverse of *seek*
  – Velocity points in opposite direction
- When character moves through target, it will turn back to approach again



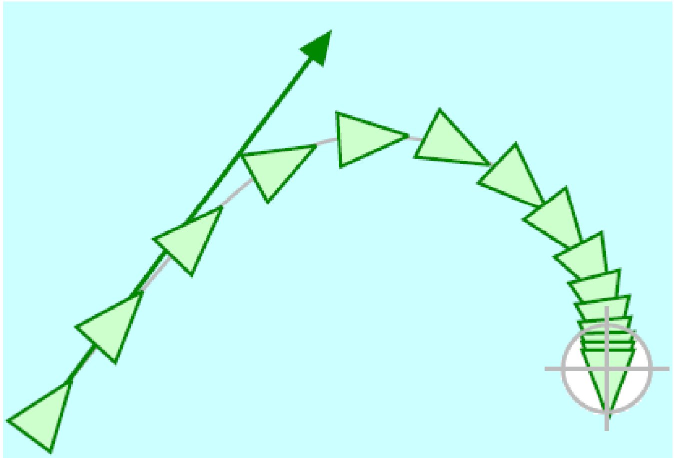
Pursuit and Evasion

- Pursuit similar to seek but target is mobile
  - Predict future position
    - Scale velocity by T
    - Add resulting offset to current position
  - Apply *seek* to future target location
- Not good estimator
  - Linear, assumes target won't turn
- Prediction interval T important
- Evasion uses flee instead of seek



© Reynolds 1999

Arrival

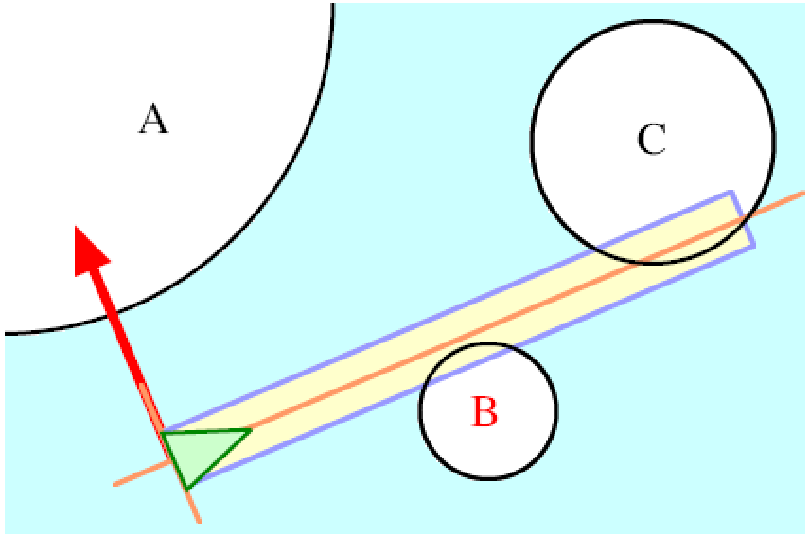


© Reynolds 1999

- Same as seek when character far from target
  - Slow down as it nears the target and stop
- Stopping radius
  - Inside this radius, desired velocity ramped down to 0

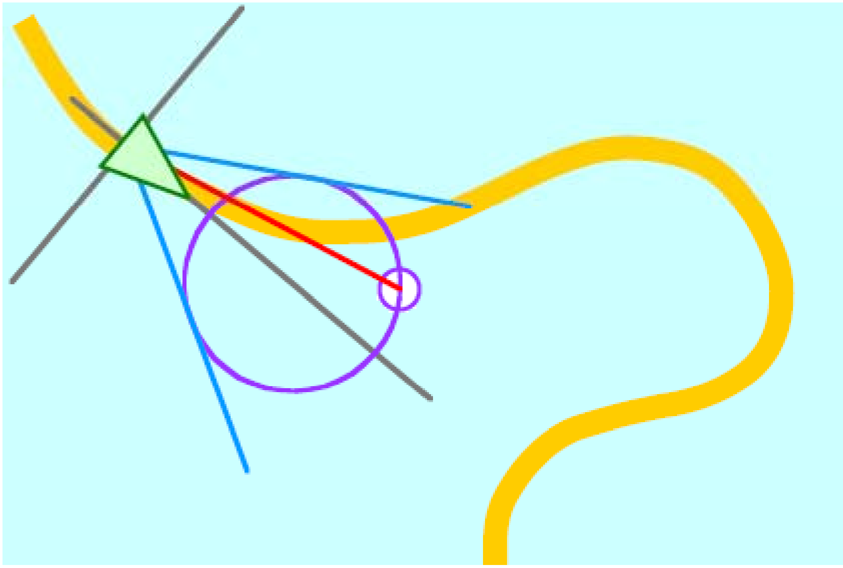
Obstacle Avoidance

- Distinguish from flee
  - Takes place only when obstacles lie directly in front
- Spherical approximation
  - Cylinder projected in front of character
  - Diameter matches character width
  - Determine what spheres intersect with the cylinder
  - Choose nearest



© Reynolds 1999

Wander

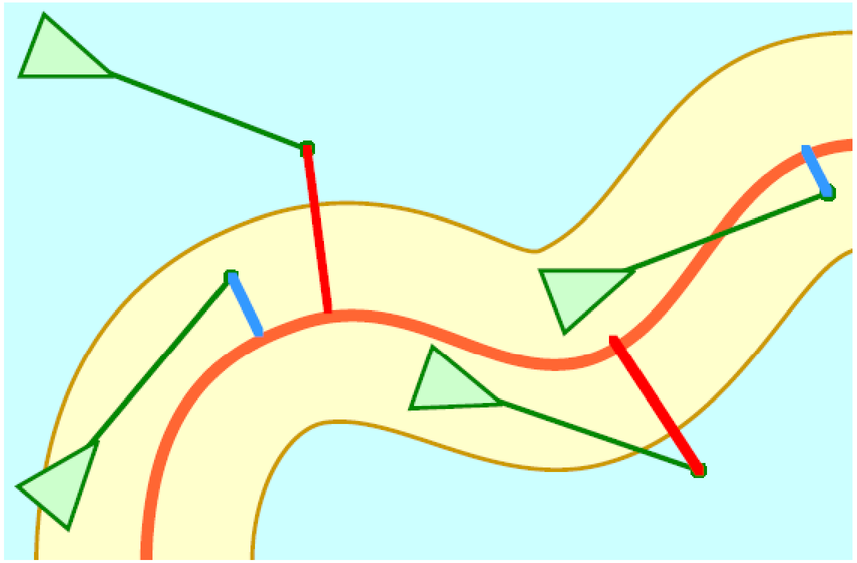


© Reynolds 1999

- Type of random steering
  - Apply small random displacements to steering direction
- Constrain steering force to surface of a sphere
  - Located in front of character
  - Add random displacement to previous
  - Constrain to sphere
  - Sphere radius controls wandering 'strength'
  - Displacement magnitude determines wander 'rate'

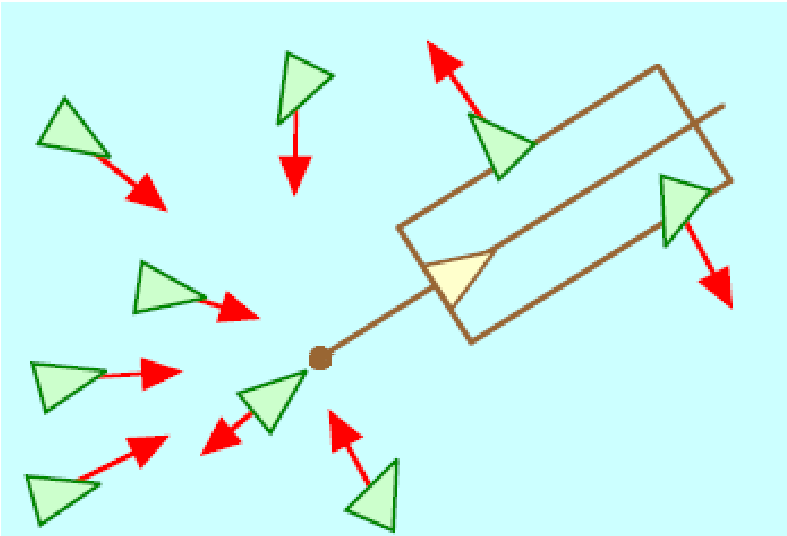
Path following

- Steer along a predefined path
  - Not rigid steering
  - Paths remain near, but characters may deviate
  - Path defined by a *spine* and a *radius*
    - Spine could be spline, connect line segments
  - Velocity-based prediction of character future position
    - Project onto nearest path position
    - Compare with radius to see if corrective steering required
    - Correct with Steer behaviour



© Reynolds 1999

Leader following



© Reynolds 1999

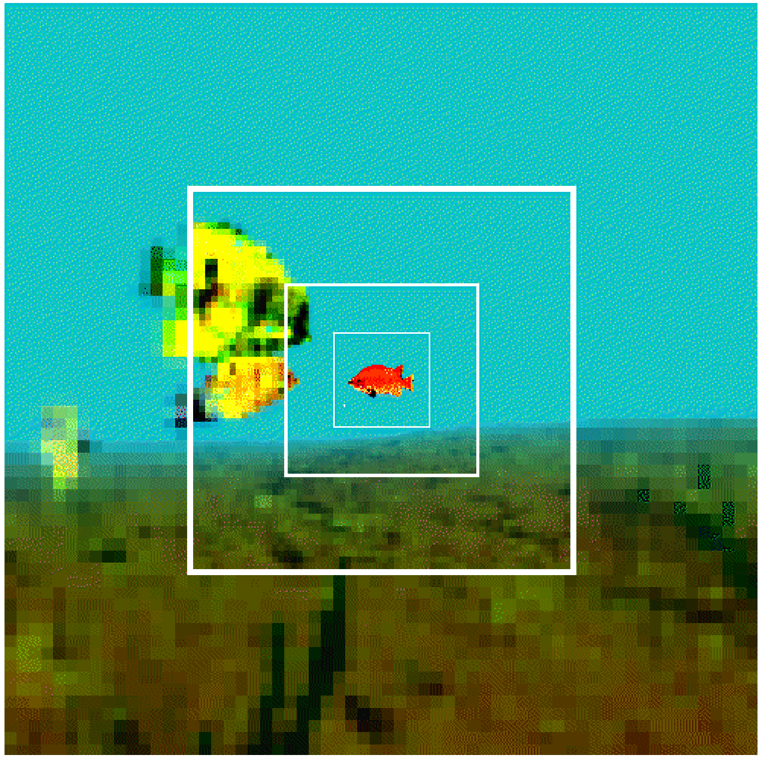
- One character designated as leader
  - Followers stay near but do not crowd
    - Arrival target set to be behind leader
  - Stay out of leaders way
    - Characters in front steer laterally before resuming arrive
  - Separation rule to avoid crowding

Artificial fishes

- Tu and Terzopoulos, SIGGRAPH 1994
- Perception, motor control, physically-based fish model

Artificial fishes

- Intentions generated
  - Based on habits, mental state and sensory perception
  - Chooses and executes a behaviour routine
  - This runs appropriate motor controllers
- Sensory perception
  - 300 degree F.O.V.
  - Selective attention mechanism and foveated vision (nice!)
- Physics-based fish model
  - Spring-mass system



ALIVE: Silas T. Dog

- Blumberg and Galyean, SIGGRAPH 1995
- Layered architecture with multiple levels of control
- Perception, behaviour control

ALIVE: Silas T. Dog

- Key contribution:
  - Multiple levels of control
    - Motivational: “you are hungry”
    - Task: “go to that tree”
    - Direct: “wag your tail”
  - Autonomy *and* directability
- Sensors, behaviours, motor system
  - Behaviour based on filtered sensory info and weighted goals and motivations
  - Synthetic vision for navigation
    - Influence from robotics

Improv

- Perlin and Goldberg, SIGGRAPH 1996
- Behaviour-based animation of (virtual) actors

Improv

- Animation engine
- Behaviour engine
  - Authors create *scripts* governing how actors communicate and make decisions
  - Determine appropriate animated actions to play at any one time
- Behaviour is non-deterministic, decisions fuzzy

```
{ choose from ("Fight" "Flee") based on "how
courageous" }

define DECISION−RULE: "how courageous"
{
factor { my "Courage"
equals its "Courage Level" to within .5 }
}
```

```
define SCRIPT "Listen To Joke"
{
{
choose from { entire set of "Stances" } based
on
"appropriate listening gestures"
choose from { entire set of "Gestures" } based
on
"appropriate listening gestures"
}{
continue for between 3 and 12 seconds }
{ repeat }
}
```

Cognitive Modelling

© Funge et al. 1999

- Funge et al., SIGGRAPH 1999
- Goes deeper into the A.I. side
  - Reasoning, knowledge representation and planning (gasp!)

Cognitive Modelling

- Adds:
  - Domain knowledge specification
    - Knowledge about the world and how it can change
  - Character control
    - Behaving in a certain way to achieve certain goals
  - Situation calculus and cognitive modelling language (CML)
    - **action** drop(*x*) **possible when** Holding(*x*);
    - **occurrence** drop(*x*) **results in** !Holding(*x*);
- Complimentary to behavioural animation
  - More sophisticated reasoning and representation
  - See first slide

Other Elements

- Also of importance to behavioural systems:
  - Adding behavioural hints to the V.E.
    - Support agent behaviour by tagging objects and zones
    - Describe what objects can be used for and how to use them
    - Differentiate between different types of zones in the environment
  - Path-finding
    - Compute a path between a current position and a destination, avoiding obstacles
    - Other constraints
      - E.g. Shortest paths, positions providing good cover from enemies

References

- Papers

- [Reynolds 1987] C. W. Reynolds. **Flocks, herds, and schools: A distributed behavioral model.** SIGGRAPH 1987.
- Reynolds 1999] C. W. Reynolds. **Steering behaviours for autonomous characters.** GDC 1999.
- [Tu and Terzopolous 1994] X. Tu and D. Terzopoulos, **Artificial fishes: Physics, locomotion, perception, behavior.** SIGGRAPH 1994.
- [Blumberg and Galyean 1995] B. Blumberg and T. Galyean, **Multi-level direction of autonomous creatures for real-time virtual environments.** SIGGRAPH 1995.
- [Perlin and Goldberg 1996] K. Perlin and A. Goldberg, **Improv: A system for scripting interactive actors in virtual worlds.** SIGGRAPH 1996.
- [Blumberg 1997] B. Blumberg, **Go with the flow: Synthetic vision for autonomous animated creatures.** Agents 1997.
- [Funge et al. 1999] J. Funge, X. Tu, D. Terzopoulos, **Cognitive modeling: Knowledge, reasoning and planning for Intelligent Characters.** SIGGRAPH 1999.
- [Farenc et al. 1999] N. Farenc, R. Boulic, D. Thalmann, **An informed environment dedicated to the simulation of virtual human in urban context**. Eurographics 1999.
- [Tecchia et al. 2002] F. Tecchia, C. Loscos, Y. Chrysanthou, **Visualizing crowds in real-time**, Computer Graphics Forum 21(4), 2002.
- [Leonard 2003] T. Leonard, **Building an AI Sensory System: Examining The Design of *Thief: The Dark Project*.** GDC 2003.

- Books

- [Funge 2004] J. Funge, **Artificial Intelligence for Computer Games: An Introduction**. A.K. Peters, July 2004.