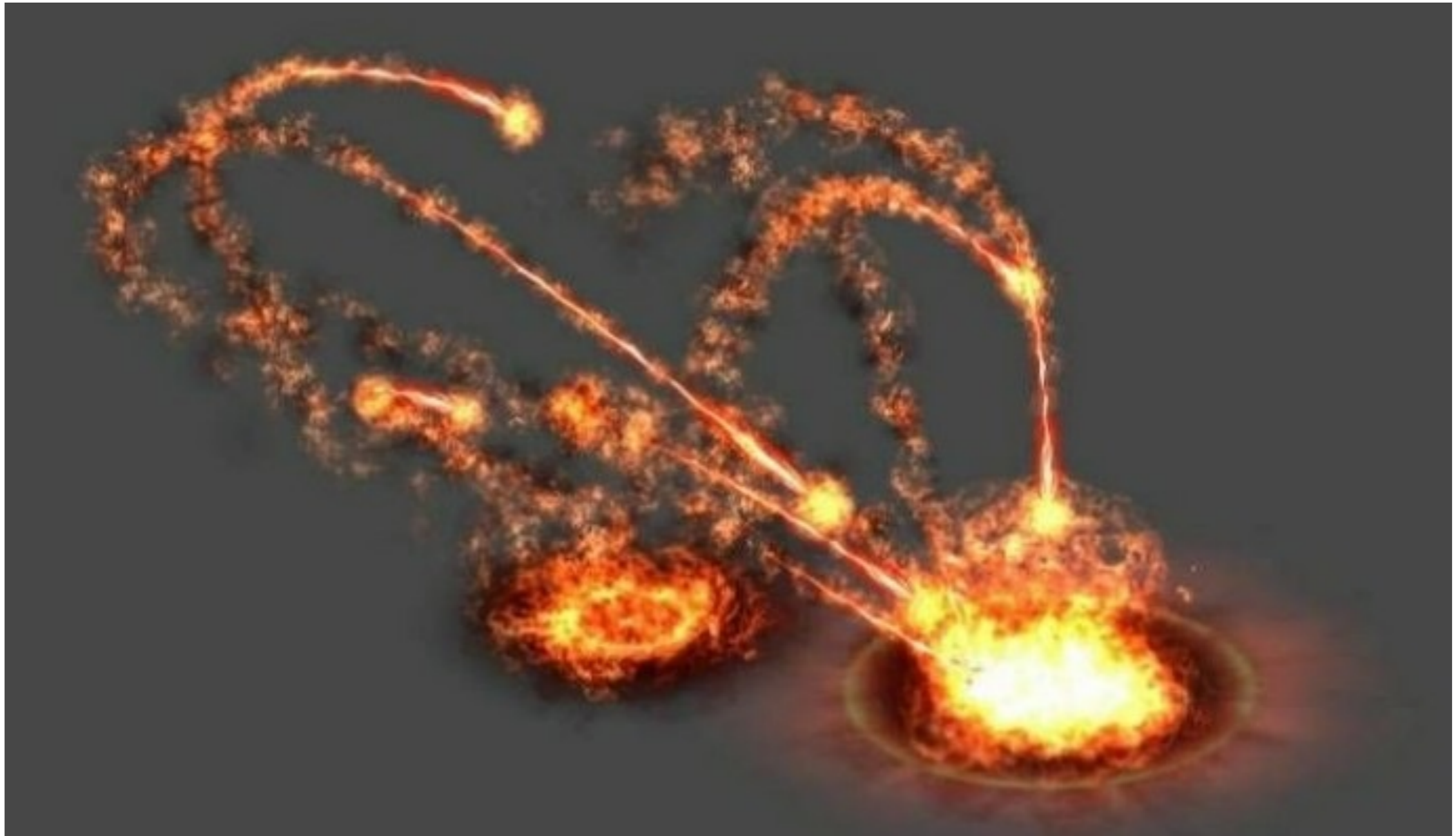


CS354R

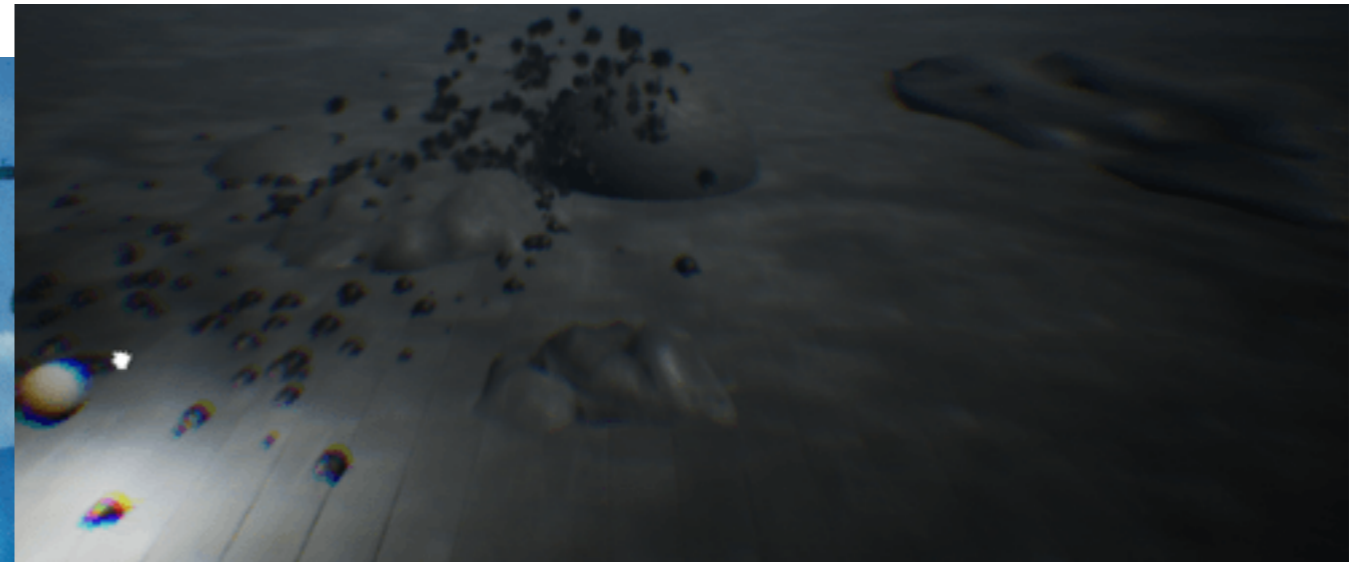
DR SARAH ABRAHAM

PARTICLES AND FLOCKING BEHAVIOR

PARTICLE EFFECTS



PARTICLE EFFECTS IN ACTION



GENERAL PARTICLE SYSTEMS

- ▶ Objects are considered point masses with orientation
- ▶ Simple rules control how the particles move
- ▶ Particles can be controlled/rendered to simulate different things:
 - ▶ Fireworks
 - ▶ Waterfalls, spray, foam
 - ▶ Explosions (smoke, flame, chunks of debris)
 - ▶ Clouds
 - ▶ Crowds, herds
- ▶ Widely used in movies as well as games

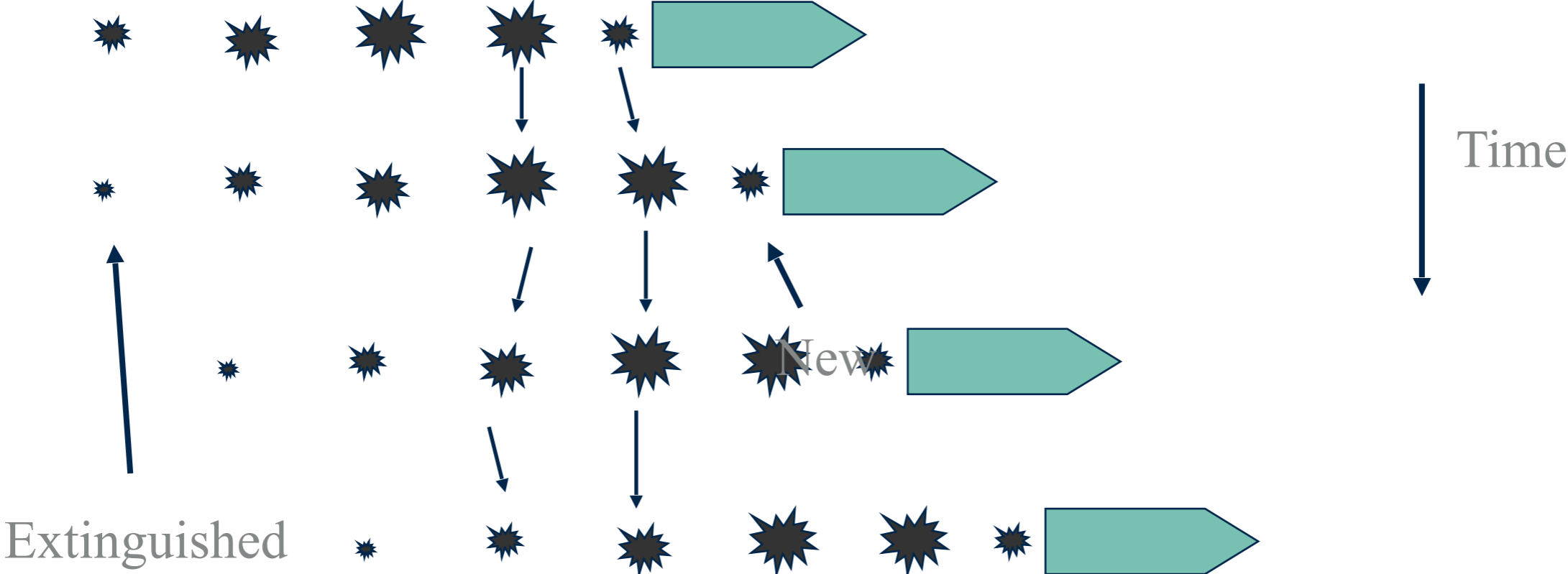
PARTICLE SYSTEM STEP BY STEP

1. Inject new particles into the system and assign individual attributes
 - ▶ There may be one or more sources
 - ▶ Particles might be generated at random (clouds), in a constant stream (waterfall), or via a script (fireworks)
2. Remove any particles that have exceeded their lifetime
 - ▶ May have a fixed lifetime, or die on some condition
3. Move all the current particles according to their script
 - ▶ Script typically involves neighboring particles and environment
4. Render all the current particles
 - ▶ Many options for rendering (shaders, textures etc)

EXAMPLE: ROCKET SMOKE TRAILS

- ▶ Particles are spawned at a constant rate
- ▶ They have zero initial velocity, or maybe a small velocity going away from the rocket
- ▶ Rules:
 - ▶ Particles can rise or fall (drift with the wind)
 - ▶ Attach a density that grows quickly then falls over time
- ▶ Extinguish when density becomes small
- ▶ Render with billboard facing the viewer, scaled according to the density of the puff

SMOKE TRAILS



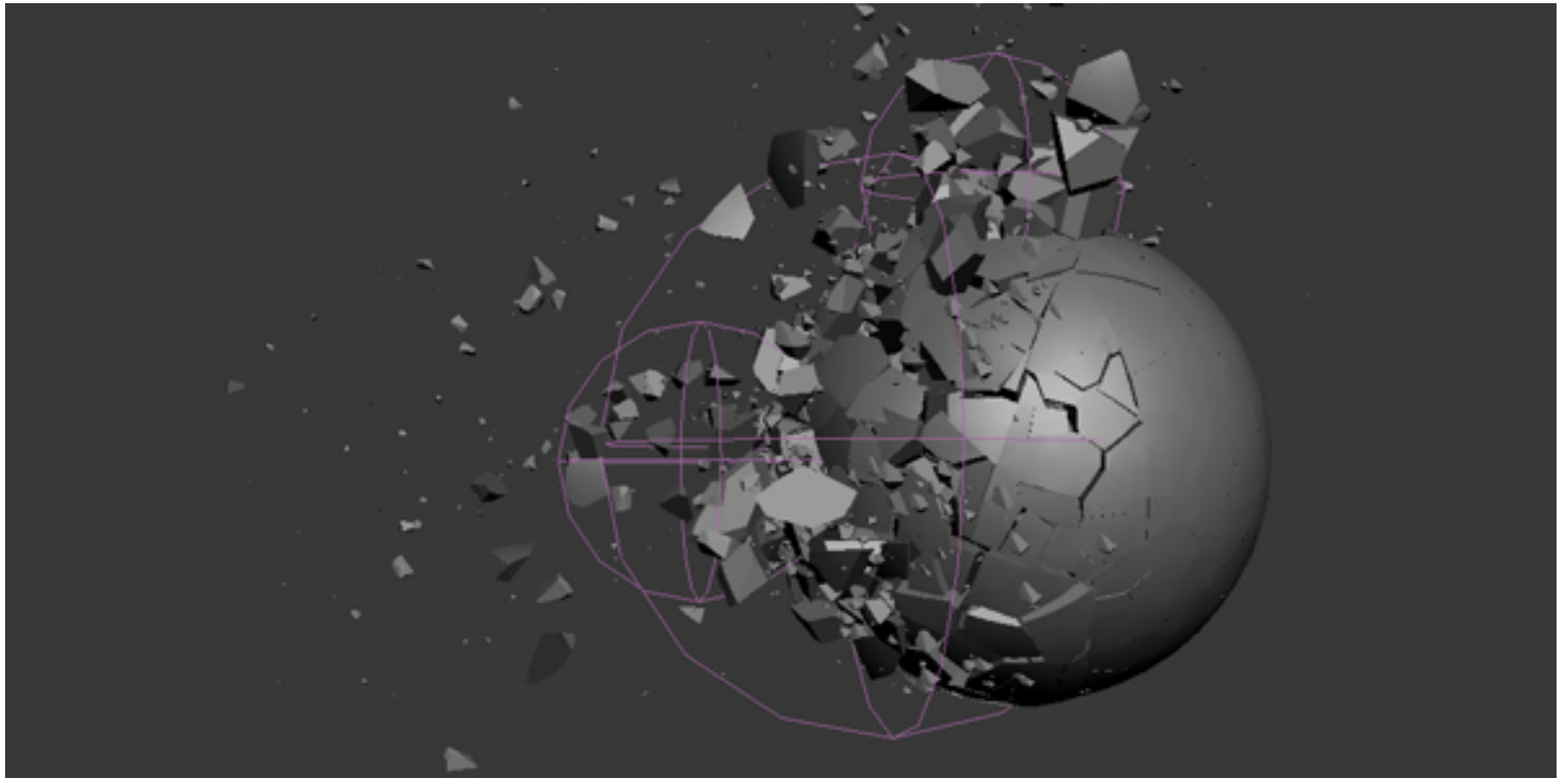
PARTICLE EMITTERS

- ▶ Provide sources for spawning particles
- ▶ Emitters can specify:
 - ▶ Rate of particle emission
 - ▶ Shape of particle emission
 - ▶ Direction of particle emission
- ▶ Can also specify parameterization of individual particle properties

EXAMPLE: OBJECT FRACTURING

- ▶ System starts when the target is hit
- ▶ Target is broken into pieces and a particle is assigned to each piece
- ▶ Each particle gets an initial velocity away from the center of the explosion
- ▶ Particle rules:
 - ▶ Move ballistically unless there is a collision
 - ▶ Computer rigid body rotation or generate random rotation
 - ▶ Resolve collisions by reflecting the velocity about the contact normal
- ▶ Rendering draws the appropriate piece of target at the particle's location

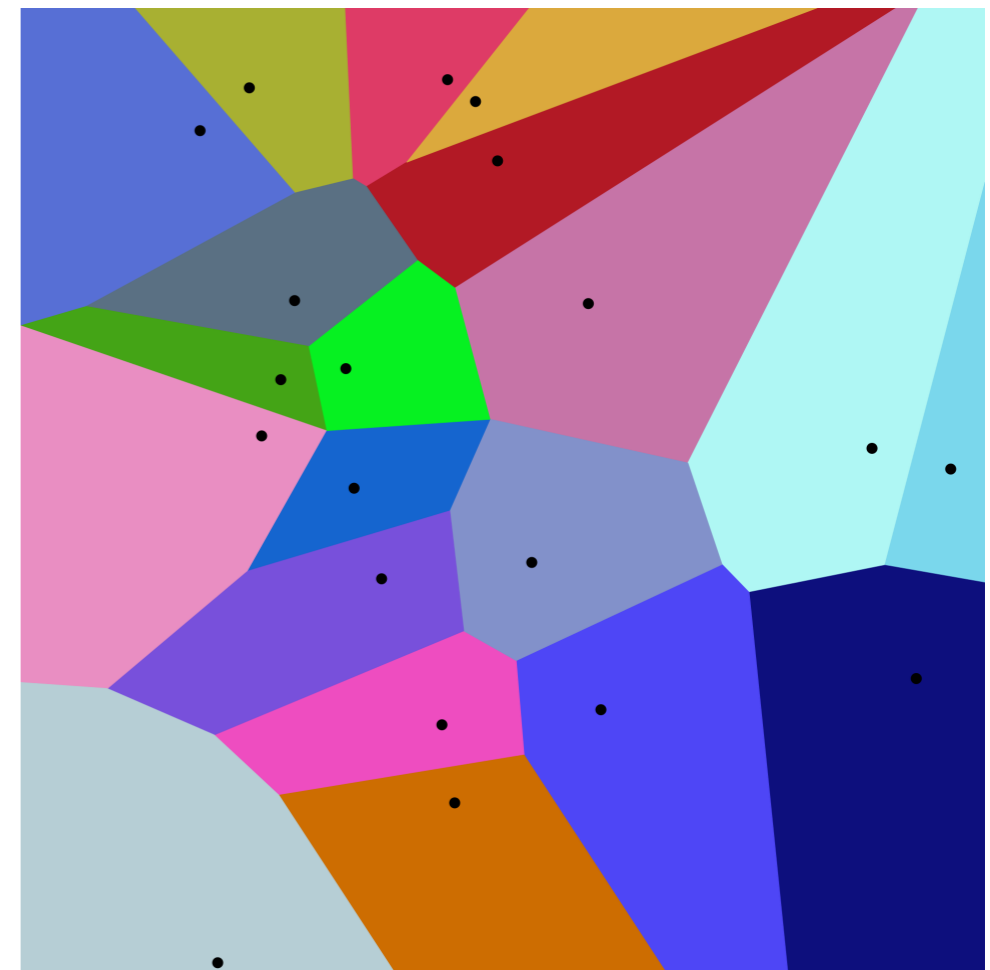
OBJECT FRACTURING



Laurent Renaud (<http://cgcookie.com/max/2009/08/18/creating-an-exploding-planet/>)

HOW TO FRACTURE?

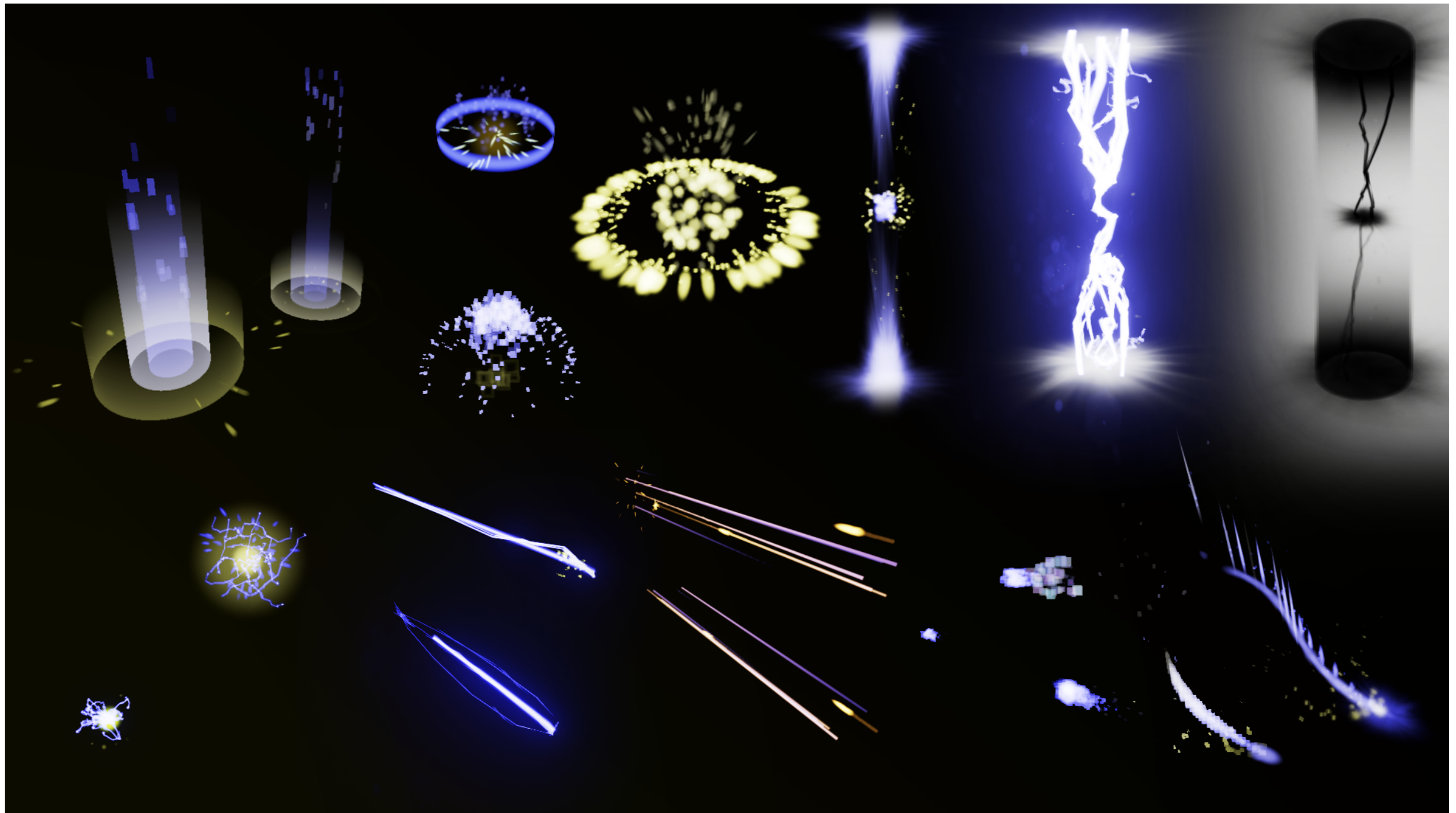
- ▶ Voronoi commonly used
 - ▶ Can be done in realtime or using pre-processing
- ▶ Voronoi partitions created by seeding the surface of a plane (or a 3D space) with points
 - ▶ Every point along the surface (or in the 3D space) associated with the closest seeded point
- ▶ Voronoi useful in procedural techniques more generally as well!



VISUALIZING PARTICLES

- ▶ Particles can be visualized in a number of ways
 - ▶ Billboarding (applying a texture to an individual particle)
 - ▶ Point shaders (applying a shader to an individual particle)
 - ▶ Mesh shaders (applying a shader to a mesh based on the particle positions)
 - ▶ Post processing (applying a post processing effect in screen space to represent particles)

VISUALIZING PARTICLES



PARTICLES IN GAMES

- ▶ They're everywhere!
 - ▶ https://www.youtube.com/watch?v=6_NsaYtooQA

PARTICLE MANAGEMENT

- ▶ Particle systems should include some sort of pool for resource management
 - ▶ Particle lifespans relatively short
 - ▶ Particles should be reused as much as possible
 - ▶ Good caching helps with particle system efficiency
- ▶ Particle systems well-suited to parallelization
 - ▶ Can be implemented in conjunction with multi-threading/
multi-core/GPUs

FLOCKING BEHAVIOR

- ▶ Particles can also model flocks, swarms, crowds etc

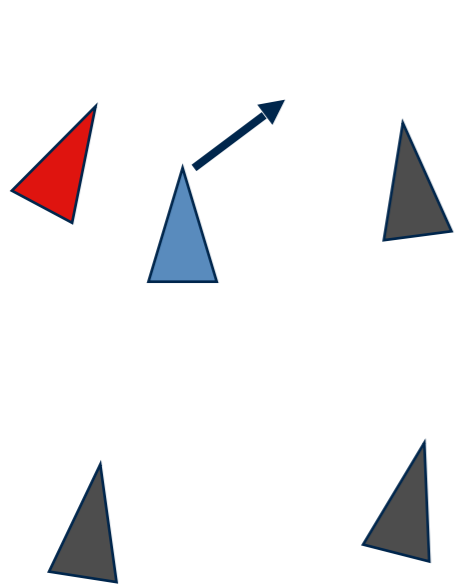


(<https://portraitsofwildflowers.wordpress.com/2011/12/10/grackles-revisited/>)

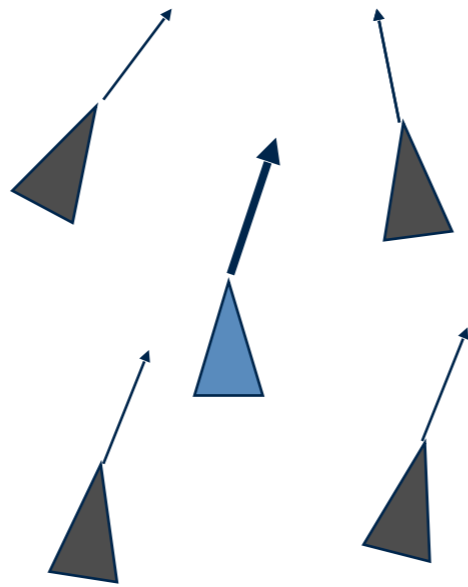
FLOCKING MODELS (REYNOLDS '87)

- ▶ Potential fields are most often used in avoiding collisions between the members of a group
 - ▶ Member pushes on its neighbors to keep from colliding
- ▶ Additional rules for groups can be defined (result is flocking, herding, schooling, etc)
- ▶ Each rule contributes a desired direction, which are combined in some way to come up with the acceleration
- ▶ The aim is to obtain emergent behavior:
 - ▶ Define simple rules on individuals that interact to give interesting global behavior
 - ▶ e.g individual birds form a flock, but we never explicitly specify a leader, or shape, or speed

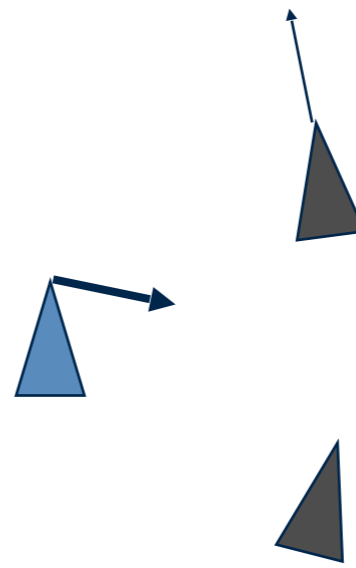
FLOCKING RULES ILLUSTRATED



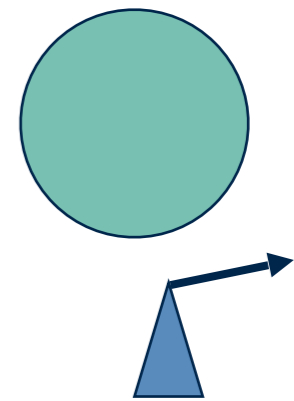
Separation: fly away away from neighbors that are “too close”



Alignment: steer toward average velocity



Cohesion: steer toward average position



Avoidance: steer away from obstacles

FLOCKING RULES EXPLANATION

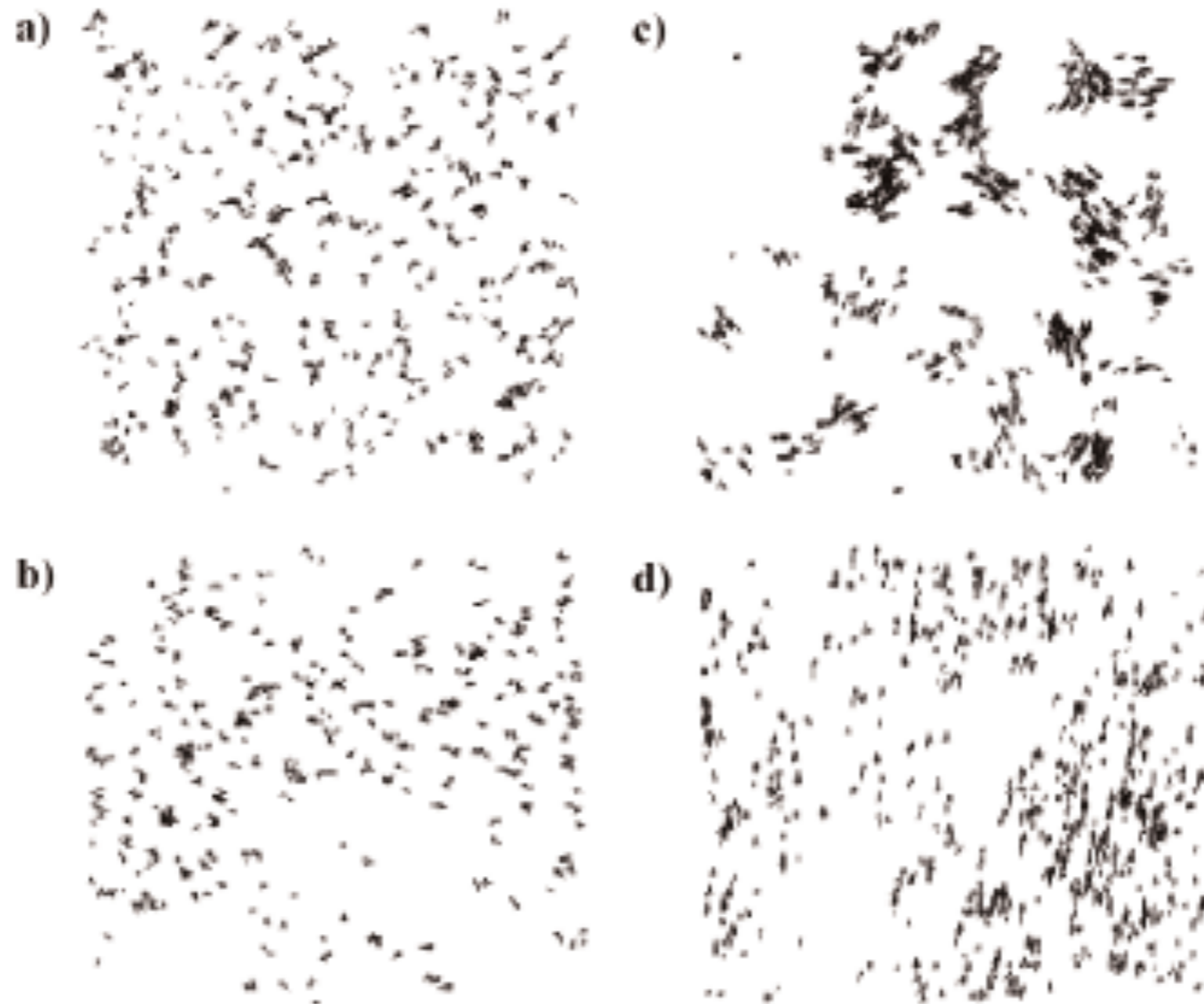
- ▶ Separation: Try to avoid running into local flock-mates
 - ▶ Works just like potential fields
 - ▶ Normally, use a perception volume to limit visible flock-mates
- ▶ Alignment: Try to fly in same direction as local flock-mates
 - ▶ Gets everyone flying in the same direction
- ▶ Cohesion: Try to move toward the average position of local flock-mates
 - ▶ Spaces everyone out evenly, and keep boundary members toward the group
- ▶ Avoidance: Try to avoid obstacles
 - ▶ Just like potential fields!

BALANCING FLOCKING

- ▶ Consider commands as accelerations
- ▶ Give a weight to each desire
 - ▶ e.g. high for avoidance, low for cohesion
- ▶ Option 1: Apply in order of highest weight, until a max acceleration is reached
 - ▶ Ensures that high priority things happen
- ▶ Option 2: Take weighted sum and truncate acceleration
 - ▶ Makes sure some part of everything happens

FLOCKING DEMO

▶ <https://www.youtube.com/watch?v=rN8DzlgMt3M>



Craig Reynolds (Boids '87)

FLOCKING EVALUATION

- ▶ Advantages:
 - ▶ Complex behavior from simple rules
 - ▶ Many types of behavior can be expressed with different rules and parameters
- ▶ Disadvantages:
 - ▶ Can be difficult to set parameters to achieve desired result
 - ▶ All the problems of potential fields regarding strength of forces

BEYOND BOIDS

- ▶ Flocking behaviors vary based on the agents being simulated
- ▶ Adjusting the rules (or evaluation of rules) allows for greater variety in simulation



ANOTHER EXAMPLE...



Mythic Ocean (<https://www.youtube.com/watch?v=dHriVqfqDMI>)

FURTHER READING

- ▶ Flocks, Herds, and Schools: a Distributed Behavioral Model (<http://www.cs.toronto.edu/~dt/siggraph97-course/cwr87/>)
- ▶ Particle Systems (<https://natureofcode.com/book/chapter-4-particle-systems/>)