Group Behaviors

- Motivation, Group/Crowd
- Procedural Animation – Boids Model
  - Particle Systems
  - Physical Dynamics
- Example – *AI implant* system
- Example – Real Time Strategy game
- Other examples
Sources

- Game Developers Conference, and Gamasutra articles
- Bouvier et al., Heigeas et al., Particle Systems
- Hodgins and Brogan, Physical Dynamics
- Biographics Technologies, *AI implant*
- Pottinger, papers about crowds in “Age of Empires”
Motivation

- Automate motion of individuals in large group
Group/Crowd

- Large group of individuals in same physical environment
- Common goal
- May act in different way than when they are alone
- Variety in individuals (not clones of each other), but still relates to group
- Games: Fast; tradeoff for realistic motion
Possible group behavior

- Seek goal position
- Flocking
- Dispersion/Aggregation
- Following leaders
- Collision avoidance
Procedural animation

- initial state of objects in system, rules for updating, states for subsequent step based on conditions in previous step
Procedural animation

- Useful for generating lifelike motion with relatively little input
- But lack of explicit control over look of individual frames
Boids model – vehicle model

- Mass, position, velocity, max force, max speed, orientation
Boids model – 3 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 local flockmates
Boids model

- 3 rules are necessary, and also sufficient based on experimental results – Video
Boids Model

- Complex behavior from simple rules
- Rules applied to individuals → lead to global behavior (emergence)

- Demo – Flocking
  http://www.red3d.com/cwr/steer/
Obstacle avoidance

- Additional rule
- Demo – Obstacle Avoidance, Unaligned obstacle avoidance
“With the proper flocking routines in place, enemies hunt in packs, cover each other, and generally act like they place some value on both their own lives and the lives of their comrades.”  
[Gamasutra article]
Path Following

- Predict future position, project onto path, steer towards path if necessary
- Approach and then follow path
- Demo – path following, crowd path following
Leader Following

- Followers – stay near leader, but do not crowd leader, and move out of way
- Requires “Arrival” behavior – steers toward target, and slowing down along way
Flow field following

- Predefined flow field – maps to flow vector
- Steers towards flow vector
- Can be specified by artist (non-programmer), by painting general direction
Queuing through a narrow doorway

- Combination of seeking and decelerating
- Avoiding wall and each other

- Demo - Queuing
Combine these rules

- Combine in different ways to get more complex behaviors

- Contradictory rules – need to prioritize them
Real-time interaction

- Environment with user-controlled car and birds (following 3 rules)
- PS2 – with 280 birds, 60 Hz
Finite-State Machine for birds

- Default behavior is *walk*
- Transition to *fly* if
  - Car comes close
  - Nearby birds *fly*
  - Loud noise nearby
- Stay at *fly* if car or loud noise keeps "annoying" it
Birds – annoyance value

- “Annoyed” first time – fly short distance away, then land again
- Annoyance value goes up every time, but slowly decays
- Together with flocking rules and obstacle avoidance
Bird motion

- Generated from small set of pre-animated cycles, and parameters like character’s position and walk/fly state

- Walking, Flying, Gliding, and WingFold
Optimization 1

- Simulation is 60 Hz, but control program runs at 10 Hz
- Apply rules once every 6 simulation steps
- Between these steps, use same steering velocity
- For variety, phase of these cycles randomized for each bird
Optimization 2

- Have to find the neighbors – $O(n^2)$
- Max number of bird $k$ in each square – $O(nk)$, or $O(n)$
Video

realtimebirds.avi
Particle Systems

- Multiple particles in each frame or time step
- Rules for updating them based on particles (velocities, diffusion properties, lifespan) and forces (gravity, wind)
- Often used for clouds, water, fire
Particle System ➔ Crowd

- Human beings modeled as interacting set of particles
- Forces of attraction/repulsion between particles and/or objects in scene
- Masses of particles can be those of individuals
- Attraction forces of certain groups of people can be higher/lower
Example crowd

![Diagram of a crowd](image)
Physical Dynamics

- Assign properties to objects (mass, resilience) and environment (gravity, density of air)
- Initial state
- Update rules
Physical Dynamics

- For each creature:
  - Find visible creatures
  - Calculate desired position and velocity (spring/damper)
Physical Dynamics

- Equations of motion for rigid model of one-legged creature
- Given velocity $\rightarrow$ control algorithms generate motion for different speeds and flight durations
Example

- Avoid obstacle by aiming for point to the side of obstacle, and incorporate this into “desired position”

- Videos – 105HopFrontView/SideView
AI implant system

- Commercial system for real-time crowd simulation of people and vehicles
- Used for games in PS2 and Xbox

Screenshots of the AI development environment.
Navigation

- Navigation meshes

Figure 3. The navMesh corresponds to all navigable space (e.g., sidewalks and roads around buildings).
Reactive decisions

- Gather environment data during runtime and apply data to set of pre-defined rules (repath if something unexpected before happens)

- Animation system to control motion of humans/vehicles
Crowd creation

- A few base templates of characters from which large numbers are instantiated
- Differentiated by parameters (animation control, reactive rules, mesh)
- Tool for artists to “paint” parameters or instantiation of many characters
Result scene

- Follow waypoints, can sense other characters, can slow down

*Ambient milling behaviour: pedestrians staying on the sidewalk and crossing at the cross walk*

*Panic behaviour: running away from traumatic event (orange ball).*
Videos

- PanicDemo640_480.mov
- raceTrack.mp4
Real-Time Strategy Games

- Groups of characters
Groups and Formations

- **Groups**
  - List of units
  - Max/avg speed
  - Centroid or leader

- **Formation**
  - Orientation (front)
  - Each unit has unique position and orientation
Group movement

- Units should move with similar speed and time

- Units should move in similar path (leader finds path, and others can follow that path)
Formation

- Ways to get into formation
  - Minimize distance travelled
  - Collisions
  - Travel time

- Tradeoff between computation time and realistic result
Formation Management

- Scaling positions
Formation Management

- Splitting up and re-forming

- When to regenerate path instead
Other examples – movies

- Many examples
Other examples – “Rip-Off”

- First game demonstrating flocking
Other examples – “Rip-Off”

- “arrival” – enemy go towards target and slow down
- “separation” – within radius of another rank, an enemy tank will steer away
- If within radius of player tank, shoot at player