Optimized Search Manager

Xavier Olivenza
Research
CITM 2016-17

What is a search engine for?

-Search for an entity in an area/range

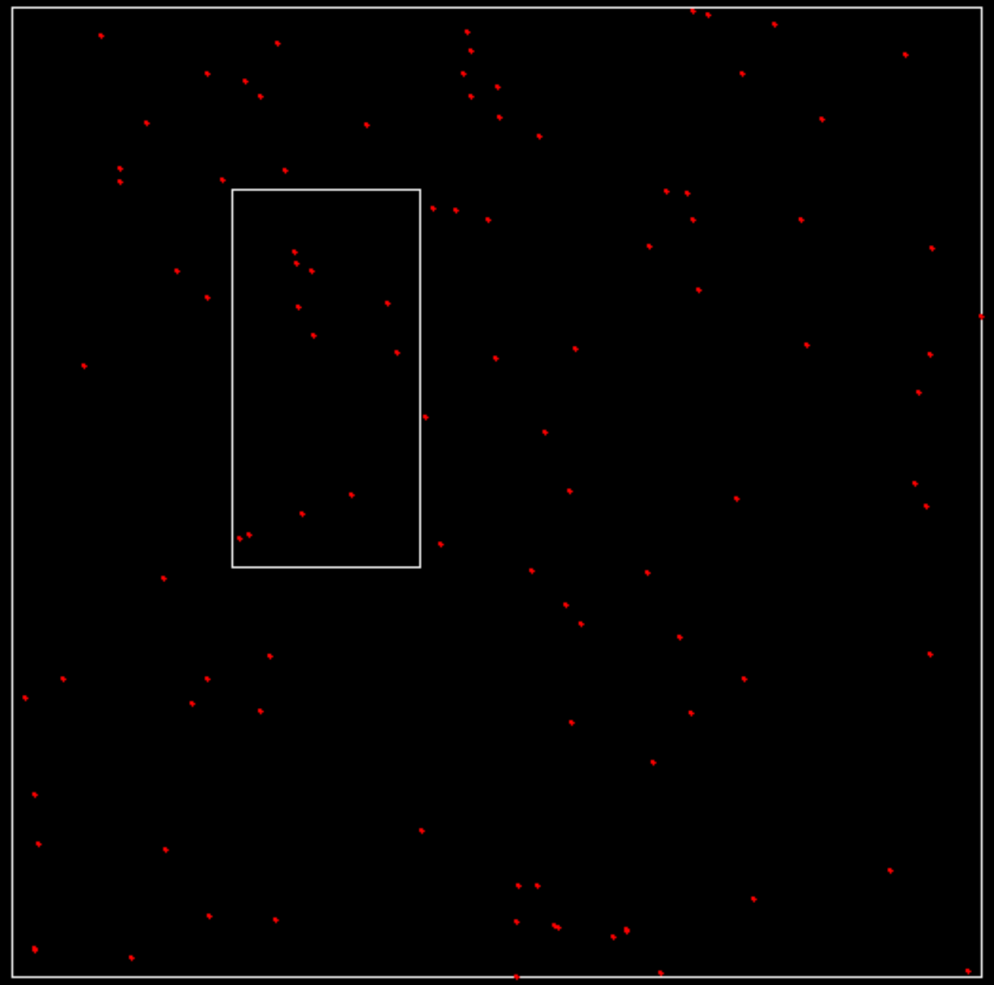
Where I can use this?

Brute force

-You have to do as many checks as entities are evaluated

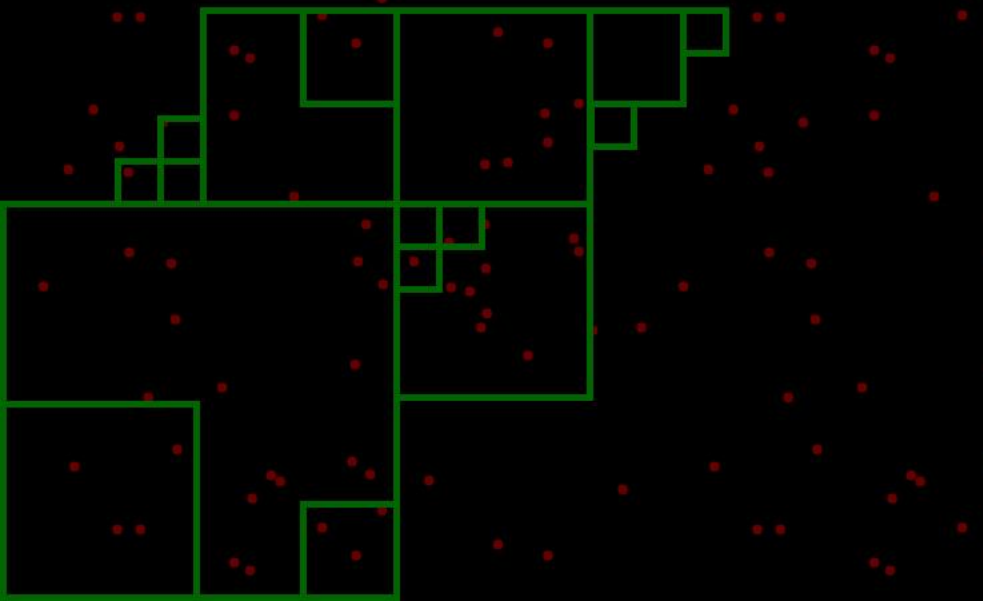
...

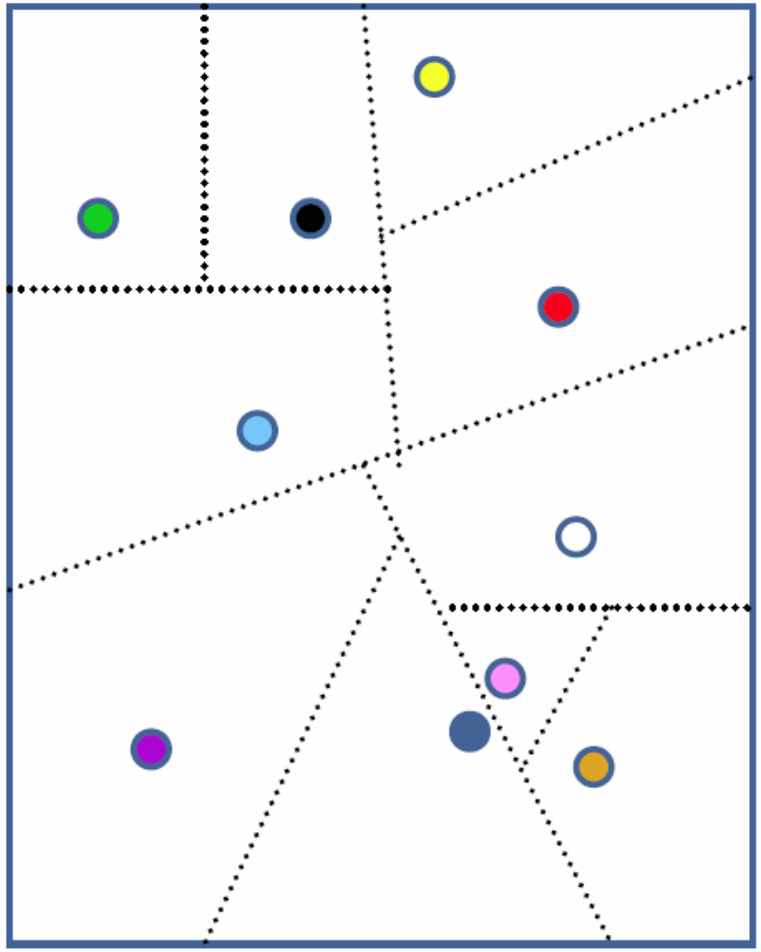
-What happens if we group and order the entities?

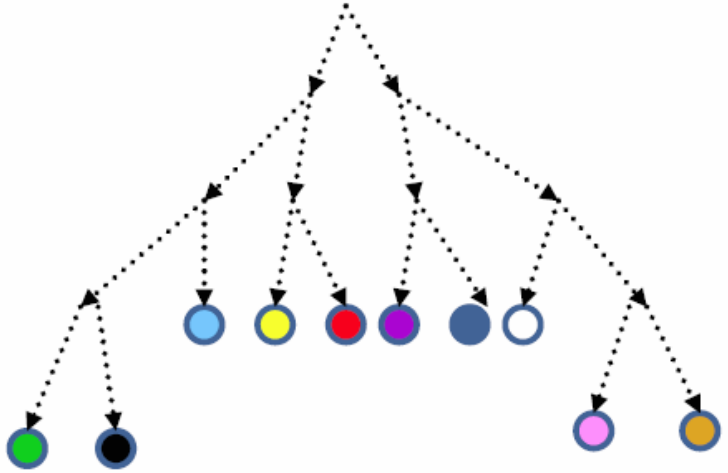


Space Partitioning

·Dividing a space into two or more subsets which do not overlap
·Algorithms that tend to be hierarchical
Most common partition data structures:
· BSP Trees
· Quadtrees
· Octrees
· K-dimensional trees
· R-Trees

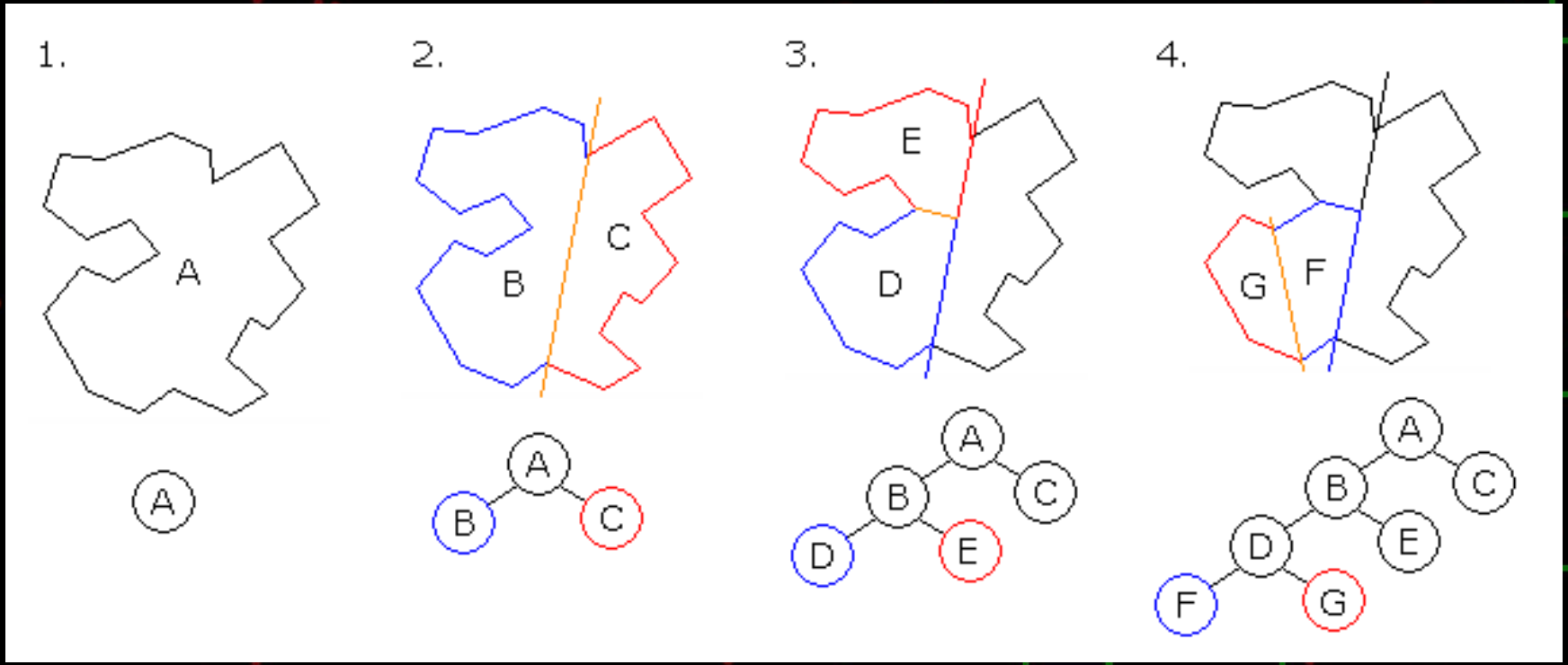






Binary Space Partitioning (BSP)

- Generalization
- Origin: Quickly draw polygonal 3d scenes
- Slow generation -> Pre-calculate



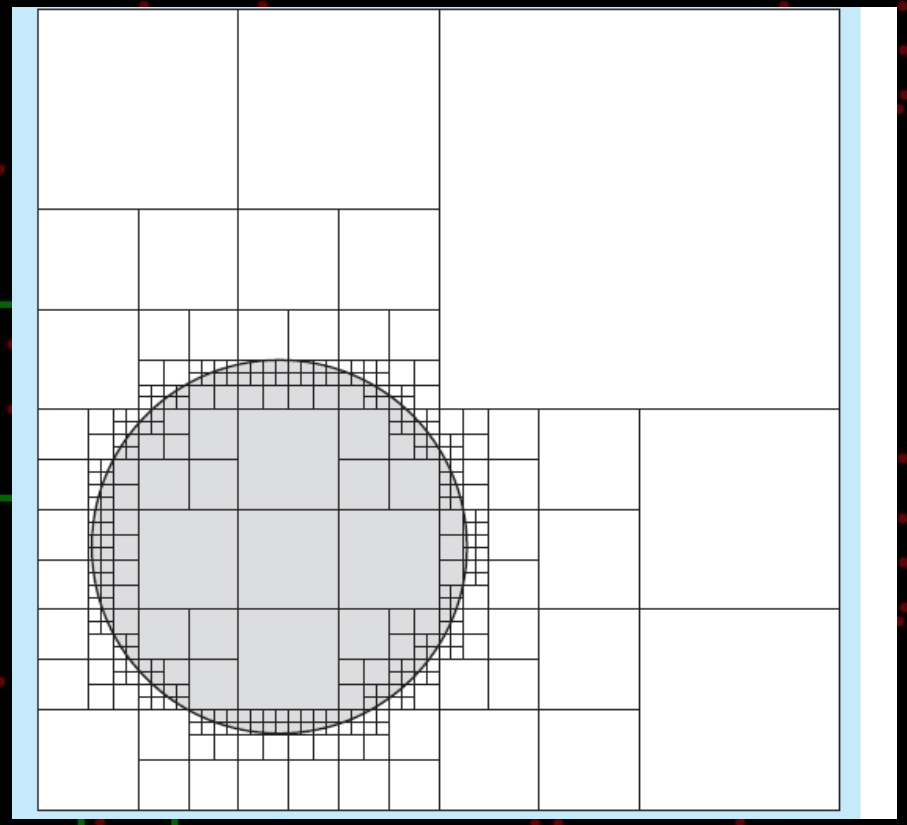
BSP, Where is used?

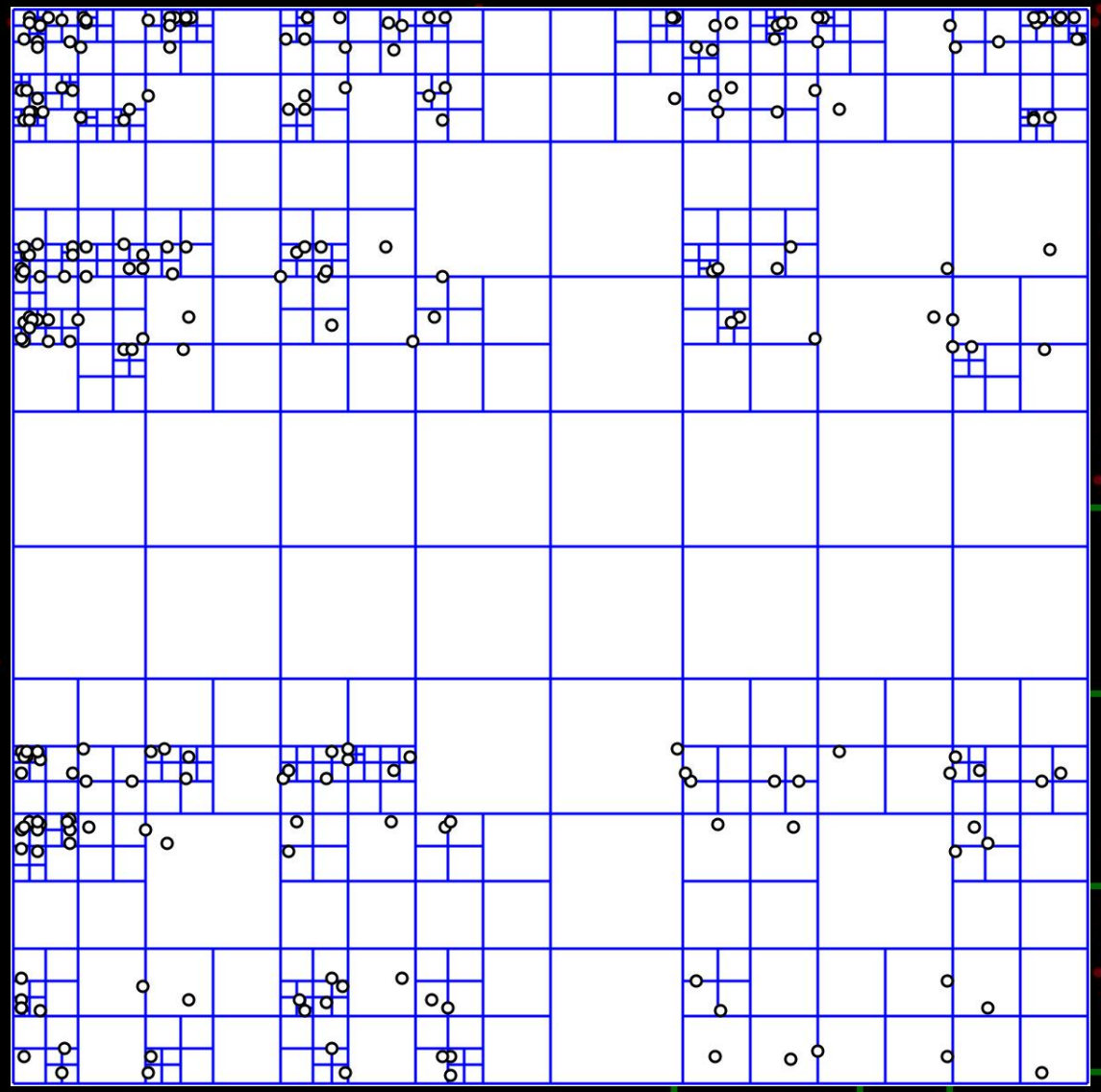
Id-Teck 1 Doom

Quake Engine + descendants

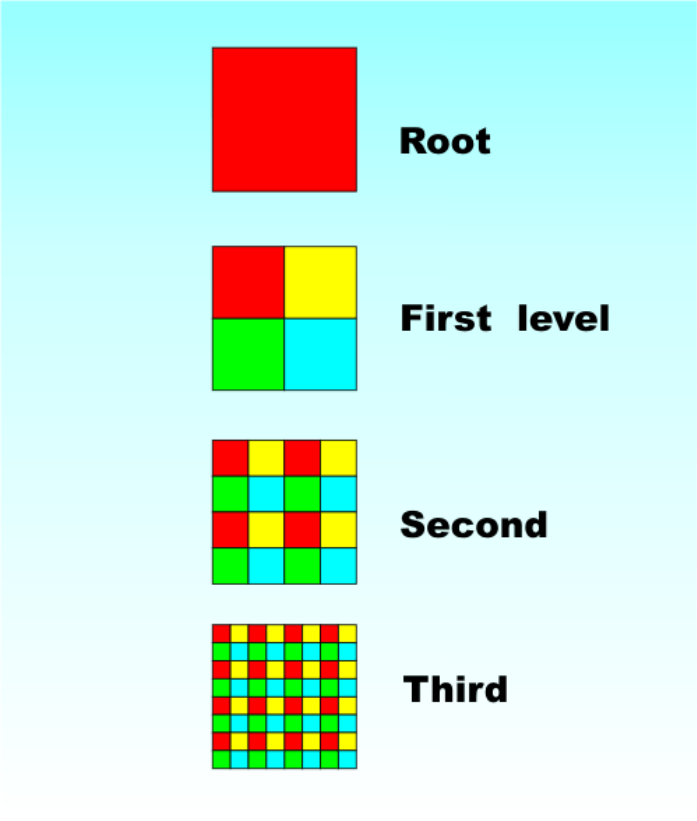
Quadtree

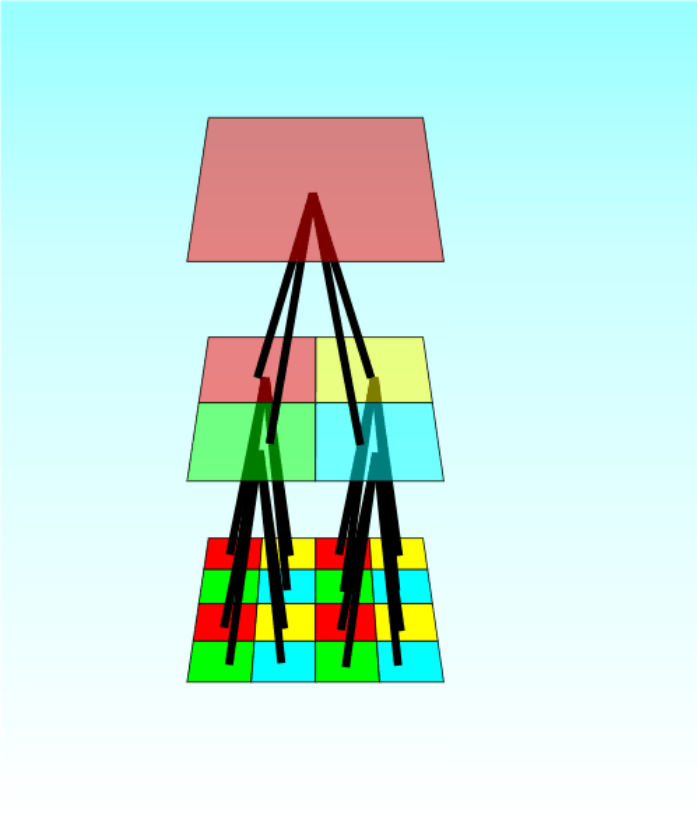
- Generally 2D
- When it reaches max node capacity -> split

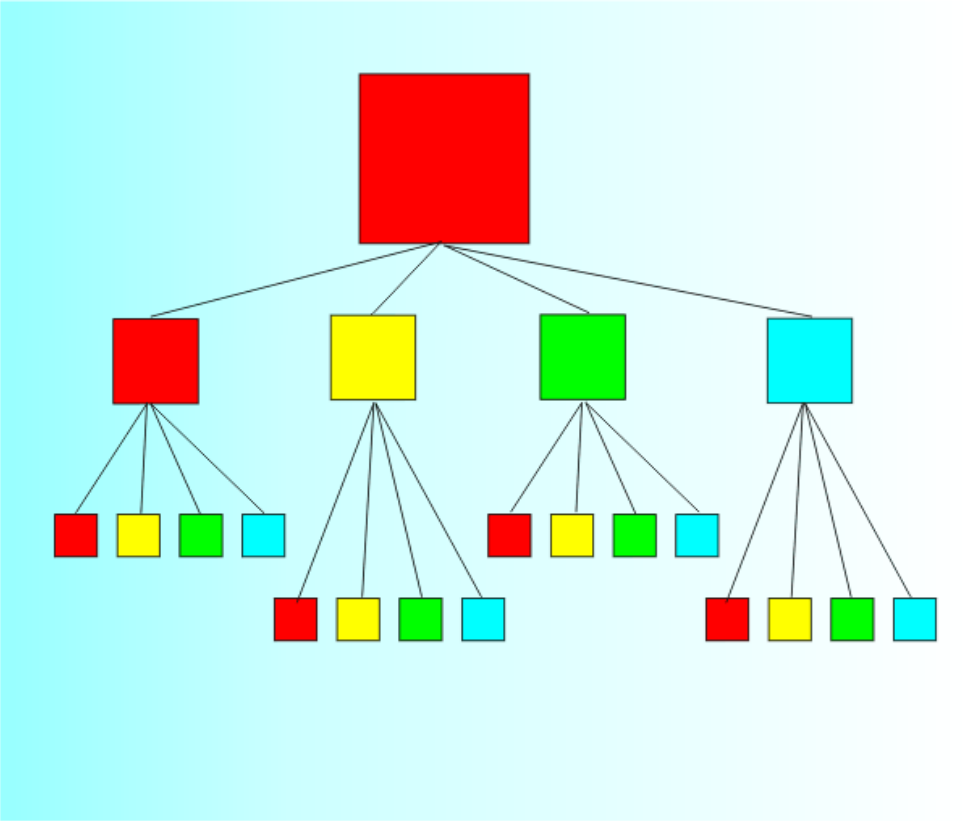




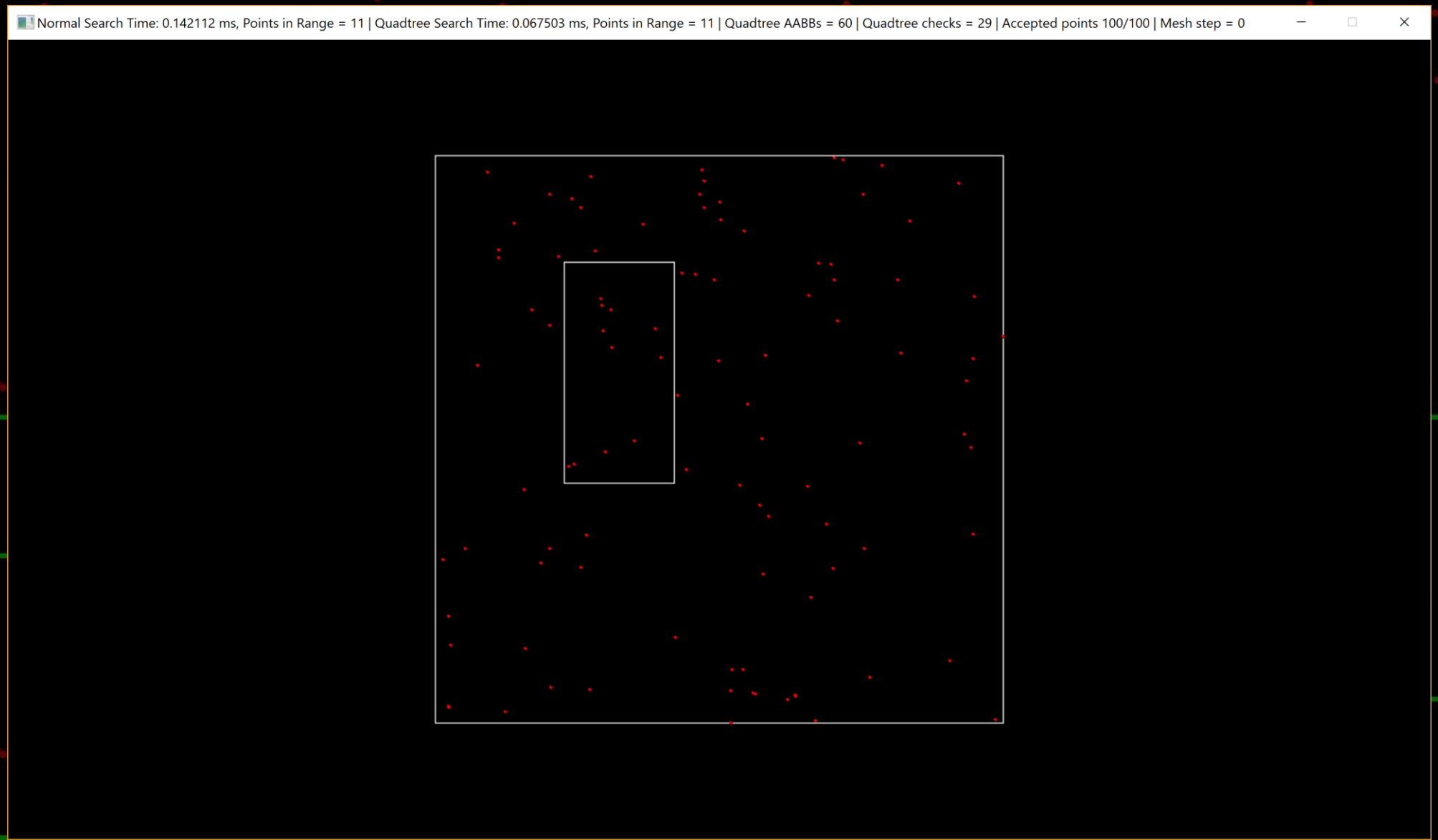
Quadtree, Tree



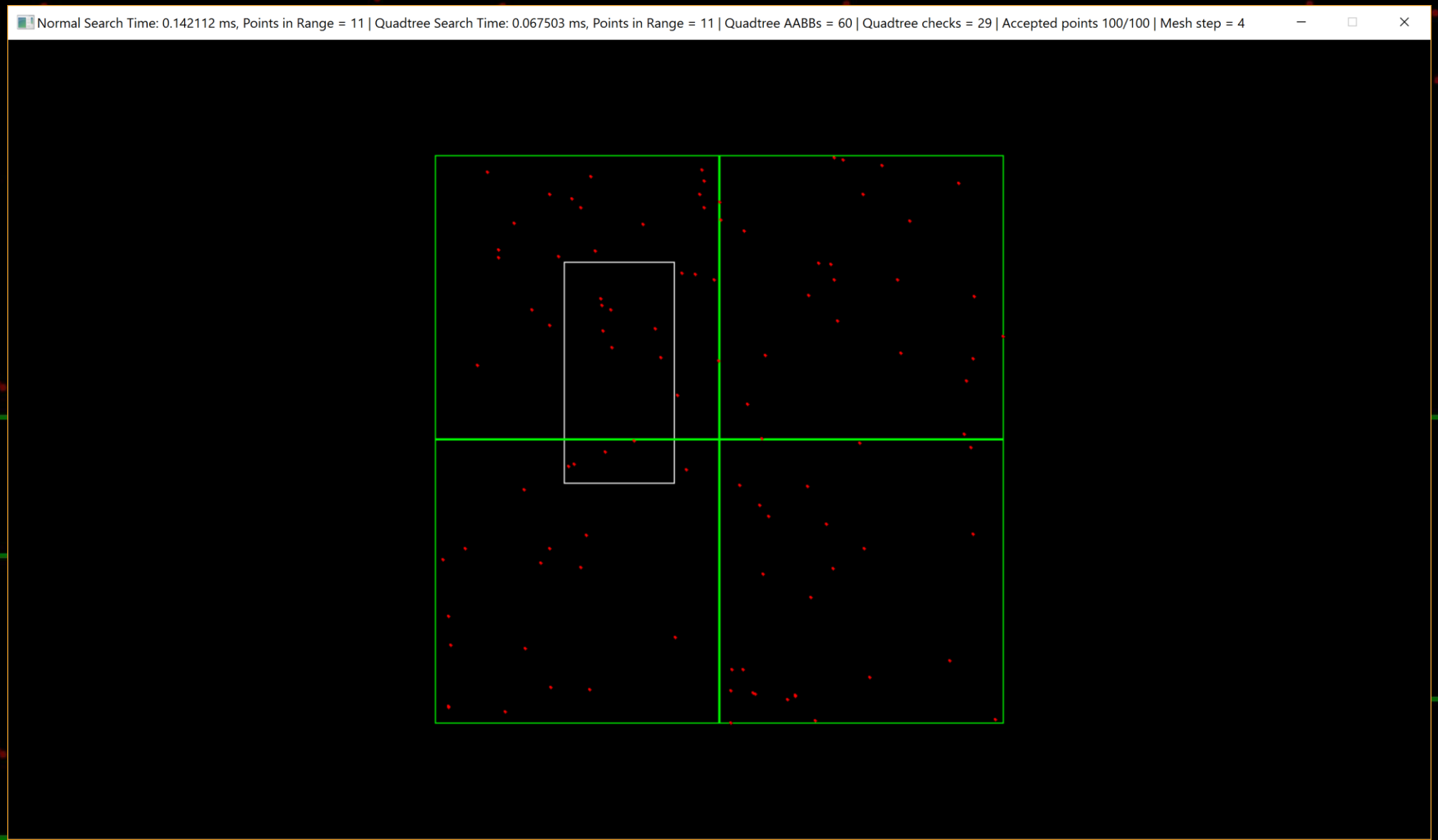




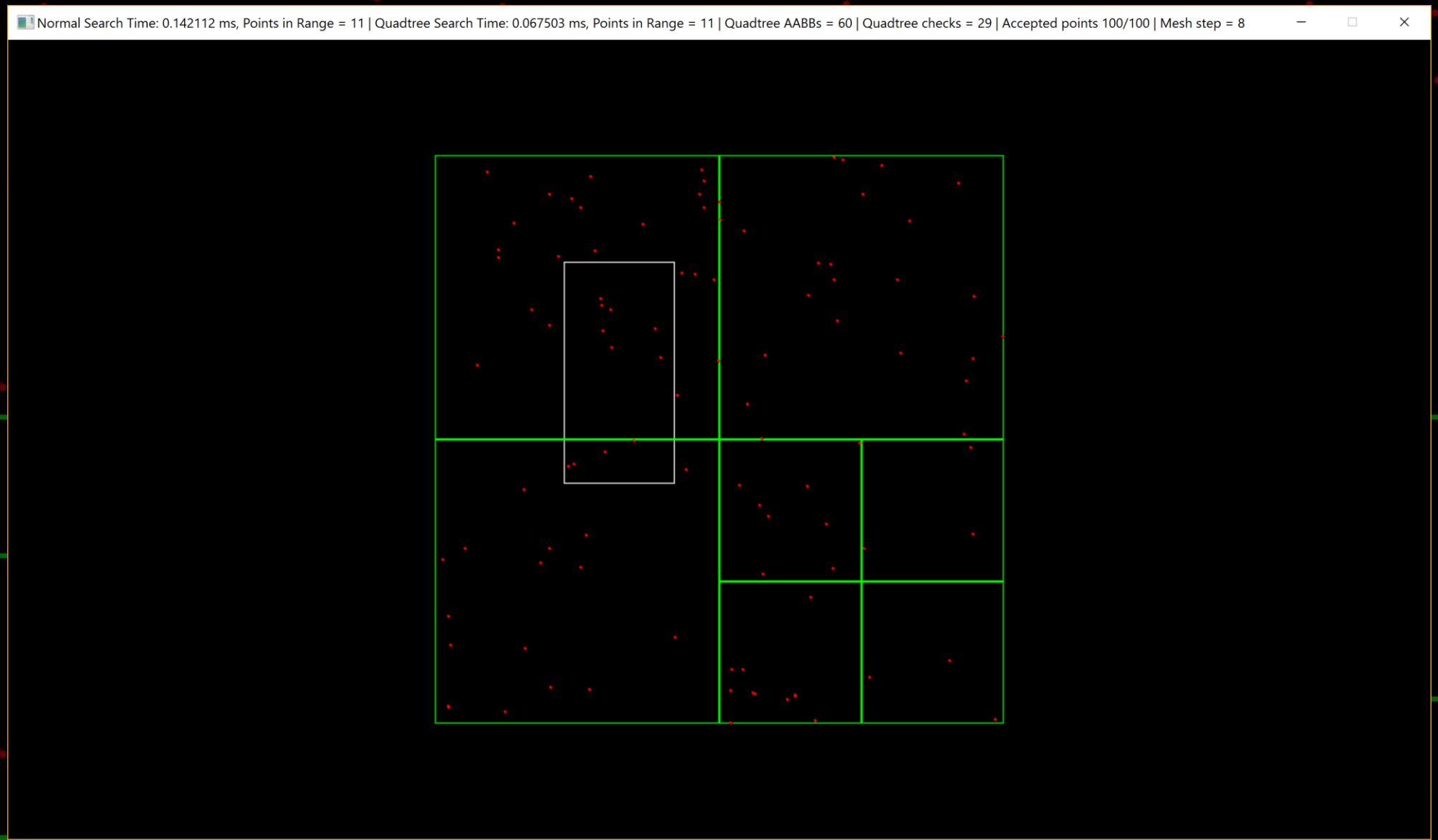
Quadtree, Tree



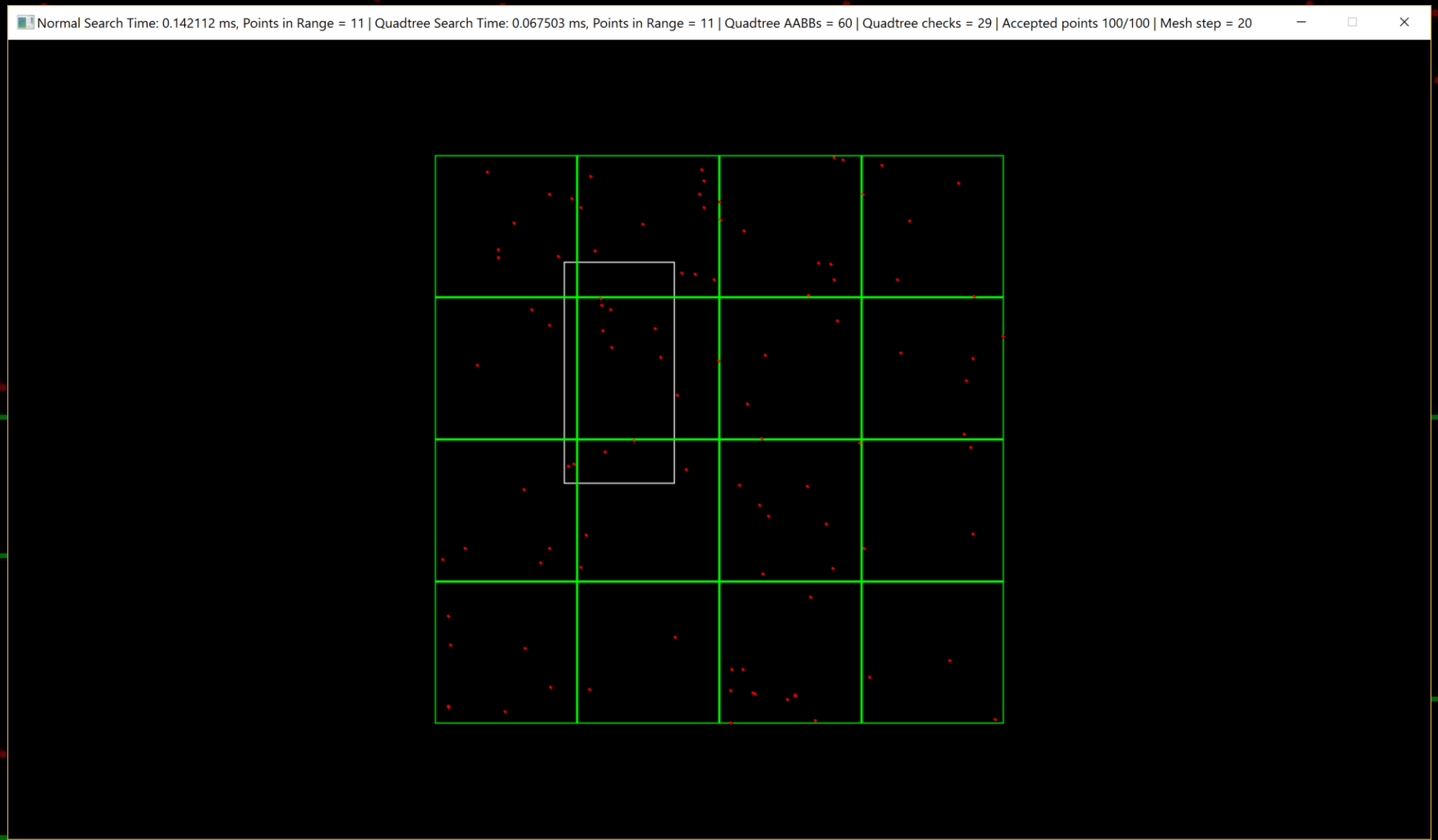
Quadtree, Tree



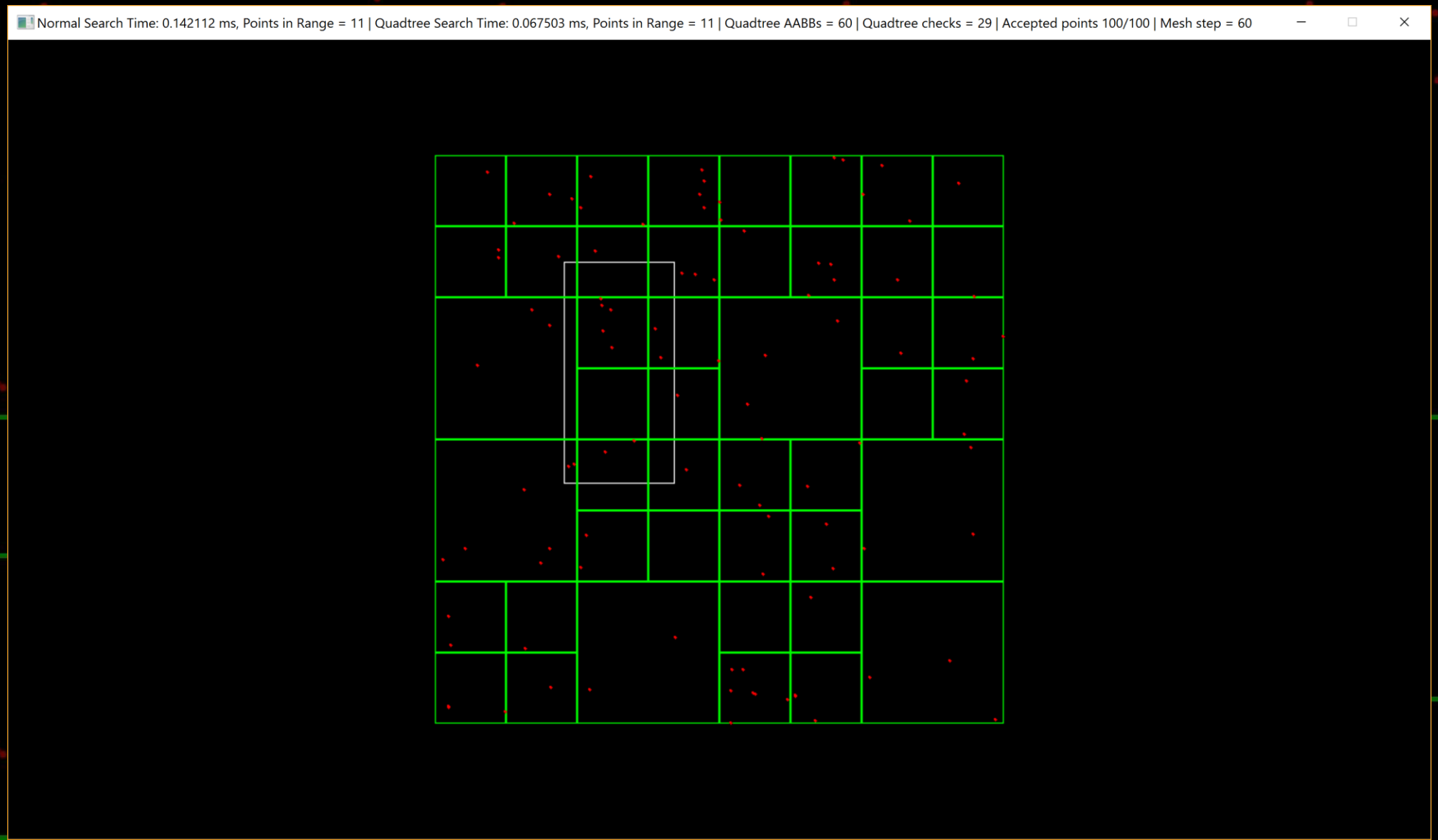
Quadtree, Tree



Quadtree, Tree

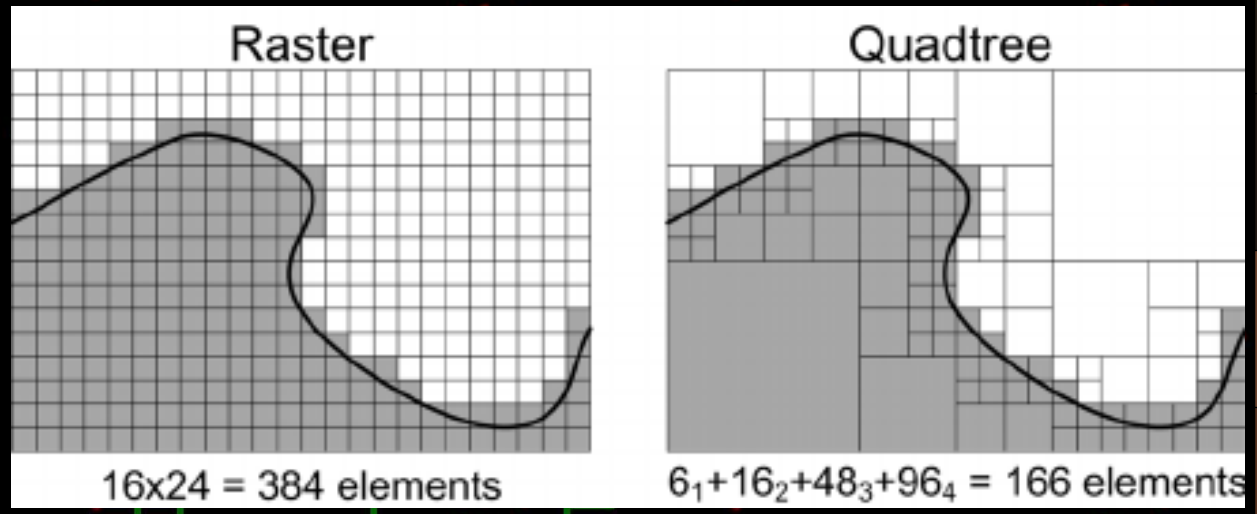


Quadtree, Tree



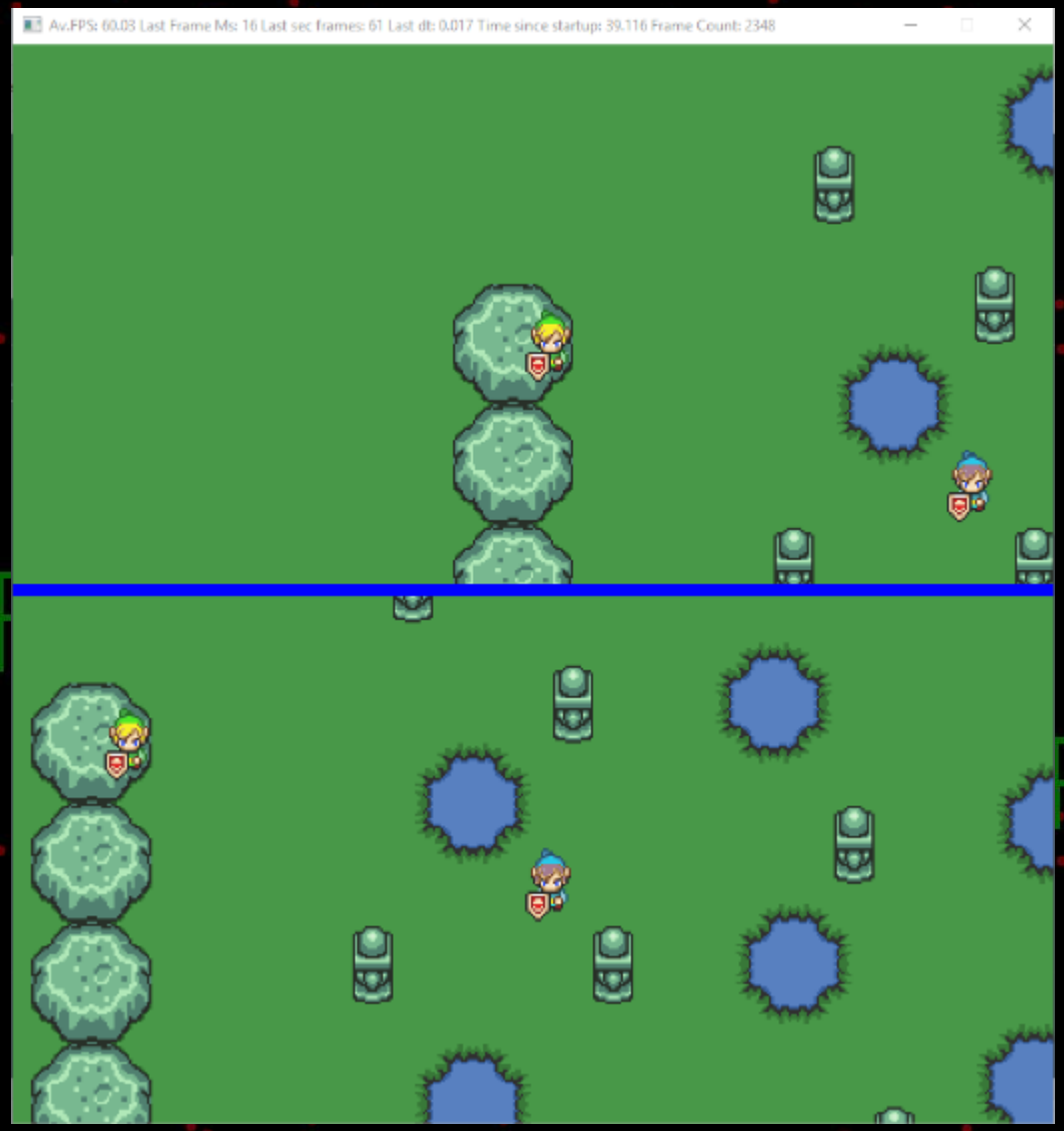
Quadtree, Where is used?

Image processing/compression



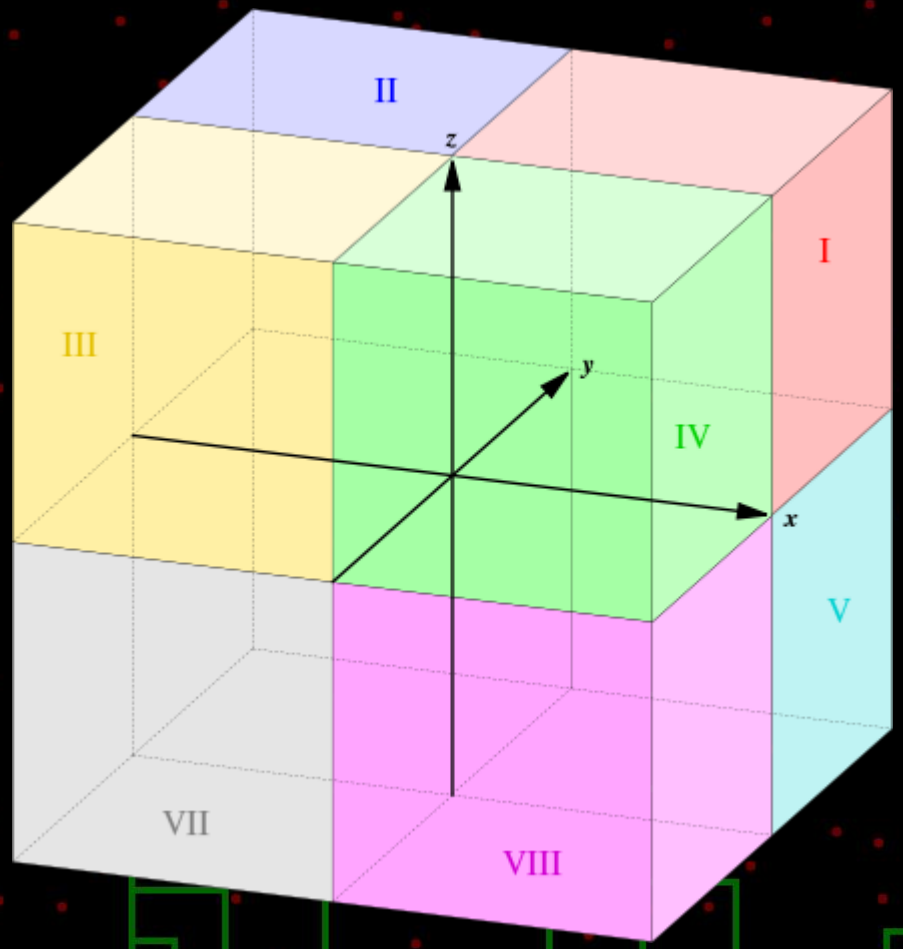
Quadtree, Where is used?

- Collisions
- Camera culling

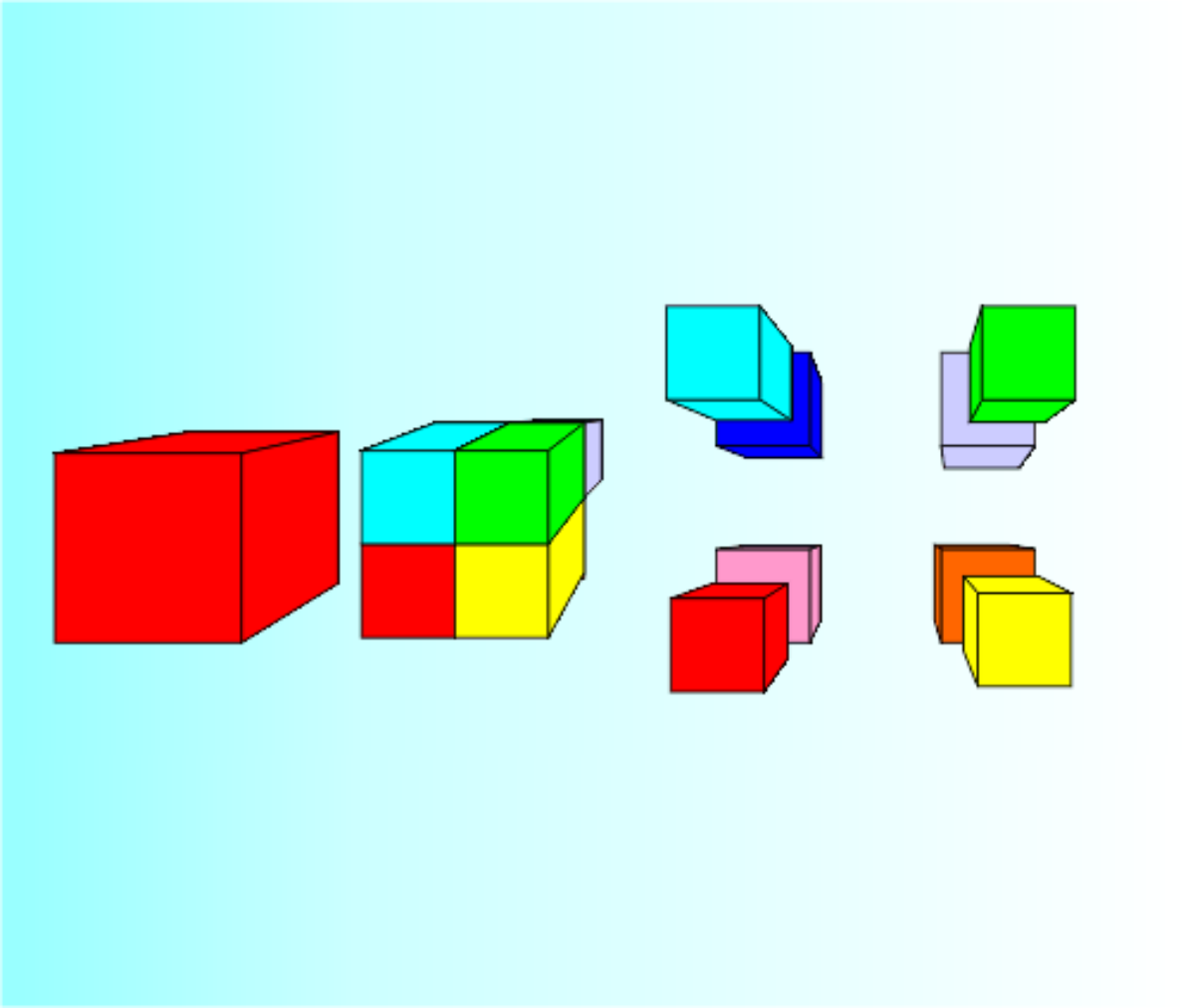


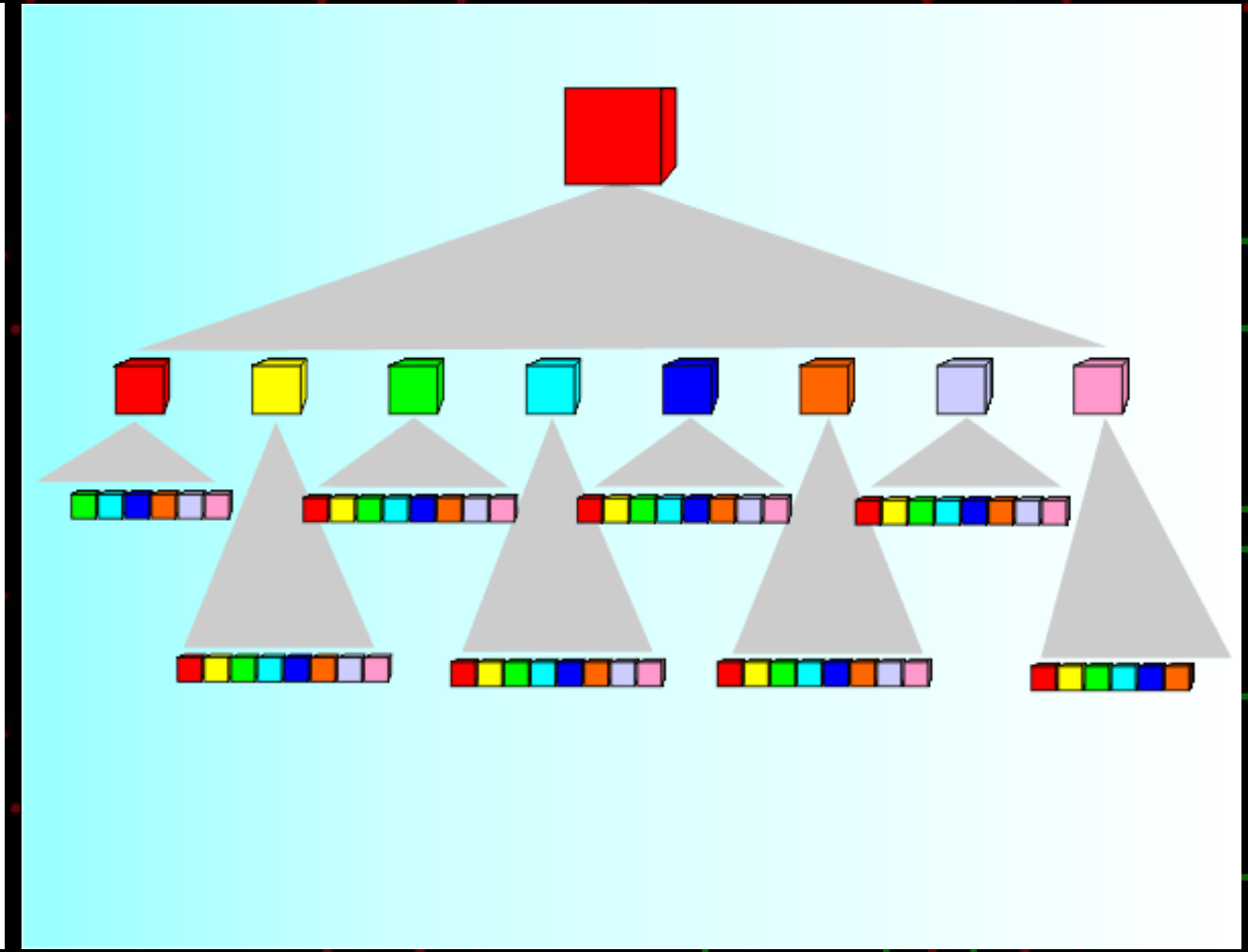
Octree

- Quadtree analogue in 3D
- 3D graphics and video game engines



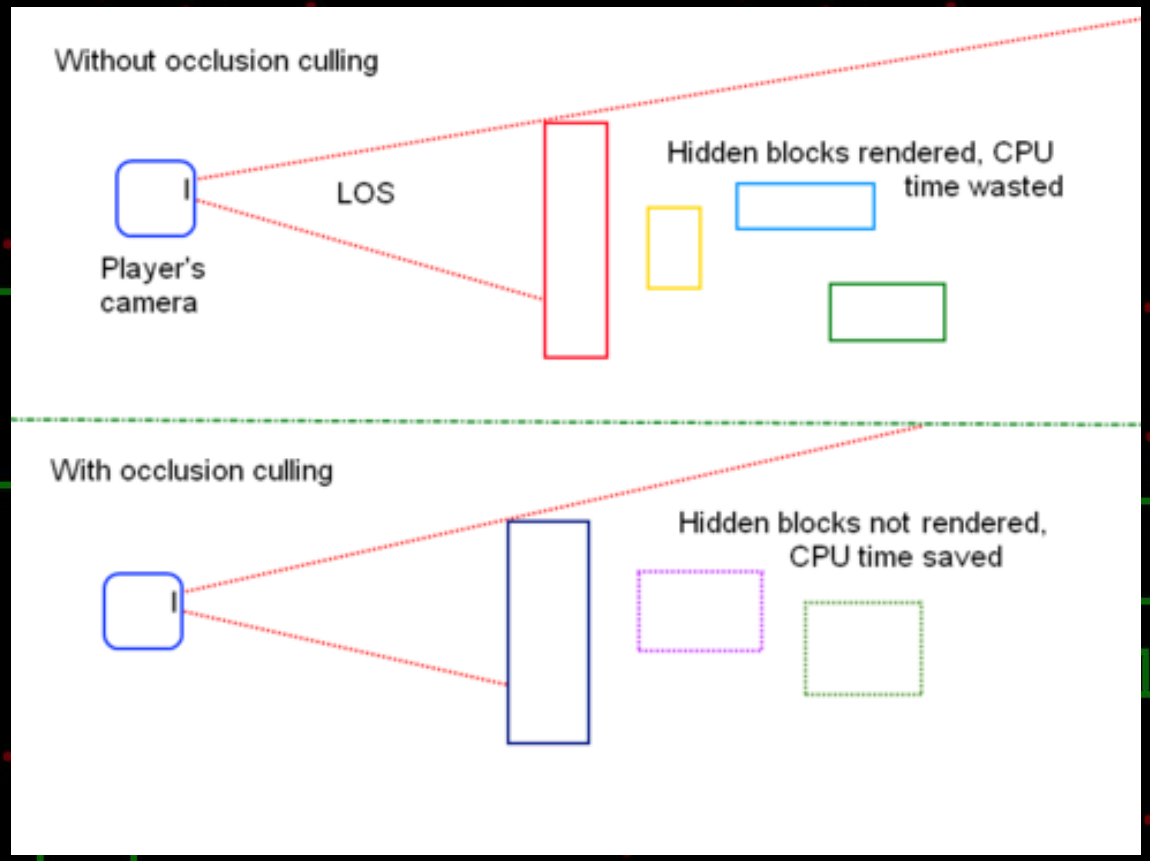
Octree, Tree

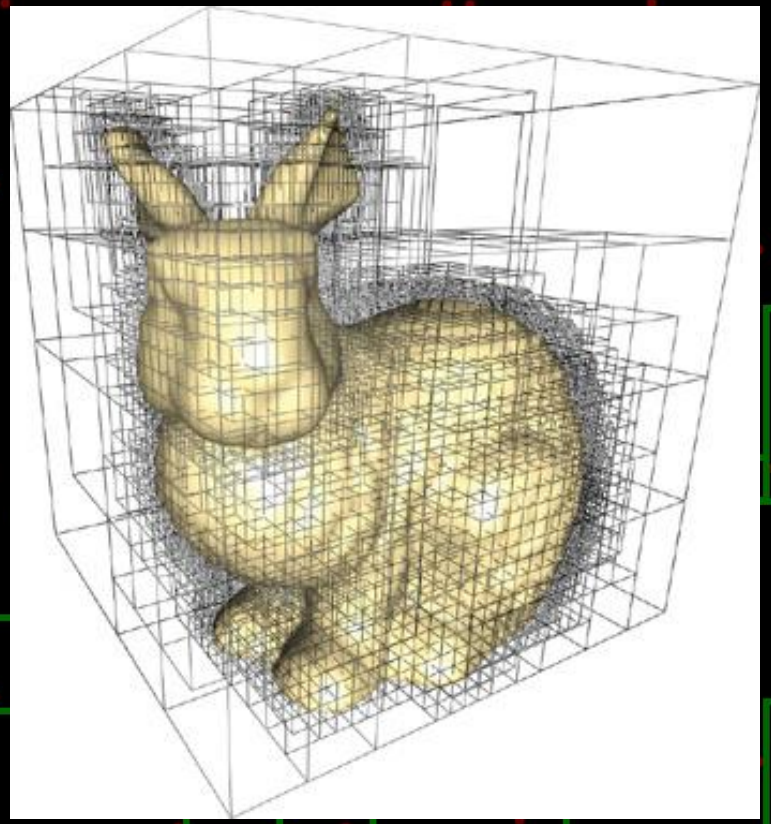




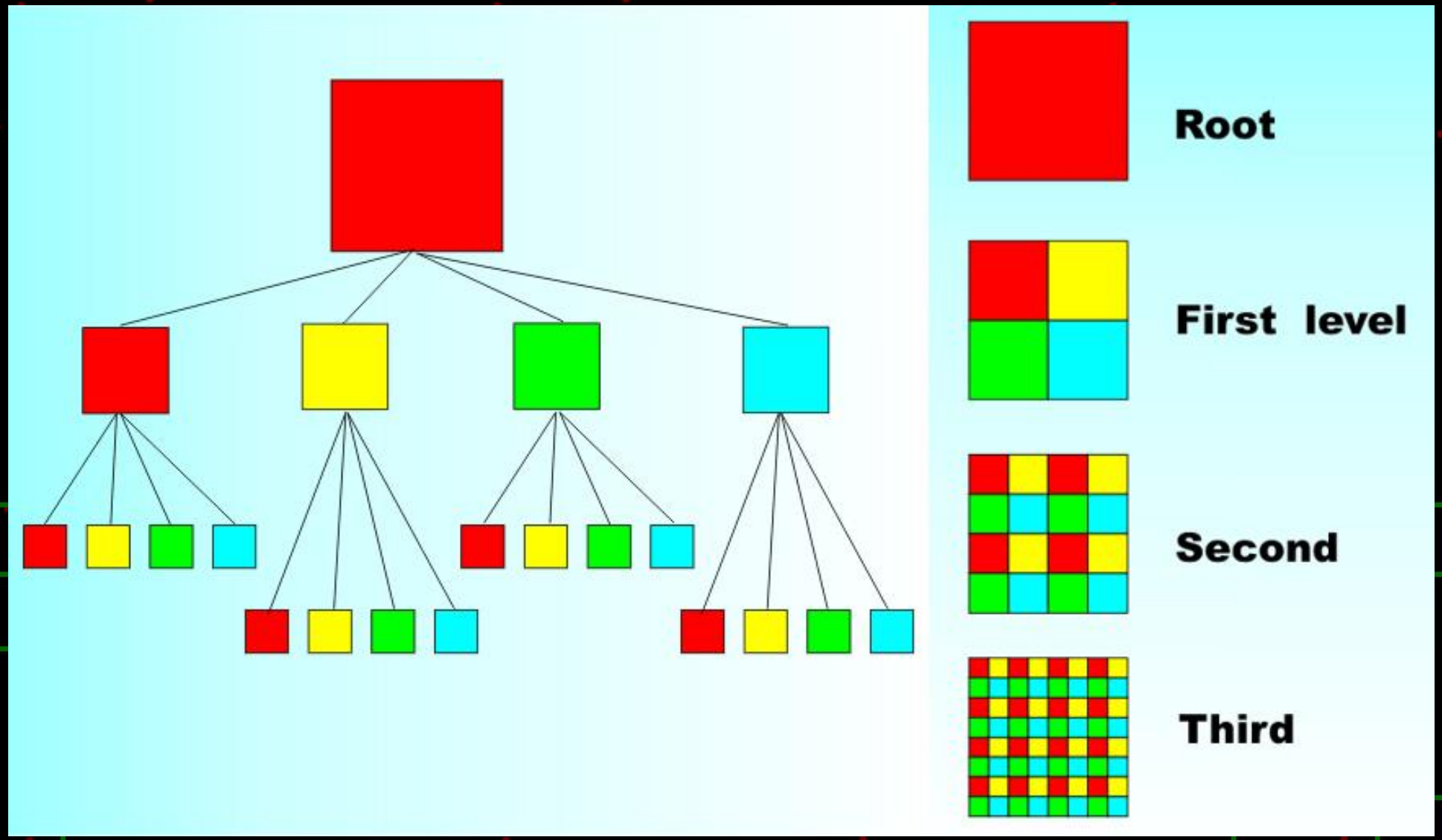
Octree, Where is used?

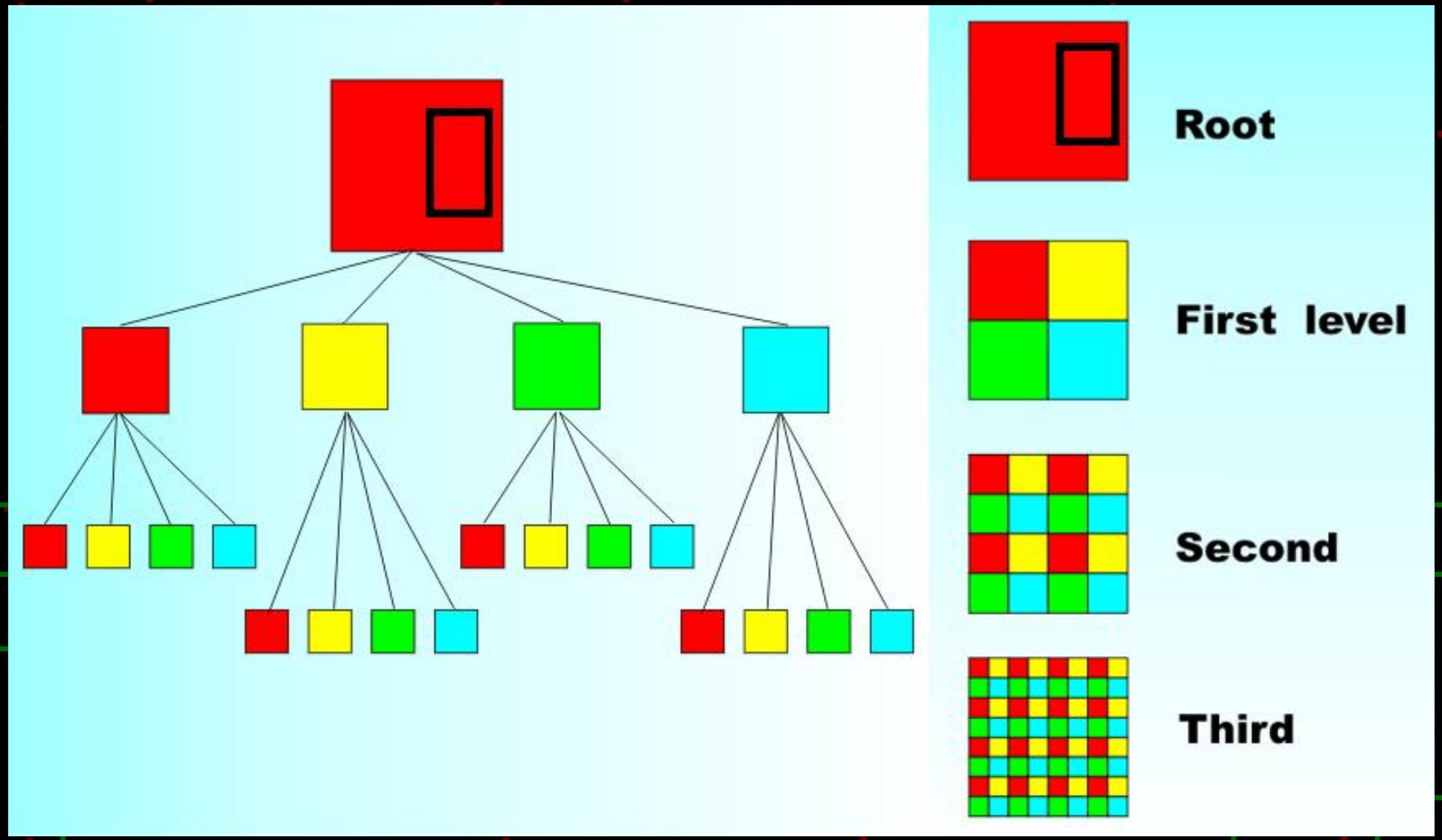
- 3D Graphics
- Efficient collision detection in three dimensions
- Occlusion Culling (OC)

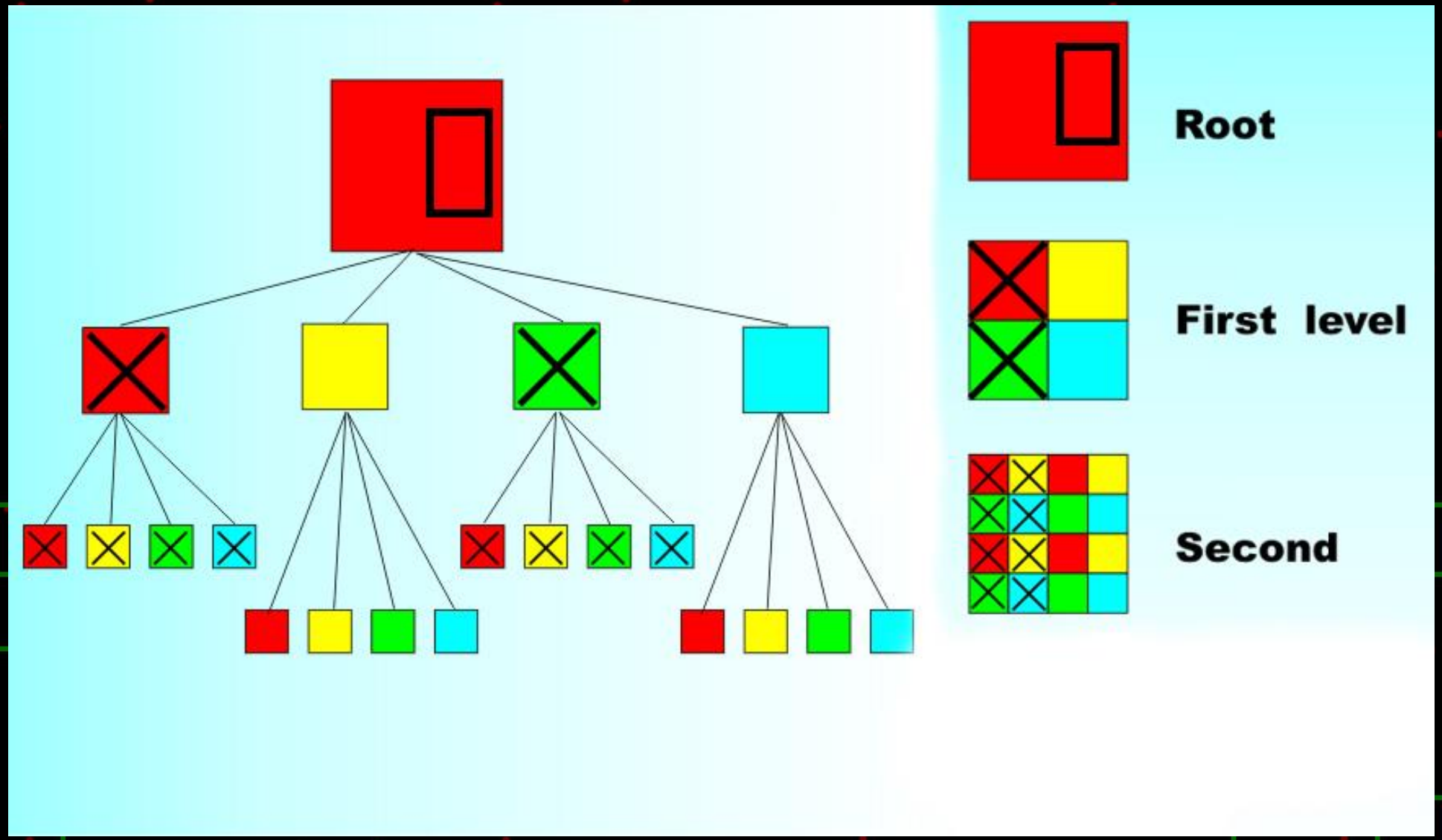


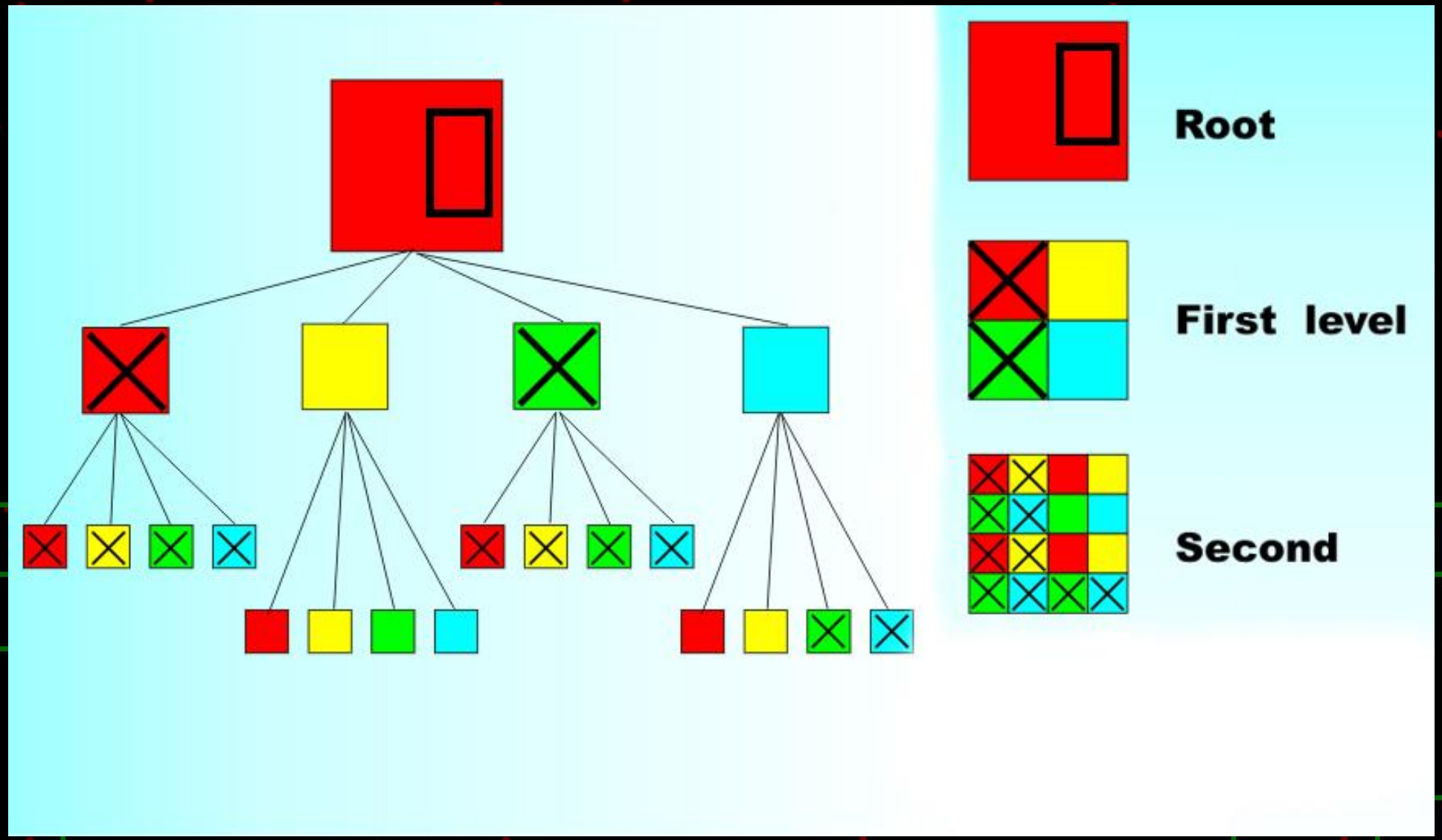


Why they are faster?

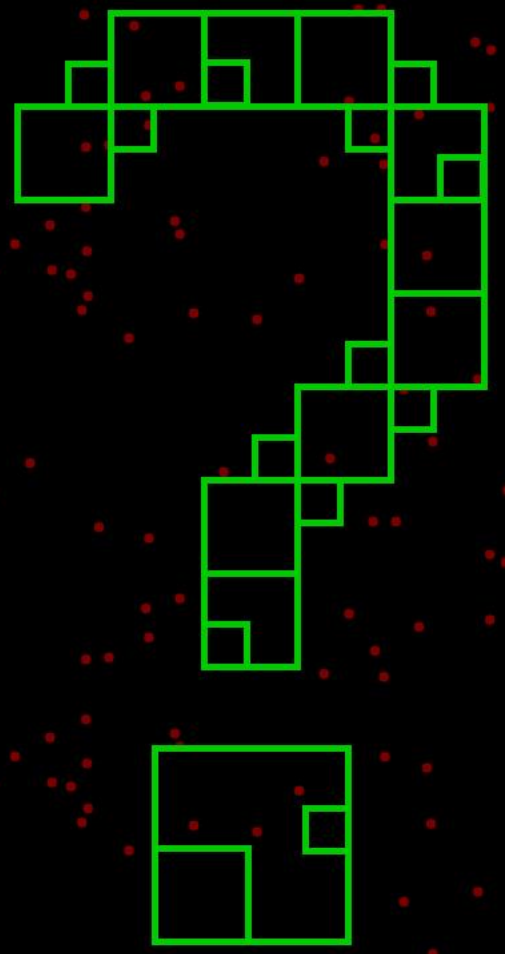

Root
First level
Second
Third


Root
First level
Second
Third


Root
First level
Second


Root
First level
Second

Any question?



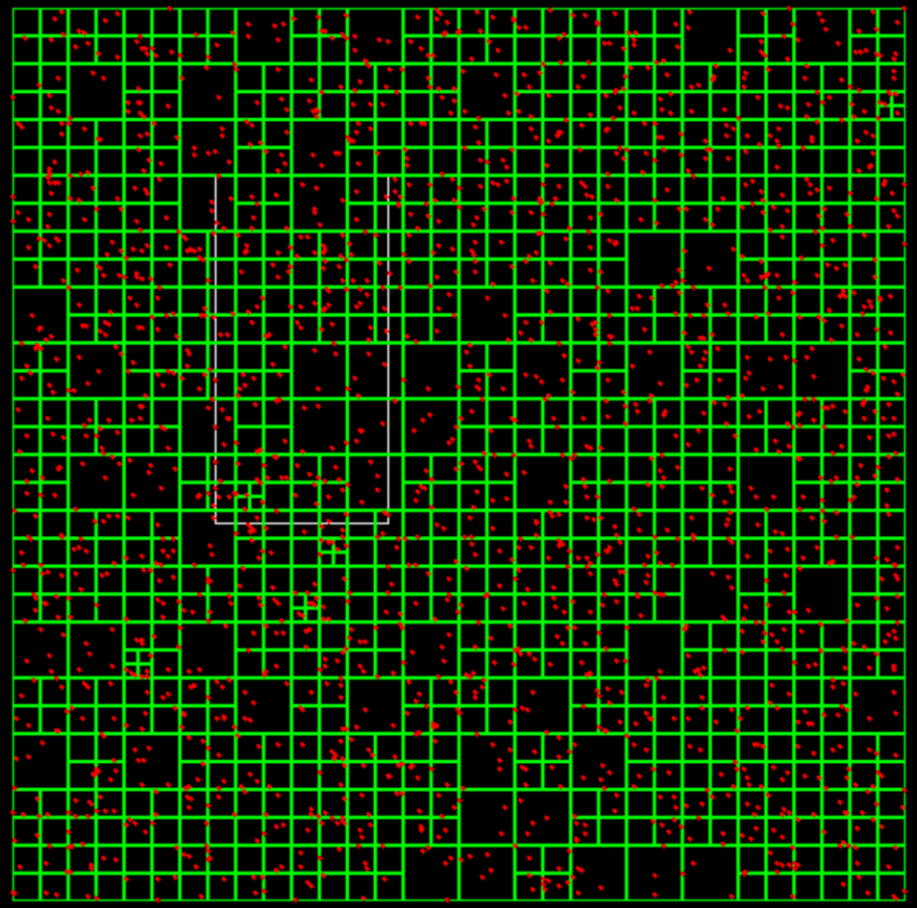
Before TODOs

- Olivenza_Xavier_Research_Optimized_Search_Manager_second_Iteration_faster
- Olivenza_Xavier_Research_Optimized_Search_Manager_second_Iteration_faster_with_TODOS
- Olivenza_Xavier_Research_Release
- Olivenza_Xavier_Research_Optimized_Search_Manager_ENG
- Olivenza_Xavier_Research_Optimized_Search_Manager_ESP
- Olivenza_Xavier_Research_Optimized_Search_Manager_PPTX
- Olivenza_Xavier_Research_Optimized_Search_Manager_PPTX

R,S,F1,F2,Q,A

Before TODOs

Normal Search Time: 1.242694 ms, Points in Range = 162 | Quadtree Search Time: 0.471734 ms, Points in Range = 162 | Quadtree AABBs = 1164 | Quadtree checks = 165 | Accepted points 2000/2000 | Mesh ste...



TODOs with the Quadtree implementation

Now get the release of my repository and proceed to do the TODOs in SDLQuadtree.cpp

TODO 1

```
bool AABB::Insert(iPoint* newpoint)
{
// TODO 1: The new point is inside the quadtree AABB?

return false;
}
```

TODO 2

```
bool AABB::Insert(iPoint* newpoint)
{
// TODO 1: The new point is inside the quadtree AABB?

// TODO 2:  Add the point to the node if there is space for it (remember Max_Elements_in_Same_Node)

return false;
}
```

TODO 3

```
void AABB::subdivide()
{
// TODO 3: Calculate the size and position of each of the 4 new nodes
}
```

TODO 4

```
void AABB::subdivide()
{
// TODO 3:  Calculate the size and position of each of the 4 new nodes

// TODO 4: Now we have all the positions and size of each number we
// can create each child with its AABB
// Don't forget to set child root
}
```

TODO 5

```
bool AABB::Insert(iPoint* newpoint)
{
// TODO 1: The new point is inside the quadtree AABB?

// TODO 2: Add the point to the node if there is space for it (remember Max_Elements_in_Same_Node)

// TODO 5: If there is not space, add the point to a subdivision

return false;
}
```

TODO 6

```
int AABB::CollectCandidates(std::vector< iPoint* >& nodes, const SDL_Rect& r)
{
uint ret = 0;
// TODO 6: Check if range is in the quadtree AABB
return ret;
}
```

TODO 7

```
int AABB::CollectCandidates(std::vector< iPoint* >& nodes, const SDL_Rect& r)
{
uint ret = 0;
// TODO 6: Check if range is in the quadtree AABB
// TODO 7: See if the points of this node are in range, get them
//(remember Max_Elements_in_Same_Node and ret to count how many points we check)
return ret;
}
```

TODO 8

```
int AABB::CollectCandidates(std::vector< iPoint* >& nodes, const SDL_Rect& r)
{
uint ret = 0;
// TODO 6: Check if range is in the quadtree AABB
// TODO 7: See if the points of this node are in range, get them
//(remember Max_Elements_in_Same_Node and ret to count how many points we check)
// TODO 8: If the node don't hace children, we can end
return ret;
}
```

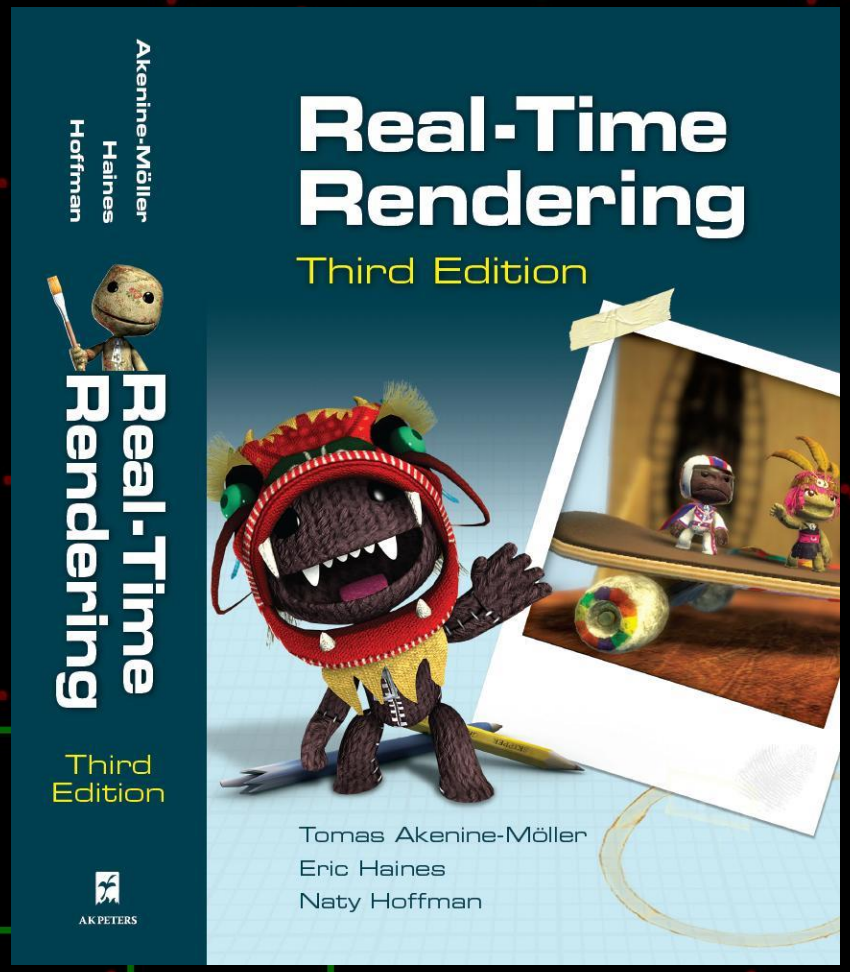
TODO 9

```
int AABB::CollectCandidates(std::vector< iPoint* >& nodes, const SDL_Rect& r)
{
uint ret = 0;
// TODO 6: Check if range is in the quadtree AABB
// TODO 7: See if the points of this node are in range, get them
//(remember Max_Elements_in_Same_Node and ret to count how many points we check)
// TODO 8: If the node don't hace children, we can end
// TODO 9: If the noide has children, get them points
return ret;
}
```

Optional Homework

·Moving entities?
Adapt the code for it.
Maybe you will need a Remove method…

Optional Homework



Search in Real-Time Video Games

Peter I. Cowling[1], Michael Buro[2], Michal Bida[3], Adi Botea[4], Bruno Bouzy[5], Martin V. Butz[6], Philip Hingston[7], Héctor Muñoz-Avila[8], Dana Nau[9], and Moshe Sipper[10]

1 University of York, UK
peter.cowling@york.ac.uk
2 University of Alberta, Canada
mburo@cs.ualberta.ca
3 Charles University in Prague, Czech Republic
Michal.Bida@mff.cuni.cz
4 IBM Research, Dublin, Ireland
adibotea@ie.ibm.com
5 Université Paris Descartes, France
bruno.bouzy@parisdescartes.fr
6 Eberhard Karls Universität Tübingen, Germany
butz@informatik.uni-tuebingen.de
7 Edith Cowan University, Australia
p.hingston@ecu.edu.au
8 Lehigh University, USA
munoz@eecs.lehigh.edu
9 University of Maryland, USA
nau@cs.umd.edu
10 Ben-Gurion University, Israel
sipper@cs.bgu.ac.il

Any last question?