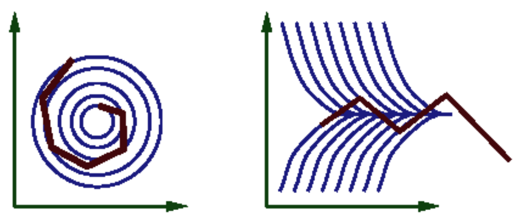
Solving the Equations of Motion

- Forward (explicit) Euler integration
  - $x(t+\Delta t) \leftarrow x(t) + \Delta t\, v(t)$
  - $v(t+\Delta t) \leftarrow v(t) + \Delta t\, f(x(t), v(t), t) / m$
- Problem:
  - Accuracy decreases as Δt gets bigger



Hodgins

Solving the Equations of Motion

Explicit Euler step

$$y_{n+1} = y_n + hf(t_n, y_n)$$

Implicit Euler step

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1})$$

Why are these methods called like this?

- **Explicit**: all quantities are known (*given explicitly*)
- **Implicit**: y_{n+1} is unknown (*given implicitly*)

$\rightarrow$ solve (nonlinear) equation(s)!

Solving the Equations of Motion

Explicit Euler step

$$y_{n+1} = y_n + hf(t_n, y_n)$$

Implicit Euler step

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1})$$

Why are these methods called like this?

- **Explicit**: all quantities are known (*given explicitly*)
- **Implicit**: y_{n+1} is unknown (*given implicitly*)
- Stability conditions for Euler
  - Explicit $y_n = (1+h\lambda)^n y_0 \quad < \infty \quad \Leftrightarrow \quad |1+h\lambda| \quad < 1$
  - Implicit $y_n = (1-h\lambda)^{-n} y_0 \quad < \infty \quad \Leftrightarrow \quad |1-h\lambda|^{-1} < 1$

$\Rightarrow$ **Implicit Euler is stable for all $h > 0$!**

More Complex Particle Systems

Particle System Forces

- Gravity
  - Force due to gravitational pull (of earth)
  - g = acceleration due to gravity (m/s^2)

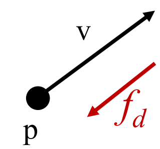
$$f_g = mg$$

g = (0, -9.80665, 0)

Particle System Forces

- Drag
  - Force due to resistance of medium
  - k_{drag} = drag coefficient (kg/s)

$$f_d = -k_{drag} v$$



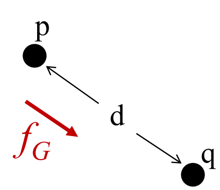
  - Air resistance sometimes taken as proportional to v^2

Particle System Forces

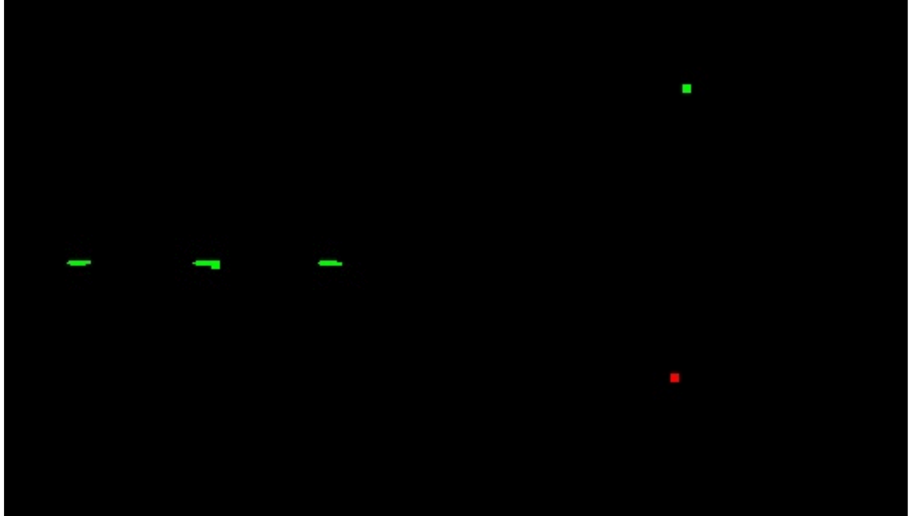
- Gravitational pull of other particles
  - Newton's universal law of gravitation

$$f_G = G \frac{m_1 \cdot m_2}{d^2}$$

$$G = 6.67428 \text{ x } 10^{-11} \text{ N m}^2 \text{ kg}^{-2}$$



Examples: Fountain, Gravity



http://www.cs.princeton.edu/courses/archive/spring14/cos426/assignment4/examples/examples.html

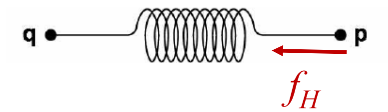
Spring-Mass Models

Particle System Forces

- Springs
  - Hooke's law

$$f_H(p) = k_s\left(d(p,q) - s\right)\ D$$

$$D = (q-p)/\|q-p\|$$
$$d(p,q) = \|q-p\|$$
$$s = \text{resting length}$$
$$k_s = \text{spring coefficient}$$



Binary, *n*-ary Forces

Much more interesting behaviors to be had from particles that interact

Simplest: binary forces, e.g. springs

$$\vec{f}_i(\vec{x}_i, \vec{x}_j) = -k_s(\|\vec{x}_i - \vec{x}_j\| - r_{ij})\frac{\vec{x}_i - \vec{x}_j}{\|\vec{x}_i - \vec{x}_j\|}$$

Nice example project with mass-spring systems:

- https://vimeo.com/73188339

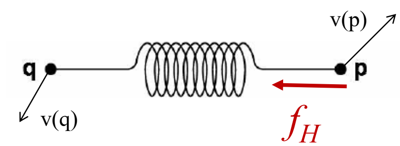
More sophisticated models for deformable things use forces relating 3 or more particles

Particle System Forces

- Springs
  - Hooke's law with damping

$$f_H(p) = \left[k_s\left(d(p,q) - s\right) + k_d\left(v(q) - v(p)\right)\cdot D\right]\ D$$

$$D = (q-p)/\|q-p\|$$
$$d(p,q) = \|q-p\|$$
$$s = \text{resting length}$$
$$k_s = \text{spring coefficient}$$
$$k_d = \text{damping coefficient}$$
$$v(p) = \text{velocity of p}$$
$$v(q) = \text{velocity of q}$$



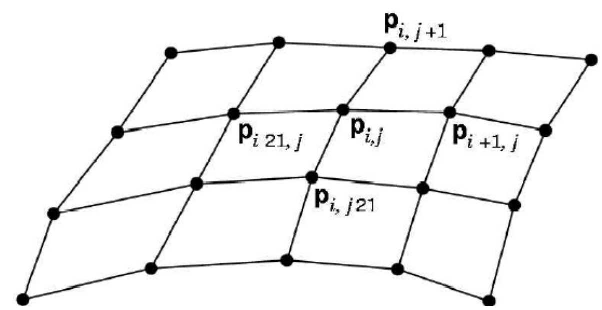
$$k_d \sim 2\sqrt{mk_s}$$

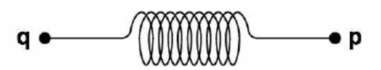
Examples: Rope

http://www.cs.princeton.edu/courses/archive/spring14/cos426/assignment4/examples/examples.html

Particle System Forces

- Spring-mass mesh





Hodgins

Solving Spring-Mass Systems

- Can be set up as an Euler integration
- $\mathbf{f}_0$ accumulates the spring forces, $\mathbf{M}^{-1}$ is a matrix of the masses at each point
- $\Delta\mathbf{x}$, $\Delta\mathbf{v}$ describe the discretized position and velocity update

$$\begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{v} \end{pmatrix} = h \begin{pmatrix} \mathbf{v}_0 \\ \mathbf{M}^{-1}\mathbf{f}_0 \end{pmatrix}$$

- Compare with implicit form:

$$\begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{v} \end{pmatrix} = h \begin{pmatrix} \mathbf{v}_0 + \Delta\mathbf{v} \\ \mathbf{M}^{-1}\mathbf{f}(\mathbf{x}_0 + \Delta\mathbf{x}, \mathbf{v}_0 + \Delta\mathbf{v}) \end{pmatrix}$$

Baraff, Witkin. SIGGRAPH 1998. https://www.cs.cmu.edu/~baraff/papers/sig98.pdf

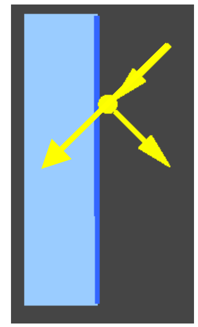
Examples: Cloth, Flag

http://www.cs.princeton.edu/courses/archive/spring14/cos426/assignment4/examples/examples.html

Collision

Particle System Forces

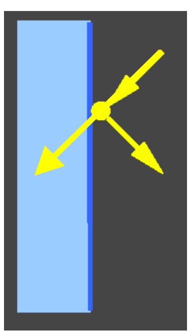
- Collisions
  - Collision detection
  - Collision response



Witkin

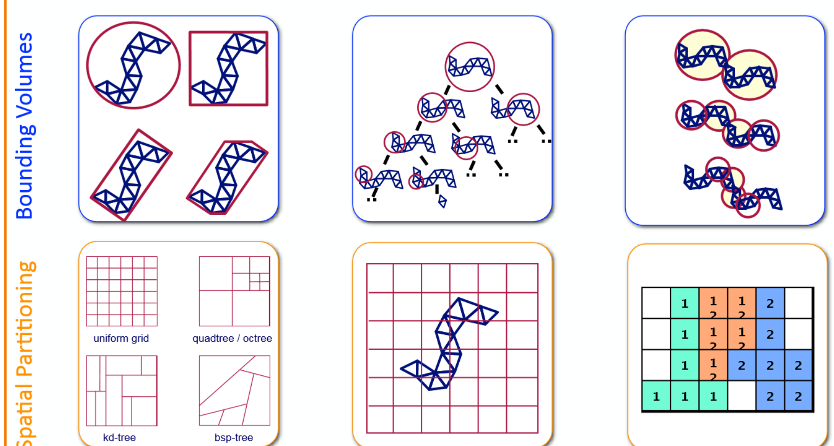
Particle System Forces

- Collision detection
  - Intersect ray with scene
  - Compute up to Δt at time of first collision, and then continue from there

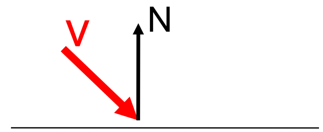


Witkin

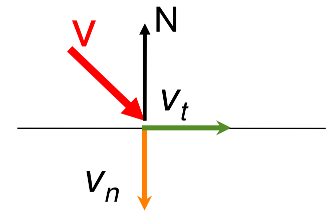
Collision Detection



Collision Response for Particles



Collision Response for Particles



$v=v_n+v_t$

normal component
tangential component

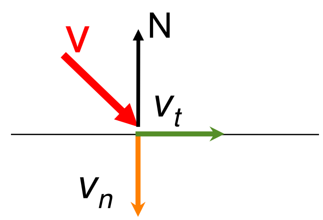
Collision Response for Particles

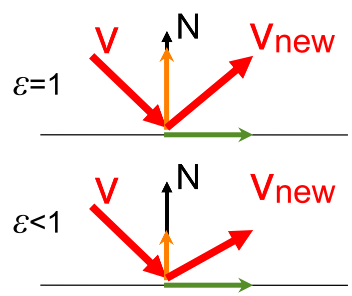
- Tangential velocity v_t *often* unchanged
- Normal velocity v_n reflects:

$$v = v_t + v_n$$

$$v \leftarrow v_t - \varepsilon v_n$$

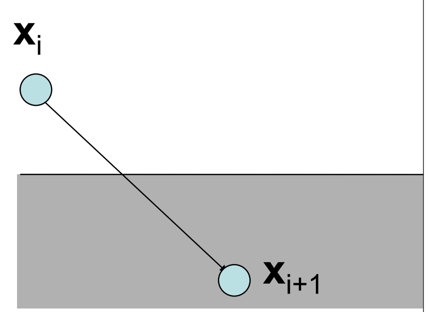
- Coefficient of restitution ε
- When $\varepsilon = 1$, mirror reflection





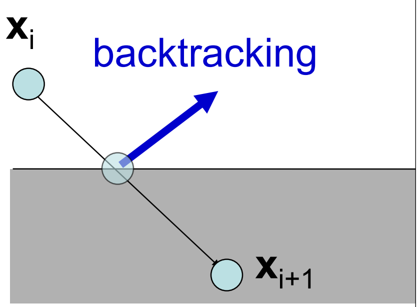
Collisions – Overshooting

- Usually, we detect collision when it is too late: we are already inside



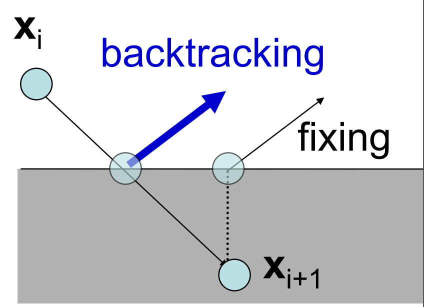
Collisions – Overshooting

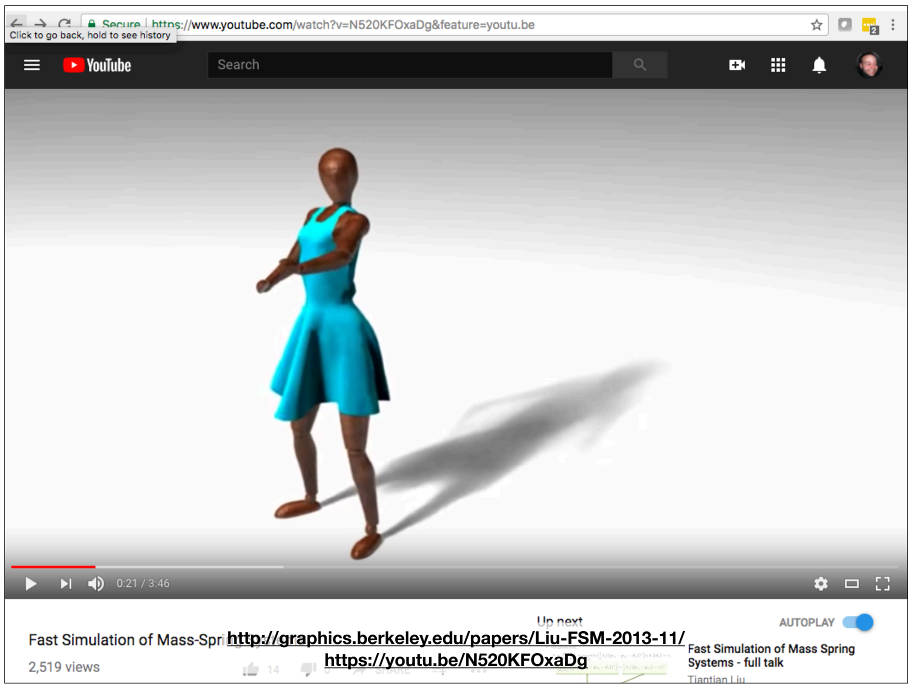
- Usually, we detect collision when it is too late: we are already inside
- Solution: Back up
  - Compute intersection point
  - Ray-object intersection!
  - Compute response there
  - Advance for remaining fractional time step



Collisions – Overshooting

- Usually, we detect collision when it is too late: we are already inside
- Solution: Back up
  - Compute intersection point
  - Ray-object intersection!
  - Compute response there
  - Advance for remaining fractional time step
- Other solution: Quick and dirty hack
  - Just project back to object closest point





http://graphics.berkeley.edu/papers/Liu-FSM-2013-11/
https://youtu.be/N520KFOxaDg

Animation So Far

- Interpolation based animation:
  - Interpolation of images, for example based on keyframes
  - Interpolation of motions, for example scripted character animation by designing transformation on per entity basis
- Physically-based animation techniques where we solve an ODE of some sort
  - We've started talking about this with particle-based physics
  - Today we'll talk about another scheme for it
- Next: we'll talk about non-physical animations through behaviors.

Animating Based on Behavioral Rules

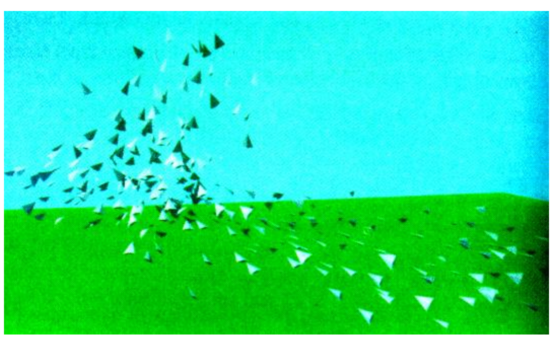
Boids

- Powerful, simple model
  - No central control
  - Only simple rules for each individual
  - Complex, emergent phenomena
  - Self-organization, swarm intelligence

Reynolds

Boids

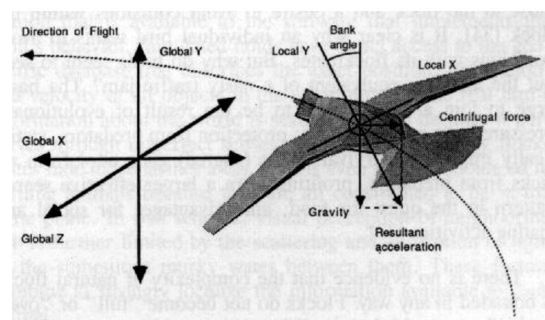
- Computer graphics motivation
  - Scripting of the path of many individual objects using traditional computer animation techniques is tedious.



Reynolds

Boids

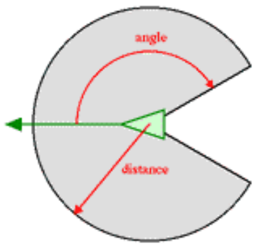
- Like a particle system, except …
  - Each boid may be an entire polygonal object with a local coordinate system (rather than a point)

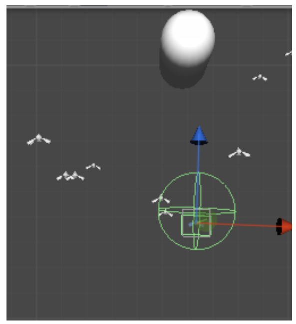


Reynolds

Boids

- Like a particle system, except …
  - Each boid can “perceive” a local region around it, e.g., a spherical neighborhood

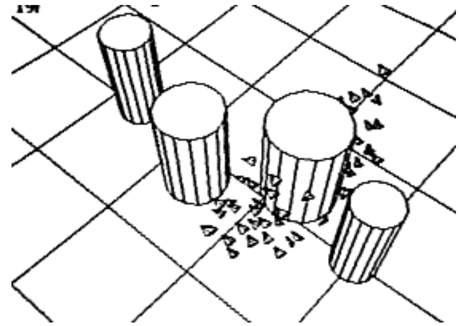




http://www.arges-systems.com

Boids

- Like a particle system, except …
  - Each boid exerts “intentional forces”



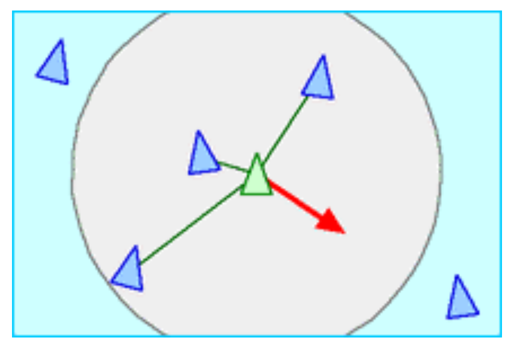
Reynolds

Flocking

- Complex flocking behaviors can be modeled with simple “intentional forces”
  - Separation
  - Alignment
  - Cohesion

Flocking – 3 Behaviors (1)

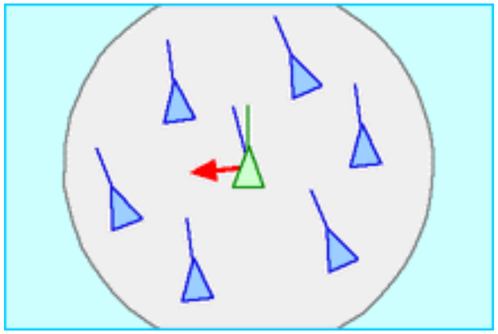
- Separation = collision avoidance: avoid collisions with nearby flockmates



http://www.red3d.com

Flocking – 3 Behaviors (2)

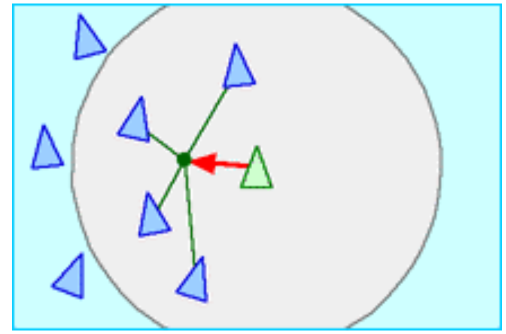
- Alignment = velocity matching:
  attempt to match velocity with nearby flockmates



http://www.red3d.com

Flocking – 3 Behaviors (3)

- Cohesion = flock centering:
  attempt to stay close to nearby flockmates



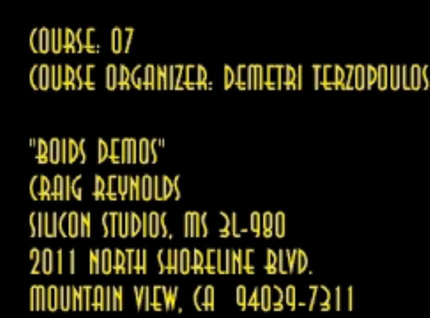
http://www.red3d.com

Other Examples (single behavior)

- Example behaviors
  - Seek
  - Flee
  - Evasion
  - Pursuit
  - Wander
  - Arrival
  - Obstacle Avoidance
  - Containment
  - Wall Following
  - Path Following

http://www.red3d.com/cwr/steer/

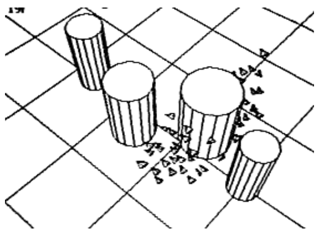
Examples



http://www.red3d.com/cwr/boids/

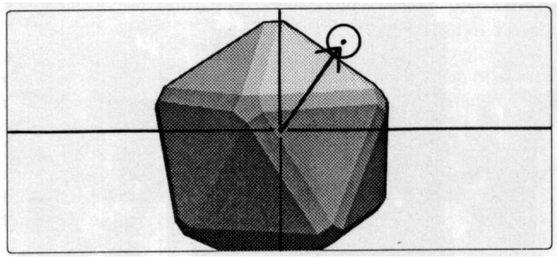
Obstacle Avoidance (1)

- Force field approach
  - Obstacles have a field of repulsion
  - Boids increasingly repulsed as they approach obstacle
- Drawbacks:
  - Approaching a force in exactly the opposite direction
  - Flying alongside a wall



Obstacle Avoidance (2)

- Steer-to-avoid approach
  - Boid only considers obstacles directly in front of it
  - Finds silhouette edge of obstacle closest to point of eventual impact
  - A vector is computed that will aim the boid at a point one body length beyond the silhouette edge

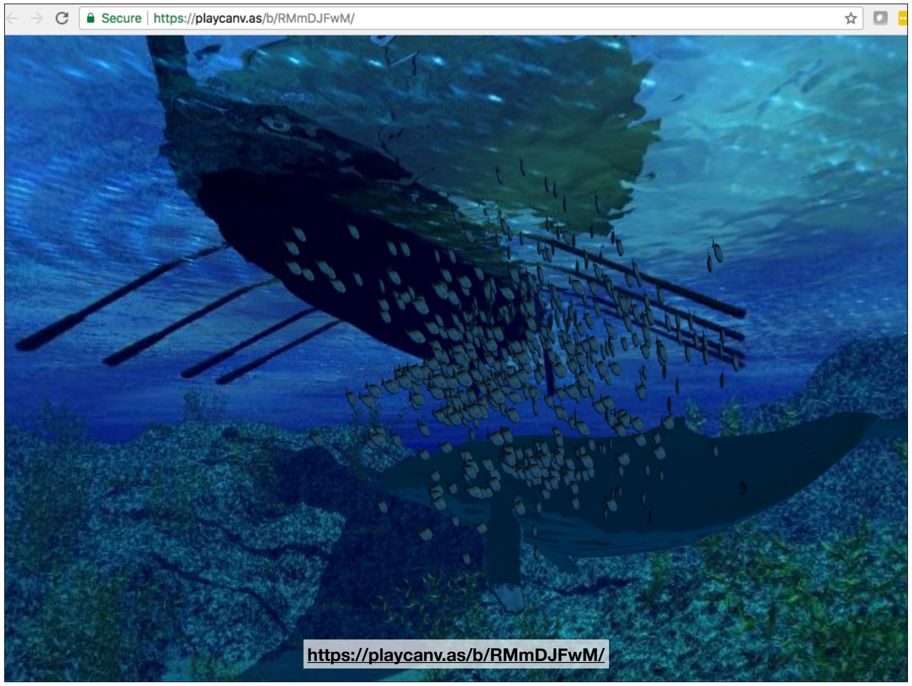


http://www.red3d.com

Arbitrating Independent Behaviors

- Navigation module of boid brain to collect relevant acceleration requests and then determine single behaviorally desired acceleration
  - Weighted average according to priority
- Emergency acceleration allocated to satisfy pressing needs first
  - Example: Centering ignored in order to maneuver around obstacles

http://ncase.me/improv-wip/


Secure | https://playcanv.as/b/RMmDJFwM/
https://playcanv.as/b/RMmDJFwM/