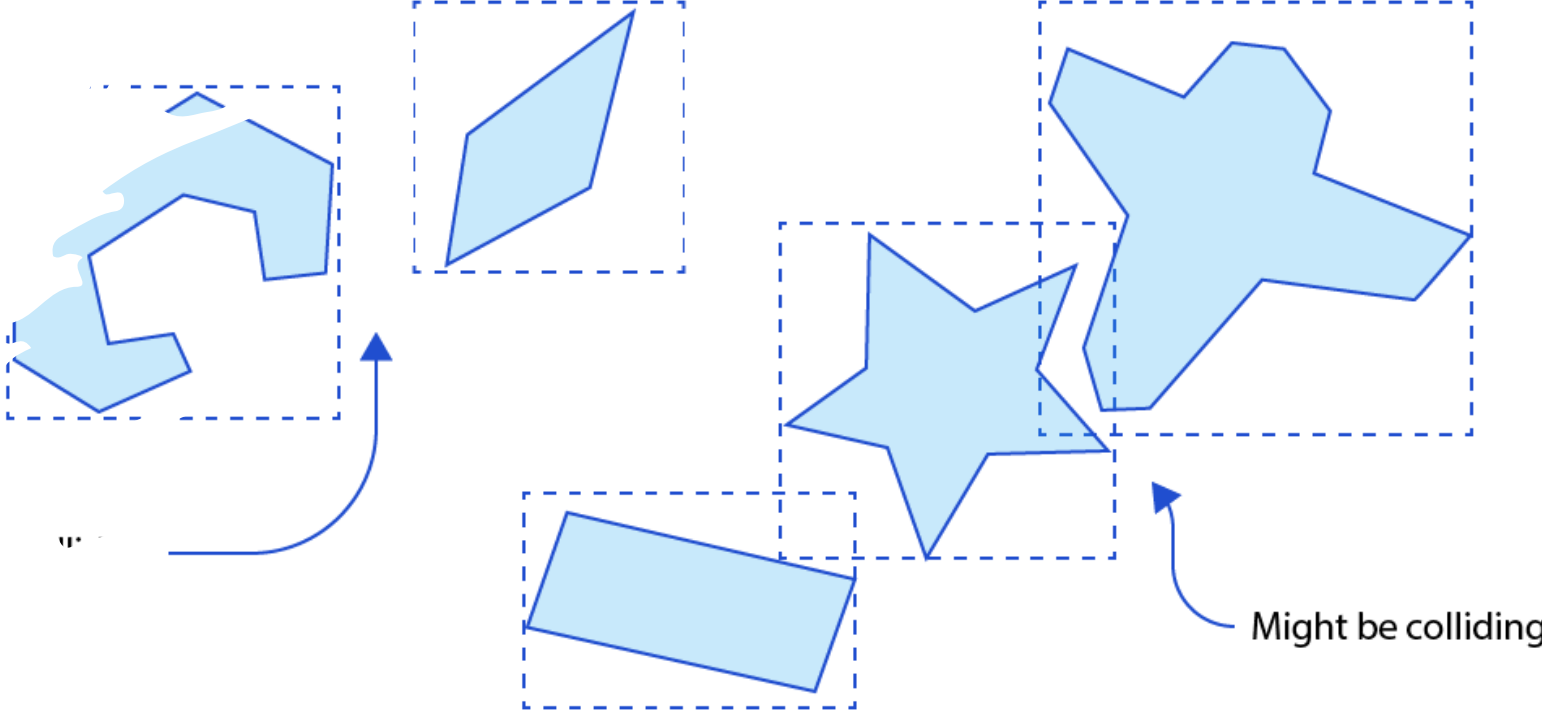
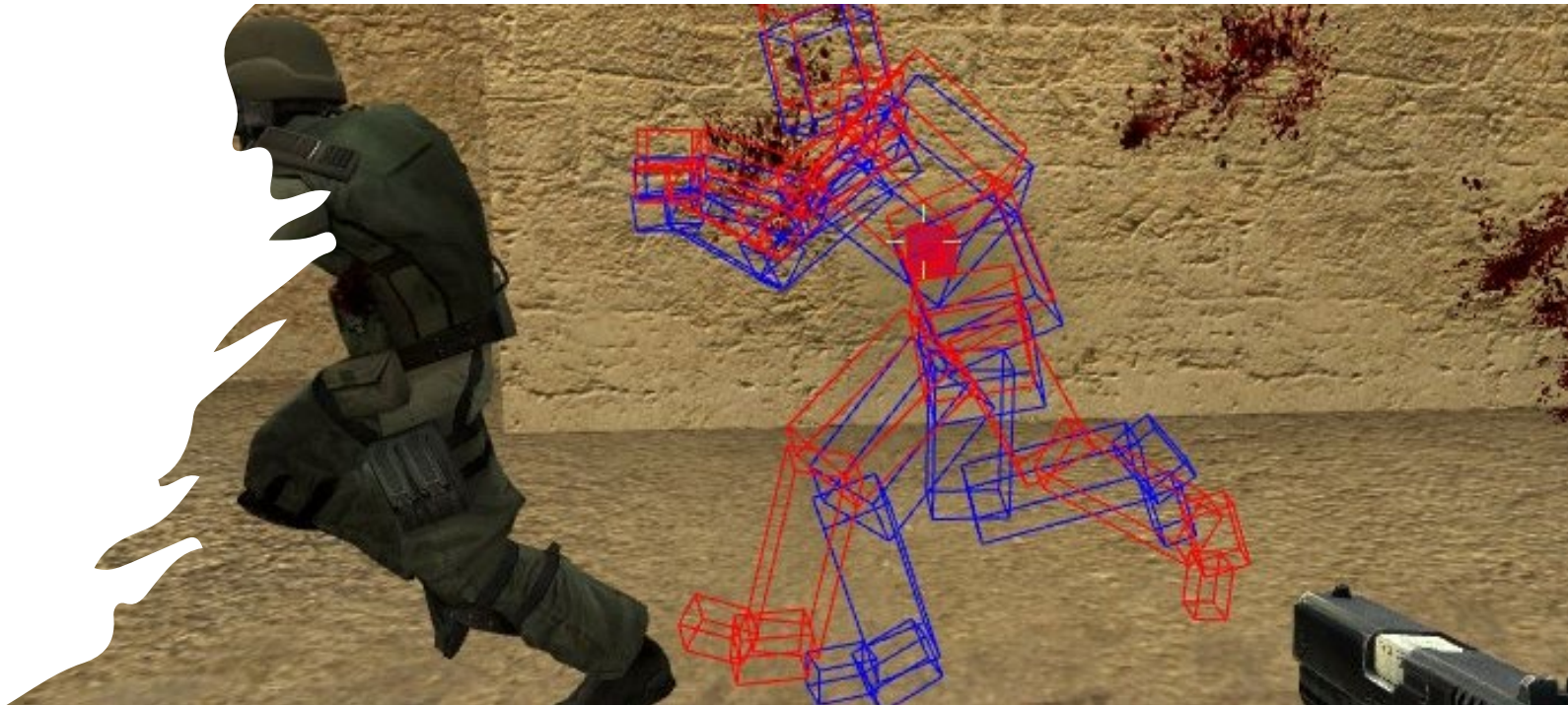
Collision in Computer Graphics (In Simulation or Games)





Simulations (Sims)

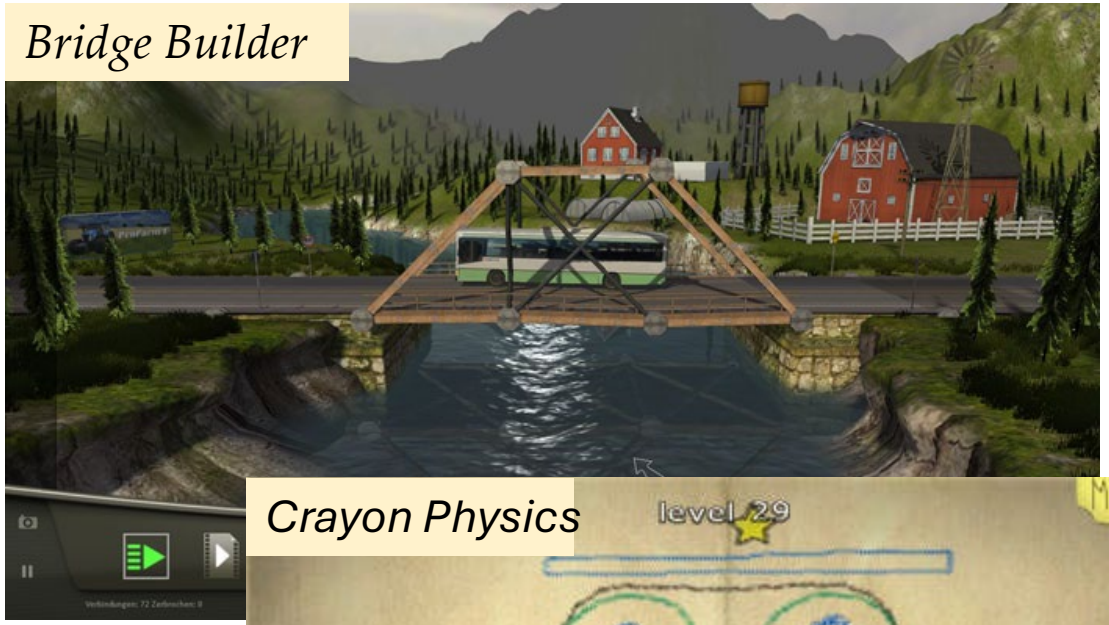
Flight Simulator

Gran Turismo

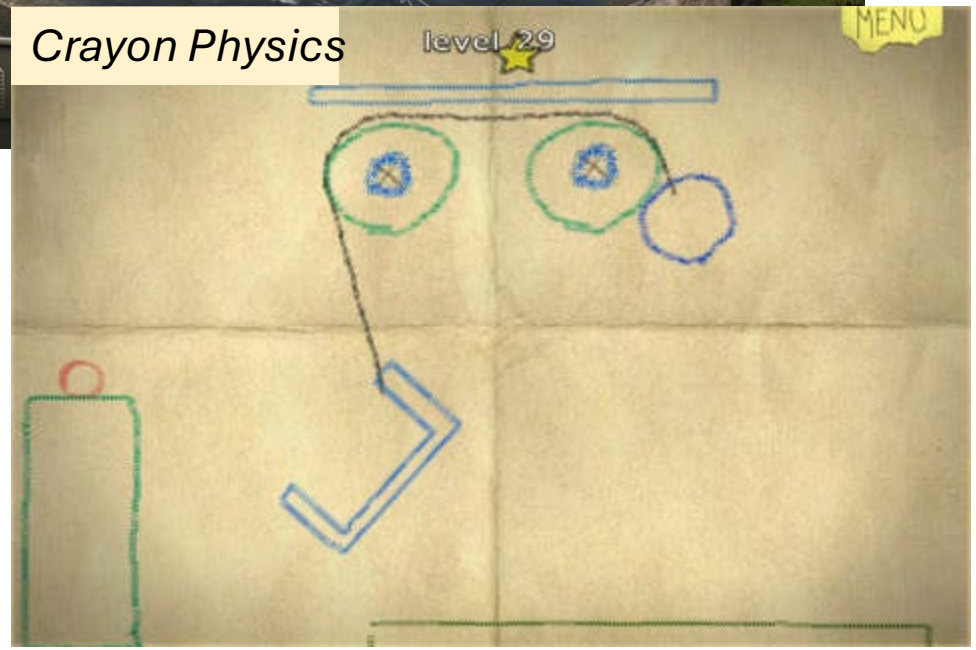
Need For Speed

Physics Puzzle Games

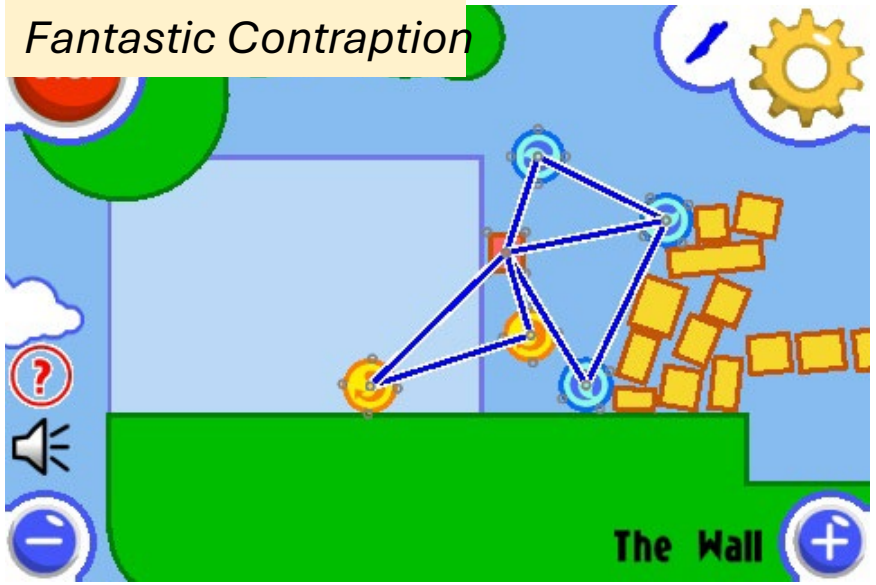
Bridge Builder



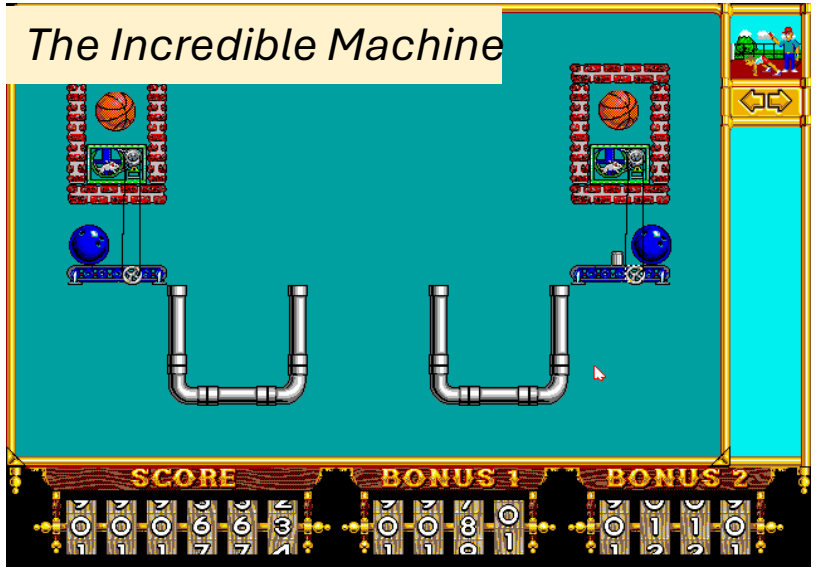
Crayon Physics



Fantastic Contraption



The Incredible Machine



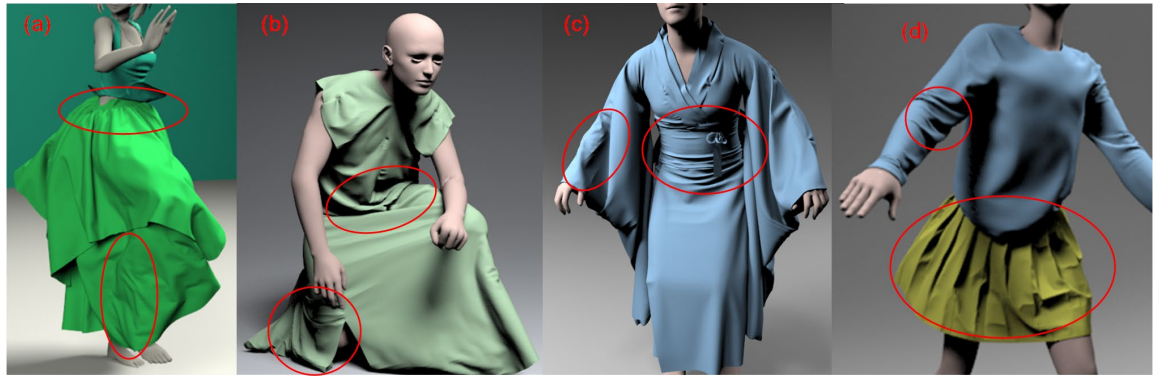
Sandbox Games

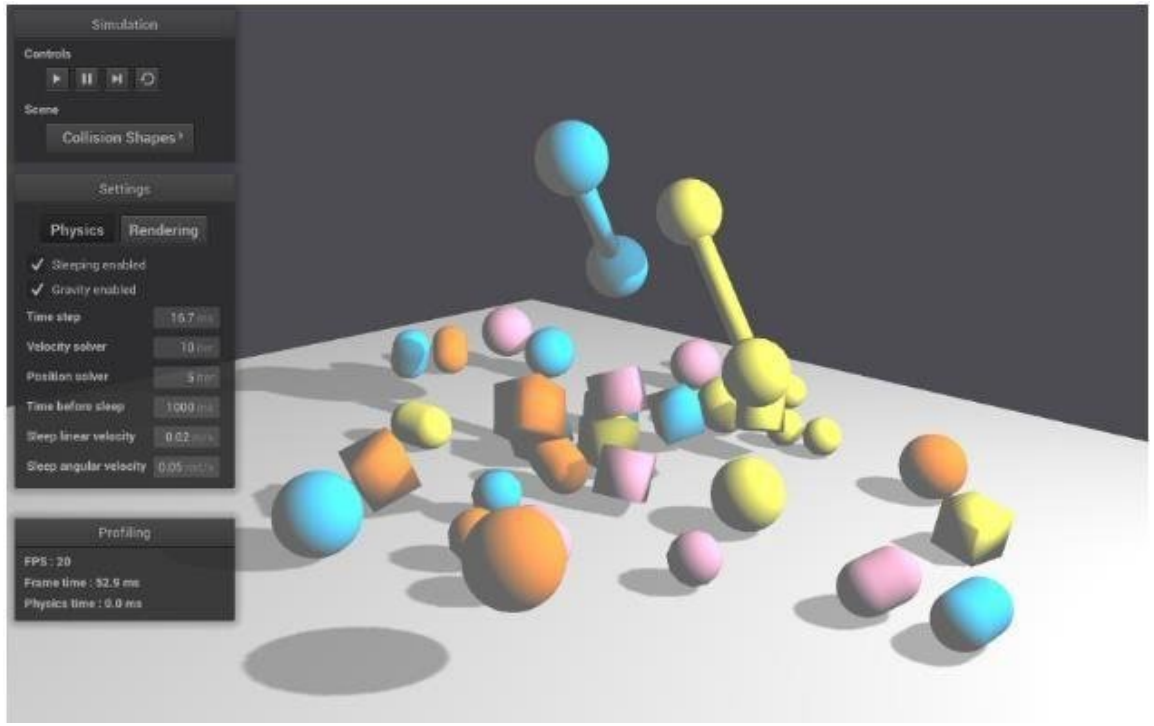
LittleBigPlanet

GTA 5

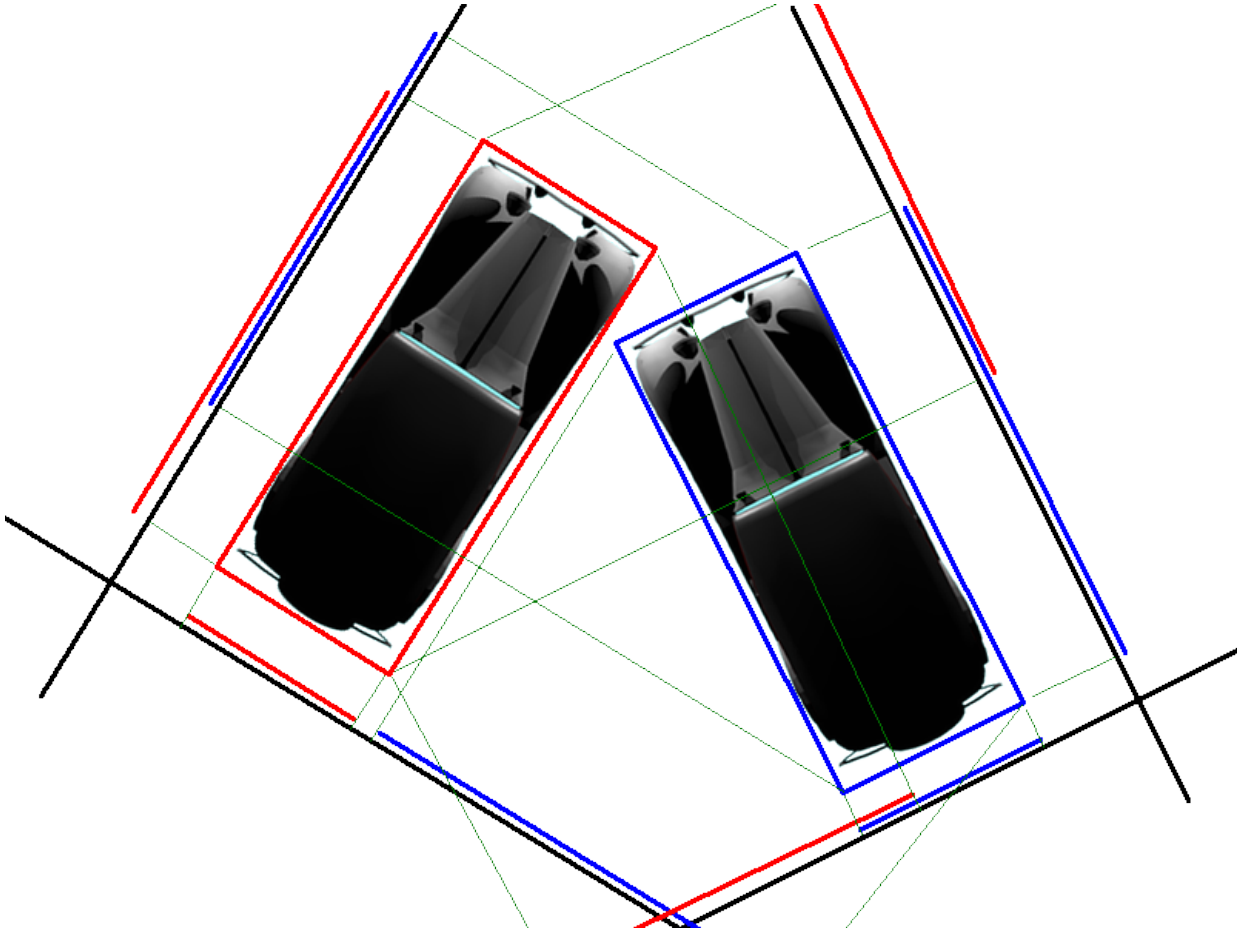
Spore

In Simulation

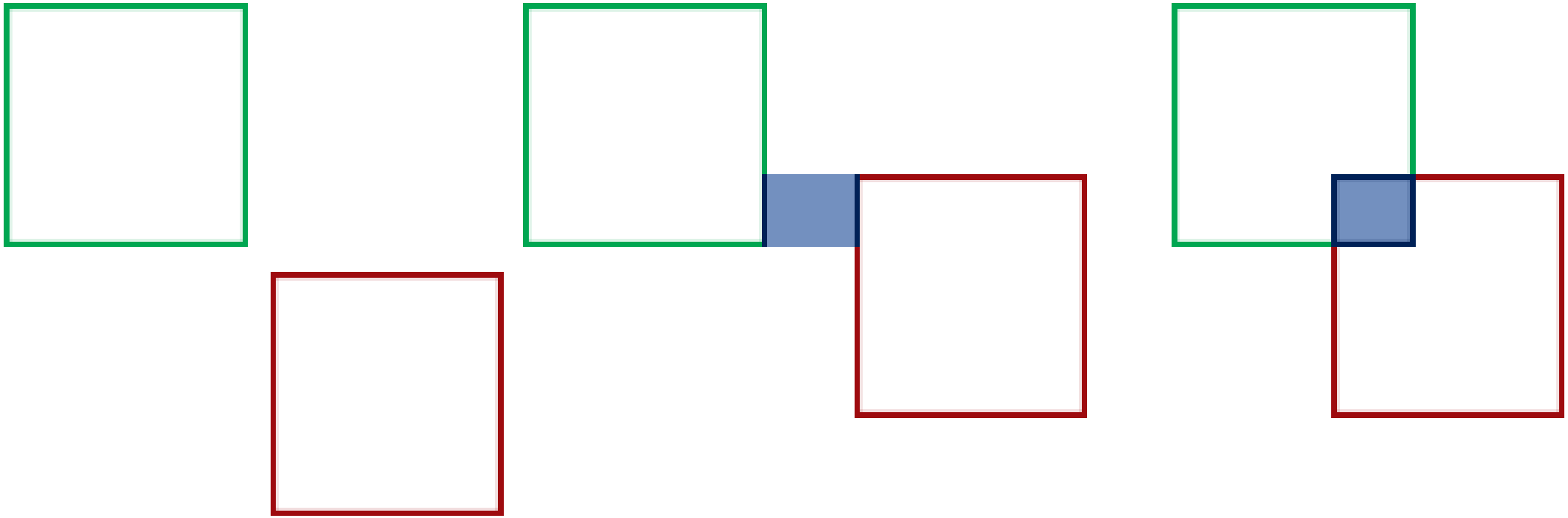




What is collision?



Collision detection is the computational problem of detecting the intersection of two or more objects

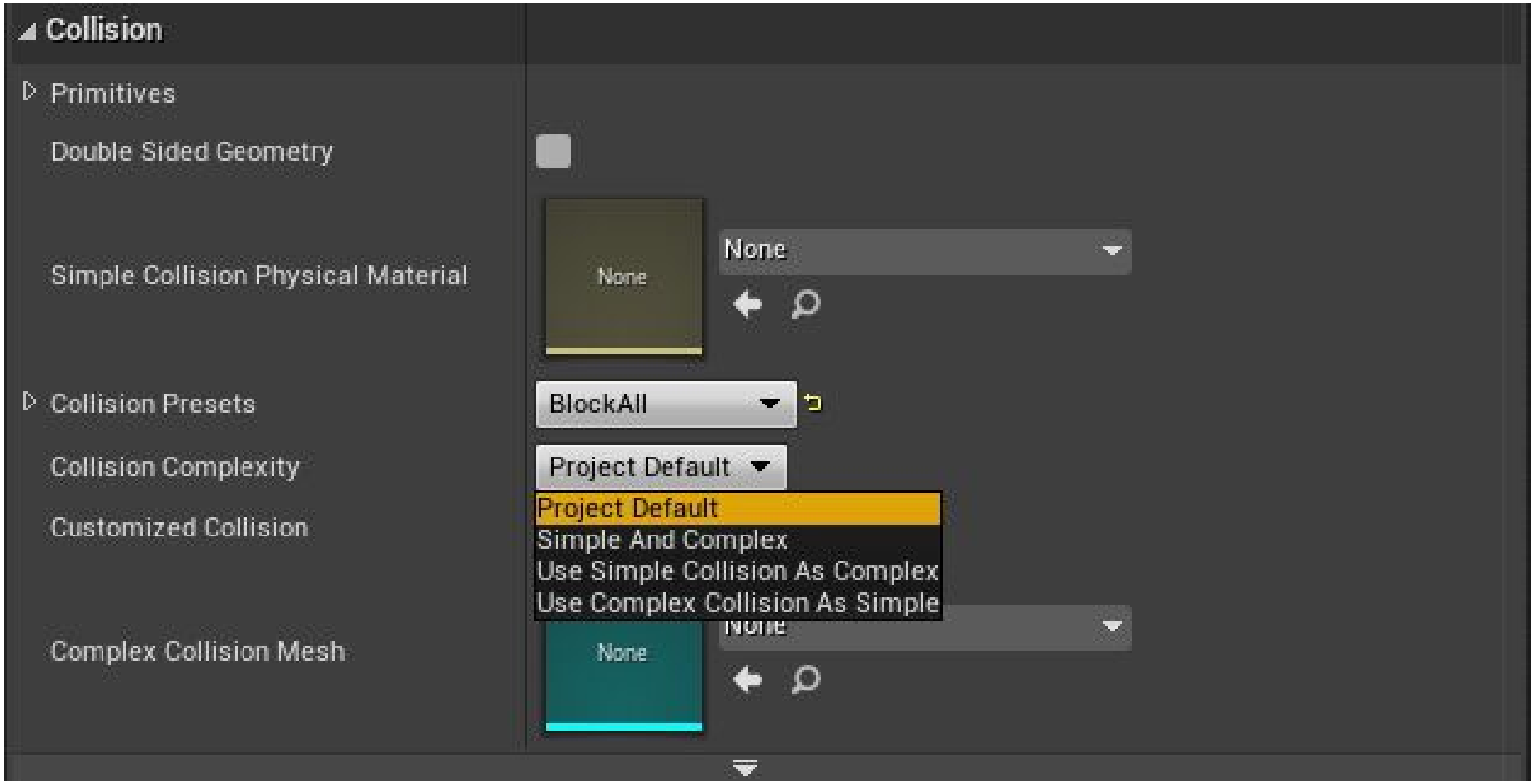

NO OVERLAP
ONE AXIS OVERLAP
TWO AXES: COLLISION

Two ways

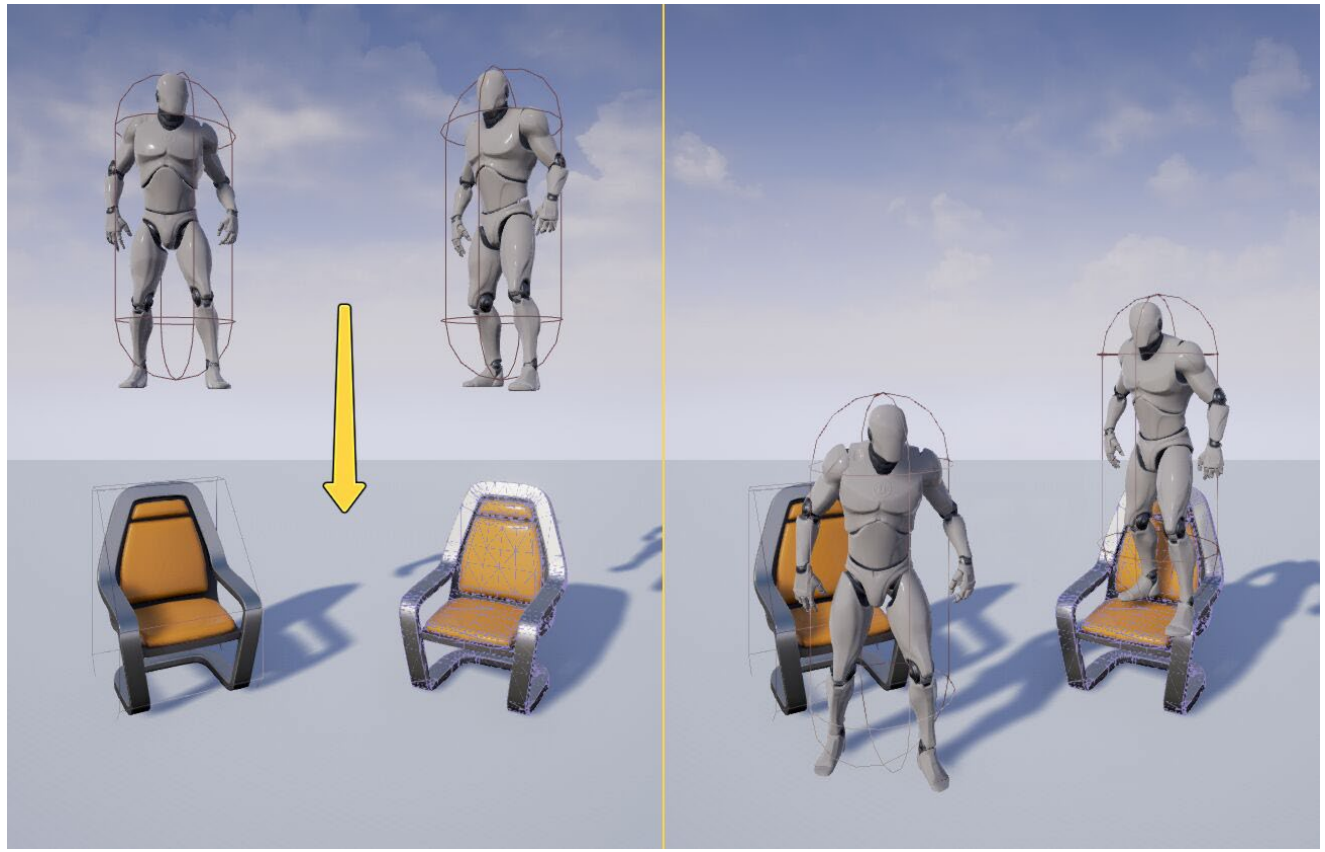
Bounding Volume

Space Partitioning

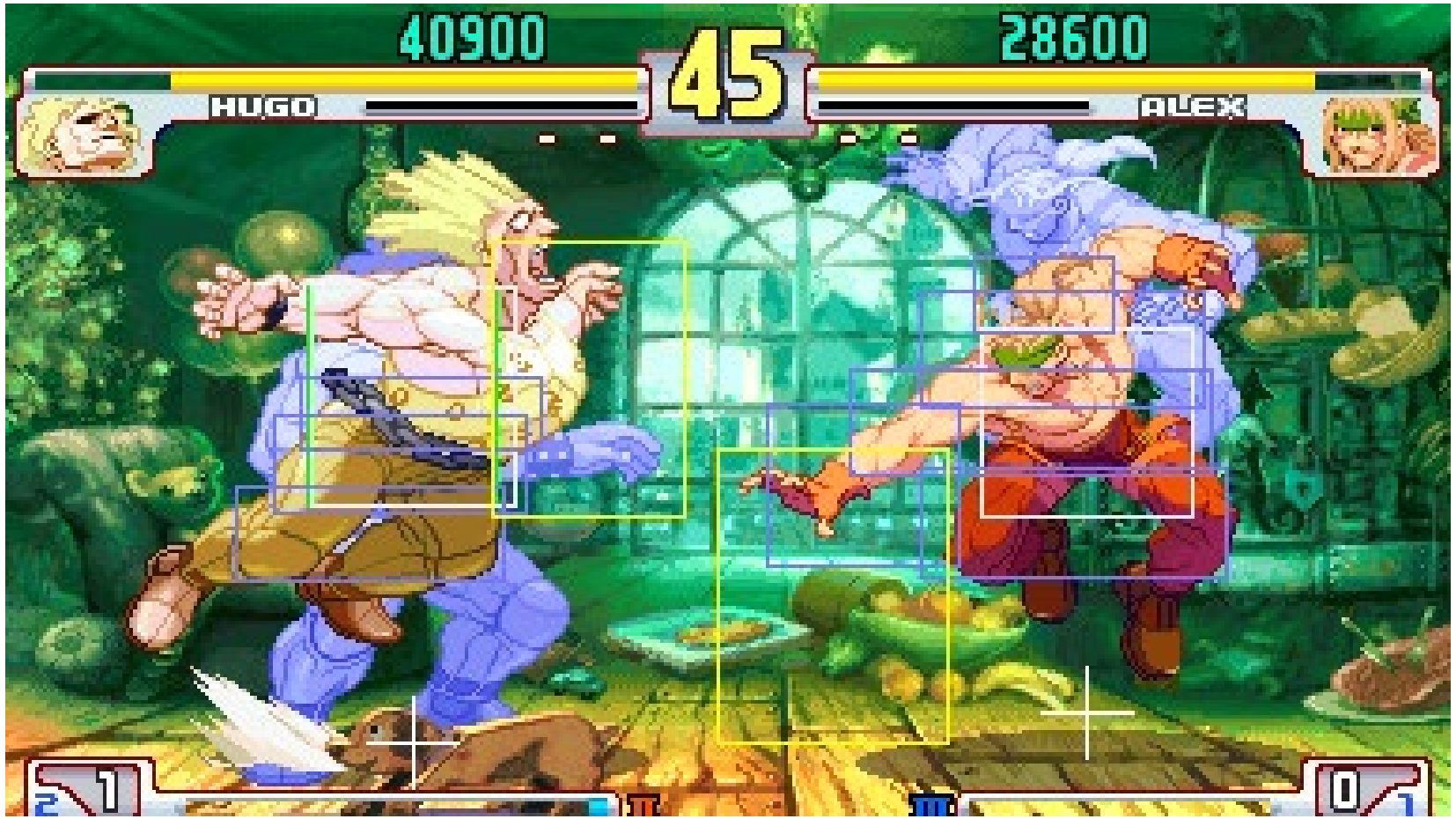
Or enable Collision Complexity and use complex collision as simple



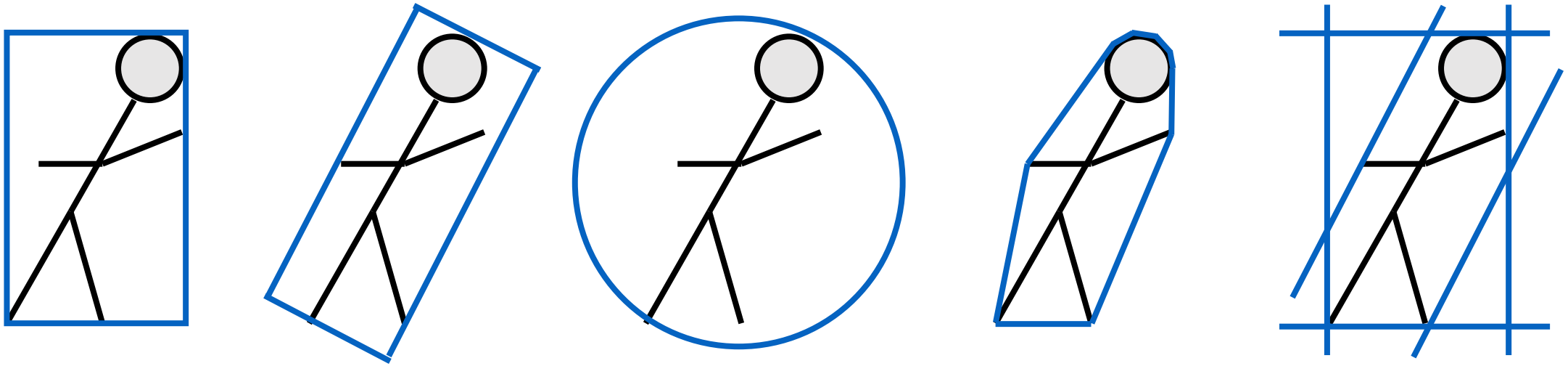
- For example, in the image below the chair on the left has **simple collision**, and when the Pawn above it falls onto it, it will slide off the large angled surface that covers the seat. However; the chair on the right is using **Use Complex Collision As Simple**, and when the Pawn above it falls, it will land on the seat of the chair and stay there.
-



Bounding Volume



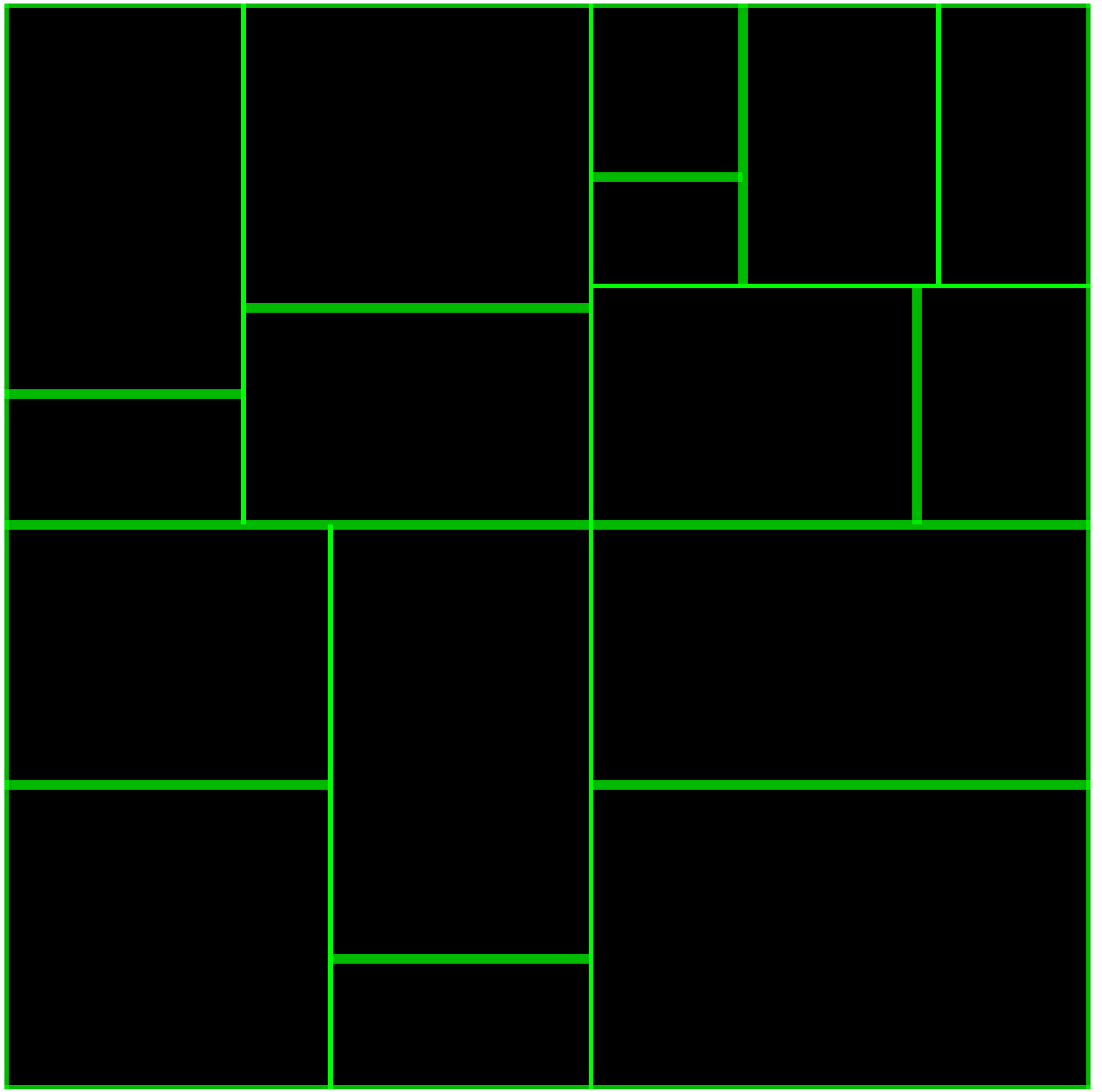
Bounding Volume



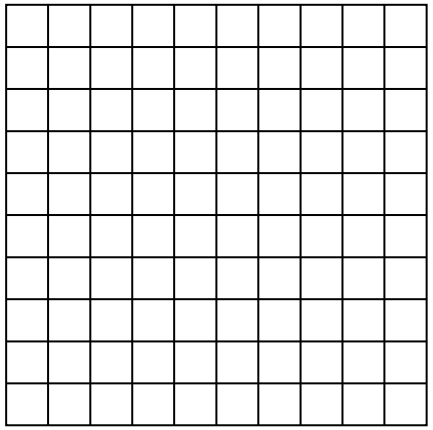
Type of Bounding Volumes

- Spheres
- Ellipsoids
- Axis-Aligned Bounding Boxes (AABB)
- Oriented Bounding Boxes (OBBs)
- Convex Hulls
- k-Discrete Orientation Polytopes (k-dop)
- Spherical Shells
- Swept-Sphere Volumes (SSVs)
  - Point Swept Spheres (PSS)
  - Line Swept Spheres (LSS)
  - Rectangle Swept Spheres (RSS)
  - Triangle Swept Spheres (TSS)

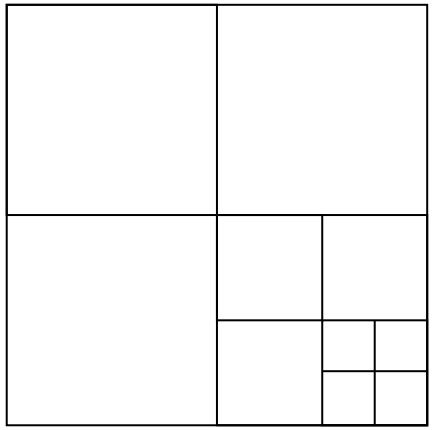
Space Partitioning



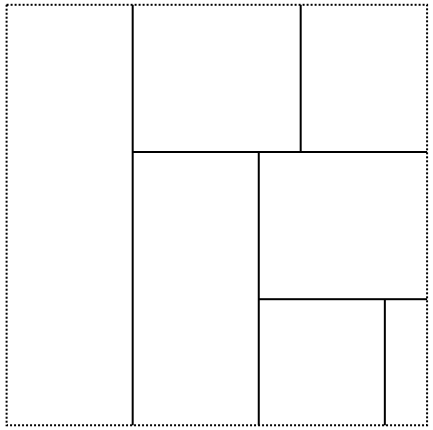
Space Partitioning



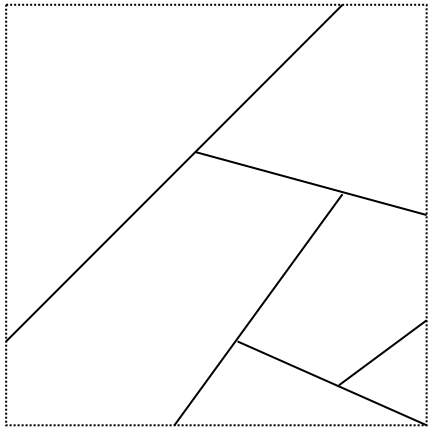
Uniform subdivision



Quadtree



kD-tree



BSP-tree

What's the best choice?

Depends on the application

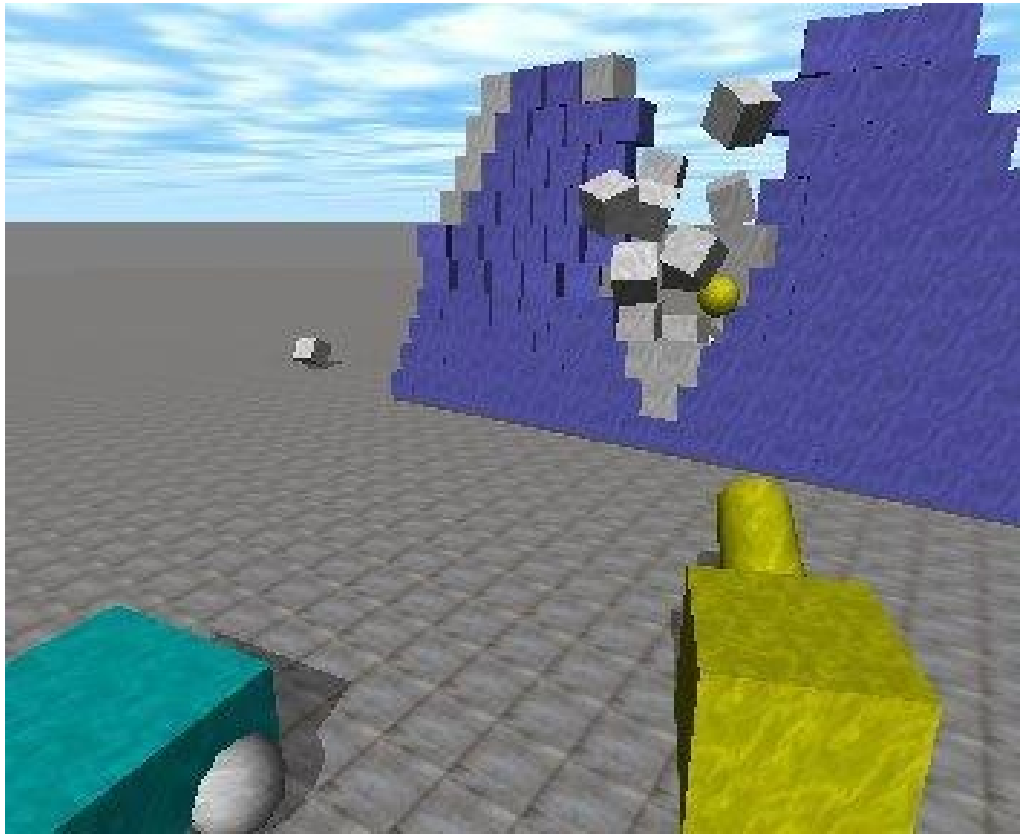
- trial and error?
- "Gut" feeling?
- Careful analysis based on a cost function?

Factors:

- Complexity of implementation
- Storage overhead
- Computational overhead
- Type of geometry: static or dynamic

Some Physic/Collision Library/Engine

ODE



OPEN DYNAMICS ENGINE™

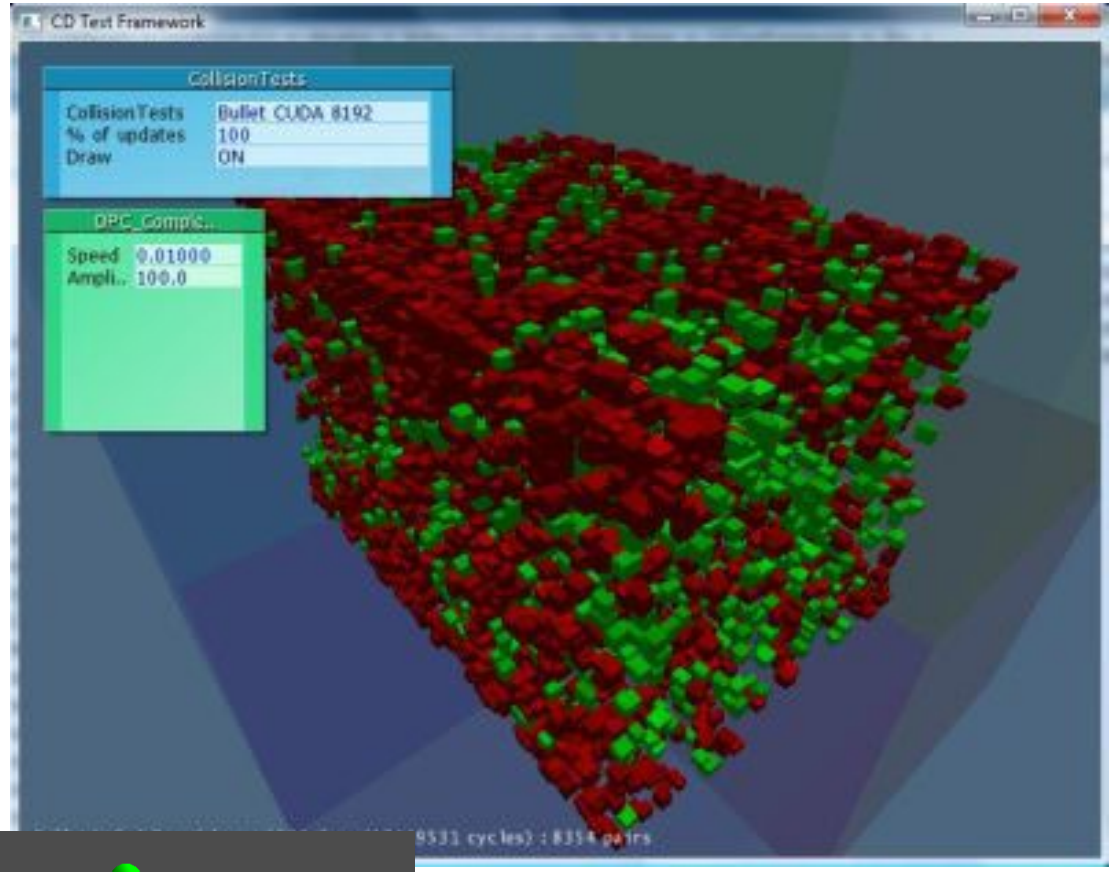
ODE stands for “Open Dynamics Engine ” (http://www.ode.org). As its name implies, ODE is an open-source collision and rigid body dynamics SDK. Its feature set is similar to a commercial product like Havok.

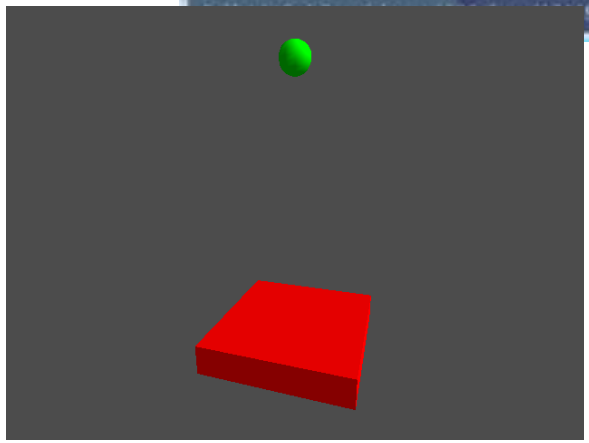
Its benefits include being free (a big plus for small game studios and school projects!) and the availability of full source code (which makes debugging much easier and opens up the possibility of modifying the physics engine to meet the specific needs of a particular game).

Bullet

Bullet is an open-source collision detection and physics library used by both the game and film industries. Its collision engine is integrated with its dynamics simulation, but hooks are provided so that the collision system can be used standalone or integrated with other physics engines.

It supports *continuous collision detection* (CCD)—also known as *time of impact* (TOI) collision detection—which as we'll see below can be extremely helpful when a simulation includes small, fast-moving objects.

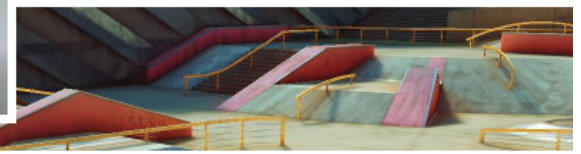




TrueAxis

Find us on Facebook YouTube Channel

TRUE AXIS



Home | Jet Car Stunts | Jet Car Stunts 2 | Space Tripper | True Skate | True Axis Physics SDK

Available Now

Awards

App Store BEST OF 2012

POCKET GAMER AWARDS 2013 NOMINEE

iTunes Rewind 2009 : Apps Best of 2009

GAME OF THE MONTH

IGN.COM EDITORS' CHOICE award

2009 Best App Ever Awards 3rd Place

BEST APPS OF 2009 AppSmile.com

touchgen 2009 winner

POCKET GAMER AWARDS

POCKET GAMER biz Top 50

Available on the App Store

ANDROID APP ON Google play

amazon apps Available at amazon

XBOX LIVE on Windows Phone

News

2014 June 26: New pa
True Skate becomes the
League Skateboarding r

Read more at toucharca

2014 Jue 25: True Ska
True Skate hits the Ama
Store.

2014 Jan 02: Jet Car S
on iOS!
Get it on the App Store

2013 Dec 23: Jet Car S
Date Anounced.
JCS2 is coming to iOS o
2014! Android will follow

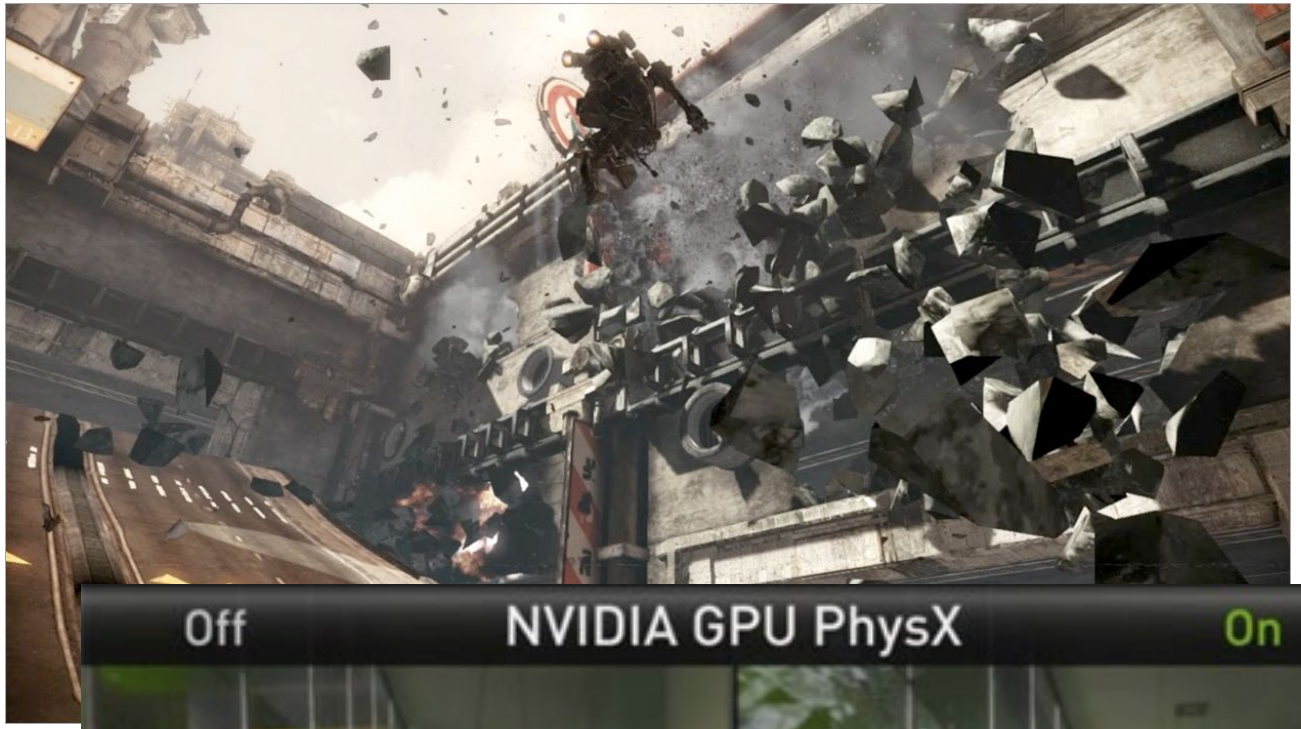
2013 Dec 19: A long c
Jet Car Stunts 2 is nearl
release, it looks like a c
but there is a lot of cont
still don't have a release
can't release in 2013, e
2014. Also, there are 4
coming to True Skate. E
January 2014, then mor
afterwards. We have be
up to date on our faceb

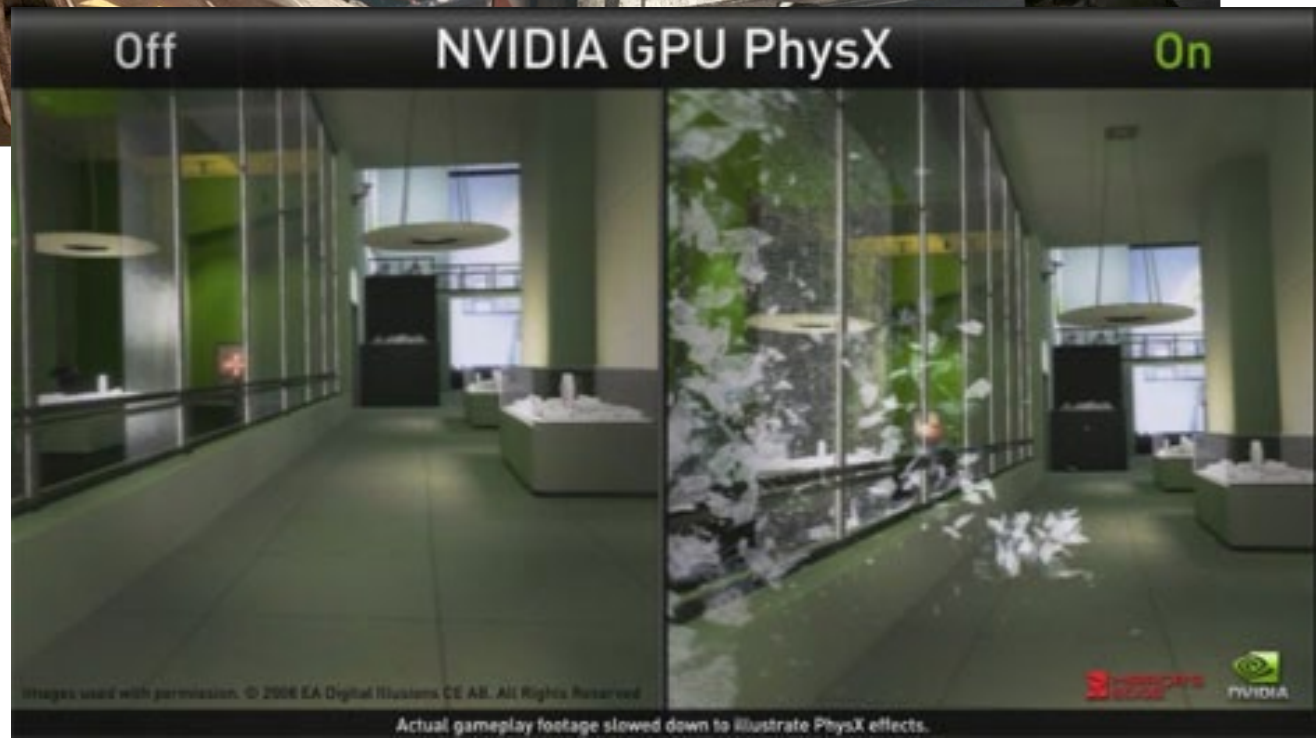
2013 Jun 19: True Ska
on Android
Find True Skate for and

PhysX

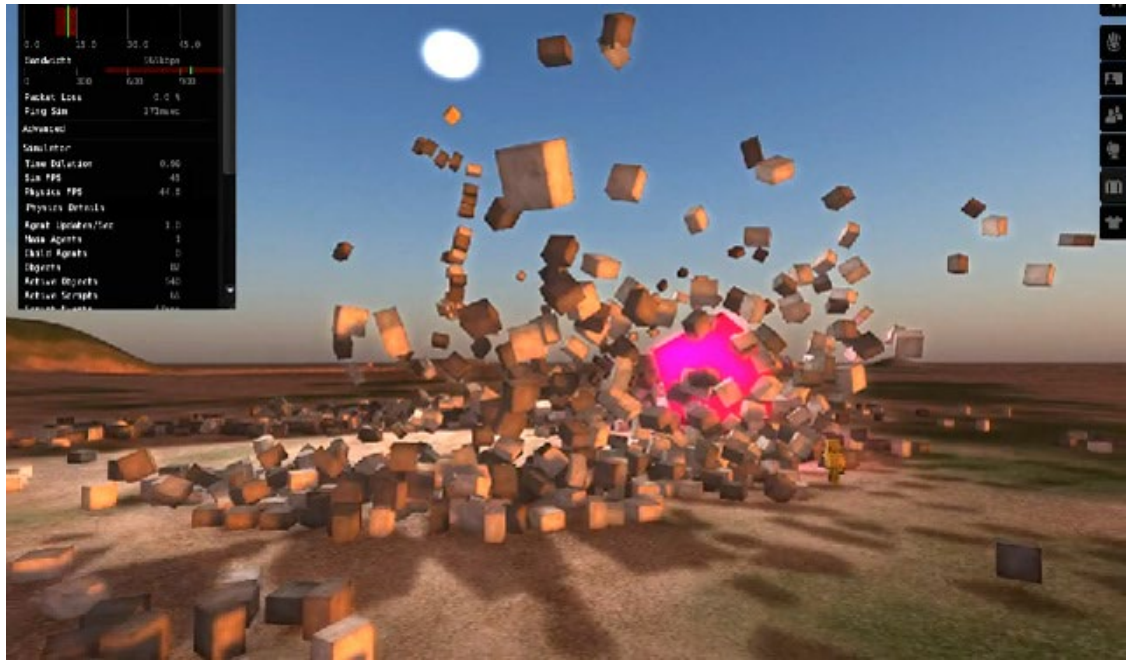
PhysX started out as a library called Novodex , produced and distributed by Ageia as part of their strategy to market their dedicated physics coprocessor.

It was bought by NVIDIA and is being retooled so that it can run using NVIDIA’s GPUs as a coprocessor.



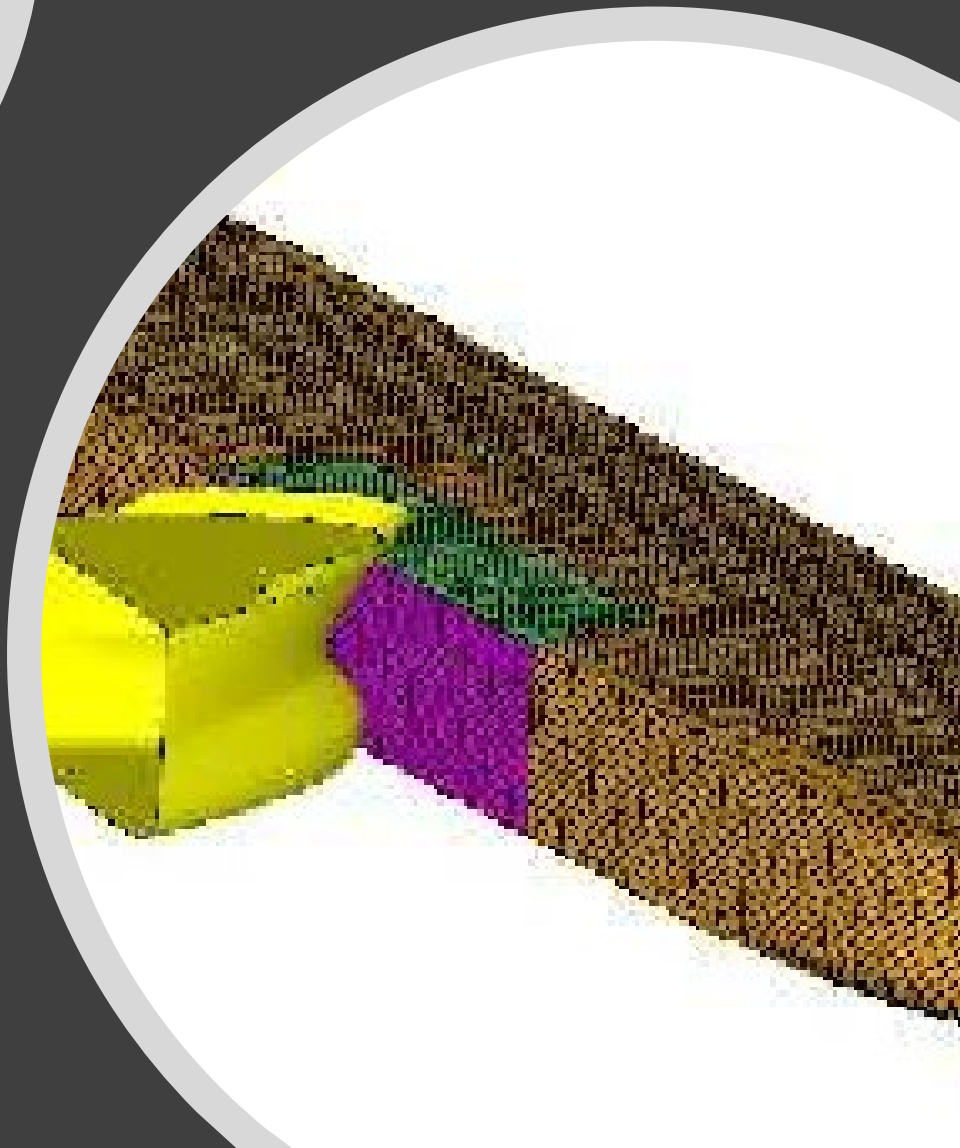


Havok is the gold standard in commercial physics SDKs, providing one of the richest feature sets available and boasting excellen performance characteristics on all supported platforms. (It's also the most expensive solution.)

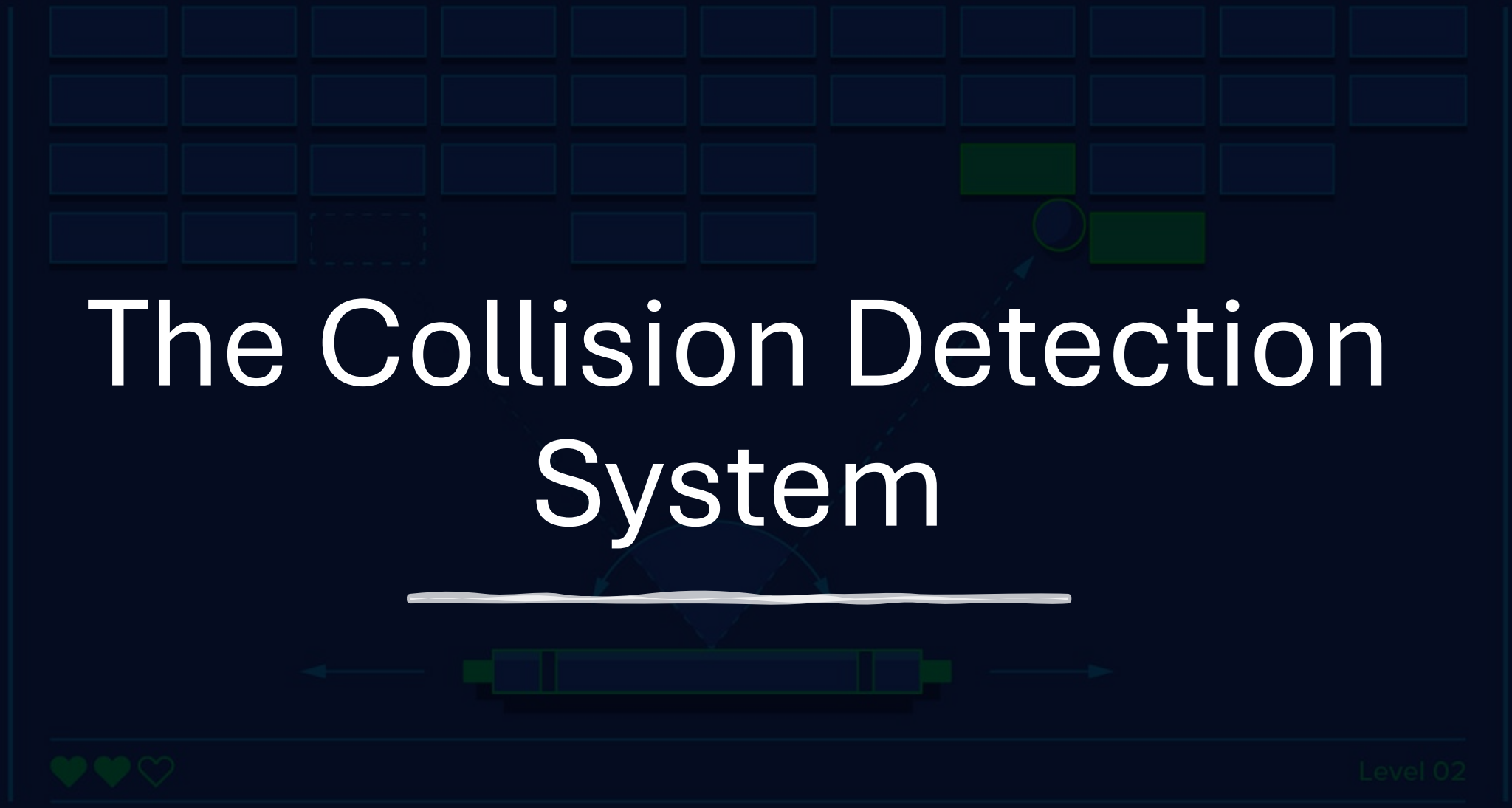


Havok is comprised of a core collision/physics engine, plus a number of optional add-on products including a vehicle physics system, a system for modeling destructible environments, and a fully featured animation SDK with direct integration into Havok's rag doll physics

Why we need collision detection?
- Games



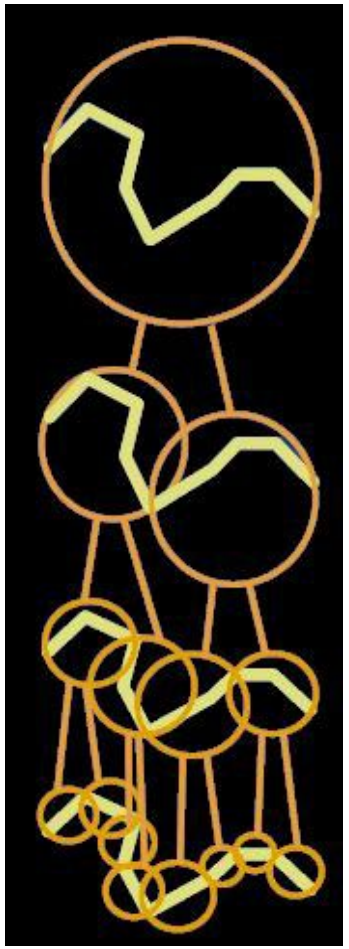
Why we need collision detection?
- Simulation and Animation



The Collision Detection System

Bounding Volumes

- Objects are often not colliding
  - Need fast reject for this case
  - Surround with some bounding object
    - Like a sphere
- Why stop with one layer of rejection testing?
  - Build a bounding volume hierarchy (BVH)
    - A tree



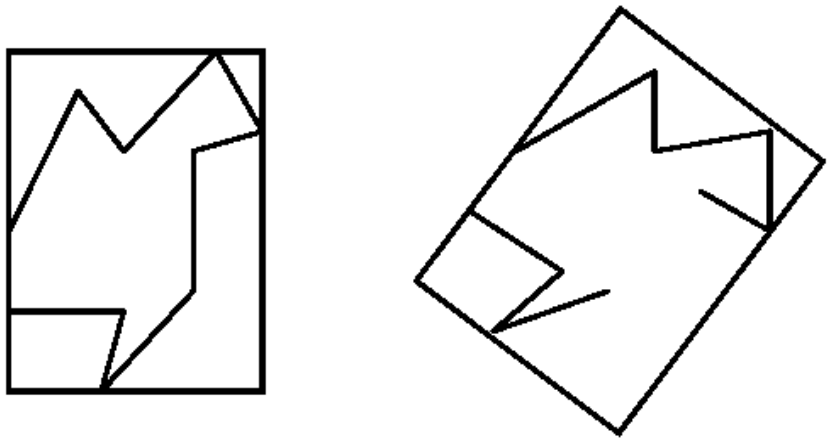
Type of Bounding Volumes

- Spheres
- Ellipsoids
- Axis-Aligned Bounding Boxes (AABB)
- Oriented Bounding Boxes (OBBs)
- Convex Hulls
- k-Discrete Orientation Polytopes (k-dop)
- Spherical Shells
- Swept-Sphere Volumes (SSVs)
  - Point Swept Spheres (PSS)
  - Line Swept Spheres (LSS)
  - Rectangle Swept Spheres (RSS)
  - Triangle Swept Spheres (TSS)

BV Choices

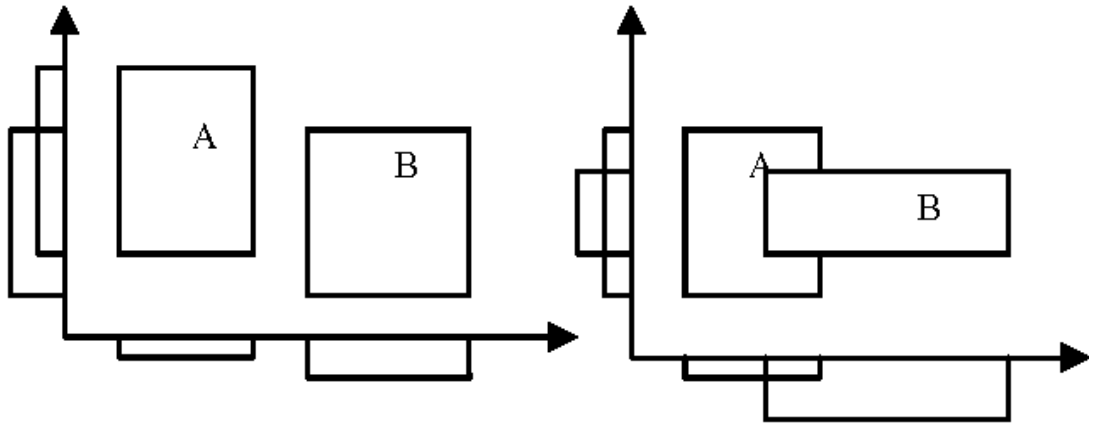
- OBB or AABB
  - OBB slightly better for close proximity
  - AABB better for handling deformations

gure 2.



(a) AABB (b) OBB

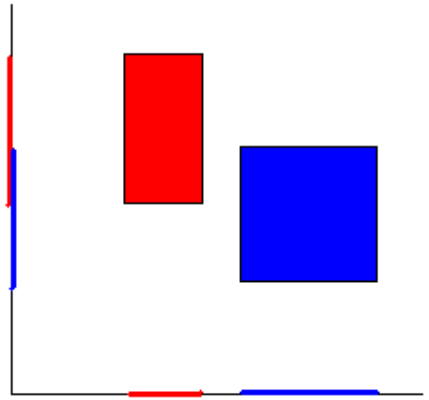
Figure 1. AABB and OBB bounding box

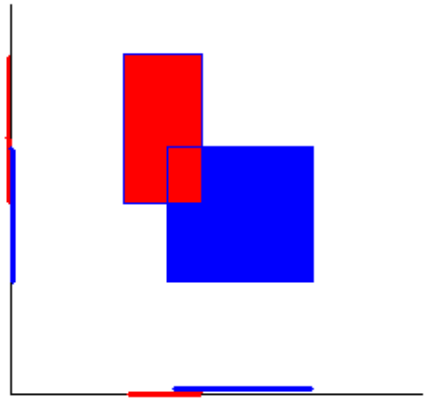


(a)A and B are disjoint (b)Aand B are intersectant

Axes Aligned Bounding Box (AABB)

Collision test: project BBs onto coordinate axes. If they overlap on each axis, the objects collide.

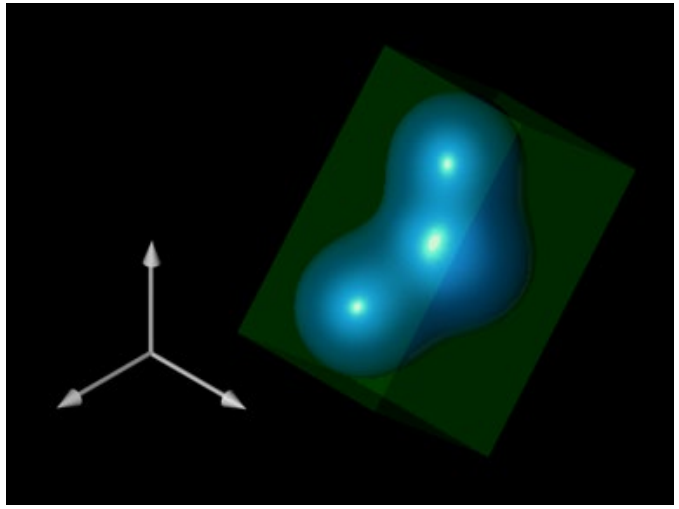




Oriented Bounding Box (OBB)

Align box to object such that it fits optimally in terms of fill efficiency

Computationally expensive
Invariant to rotation
Complex intersection check



Axis-Aligned Bounding Box (AABB)

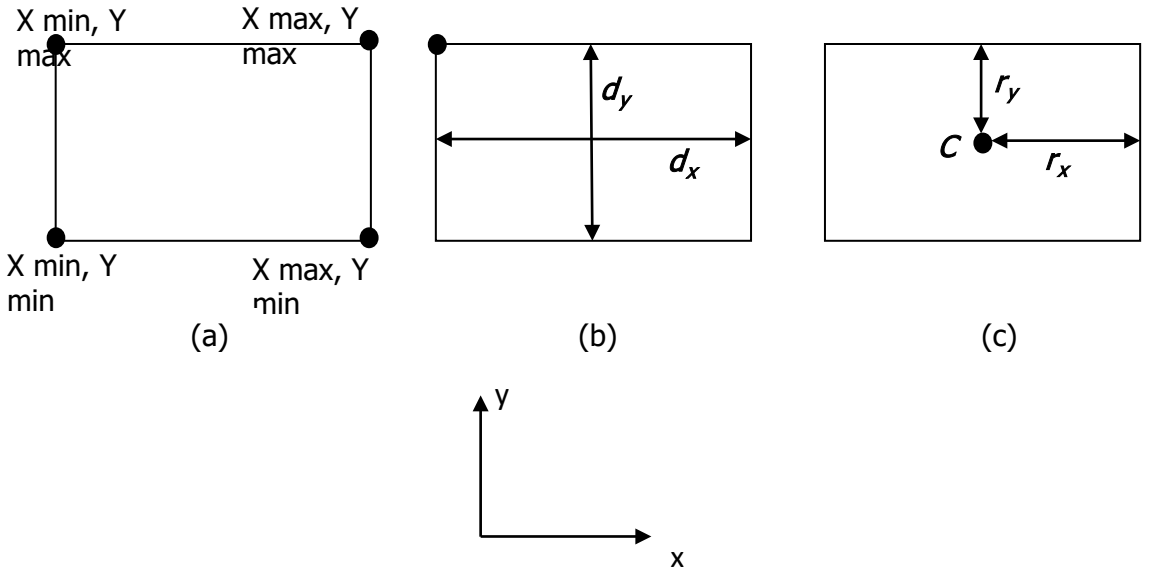
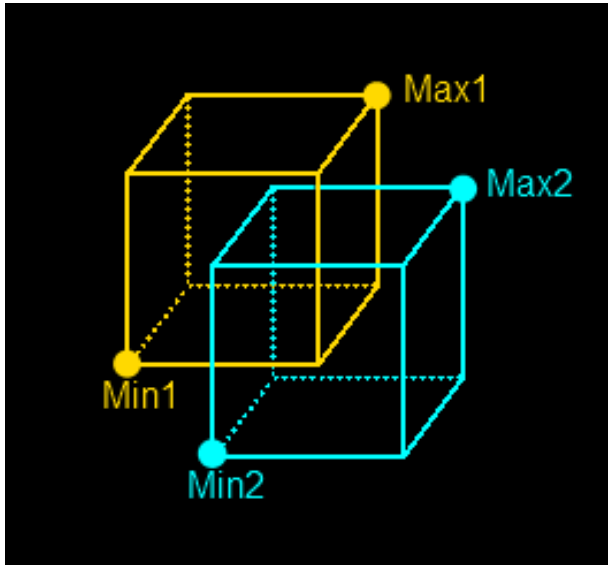
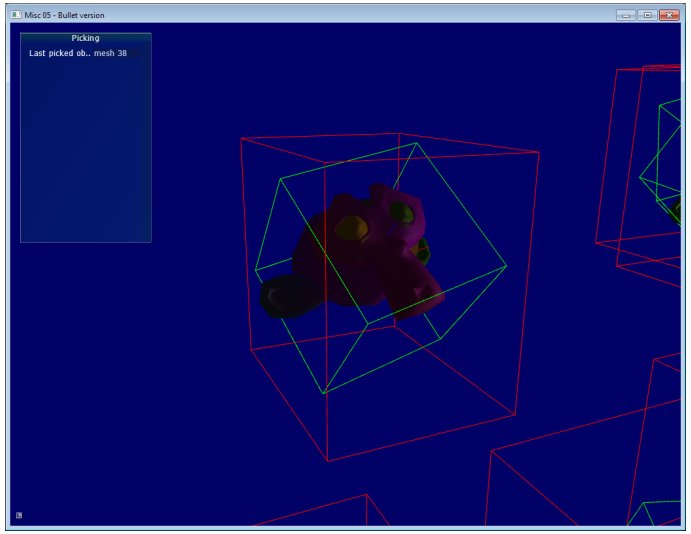


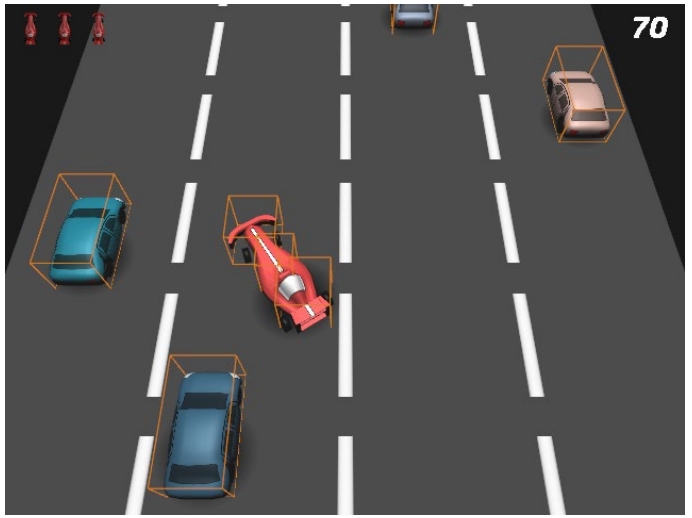
Figure 2.15: AABB representation: (a) Min-max; (b) Min-widths; (c) Centre-radius. X and Y coordinates system is shown in order to visualize the corresponding AABB into Cartesian coordinate in this example (source (Ericson, 2004)).

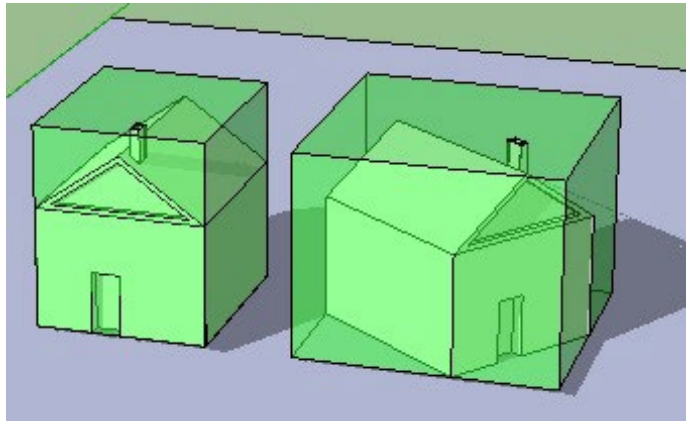
Axis-aligned bounding box (AABB) is the most common BV used for collision detection. It is a rectangular box with six surfaces which normally parallel with the standard axes. Also it can be represented as two points' minimum and maximum (min-max) in world coordinate space.

AABB


Max1
Max2
Min1
Min2


Misc 05 - Bullet version
Picking
Last picked ob.. mesh 38


70

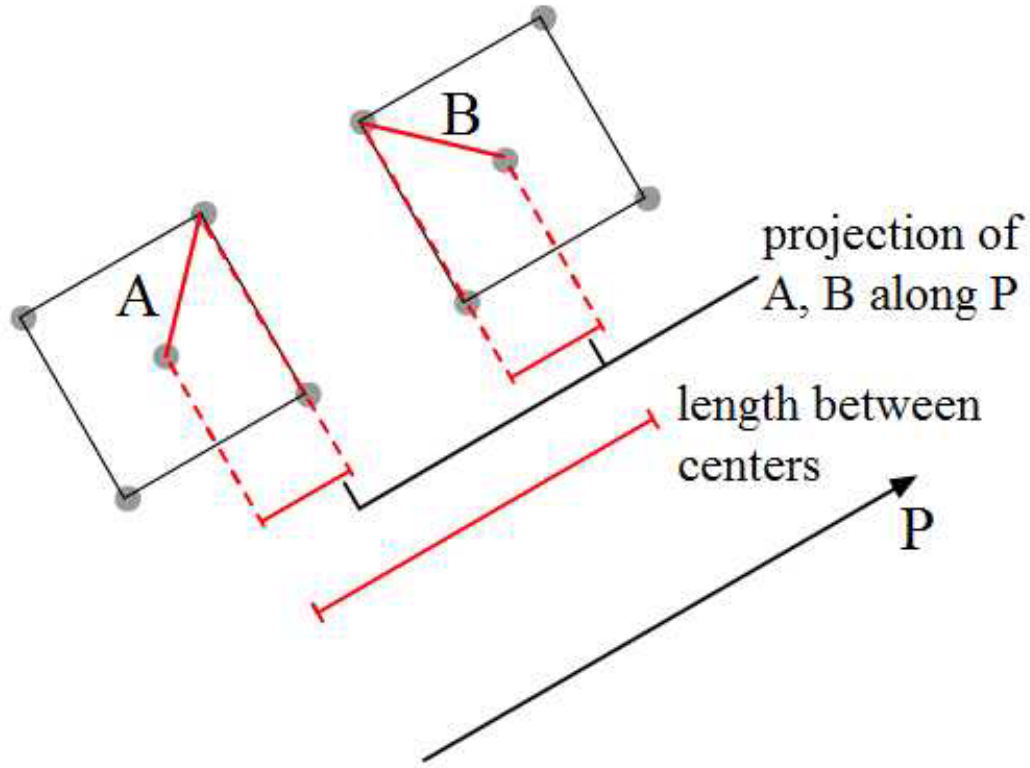


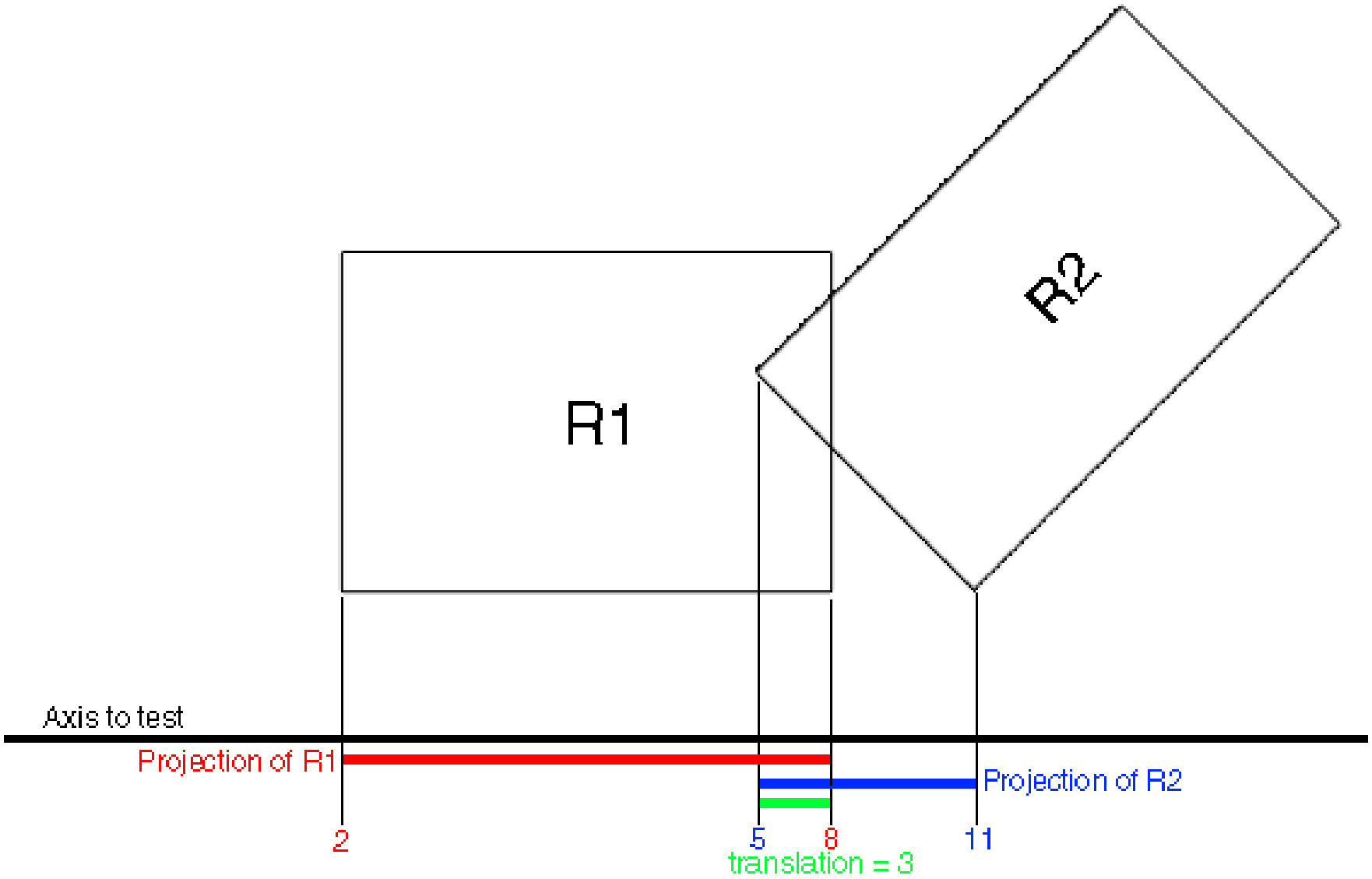
Bounding Volumes

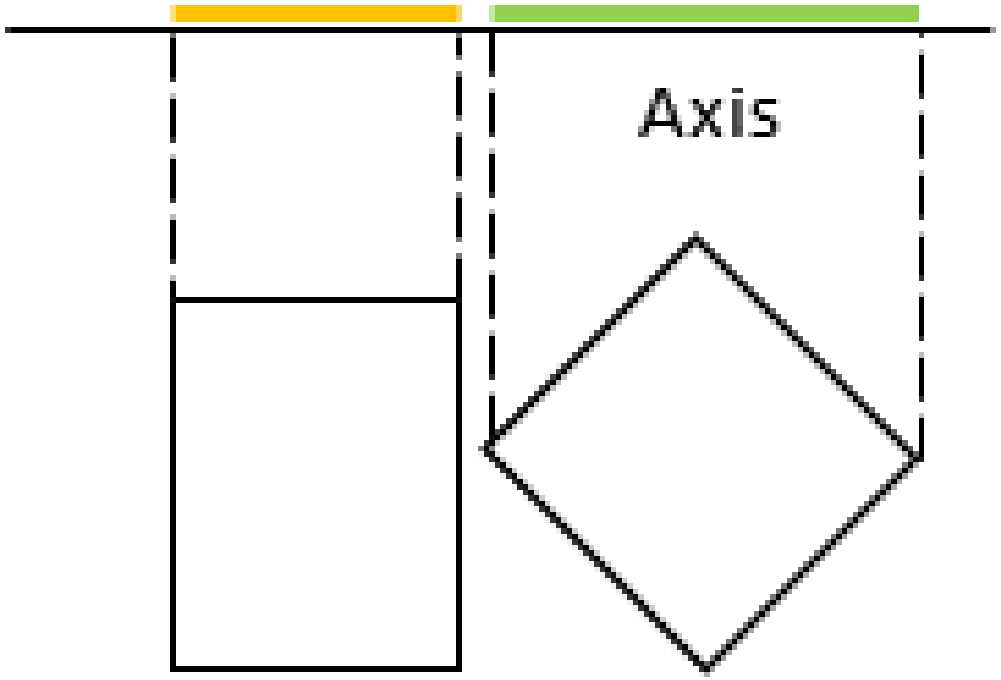
- The overlap test is based on the

- Separating Axes Theorem
- (S. Gottschalk. Separating axis theorem. Technical Report TR96-024,Department
- of Computer Science, UNC Chapel Hill, 1996)

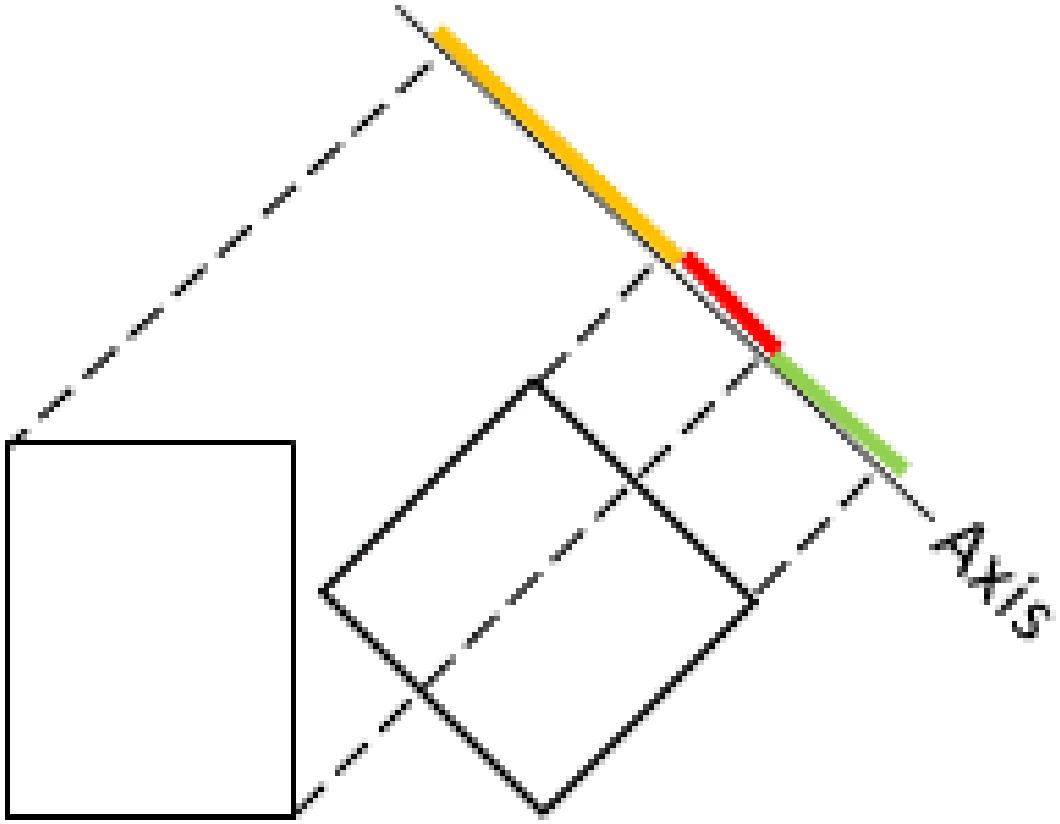
- Two convex polytopes are disjoint iff there exists a separating axis orthogonal to a face of either polytope or orthogonal to an edge from each polytope.



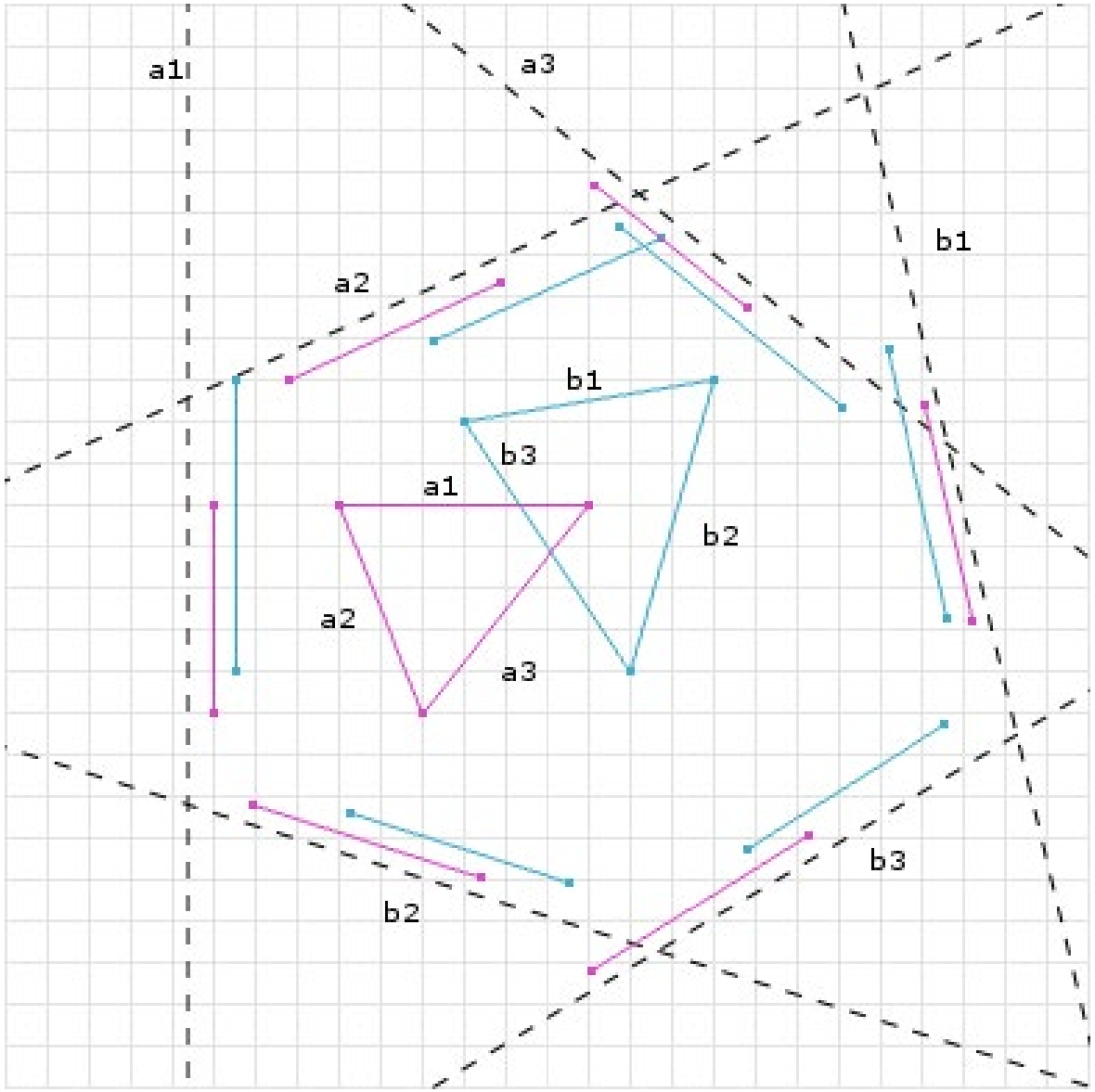

R1
R2
Axis to test
Projection of R1
Projection of R2
2
5
8
11
translation = 3



No overlap on this axis

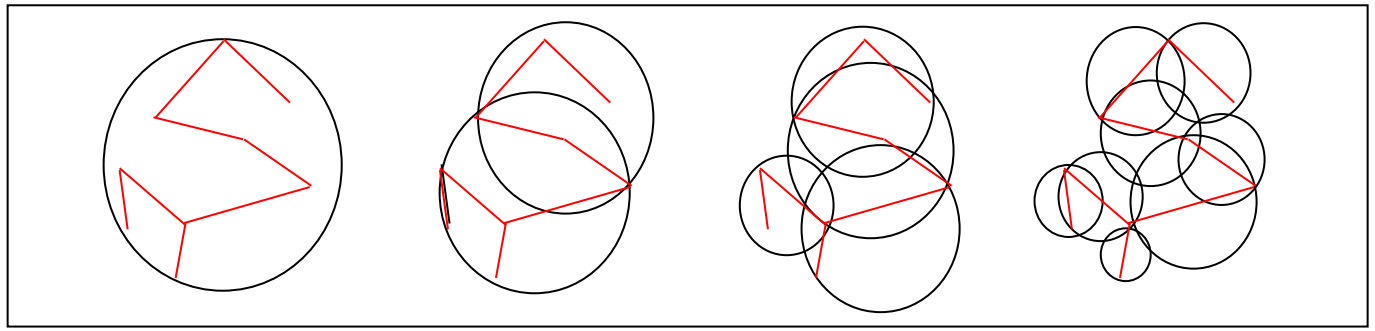


Intersection on this axis


a1
a3
b1
a2
b1
b3
a1
b2
a2
a3
b3
b2

Bounding Volume Hierarchies

- Model Hierarchy:
  - each node has a simple volume that bounds a set of triangles
  - children contain volumes that each bound a different portion of the parent's triangles
  - The leaves of the hierarchy usually contain individual triangles
- A binary bounding volume hierarchy:



Introduction

- Bounding Volume Hierarchies vs. Spatial Partitioning
  - What are they and how do they compare?
- Motivation: *Need for Speed!*
  - Demonstration through applications:
    View-frustum culling, ray-tracing, collision detection
- How can hierarchies help?
  - Apply to example applications
- Building bounding volume hierarchies
- Building spatial partitionings
- What's the best choice?
- Can we do better?

What are they? How do they Compare?

Bounding Volume Hierarchies

- Hierarchical object representation
- Object subdivision
- Hierarchical clustering of objects
- Object levels of detail
- Classifies regions of space around objects

Examples:

- OBB-trees
- AABB-trees
- Sphere-trees
- k-DOPs

Spatial Partitioning

- Hierarchical spatial representation
- Spatial subdivision
- Hierarchical clustering of space
- Spatial levels of detail
- Classifies objects around regions of space

Examples:

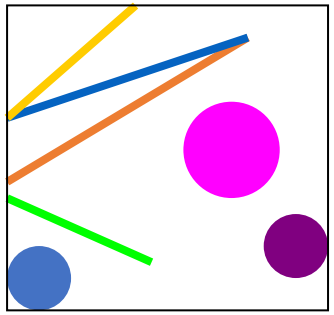
- Uniform grids
- Quadtrees & Octrees
- BSP-trees
- kD-trees

Examples

Bounding Volume Hierarchies

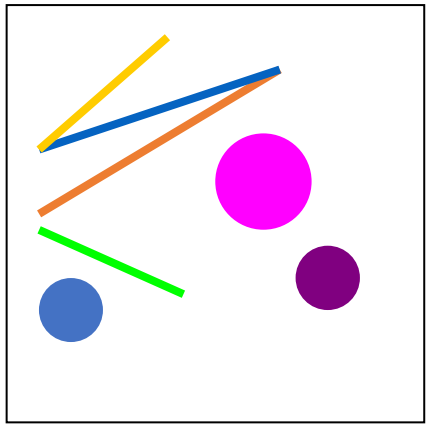
Tightly fits objects

Redundant spatial representation



Spatial Partitioning

- Tightly fills space
- Redundant object representation

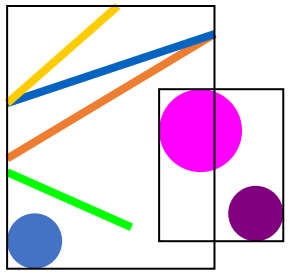


Examples

Bounding Volume Hierarchies

Tightly fits objects

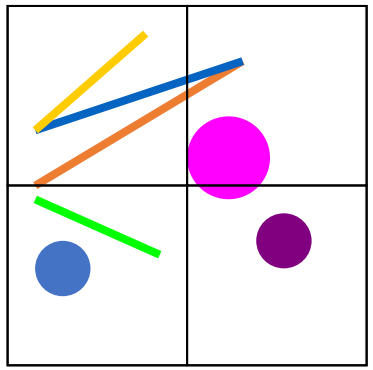
Redundant spatial representation



Volumes overlap multiple objects

Spatial Partitioning

- Tightly fills space
- Redundant object representation



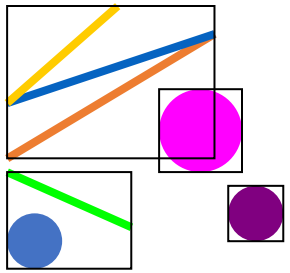
Objects overlap multiple volumes

Examples

Bounding Volume Hierarchies

Tightly fits objects

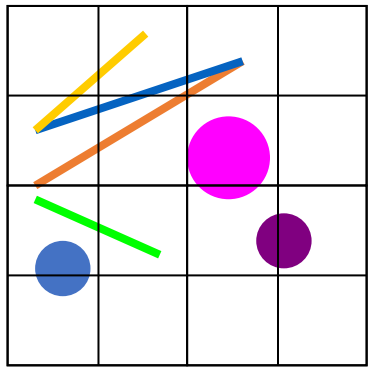
Redundant spatial representation



Volumes overlap multiple objects

Spatial Partitioning

- Tightly fills space
- Redundant object representation



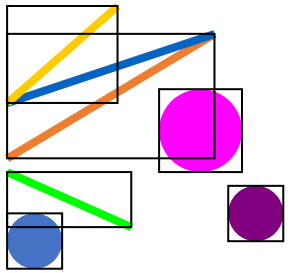
Objects overlap multiple volumes

Examples

Bounding Volume Hierarchies

Tightly fits objects

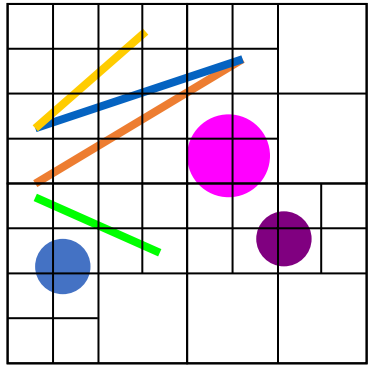
Redundant spatial representation



Volumes overlap multiple objects

Spatial Partitioning

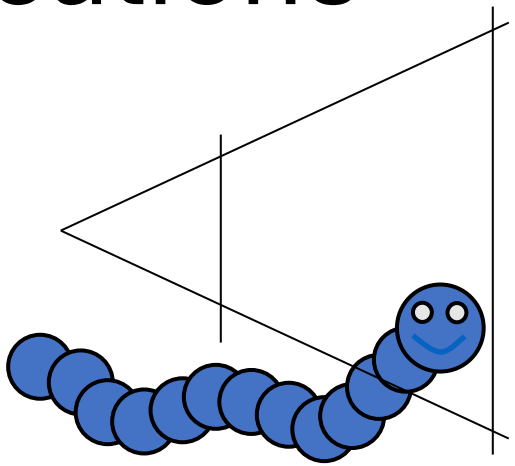
- Tightly fills space
- Redundant object representation



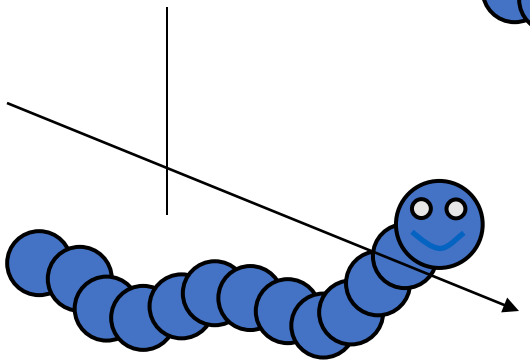
Objects overlap multiple volumes

Motivation: Example Applications

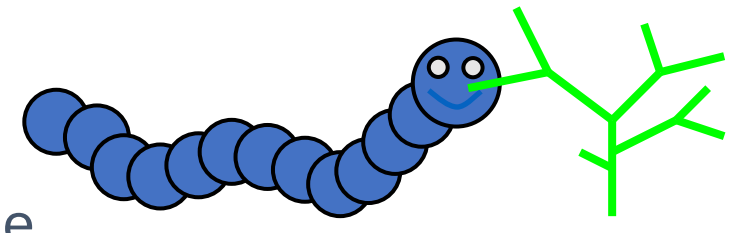
View-frustum culling O(n)



Ray-tracing O(n) per ray



Collision detection $O(n^2)$



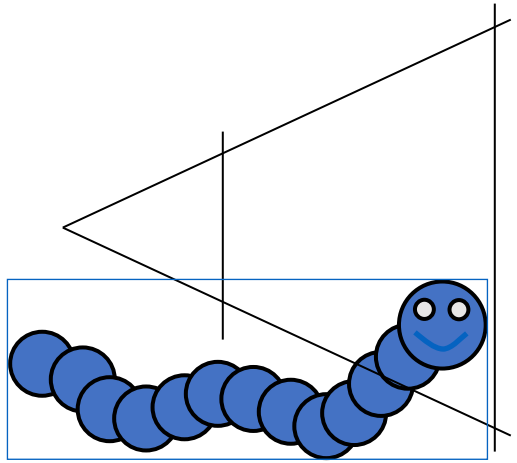
n: number of objects in the scene

How do we speed it up?

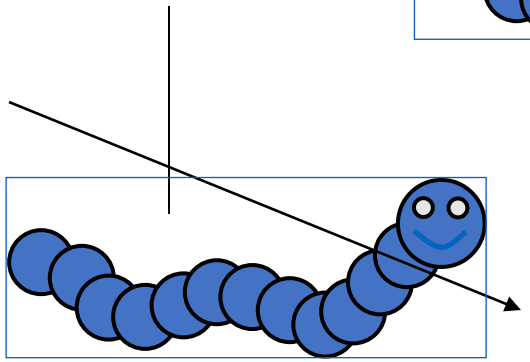
- More efficient intersection calculations
- Avoid intersection calculations
  - Make a single intersection calculation to decide
    for an entire cluster of objects or space
  - Cluster hierarchically

How can bounding volume hierarchies help?

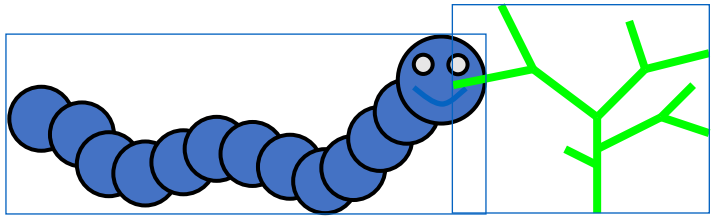
View-frustum culling



Ray-tracing

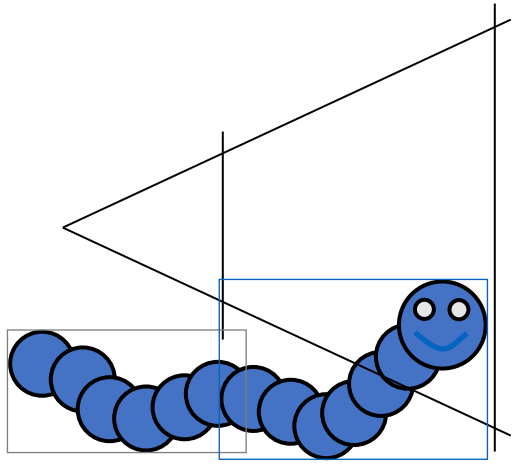


Collision detection

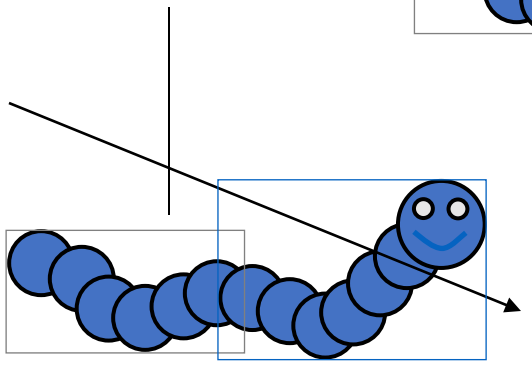


How can bounding volume hierarchies help?

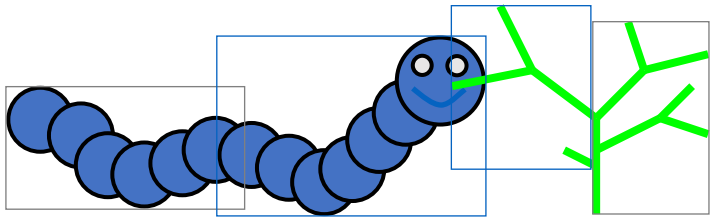
View-frustum culling



Ray-tracing

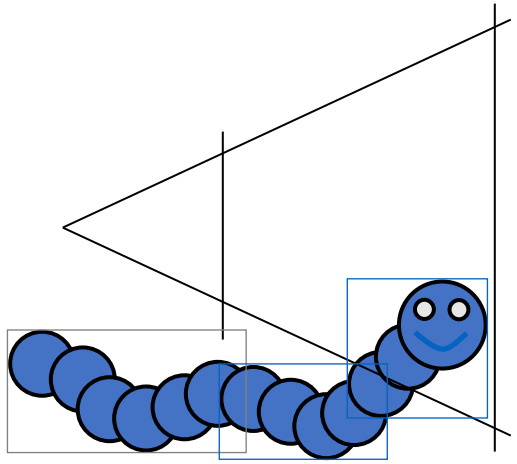


Collision detection

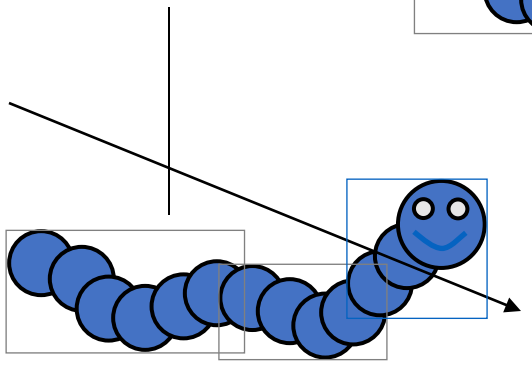


How can bounding volume hierarchies help?

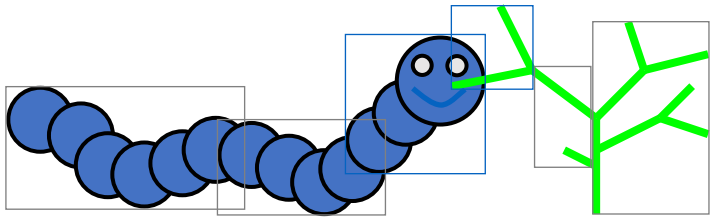
View-frustum culling



Ray-tracing

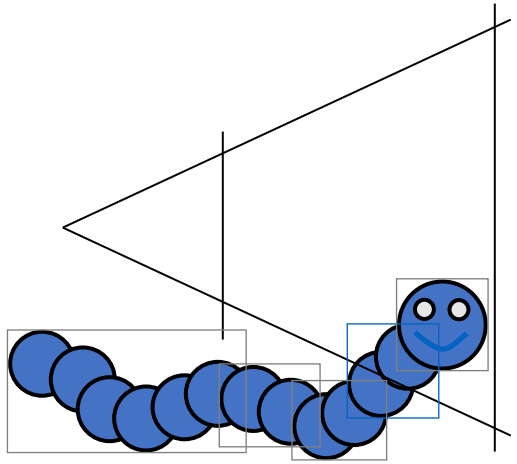


Collision detection

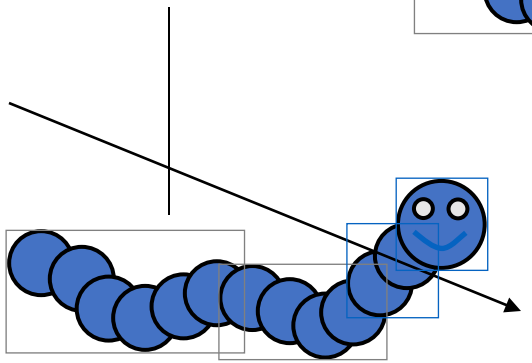


How can bounding volume hierarchies help?

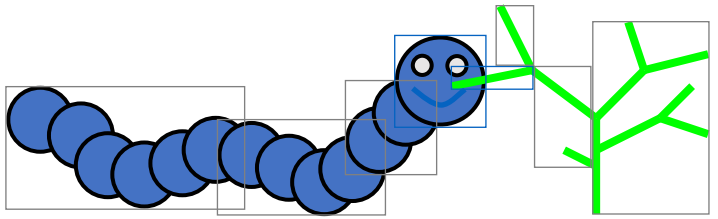
View-frustum culling



Ray-tracing



Collision detection

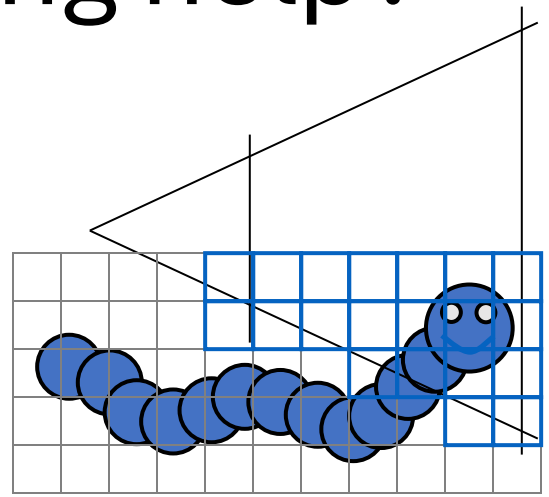


How can bounding volume hierarchies help?

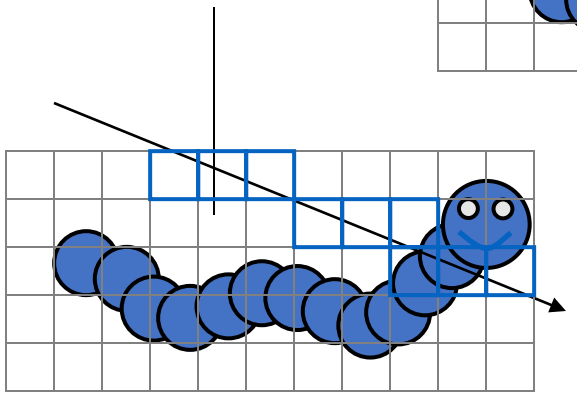
Logarithmic search for intersecting primitives!

How can spatial partitioning help?

View-frustum culling

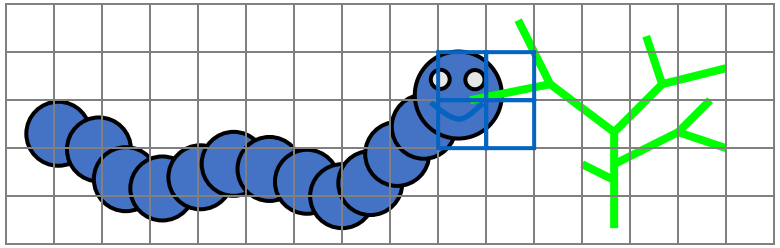


Ray-tracing



Uniform spatial partitioning

Collision detection



How can spatial partitioning help?

Performance varies for uniform partitioning, but hierarchical approaches also give logarithmic search for intersecting primitives!

What are the potential problems?

- What are the hidden costs?
  - When nothing intersects?
  - When nearly everything intersects?
  - What are the worst cases?
- Is it worth it?
- What applications get the most benefit?
- What about just using my modeling hierarchy?
  - Too shallow (not fine enough level of detail
  - Designed for object manipulation rather than minimizing intersections
  - Insensitive to actual positions of objects

Building Bounding Volume Hierarchies

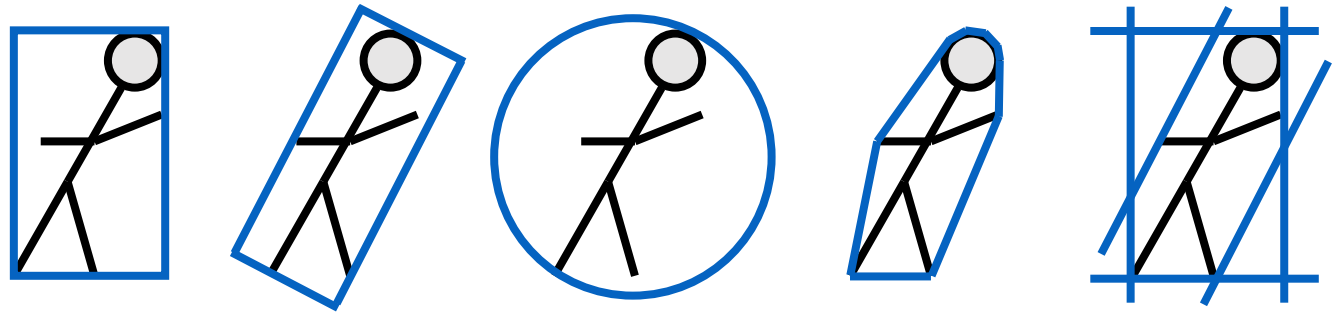
Choose a bounding volume type

- Axis-aligned bounding box (AABB)
- Oriented bounding box (OBB)
- Sphere
- Convex Hull
- k-DOP (discrete oriented polytope)

Choose a clustering strategy

- Top-down:
  how do we partition objects among children?
- Bottom-up:
  how do we find leaf clusters and merge into parents?

Bounding Volume Type



Intersection cost vs. tightness of fit vs. storage overhead vs. implementation complexity

How do we find the best fit for a particular bounding volume?

- AABBs and convex hulls are clear.
  What about spheres, k-DOPs, and OBBs?

How do we compare the quality of fit between different BVs?

- Min volume, min surface area, etc.

Hierarchical Clustering Strategy

Top-down: how do we partition objects among children?

splitting axis	split point
longest dimension, largest spread of objects, etc.	mean, median, largest gap, etc.

Hierarchical Clustering Strategy

Bottom-up: how do we find leaf clusters and merge into parents?

Leaf object clusters

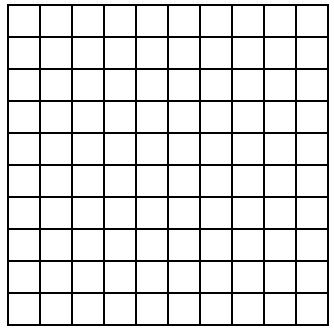
- single primitive, specific minimum size cluster, etc.

Merging children into parent

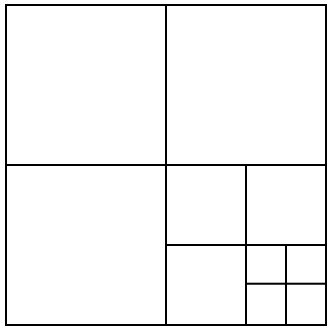
- Nearest neighbors: uniform subdivision, Voronoi diagram

Building Spatial Partitionings

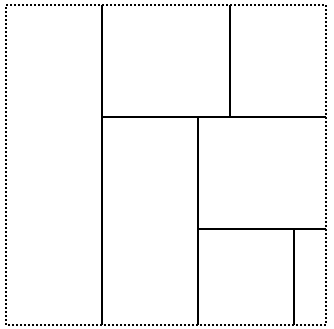
Decide how to recursively subdivide space (top-down)



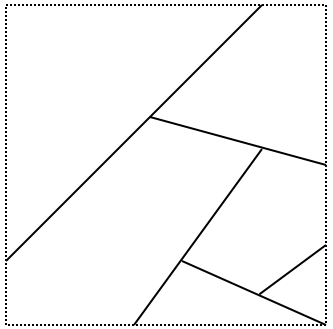
Uniform subdivision



Quadtree

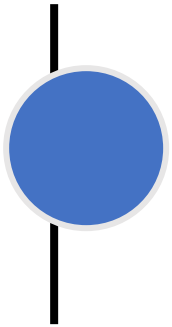


kD-tree



BSP-tree

Decide how to classify objects into regions of space with respect to partitioning plane



Store in both regions,
Store with partition, or
Split geometry

What’s the best choice?

- Depends on the application
  - trial and error?
  - “Gut” feeling?
  - Careful analysis based on a cost function?
- Factors:
  - Complexity of implementation
  - Storage overhead
  - Computational overhead
  - Type of geometry: static or dynamic

Can we do better?

Combining bounding volume hierarchies and spatial partitioning

- Examples:
  - Occlusion culling: octrees of BSP-trees
  - Radiosity: 3D BSP-trees of 2D BSP-trees

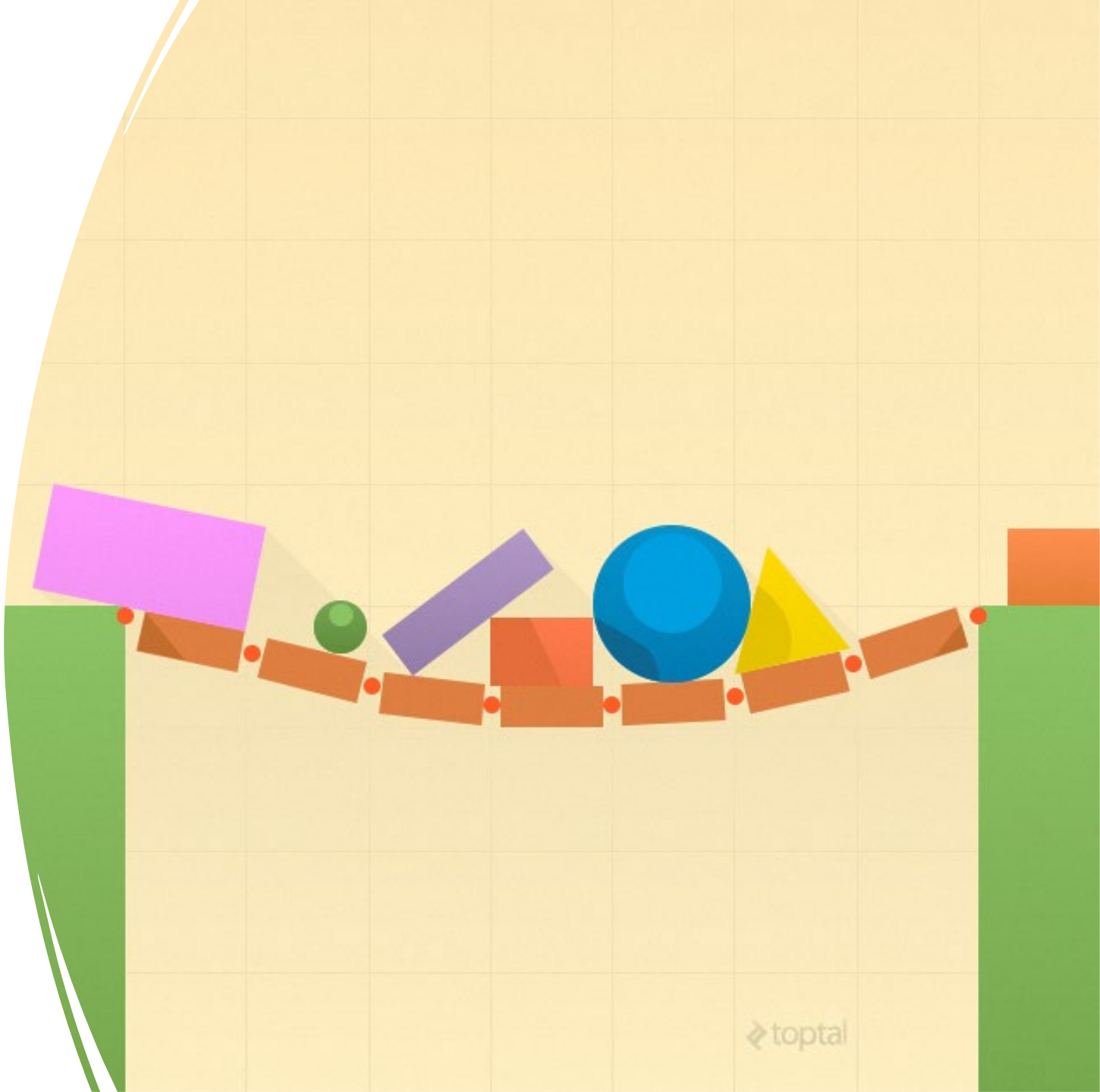
Hybrid bounding volume hierarchies

- adaptive nodes
- adaptive trees
- performance driven metrics

Self Study - Do find how AABB vs AABB overlap test is conducted

Study More

- https://www.toptal.com/game/video-game-physics-part-i-an-introduction-to-rigid-body-dynamics
- https://www.toptal.com/game/video-game-physics-part-ii-collision-detection-for-solid-objects
- https://www.toptal.com/game/video-game-physics-part-iii-constrained-rigid-body-simulation



References

- https://www.youtube.com/watch?v=4k__YjrXlus (use collision on nanite meshes for UE5)
- https://www.youtube.com/watch?v=YucYfUbazKY (create nanite meshes)