Collision Detection Algorithms

Brandan Roachell and Rob Bray
April 27, 2023

Test Questions

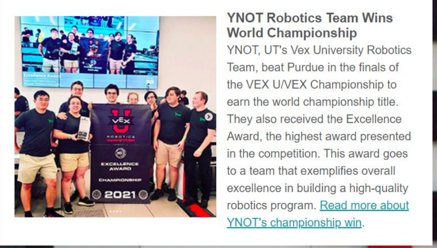
1. What is the major flaw of discrete collision detection?
2. What kind of *k*-DOP is a 3D AABB? (What is *k*?)
3. What is one way to optimize a BVH?

Brandan Roachell

- First (and only, hopefully) year master's student in Computer Science (5-Year BS/MS)
- Graduated from UTK in Dec. 2022 w/ BS in CS and a math minor
- Used to do undergrad research under Dr. Taufer, now a GTA for her data mining class
- Planning to do software engineering somewhere, but not really sure
- Interests:
  - piano
  - robotics
  - hiking?

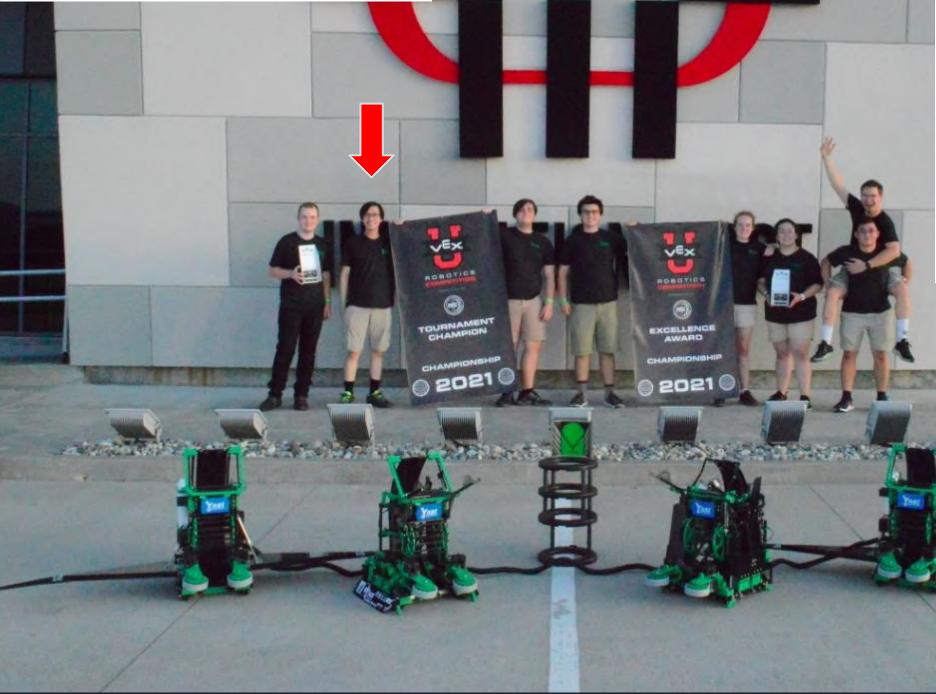
13 y/o me
(May 2015)

Chimney
Tops Trail
(Dec. 2021)

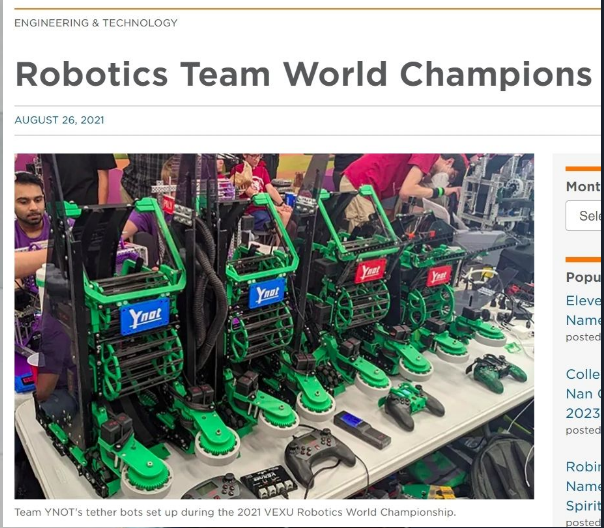


YNOT Robotics Team Wins World Championship

YNOT, UT's Vex University Robotics Team, beat Purdue in the finals of the VEX U/VEX Championship to earn the world championship title. They also received the Excellence Award, the highest award presented in the competition. This award goes to a team that exemplifies overall excellence in building a high-quality robotics program. Read more about YNOT's championship win.



2021 “World” Champs!
and Excellence Award



ENGINEERING & TECHNOLOGY

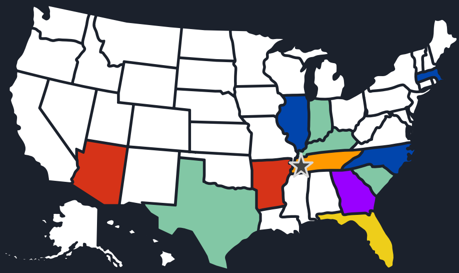
Robotics Team World Champions

AUGUST 26, 2021

Team YNOT's tether bots set up during the 2021 VEXU Robotics World Championship.

2022 Think Award

action shot (2022)



- I’m from Memphis, TN
- Places I’ve been (intended destination)
  - also a few cities in China back in 2010 (toured Beijing and visited family in Nanning) and a few cities in Germany (mainly Berlin) in summer 2019 before my first semester here
- Can’t cook but love eating / trying all kinds of foods
  - some of my absolute favorites: nigiri/sashimi, Korean BBQ, ramen, bubble tea

Houston, TX
(2021)

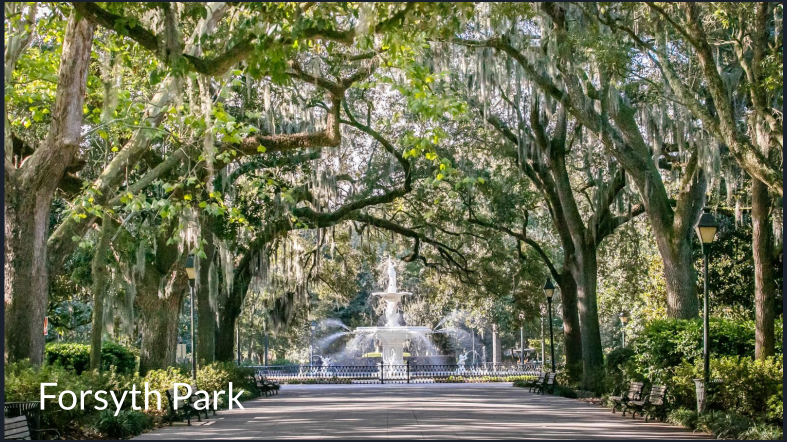
Atlanta, GA
(2022)

Fin-Two
(Downtown!)

It's #GaPreKWeek!
WESTIN
West Coast beauty seeks easygoing beachcomber
Hotels.com

Rob Bray

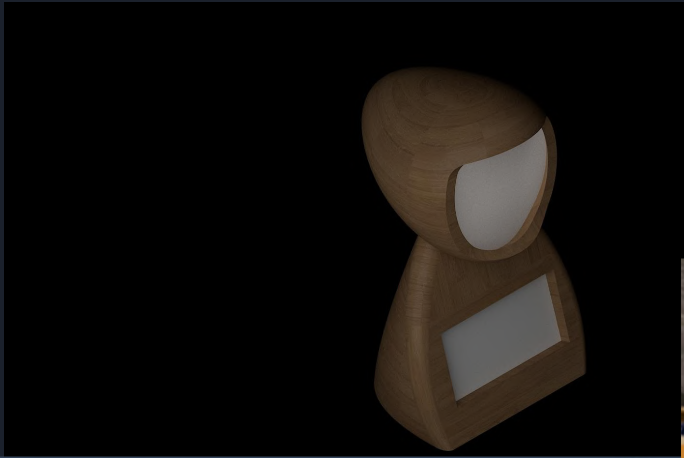
- Grew up in Savannah, GA. Lived in Hummelstown, Pennsylvania for 3 years. Moved to Tennessee for undergrad in CS at UTK
- First year masters in CS, advised by Dr. Gregor, GRA for Dr. Xiaopeng Zhao, GTA for Dr. Emrich

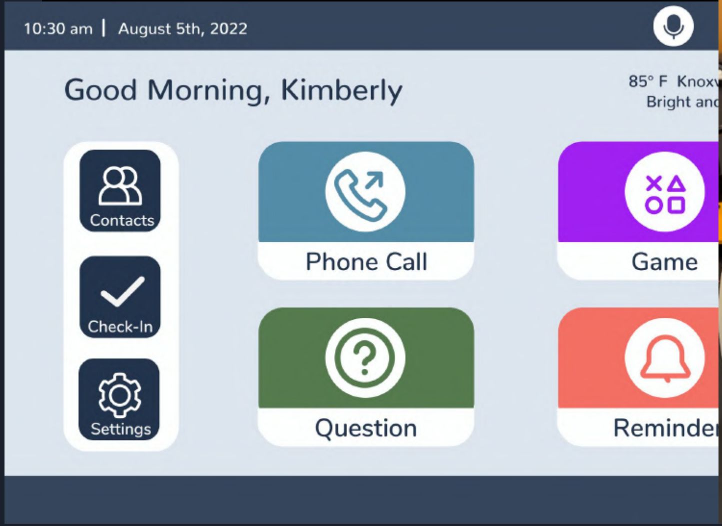

Forsyth Park

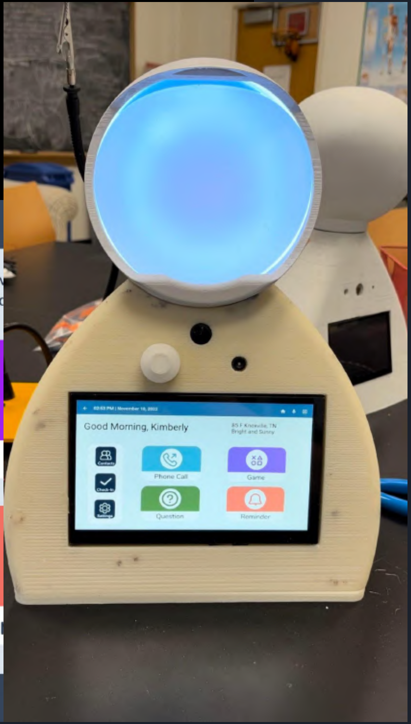
https://www.tripadvisor.com/Attraction_Review-g60814-d259614-Reviews-Forsyth_Park-Savannah_Georgia.html

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.savannah.com%2F&psig=AOvVaw0MehE0a62avBN1bLld38cq&ust=1682521410469000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCKDngu2mxf4CFQAAAAAdAAAAABAY

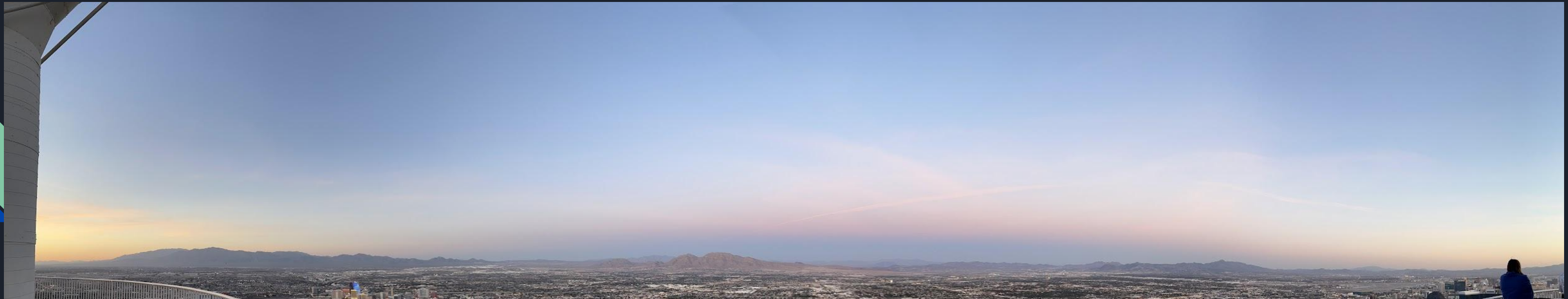
- Thesis work: FRED, the friendly robot to ease dementia. Affordable social robot based on Raspberry Pi for older adults with cognitive decline



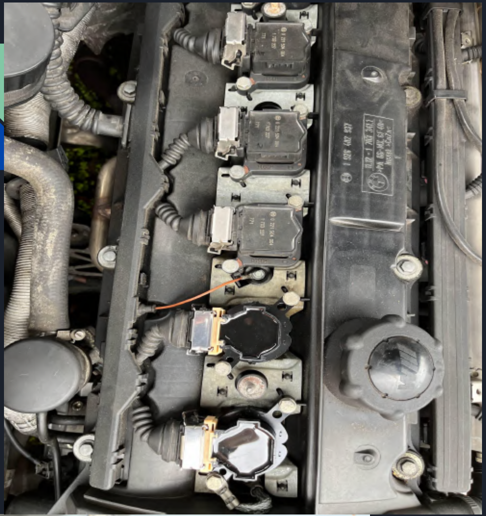


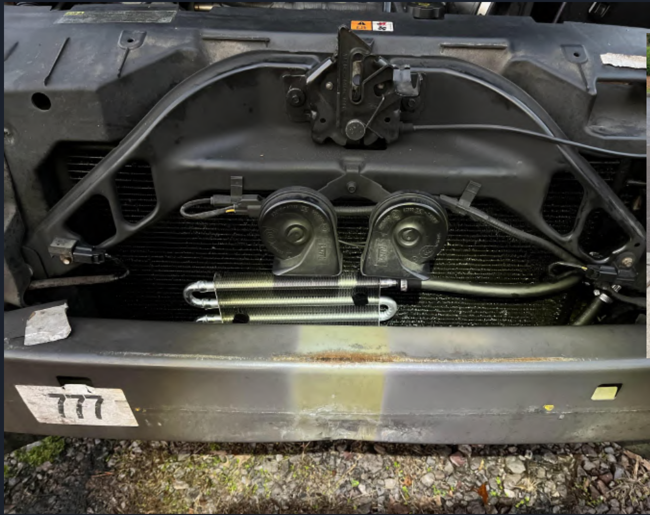


BURG



PLAY
$25
$100
DOUBLE BET
REPEAT LAST BET
The Strat Hotel Casino Skypod
CASHOUT TICKET
$31.96





777

MK N 9895

Mediterranean
507

- I love movies
- Favorite movie ever: Monty Python and the Holy Grail
- Favorite movie right now: Bullet Train

https://www.google.com/url?sa=i&url=https%3A%2F%2Fbloody-disgusting.com%2Fmovie%2F3718201%2Fbullet-train-trailer-2-brad-pitt-fights-to-survive-in-deadpool-2-directors-violent-action-thriller%2F&psig=AOvVaw2Tepl1mehgGfUyUinrdyWz&ust=1682522530896000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCOCI-oKrxf4CFQAAAAAdAAAAABAo

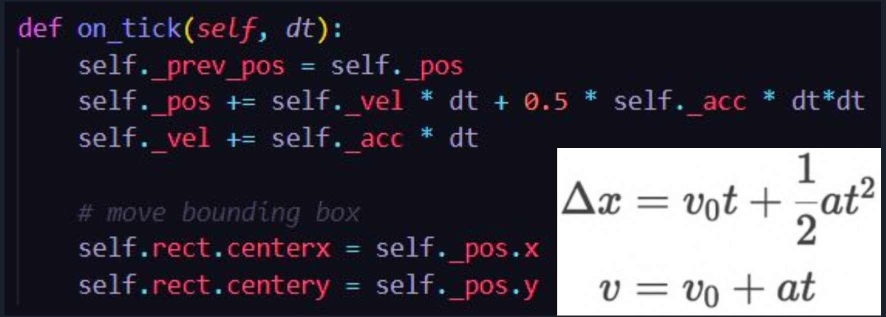
https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.imdb.com%2Ftitle%2Ftt0071853%2F&psig=AOvVaw3g0p_egR49V5ZSP0ywH5wB&ust=1682522411132000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCKjLkMiqxf4CFQAAAAAdAAAAABAD

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.sonypictures.com%2Fmovies%2Fbullettrain&psig=AOvVaw2Tepl1mehgGfUyUinrdyWz&ust=1682522530896000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCOCI-oKrxf4CFQAAAAAdAAAAABAI

Outline

- Overview of Simulating Collisions
    - Intro to Physics Simulation
    - Types of Collision Detection
- Data Structures & Algorithms
- History
- Applications
- Implementations
- Live Demo
- Results
- Open Issues
- References and Closing

Intro to Physics Simulation



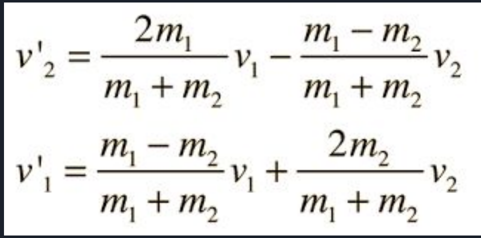
```
def on_tick(self, dt):
    self._prev_pos = self._pos
    self._pos += self._vel * dt + 0.5 * self._acc * dt*dt
    self._vel += self._acc * dt

    # move bounding box
    self.rect.centerx = self._pos.x
    self.rect.centery = self._pos.y
```

$$\Delta x = v_0 t + \frac{1}{2} a t^2$$

$$v = v_0 + at$$

- Goal: simulate a set of physical laws as accurately as possible
  - it should at least “feel right” and be stable enough
- Base level: kinematics, the motion of objects alone over time
  - gravity, friction
- Next level: collision detection
  - more complex shapes and models?
- Next level: collision resolution
  - elastic or inelastic?
  - what about angular velocity / momentum?
- More advanced simulations for fluid dynamics, soft-body dynamics, cloth, modeling light transport (ray tracing), etc.



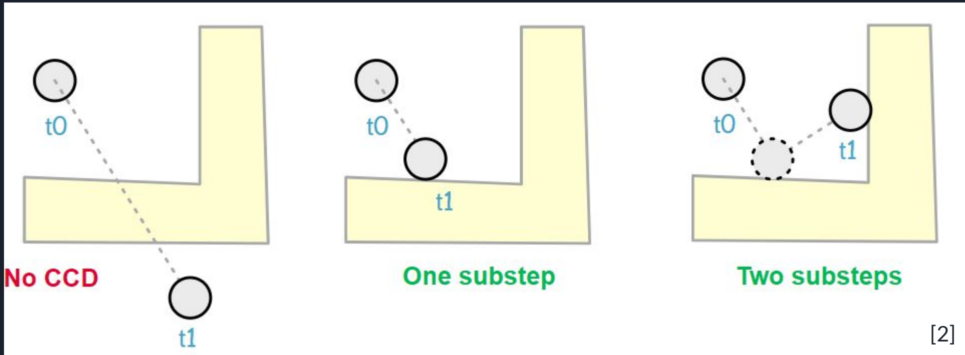
$$v'_2 = \frac{2m_1}{m_1 + m_2} v_1 - \frac{m_1 - m_2}{m_1 + m_2} v_2$$

$$v'_1 = \frac{m_1 - m_2}{m_1 + m_2} v_1 + \frac{2m_2}{m_1 + m_2} v_2$$

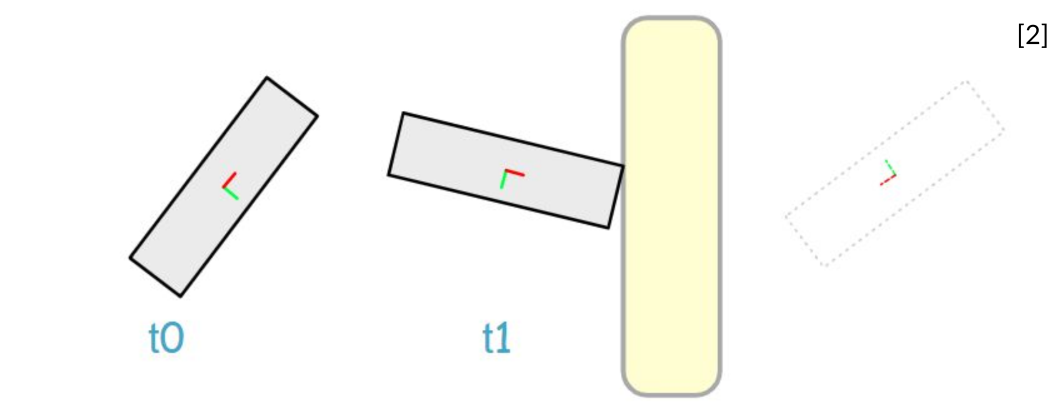
velocities after an elastic collision

Types of Collision Detection

- Discrete
  - uses the time step between current time and the last frame for kinematics (referred to as *dt*, “delta time,” or a “tick”)
  - suffers from the so-called “tunneling effect”
- Continuous
  - requires some form of interpolation between t_0 and $t_1 = t_0 + dt$
    - supersampling
    - bisection
    - ray casting
  - more substeps (costly)
    - significant impact on performance

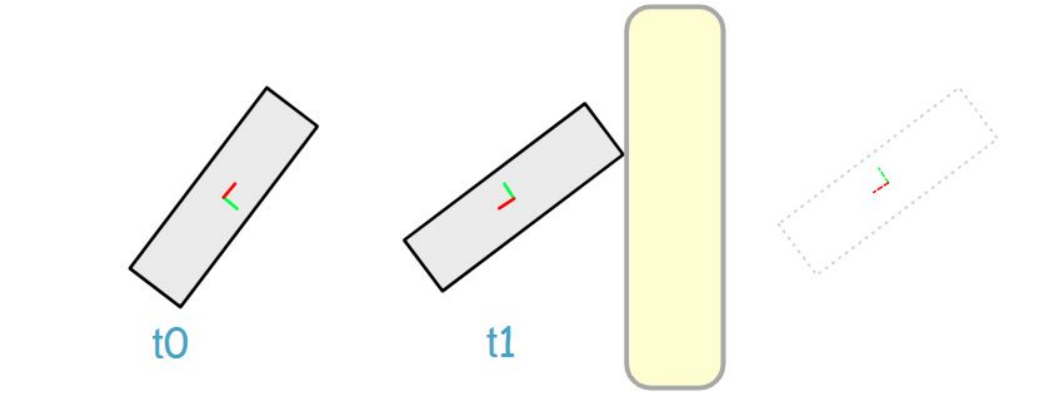


[2]



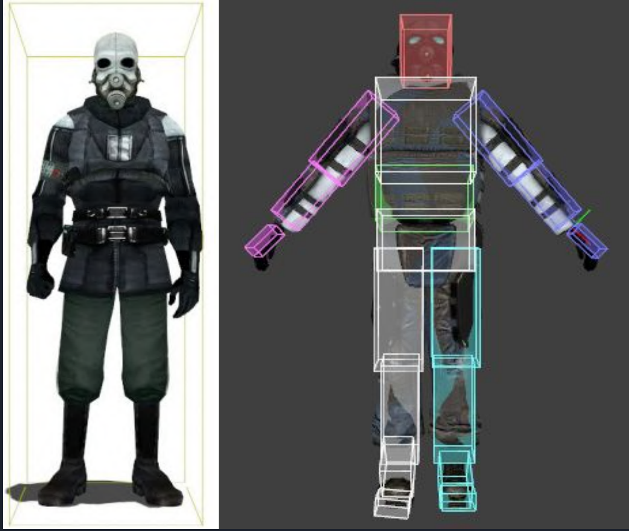
[2]

orientation usually not considered in CCD!

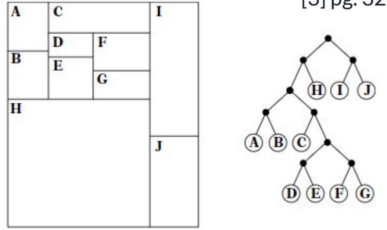


Data Structures

- Bounding box/volume vs. hitbox
  - axis-aligned bounding box (AABB)
  - oriented bounding box (OBB)
  - *k* discrete oriented polytope (*k*-DOP)
- Bounding volume hierarchy (BVH)
  - AABB tree or R-tree (“rectangle” tree)
- *k*-d trees
  - generalization of quadtrees / octrees
  - *k* represents the number of dimensions subdivided by arbitrarily positioned splits, performed in one dimension at a time (does not have to match the dimensionality of the space)

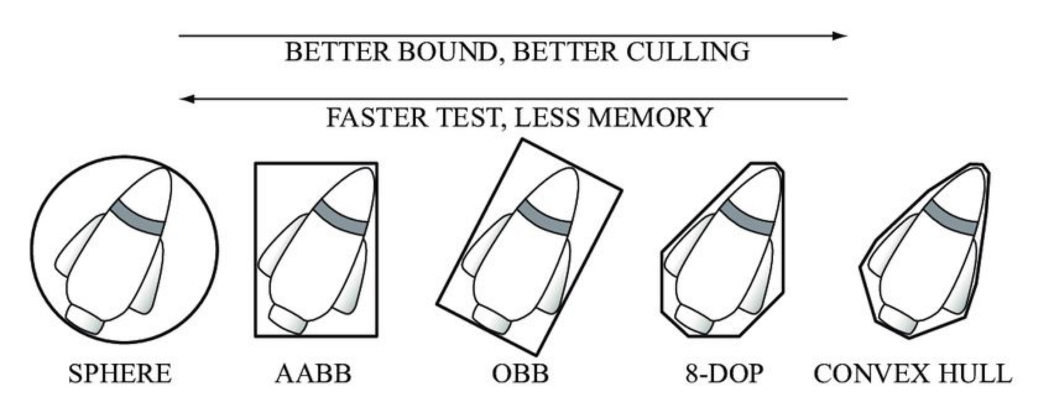


https://developer.valvesoftware.com/wiki/Bounding_box
https://developer.valvesoftware.com/wiki/Hitbox

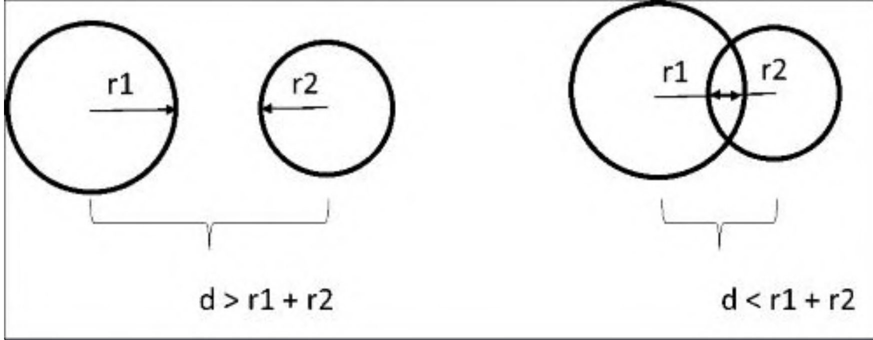


[3] pg. 320

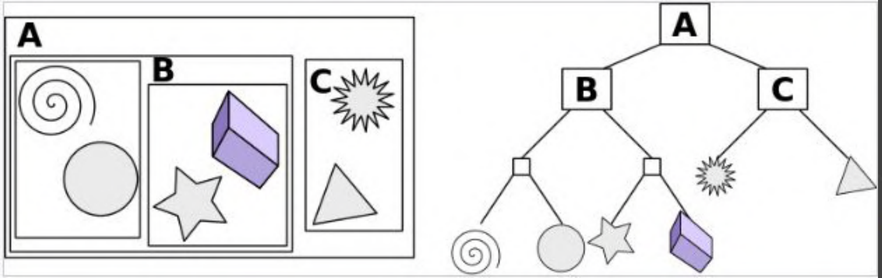
a 2D *k*-d tree



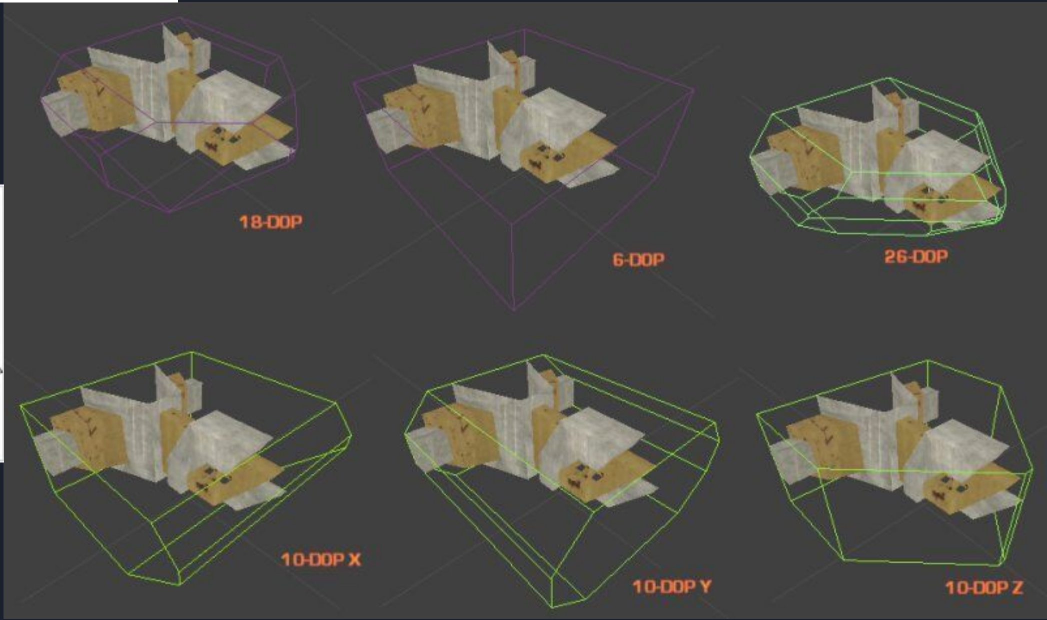
https://www.researchgate.net/figure/Bounding-volumes-sphere-axis-aligned-bounding-box-AABB-oriented-bounding-box_fig9_272093426



https://subscription.packtpub.com/book/game+development/9781783288199/5/ch05lvl1sec29/circular-collision-detection



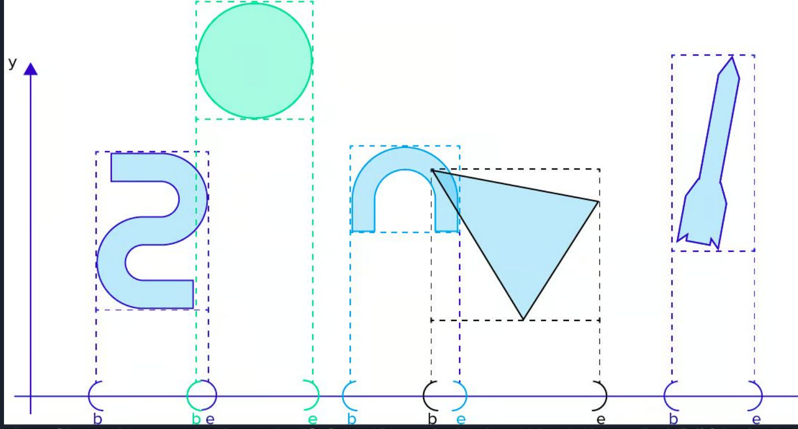
https://en.wikipedia.org/wiki/Bounding_volume_hierarchy



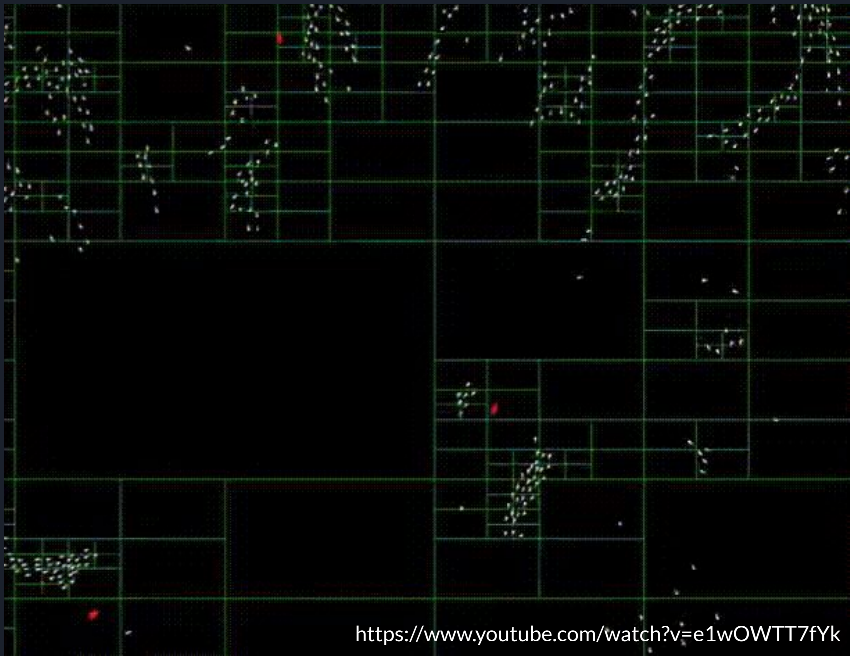
https://docs.unrealengine.com/4.27/en-US/InteractiveExperiences/Physics/Collision/HowTo/AddDOP/

Algorithms

- Pairwise checking
  - naive: try all combinations
  - broad phase algorithms to reduce the number of pairs
    - sweep and prune / sort and sweep
    - hierarchical methods
      - spatial partitioning
        - uniform grid
        - *k*-d trees
      - object partitioning
        - BVH



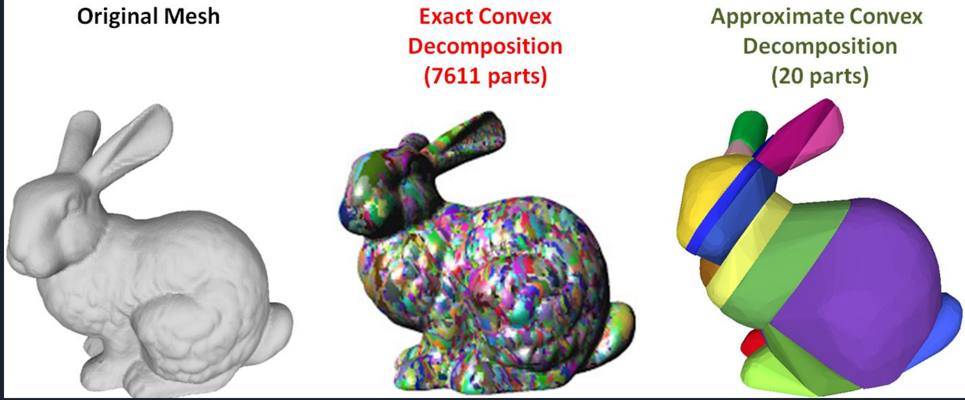
[4]



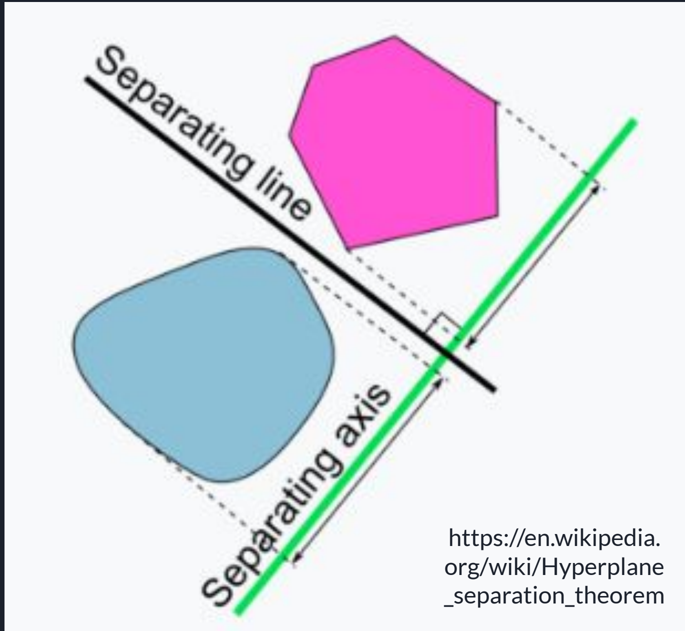
https://www.youtube.com/watch?v=e1wOWTT7fYk

Algorithms

- Pairwise checking
    - narrow phase
        - convex decomposition
            - Hierarchical Approximate Convex Decomposition (HACD)
        - check for overlap
            - Separating Axis Theorem (SAT)
            - Gilbert–Johnson–Keerthi distance algorithm (GJK)


Original Mesh
Exact Convex Decomposition (7611 parts)
Approximate Convex Decomposition (20 parts)

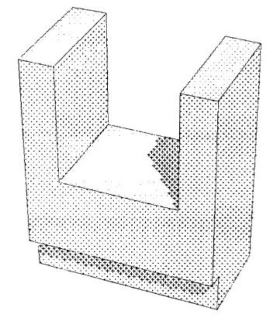
https://github.com/kmammou/v-hacd



https://en.wikipedia.org/wiki/Hyperplane_separation_theorem

History

- Arthur Appel (1968): first computer-generated image shaded by ray tracing
- Jeffrey Goldsmith & John Salmon (1987):
    - automatic generation of bounding volume hierarchies
    - surface area heuristic (SAH) as a predictor for tree generation
- J. David MacDonald & Kellogg Booth (1990): applying SAH to binary trees for space subdivision



Appel 1968 [9]

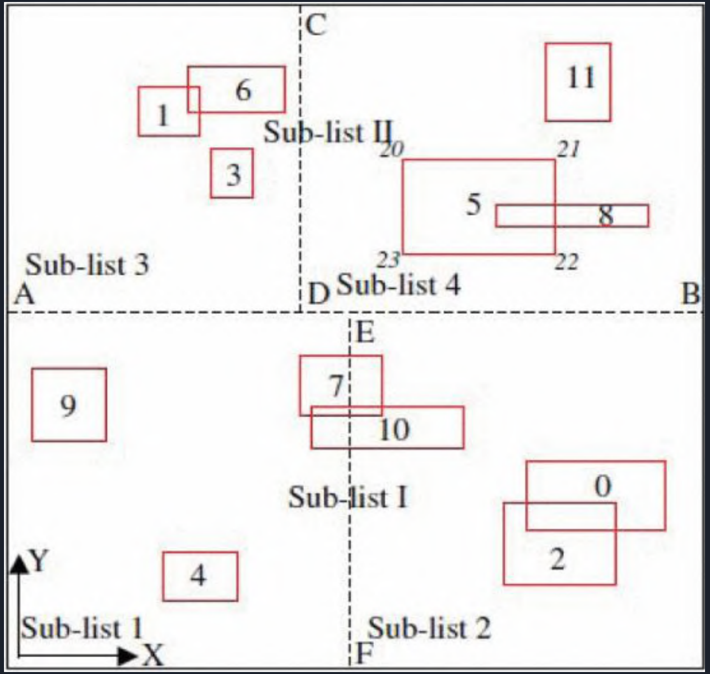
Goldsmith & Salmon 1987 [10]

MacDonald & Booth 1990 [11]

https://independent.academia.edu/GinovandenBergen

History

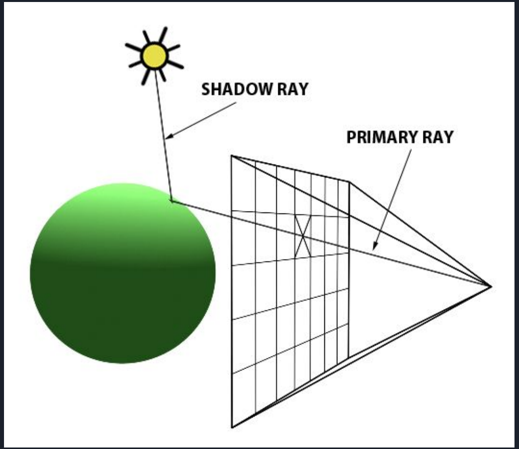
- Optimizations
  - Gino van den Bergen (1998): faster overlap testing with probabilistic SAT [12]
    - only 15% of separating axes are in the direction of two edges, so test a smaller subset for 6% failure rate
    - this performance is competitive with OBBs, where AABBs have been previously discounted
  - Yi-Si Xing, Xiaoping Liu, Shao-Ping Xu (2020): claimed *O*(*n*) algorithm similar to sweep and prune with their novel axial cut



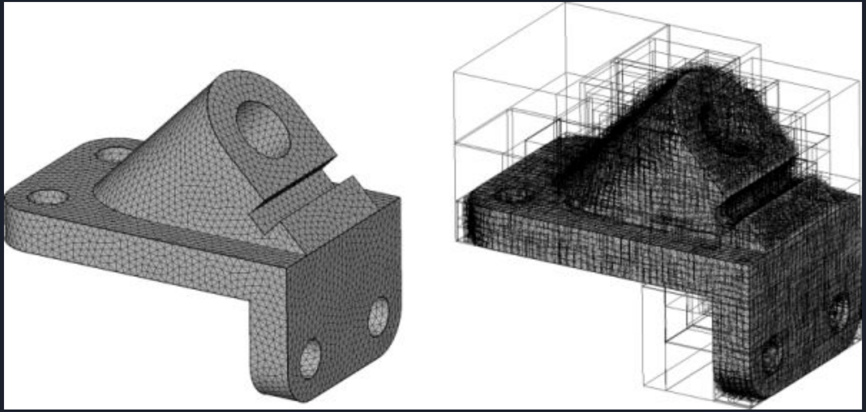
axial cut & locality overlap testing (Xing et al. 2020) [13]

Applications

- Optimizing simulations
  - acceleration of ray tracing
  - game engines
  - aerospace: 4D AABB tree for space debris collision
  - swarm robotics
- Computational geometry / computer graphics
  - specifically CAD

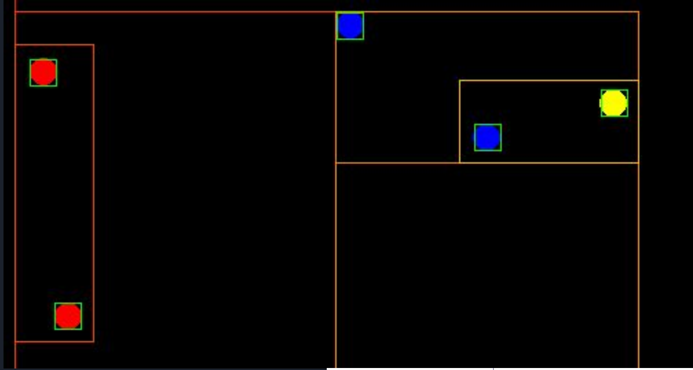


https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-ray-tracing/implementing-the-ray tracing-algorithm.html

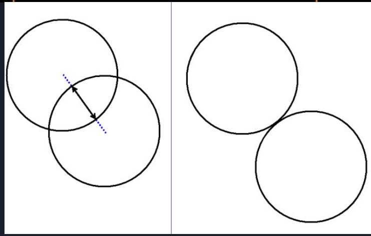


https://doc.cgal.org/latest/AABB_tree/index.html

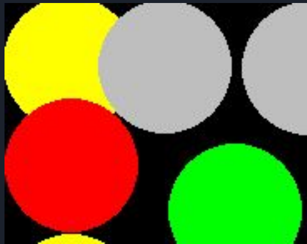
Implementations



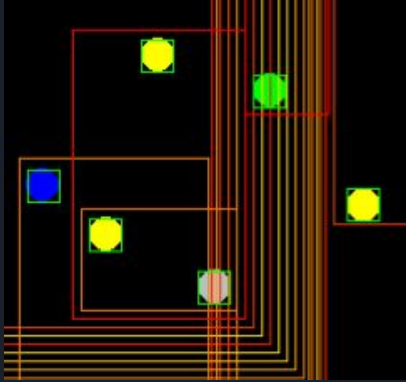
- Implemented a simple 2D discrete collision simulation
  - perfectly elastic circles
  - continuous wall collisions
  - “nudging” and density issues
- Implemented a 2D AABB (supports unions)
- Implemented a binary AABB tree
  - inserting a new leaf – *O*(log *n*)
    - recursively find the best sibling based on some cost function
    - create a new internal node, where the internal node’s children are the best sibling and the new node
    - trickle up the change by refitting the AABBs based on the new children
  - deleting a leaf – *O*(log *n*)
    - sibling replaces parent (internal node)
    - trickle up the change by refitting the AABBs



https://flatredball.com/documentation/tutorials/math/circle-collision/



Implementations



issue with improper padding

- Various strategies for optimization
  - efficient insertion / good cost function
    - need a balance between quick insertion but “good enough” AABBs
    - either the insertion itself or the cost function can be recursive, employing some heuristic for efficiency
      - minimize surface area and overlap
  - try to keep as much of the tree when handling movement
    - rebuilding the tree each frame is costly
    - reinsertion is a good strategy because it is spread out
      - occurs when a node travels outside its parent’s bounding box
    - padding / “speculative expansion” based on velocity
  - balancing
    - tree rotations
    - but reinsertion can can randomize the order naturally

Implementations

- Broad phase checking each frame
  - brute force – $O(n^2)$
  - combinations – $O(n \text{ choose } 2) = O(n(n-1)/2)$ ($= O(n^2)$)
  - AABB pruning – $O(n \log n)$
    - for each of the n objects, each one takes $O(\log n)$ to determine leaf overlaps by eliminating up to half the objects per parent AABB check
    - we do not filter out duplicate checks

Implementations

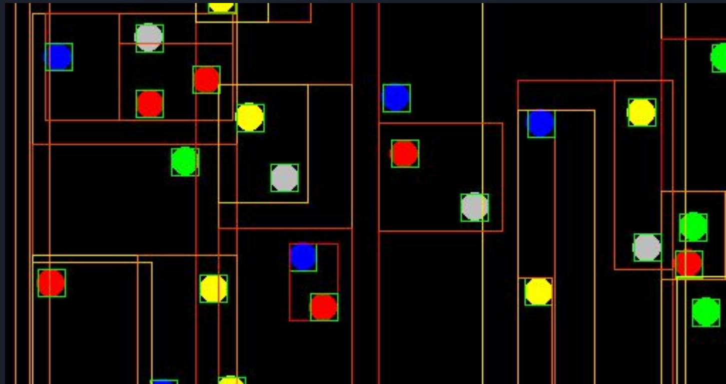
- Demo!
- Our code: https://github.com/rjbray915/CS581-Final

Results

- Testing parameters:
  - seed: 581
  - number of circles: 100
  - min radius: 10
  - max radius: 10
  - spacing: 10
  - numbers are recorded during the second 30 seconds of the simulation

	Naive	AABB Tree
Avg FPS	153.8	232.8
Avg Checks	4950.0	1012.3

- FPS speedup: 51.4%
- Checks performed: 20.5%

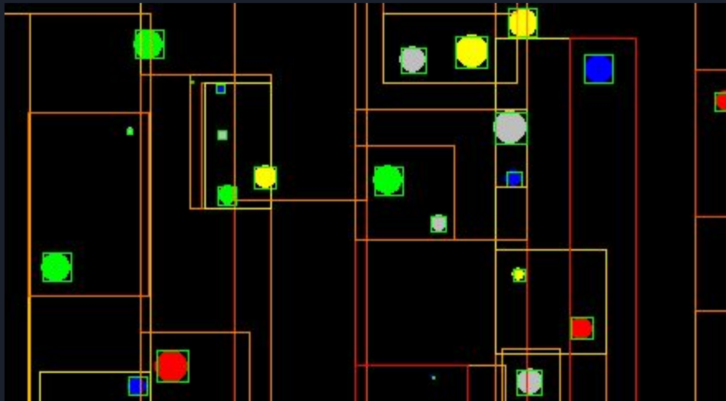


Results

- Testing parameters:
  - seed: 581
  - number of circles: 100
  - min radius: 1
  - max radius: 10
  - spacing: 10
  - numbers are recorded during the second 30 seconds of the simulation

	Naive	AABB Tree
Avg FPS	150.1	279.2
Avg Checks	4950.0	960.1

- FPS speedup: 86.0%
- Checks performed: 19.4%

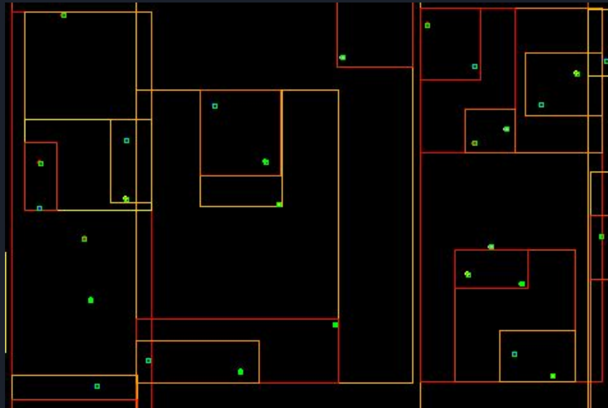


Results

- Testing parameters:
  - seed: 581
  - number of circles: 100
  - min radius: 2
  - max radius: 2
  - spacing: 10
  - numbers are recorded during the second 30 seconds of the simulation

	Naive	AABB Tree
Avg FPS	152.6	298.3
Avg Checks	4950.0	852.9

- FPS speedup: 95.5%
- Checks performed: 17.2%

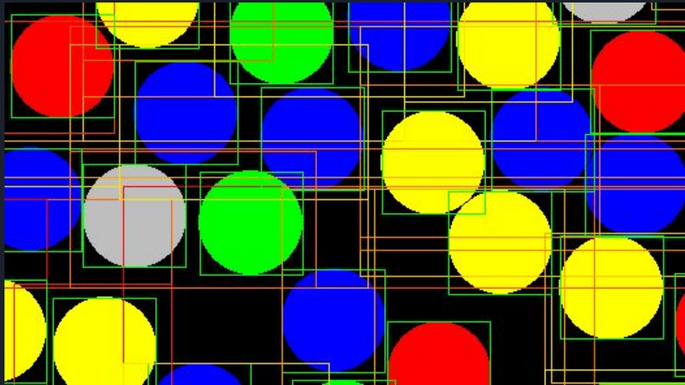


Results

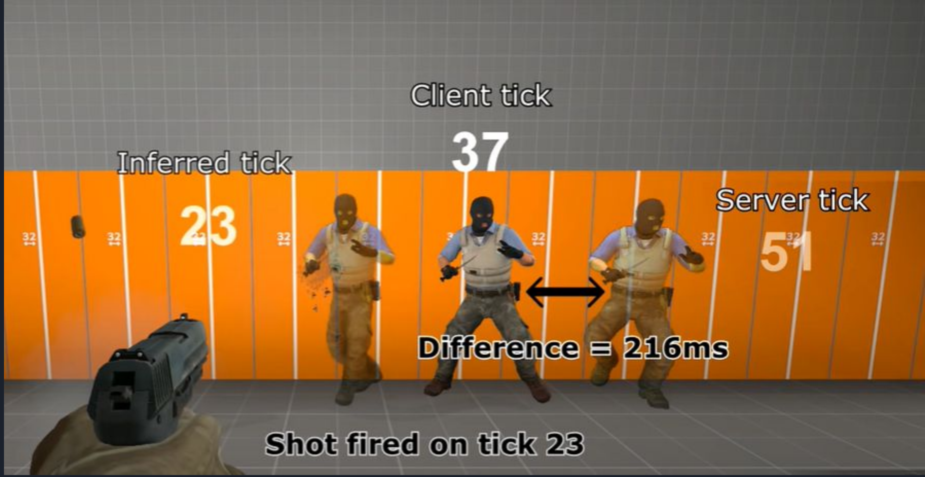
- Testing parameters:
  - seed: 581
  - number of circles: 100
  - min radius: 40
  - max radius: 40
  - spacing: 10
  - numbers are recorded during the second 30 seconds of the simulation

	Naive	AABB Tree
Avg FPS	146.4	191.1
Avg Checks	4950.0	1632.8

- FPS speedup: 30.5%
- Checks performed: 33.0%



Open Issues



https://www.youtube.com/watch?v=6EwaW2iz4iA

- Online video games
  - hit registration, lag compensation
  - server tick rate
    - Counter-Strike 2 sub-tick updates
- Continuous collision detection
  - always room for improvement in efficiency, accuracy, and precision

References

1. https://digitalrune.github.io/DigitalRune-Documentation/html/138fc8fe-c536-40e0-af6b-0fb7e8eb9623.htm
2. https://www.nphysics.org/continuous_collision_detection/
3. https://www.taylorfrancis.com/books/mono/10.1201/b14581/real-time-collision-detection-christer-ericson
4. https://en.wikipedia.org/wiki/Bounding_volume_hierarchy
5. https://www.toptal.com/game/video-game-physics-part-ii-collision-detection-for-solid-objects
6. https://www.youtube.com/watch?v=9IULfQH7E90
7. https://box2d.org/files/ErinCatto_DynamicBVH_GDC2019.pdf
8. https://github.com/kip-hart/AABBTree/
9. https://dl.acm.org/doi/10.1145/1468075.1468082
10. https://ieeexplore.ieee.org/document/4057175
11. https://link.springer.com/article/10.1007/BF01911006
12. https://www.cs.cmu.edu/~djames/pbmis/etc/jgt98deform_AABB.pdf
13. https://ieeexplore.ieee.org/abstract/document/5524093?casa_token=3_gGtO94mGwAAAAA:G8P3XI6RN6fxm5PbXQ3FlehlIvJiuaj3rZb20on79exIzjwXea-M2Gde7fZPmGiPHD2oFECXNw

Discussion

- Questions?

Test Questions Revisited

1. What is the major flaw of discrete collision detection?
2. What kind of *k*-DOP is a 3D AABB? (What is *k*?)
3. What is one way to optimize a BVH?