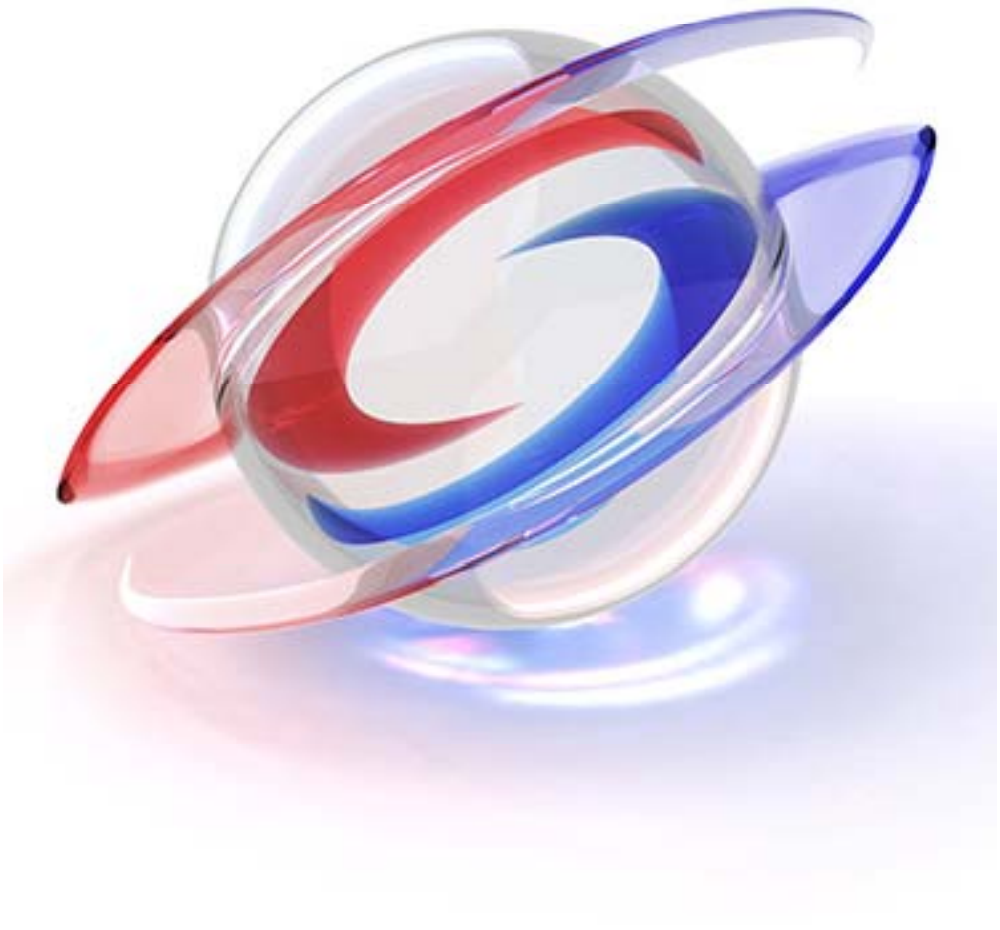
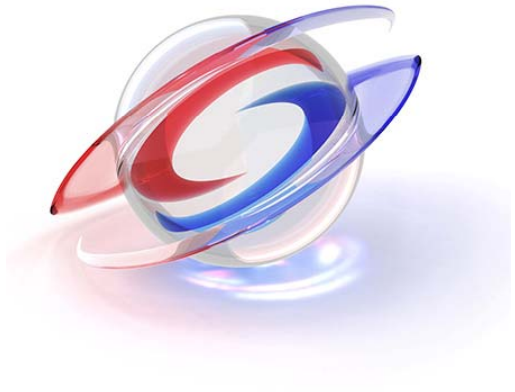


# Collision detection in PhysX

Recent Advances in Real-Time Collision and Proximity Computations for Games and Simulations
SIGGRAPH 2010 Course

Richard Tonge
*NVIDIA*

# Course Outline



- 2:00pm Course Introduction [Yoon]
- 2:10pm Introduction to Collision and Proximity Queries [Manocha]
- 2:30pm Proximity Queries for Rigid and Articulated Characters [Kim]
- 3:00pm Collision Detection for Deformable and Fracturing Models [Yoon]
- 3:30pm Break
- 3:45pm GPU-Based Proximity Computations [Manocha]
- 4:15pm Optimizing Proximity Queries for CPU, SPU and GPU [Coumans]
- 4:45pm Collision detection in PhysX [Tonge]

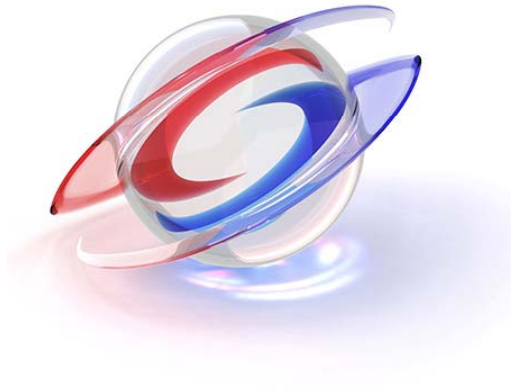
# Collision detection in PhysX

- PhysX overview
- GPU considerations
- Fluids
- Cloth
- Rigid bodies

# What is PhysX?

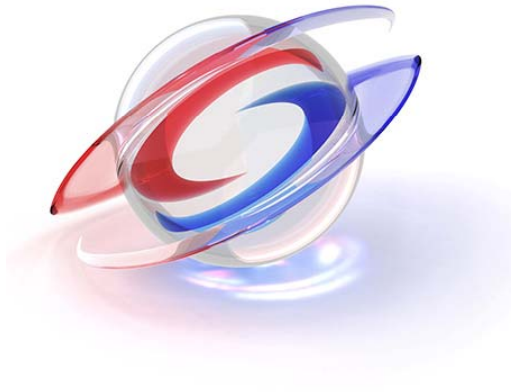
- Immersion and interactivity for games
  - Static walls vs walls that blow up when you shoot them
  - Rendered fog vs fog that swirls around the player
  - Bare rooms vs rooms with movable objects and clutter
- Large scale effects for VFX production
  - Explosions, building collapse etc.

# PhysX Features



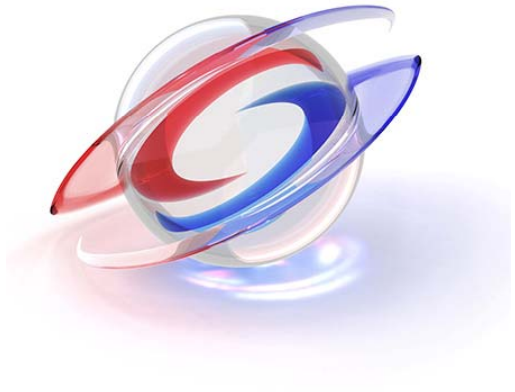
- Rigid bodies
  - Destruction, clutter, ragdolls, vehicles
- Fluids
  - Debris effects, smoke
- Deformables
  - Clothing, organic creatures, meaty chunks
- Authoring –3ds Max, Maya, Softimage, APEX

# What is PhysX?



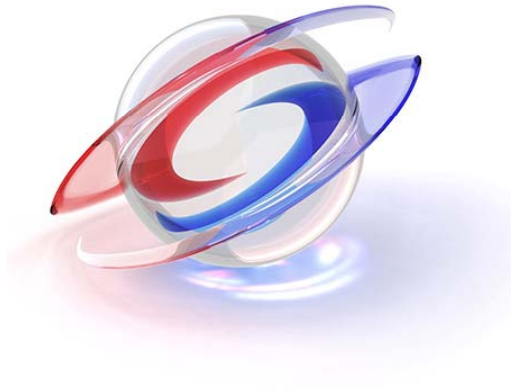
- User base
  - Over 150 games
  - Third party VFX production plugins (TV, film etc.)
- Platforms
  - PC, Mac, Xbox 360, Playstation3, Wii, iPhone
- Processors
  - Optimized for  CPUs, SPUs and GPUs

# Overview of Collision in PhysX



- Broadphase
  - AABB vs AABB, 3 axis sweep and prune
- Midphase
  - AABB tree vs (AABB, OBB, sphere,capsule,plane, ray)
  - OPCODE AABB tree
- Narrowphase
  - Coming up…

# Narrowphase



- SPH Fluids (CCD)
  - Particles vs static triangle mesh
  - Particles vs dynamic primitives
- Cloth (CCD)
  - Vertices vs static triangle mesh
  - Vertices vs dynamic primitives
- Rigid body (Discrete)
  - Convex mesh and primitives vs static triangle mesh
  - Convex, primitives vs Convex, primitives

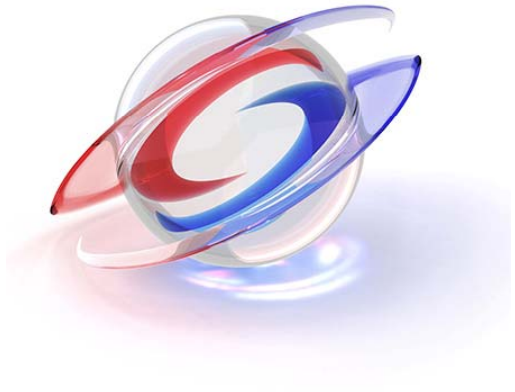
# Collision detection in PhysX

- PhysX overview
- GPU considerations
- Fluids
- Cloth
- Rigid bodies

# Why GPU PhysX?

- Physics is highly parallel
  - For 1000s of threads, GPU is the right tool for job
- Rendering is quite good these days
  - Spending GPU cycles on physics can have greater bang per buck than improving rendering

# GPU algorithm goals



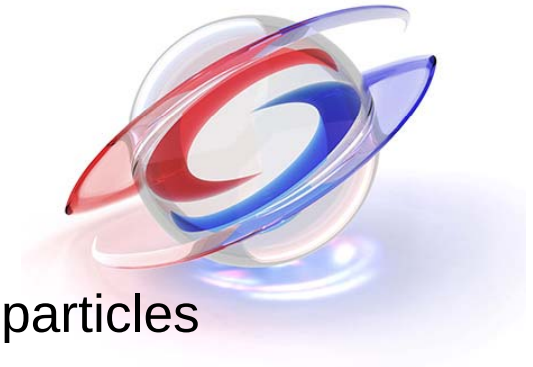
- For best GPU performance, we need
  - Work for thousands of threads
  - Minimal register usage to maximize num concurrent threads and hide latency
  - Minimal synchronization between threads
  - Similar amount of work per thread
  - Data locality

# Collision detection in PhysX

- PhysX overview
- GPU considerations
- Fluids
- Cloth
- Rigid bodies

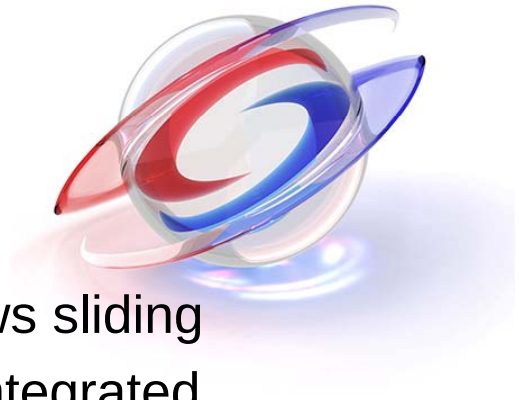
# Fluids

# Fluid Requirements

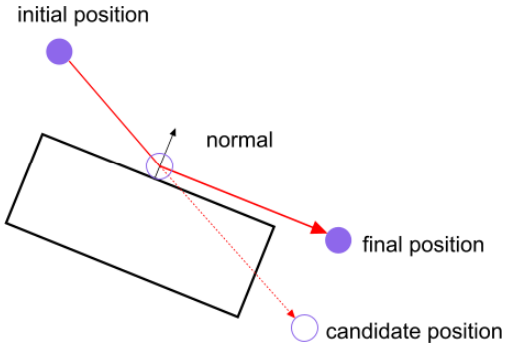


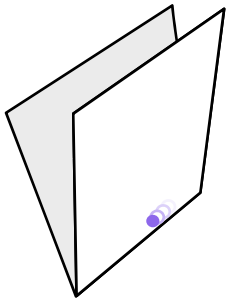
- We use an SPH model where fluid is discretized as particles
- So collision detection is just particle vs primitive and particle vs mesh
- Requirements
  - Particles must not leak or stick
  - No penetration even at high velocities
  - Static object collision must have higher priority than dynamic
  - Target 10000 to 100000 particles at 60Hz on GPU

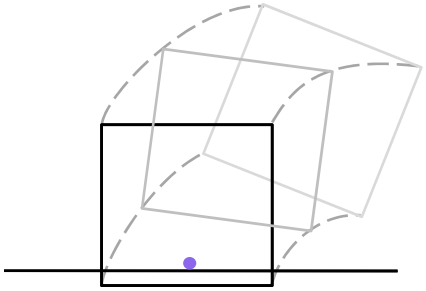
# Implementation



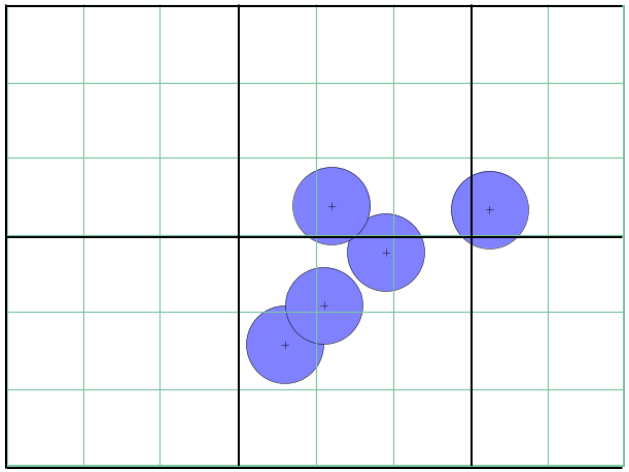
- CCD finds first impact point, contact constraints allows sliding
- Constraint solver is not required as each particle is integrated separately

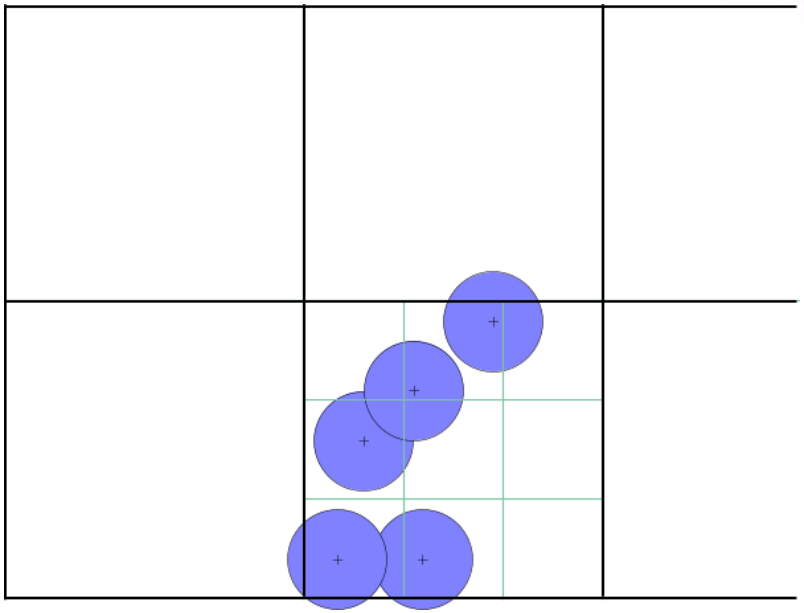


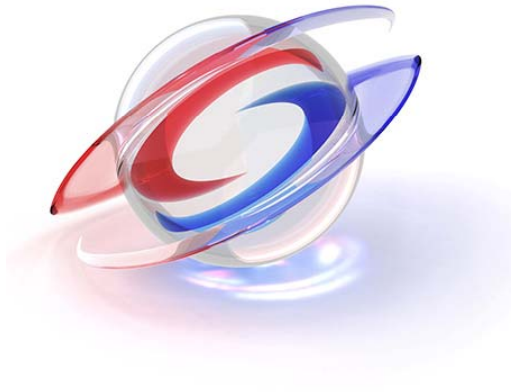


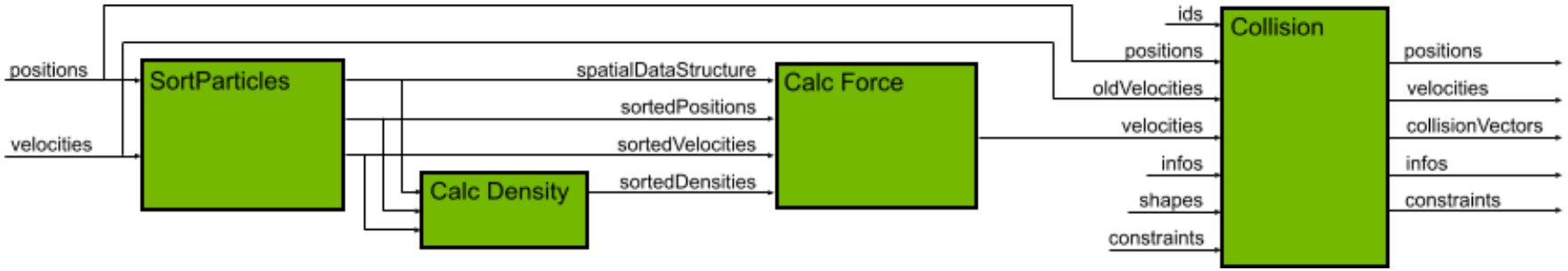
# GPU Implementation – uniform grid





# Fluids




positions
velocities
SortParticles
spatialDataStructure
sortedPositions
sortedVelocities
sortedDdensities
Calc Density
Calc Force
ids
positions
oldVelocities
velocities
infos
shapes
constraints
Collision
positions
velocities
collisionVectors
infos
constraints

# Collision detection in PhysX

- PhysX overview
- GPU considerations
- Fluids
- Cloth
- Rigid bodies

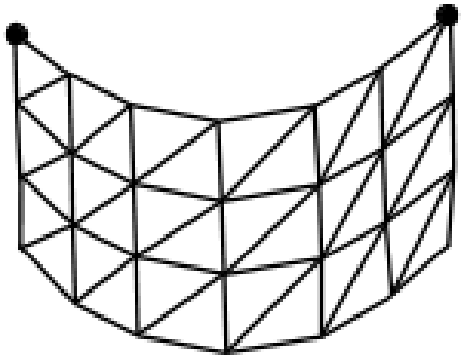
# Cloth

# Cloth Requirements

- High velocity without tunnelling
  - Typical use case is fast moving clothed character
- Minimize resting penetration with rigid objects
  - Cloth is often in contact with character
- Support tearing and rigid attachment
- Lots of independent threads for GPU
- Target 10 cloths with up to 10K vertices on GPU at 60Hz

# Cloth Model

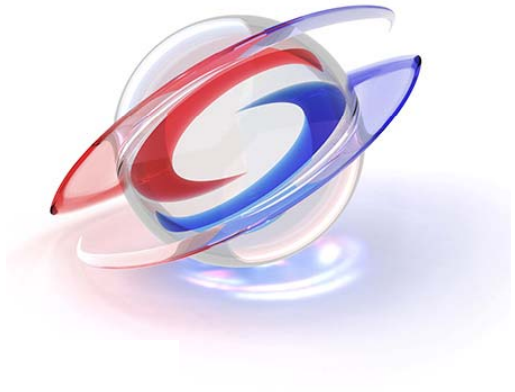
- Cloth is discretized as a triangle mesh
- Collision is CCD to eliminate tunneling
- Collision detection is performed on vertices only
  - So cloth triangles must be small to minimize penetration
  - But, it's very fast, especially on GPU
- Self collision

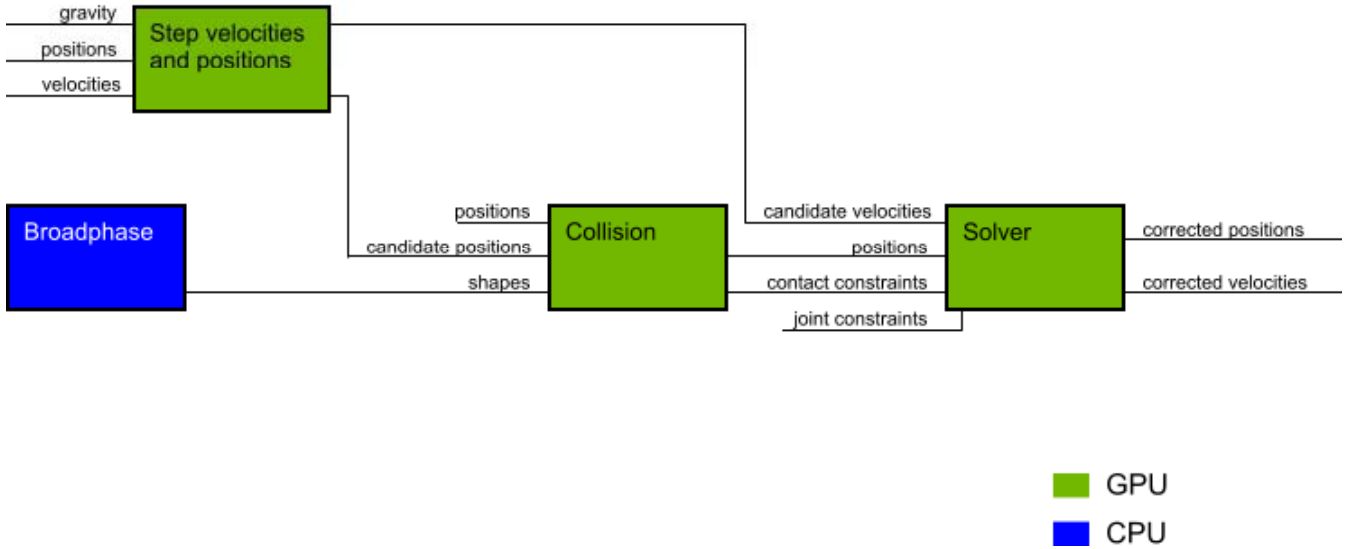


# Cloth Midphase

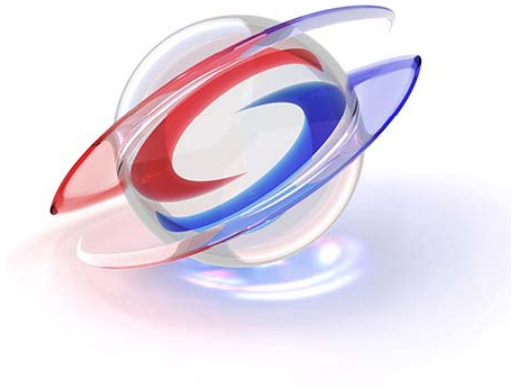
- Cloth vs triangle mesh AABB collision reported by broadphase
- Cloth ray AABB tested against shape AABBs
- Finally, ray tested against triangles or shapes

# Cloth Pipeline





# Cloth video

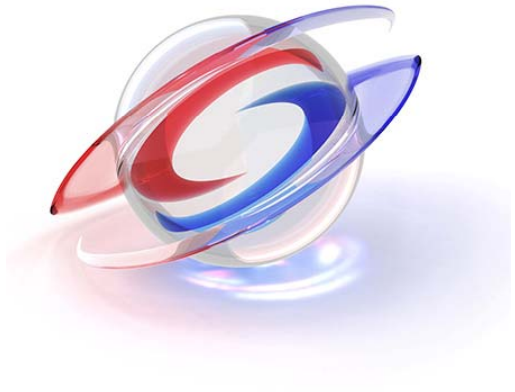


# Collision detection in PhysX

- PhysX overview
- GPU considerations
- Fluids
- Cloth
- Rigid bodies

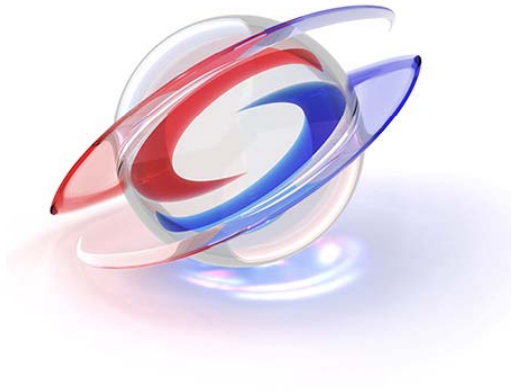
# Rigid Bodies

# Rigid Body Requirements



- Smooth sliding
- Stable stacking
- Stable resting, no jitter
- Parallelizable >1 thread per pair for GPU
- Speed targets
  - 100 -1000 bodies on consoles and PC CPU at 60Hz
  - 1000-10000 bodies on GPU at 60Hz

# Rigid Body Design Options



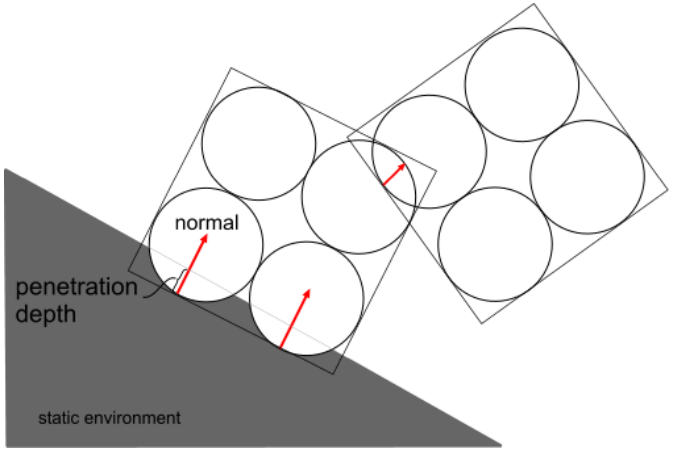
- CCD based or penetration based
- GJK or SAT
- Simultaneous or incremental manifold

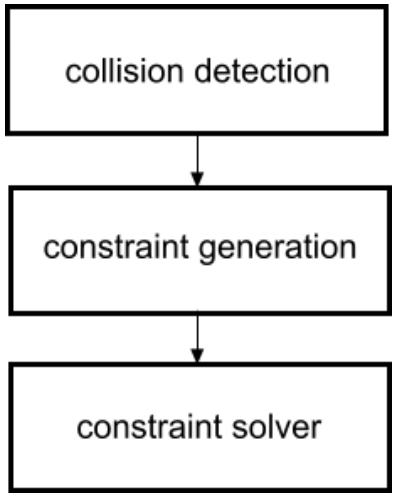
# Rigid Body Design Choices

- Penetration based
  - Bodies penetrate due to initial positioning and numerical error anyway.
- SAT with projection and polygon clipping
  - Lots of threads per pair for GPU implementation.
- Simultaneous contact manifold generation
  - Robust, easy to implement, lots of threads for GPU.

# GPU Rigid Bodies 0.9

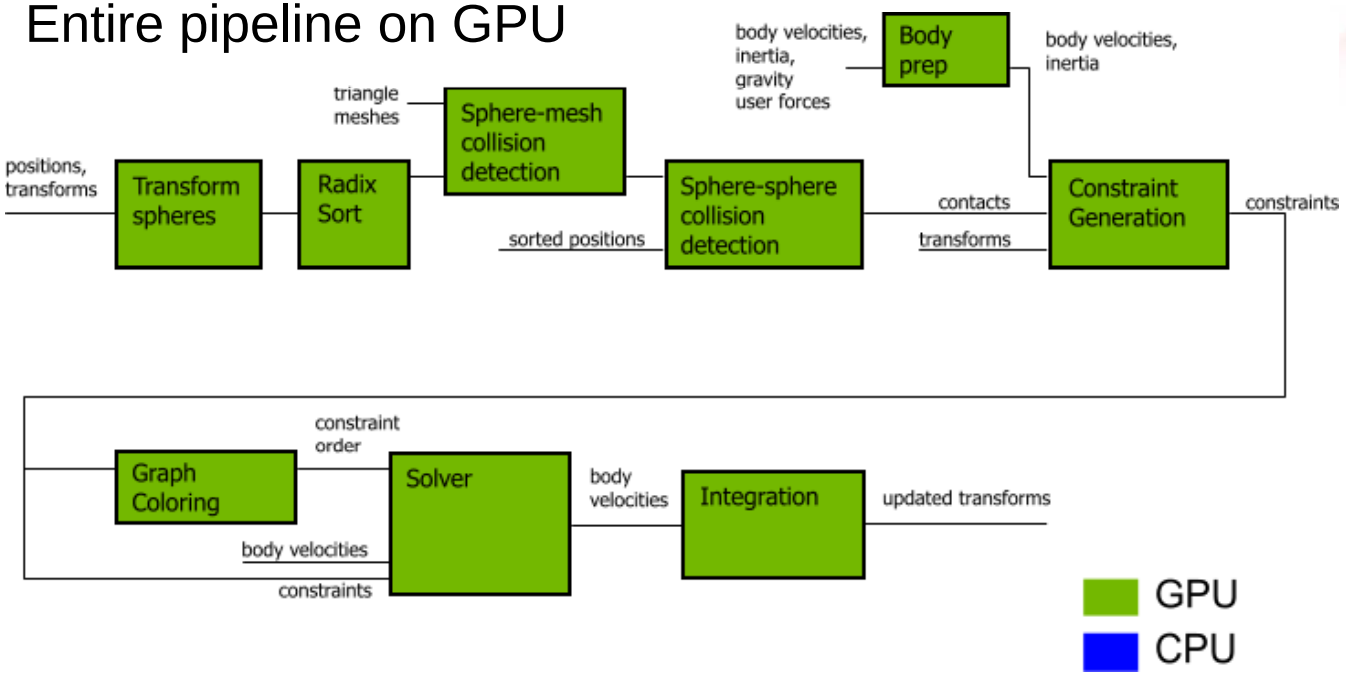
- Used in Batman: Arkham Asylum
- Bodies voxelized, spheres placed at non-empty voxels
- Collision detection is sphere-sphere and sphere-tri mesh
- Constraints solved by iterative LCP solver



# GPU Rigid Bodies 0.9

Entire pipeline on GPU



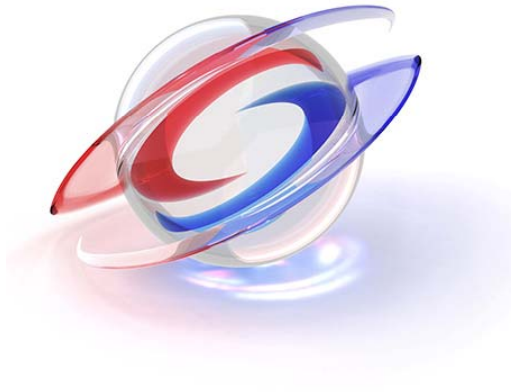
# GPU Rigid Bodies 1.0

- Uses actual geometry instead of spheres
- Collision detection is convex-convex and convex-tri mesh
- Same broadphase, midphase, narrowphase, friction model and solver as CPU PhysX
- Entire pipeline on GPU
- Coming soon…

# Example user: RayFire Tool

- 3rd party 3ds Max plugin for VFX production
- Designed by VFX artist Mir Vadim
- Used in TV, commercials, movies and games
- Utilizes PhysX for rigid body collision detection and simulation
- More info - www.mirvadim.com

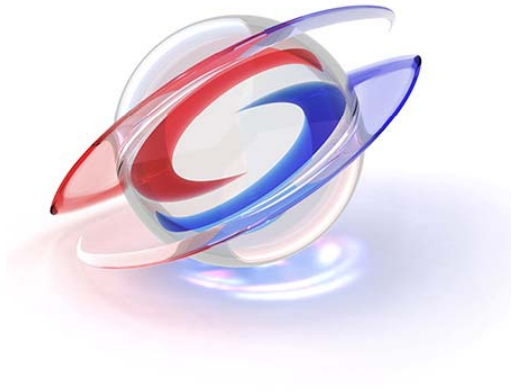
# RayFire Tool video



# About the video

- Scenes used up to 2000 PhysX convex bodies
- Static environment
  - Unlike games, detailed static meshes not required
  - Large static box blocking volumes are used instead
  - Can use static tri meshes if needed for close ups
- Demonstrates PhysX broadphase, midphase and convex-convex narrowphase

# Questions



# SIGGRAPH 2010

***WONDER!***

"***The People Behind The Pixels***"

*Community … Clarity … Content*