Computer Animation
Groups of Objects SS 15

Prof. Dr. Charles A. Wüthrich,
Fakultät Medien, Medieninformatik
Bauhaus-Universität Weimar
caw AT medien.uni-weimar.de

Particle systems

- A *particle system* is a large collection of individual elements which taken together represent a conglomerate object
- The „global“ behaviour of the particles is called *emergent behaviour*
- This can be used both for particle systems (which usually have more invividua) and for *flocking*
- Flock members have a more sophisticated behaviour than a simple element of particle system
- While particle systems behave according to physics, flocking particles add some intelligence to the behaviour of the individuals
- The more intelligence is added, the more the element moves in a more interesting way, and the more it shows *autonomous behaviour*

Particle systems

- In a particle system, due to the no of its elements, simplified assumptions are made
- Typical assumptions are
  - Particles do not collide among themselves
  - Particles do not cast indiv. shadows, but the aggregate may do
  - Particles only cast shadows on the rest of the environment, not among themselves
  - Particles do not reflect light, each is modeled as a point light source
- Often particles are modeled as having a finite life span
- To avoid dull behaviour, often randomness is added
- When a particle system is computed, the following steps are taken:
  - Generate new particles born this frame
  - Initialize attributes of new particle
  - Remove dying particles
  - Animate active particles
  - Render them

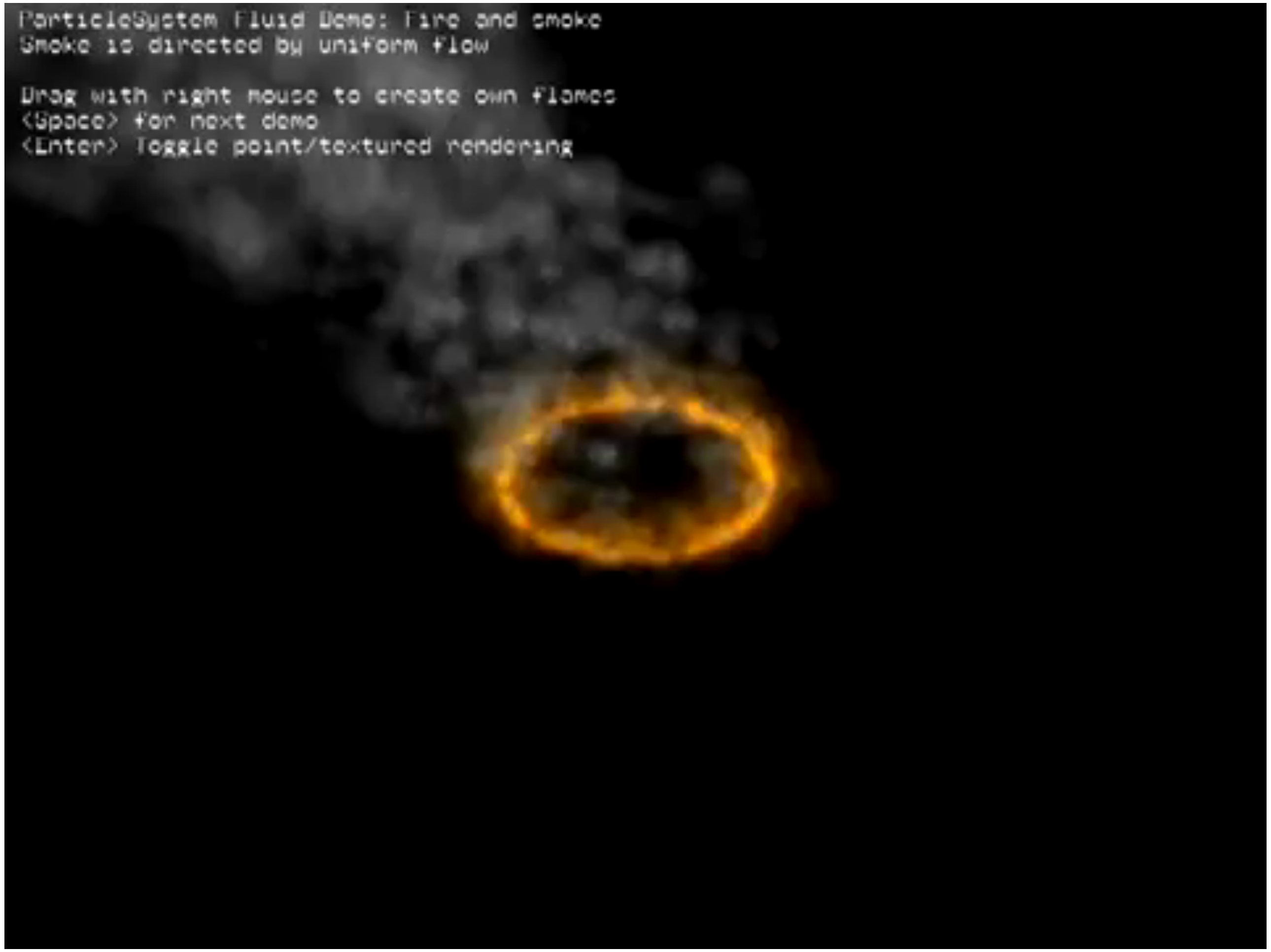
Particle generation

- Particles are usually generated according to a stochastic process
  - At each frame, a random number r_P of particles is generated
  - Generation has a user specified distribution centered at the desired number of particles per frame
  - r_P=ave+Rand(seed) · range
    where ave is the desired average and range is the desired variation range
- Sometimes it may be convenient to have this random function as a function of time, i.e. to make the number of desired particles increase in time
- If the particles are used to model a fuzzy object, then the area of the screen covered by the object A_s is used to control the number of particles
  r_P=ave+Rand(seed) · range · A_s

Particle attributes

- Attributes of the particles are typically
  - Position
  - Velocity
  - Shape parameters
  - Color
  - Transparency
  - Lifetime
- At each frame, the lifetime of each particle is decremented by one until it reaches zero
- During lifetime, particles are animated (position, velocity, shape, color, transparency)
- At each frame, forces on the particles are computed
- These result in an acceleration, which determines a velocity
- Also other attributes may be a function of time
- Rendering is often done modeling them as a point light source adding color to the pixel
- This to avoid particles to contribute to lighting computations

Particle examples



Courtesy S. Eriksson, University of AAlto © 2012

Flocks

- Here the number of members is small
- But each member has some intelligence and simple physics (avoid collision, gravity, drag)
- Aggregate behavior emerges from the members (emergent behavior)
- Each member is called a boid

- Two forces govern flock behavior:
  - collision avoidance: both with other boids and with obstacles
  - Motion has some random parameter to keep it from looking regular
  - flock centering: the boid tries to be a flock member
  - Flock centering keeps together the flock but does not have to be absolute, otherwise flocks cannot split around objects

Flocks: local behavior

- Controlling locally the behavior is the aim
- Three processes may be modeled:
  - Physics: similar to particle with gravity, collision detection and response
  - Perception of the environment: each boid views its direct neighbors and obstacles directly in front
  - Reasoning and reaction to determine the behavior
  - Additionally velocity matching is added (each boid tryies to match the speed of its neighbours)
- Global control is either applied to all boids or to a group leader
  - In this case the boids follow the leader
- The leader role can be rotated among boids in time
- Usually all this is implemented as three controllers which are priorized in the following order: collision avoidance, velocity matching and flock centering

Flock complexity

- The major problem with flocks is the fact that processing complexity is n^2.
- Even if interactions are allowed only with k nearest neighbors, those have to be found
- One way to find efficiently is to perform a 3d bucket sort and then check adjacent buckets for neighbors
- Of course, efficiency depends on the bucket size:
  - The more buckets, the less boids per bucket
- Another way of doing it is through message passing, where each boid informs the flock of its whereabouts

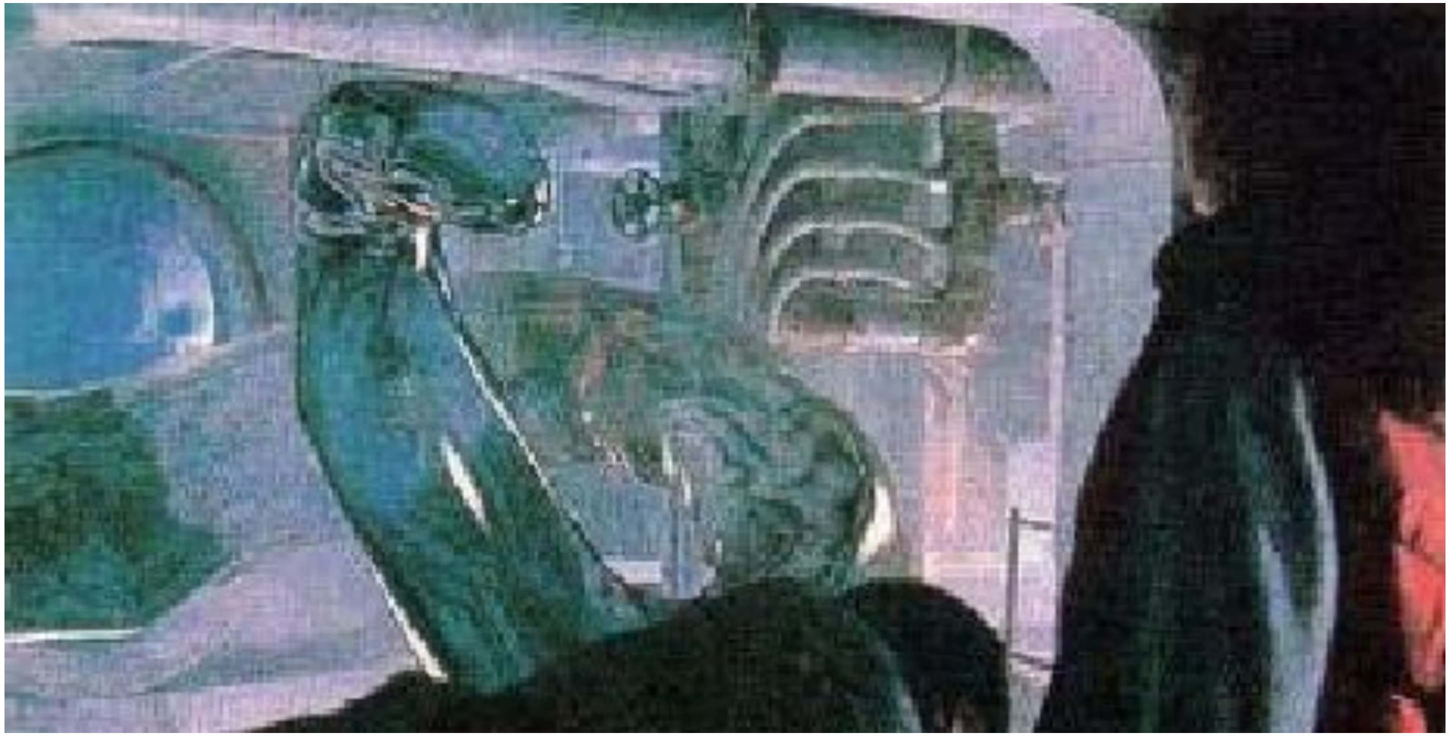
Collision avoidance

- There are several ways to avoid collisions
  - The simplest way is adding a repelling force around an object
  - However, this looks weird as the boid keeps attempting to aim at the repelling surface and contantly gets blown away
  - Another method computes if the boid trajectory hits the surface and starts a steering behavior
  - Quite complicated is the simulation of a splitting flock around an obstacle, since a balance has to be found between collision avoidance and flock cohesion

Autonomous behaviour (crowds)

- Modeling intelligent behaviour is a complex task
- Autonomous behaviour models an object knowing about its environment
- This can become as complicated as one wants
- Usually applied to animals, but also to people, cars on a road, planes, or soldiers in a battle
- Knowledge of the environment is provided by providing access to the environment geometry
- Subjective vision can be achieved by rendering the environment from the point of view of the object
- Internal state is modeled by intentions = the urge to satisfy a need
- High level goals can be decomposed in single low level tasks (levels of behaviour)
- Internal state and knowledge of the environment are input to the reasoning unit, which produces a strategy (=what needs to be done)
- Such strategy is turned into a sequence of actions by the planner, and actions are turned into movement
- If intentions are competing, they must be prioritized

End



+++ Ende - The end - Finis - Fin - Fine +++ Ende - The end - Finis - Fin - Fine +++