GDC 13 EUROPE

Liquid Intelligence

Combining AI and Physics in Vessel

John Krajewski
President, Strange Loop Games

GAME DEVELOPERS CONFERENCE EUROPE
COLOGNE, GERMANY
AUGUST 19–21, 2013
EXPO DATES: AUGUST 19–20
2013

Vessel

Part 1: Programming

Systems-based design

1. Physics Objects

3. Liquid

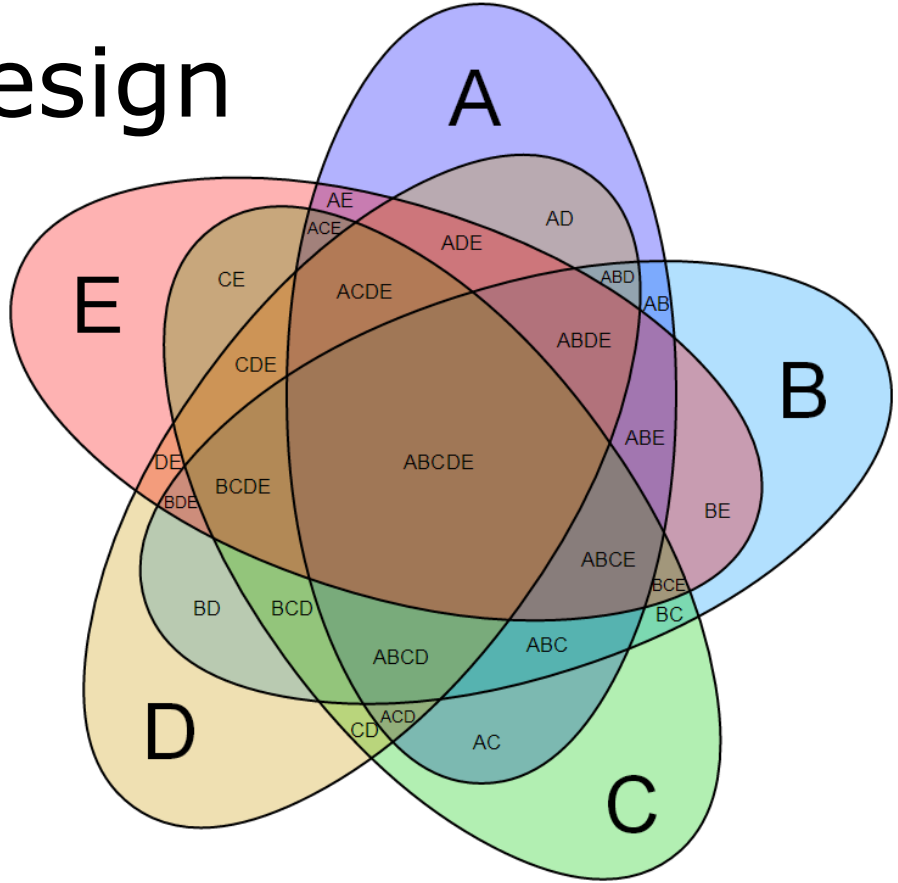
2. Constraints and Springs

4. AI

Systems-based design

- Goal for systems:

To be **independent with rich interactions.**



Physics bodies

- Impulse based
- Seamless support for non-convex shapes
- Contact graph stacking algorithm

Constraints and Springs

- Attach point between objects
- ‘Hinge’ setting
  - Sliding, rotating, axis constrained
- All interactive objects are physical

Liquids

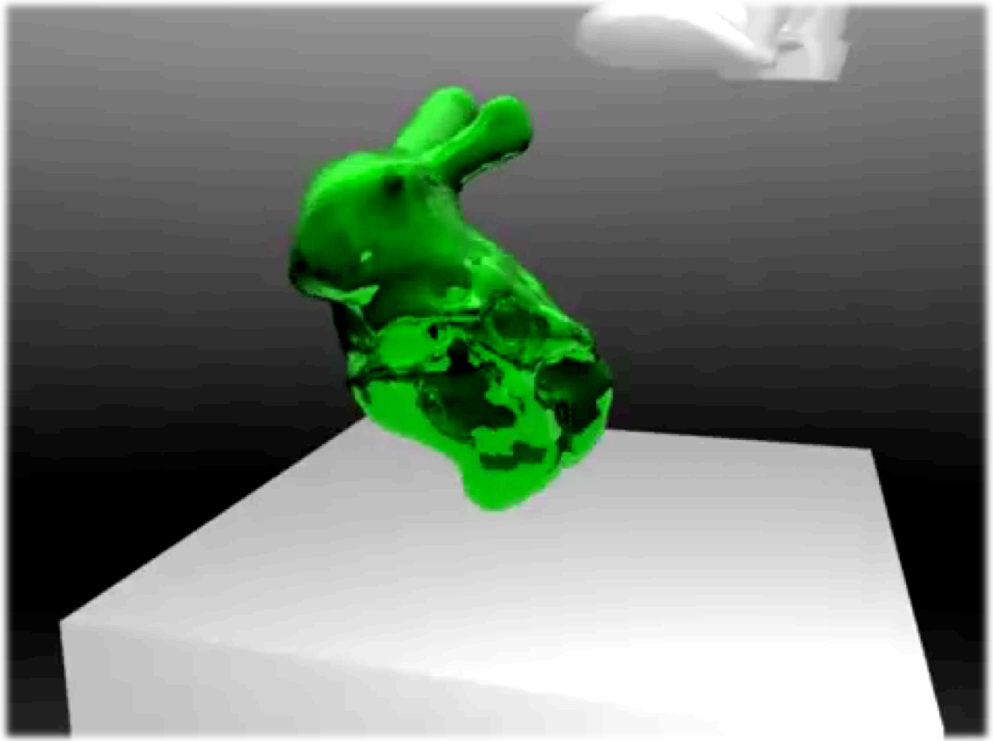
- Smoothed Particle Hydrodynamics

Liquids

Particle-based Viscoelastic Fluid Simulation

Simon Clavet, Philippe Beaudoin, and Pierre Poulin

LIGUM, Dept. IRO, Université de Montréal



Liquids

- Compute three values for each particle:
  - Density
  - Pressure
  - Near-pressure

Liquids

- Particles can both push and pull neighbors
- Multiple kernels creates surface tension effects

Liquids

- Pressure Optimization:
  - Track neighboring particles with a sparse hash grid

Liquids

- Viscosity
  - Friction between drops
  - Calculates velocity between neighboring drops, applies drag

Liquids

- Stickiness
  - Attraction to nearest wall
    - Extension of collision system
  - Dynamic springs between particles

Liquids

- Collision
  - Liquid must move and be moved by objects
  - Huge performance drain
  - Optimization: add objects into hash grid

Liquids

- Chemical Reactions

Water + Lava = Steam

Red Goo + Blue Goo = BAM

Liquids

- Lighting
  - 2D planes + normal maps

Liquids

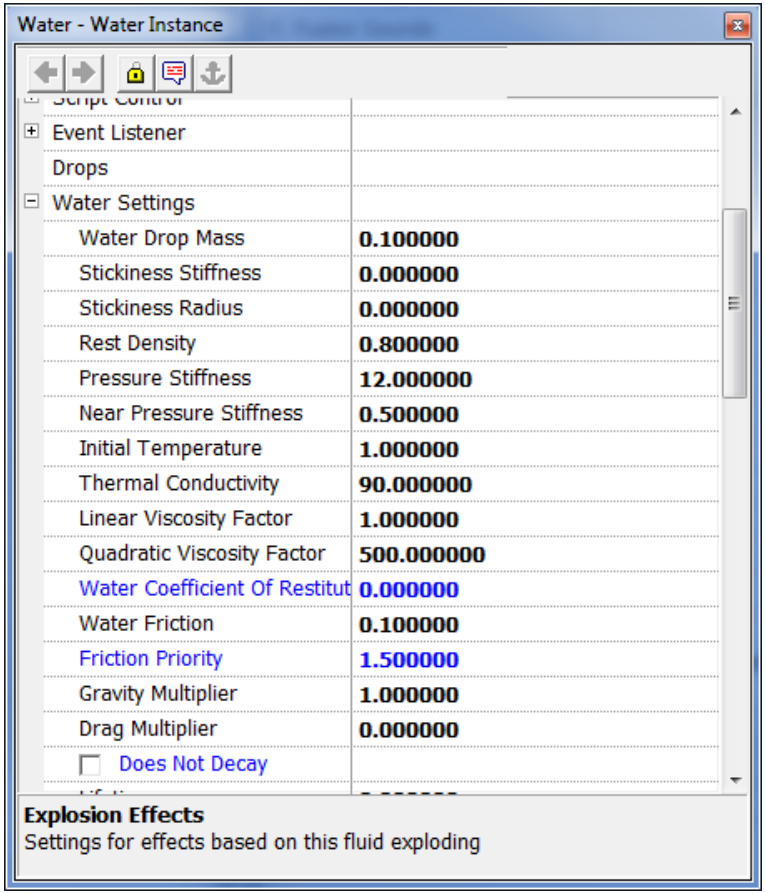
- Lighting
  - Build clusters of liquid
  - Place lights at clusters
  - Use hashtable again
  - Spatial/temporal coherence
  - Parameters (density, avg. velocity) determine light type

Liquids

- Attractors
  - Point attractors, exert a force
  - Uses spatial partitioning again

Liquids

- Endless parameter tweaks!
  - Hundreds of parameters



AI

- Liquid Creatures
  - Animated skeletons
  - Each bone can attract liquid
  - Attaches close drops, pulls them close
  - Different numbers of particles depending on the bone

AI

- Circulatory system
  - Transfer particles between bones
  - Particles flow to most needy bone
  - Fluros can function with a range of particles
  - Regrowing lost limbs
  - Seamless flow between living and non-living

AI

- Behaviors
  - Priority tree of behaviors
  - Performs highest behavior that meets conditions
  - Conditions can be physics based

AI

- Pathfinding
  - A* on a hand-placed network
  - Fluros can ‘claim’ objects like clusters or buttons
  - Behaviors very predictable so they can fit into puzzles

Part 2: Design

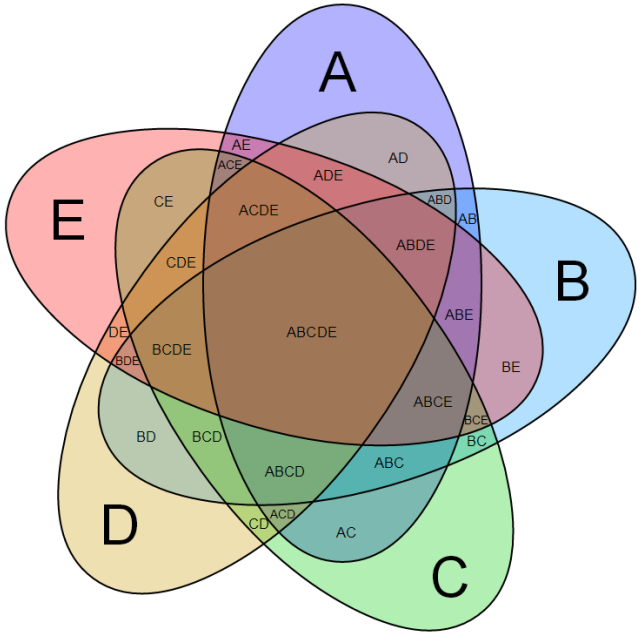
Interacting systems

"Every block of stone has a statue inside it and it is the task of the sculptor to discover it."

Michelangelo

Interacting systems

- Puzzle solutions = the results of combined systems
- Fair puzzles: the player knows the facts, they must induct the solution
- Satisfying to combine ideas
  - *"Solving Vessel's puzzles feels like a 1,000 watt light bulb flicking on in your brain." - GamesRadar*



Puzzle example: Combining Chemical Reactions and Fluros

Puzzle example: Teaching for the ‘exam’

Design – Unexpected effects

- Cycle: Create a system, play with it, wonder ‘what-if’
- Designing is just like playing the game
- Science – stretching the unexpected
- Programming and design are inseparable

Puzzle example: Stretching a small effect

Not just puzzle design

- All the aspects of the game grew out of the core systems
  - Story – Living machines
  - Visuals – A mechanical world
  - Music – Contemplative, dark, glitchy

John Krajewski - Strange Loop Games – john@strangeloopgames.com
Twitter: strangeloopgame