Behavioral animation
Animation Category Recap

• **Procedural**
  – Artist specifies translation and rotation over time

• **Physically based**
  – Artist specifies forces acting on objects
  – Motion equations dictate movement

• **Interpolation**
  – Artist draws the scene at the beginning and end of a time interval
  – Intermediate positions are calculated

• **Behavioral**
  – Artist directs autonomous agents
Behavioral Animation

• Control the motion of one or more objects using virtual actors

• Goal: realistic or believable motion so that the object appears to be autonomous

• Matt Lewis’ page on BA
  http://accad.osu.edu/~mlewis/Class/behavior.html
Behavioral animation

• Character or object motion based on:
  – Knowledge of the environment
  – Aggregate behavior
  – Primitive behavior
  – Intelligent behavior
  – Crowd management
**Actor motion:** Moving forward/backward

Position \((x, y)\)

Goal \((x_g, y_g)\)

\[ V = \text{Goal} - \text{Position} \]

\[ \text{Next Position} = \text{Position} + V \times dt \]
Actor motion: Rotation

Orientation \((ox, oy)\) = transform.forward

\[ V_{target} = \text{Goal} - \text{Position} \]

Determine which rotation \((\pm \theta)\) orients the actor closer to the goal using dot product.

In Unity, you can use RotateTowards().
class OrientedAgent2D {

    // Data
    Vector3 position;
    GameObject model; // Use this for geometry and orientation

    // Methods
    Update(float deltaTime);
    TurnLeft();
    TurnRight();
    MoveForward();
    MoveBackward();
};
Knowing the environment

- Vision and other senses
  - Information available now

- Memory
  - Information stored from the past
Actor Vision

• General vision
  – What can the actor see?

• Targeted vision
  – Can the actor see object X?

• Computation vs. accuracy
  – How much of an object needs to be seen to be identified?
  – Do we need to model visual perception?
Vision Model 1: Omniscience

Everything in the scene is known.
Vision Model 1: Field of View

Those in the field of view are visible.
Use dot product, to get the cosine of the angle between O and V. Visible, when the angle between O and V is less then or equal to $\theta/2$. 

**Vision Model 1: Field of View**

Angle of cone = $\theta$
Vector3 agentPosition, orientation, objectPosition;
float visionLimit;

Vector3 agentToVertex = objectPosition – agentPosition;
agentToVertex.Normalize();
if(Vector3.Dot(agentToVertex,orientation) > visionLimit)
    // agent can see object
Vision Model 1: Occluded Vision

Ray casting with collision detection
Sample the environment
For each object, Target-testing vision

Binary:
Can the actor see X?
Cast a ray

Detailed:
How well can the actor see X?
Use multiple rays
Target-testing vision (to know what is visible)

Sample the vision cone
Cast multiple rays
Predator Prey vision anatomy
Prey-Predator (vision)
Motion – force based

Apply a force on the predator towards the prey when in view

Force = c * \( \text{pos}_{\text{prey}} - \text{pos}_{\text{pred}} \)
Motion – kinematic based

Determine the closest prey in view then turn towards it and increase forward velocity
The Boids Model

• Invented by Craig Reynolds in 1986
• Artificial life system: bird
• The complexity of the system comes from a set of simple rules applied on the agents (birds):
  • Separation: avoid crowding
  • Alignment: follow other birds’ motions nearby
  • Cohesion: Stay together with other birds
• [https://www.youtube.com/watch?v=rN8DzIgLgMt3M](https://www.youtube.com/watch?v=rN8DzIgLgMt3M)
• [https://www.youtube.com/watch?v=lKh_IzRb9Ro](https://www.youtube.com/watch?v=lKh_IzRb9Ro)
Other flocking issues

• Global control
  – script flock leader
  – global migratory urge

• Negotiating the motion
  – Separation, alignment, and cohesion may compete/contradict

• Collision avoidance
  – Once an object is in sight, start steering to avoid

• Splitting and rejoining (difficult)
  – Collision avoidance too strong – flock may never rejoin after split
  – Flock membership too strong – flock does not split and formation changes

• Modeling flight
Motion Planning

Forces
Or
"Reasoning"
(e.g. rule-based)
Navigating obstacles

Penalty Force
(Collision)

Plan ahead
(Intelligence)

Attempt at parallel movement

Attempt at finding a passageway
Modeling flight – common in flocking
Modeling flight

Geometric flight – dynamic, incremental rigid transformation of an object moving along a tangent to a three dimensional curve
Modeling Flight – Lift

Air passing above wing most travel longer than air below it. Less pressure above wing = lift
Modeling Flight – turn by rolling

Diagram showing the forces involved in modeling flight, with lift and gravity forces depicted for a plane in a stationary position and in a turn.
Emergent behavior

• Complex systems and patterns arising out of simple rules

• Aints – AI ants
  http://www.youtube.com/watch?v=rsqsZCH36Hk