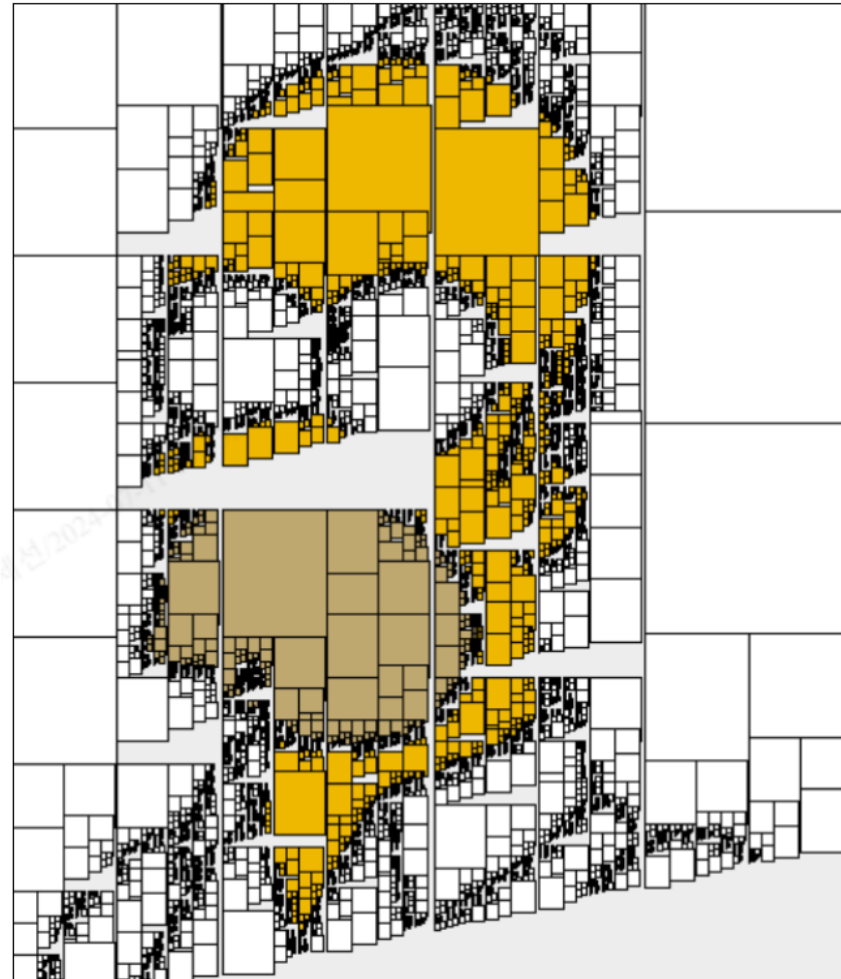


Quad Tree

Jeseon Lee

1. Background

- Quad Tree는 대표적인 공간 분할 자료 구조 중 하나임
- 특히, Quad Tree는 coNA엔진에서도 사용하는 공간 분할 자료구조이기에 꼭 알아 둘 필요가 있음



1. Background

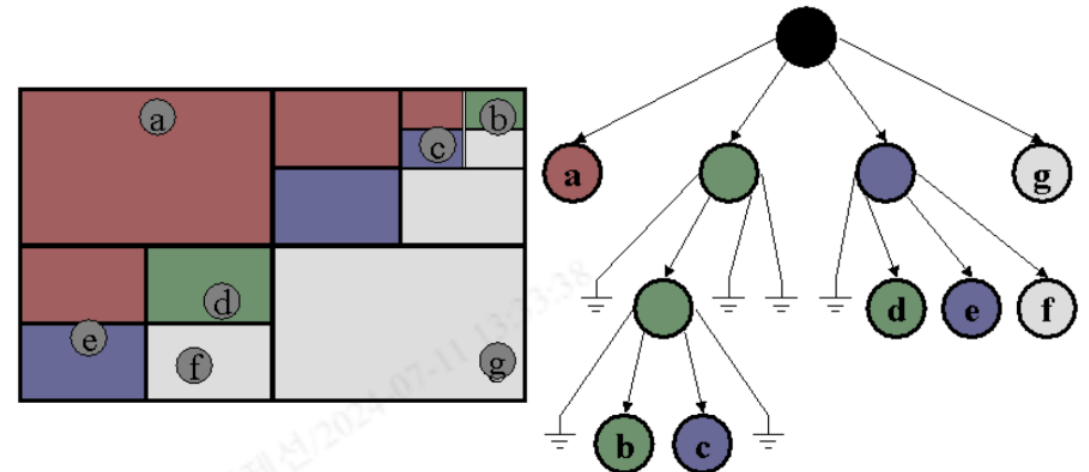
- 만약 지형 data가 하나의 통짜 Object의 경우 문제점?
 - 1) 카메라 Frustum에 들어오지 않더라도 Rendering 하게되어 불필요한 렌더링.
 - 2) 캐릭터와 지형 object 와 충돌 처리를 한다고 해도 보이지도 않는 영역과 충돌 연산이 필요.
 - => 위와 같은 이유로, 불필요한 렌더링 & 불필요한 충돌 연산으로 인해 공간 자료구조인 Quad Tree 필요성 강조



2. Quad Tree 알고리즘

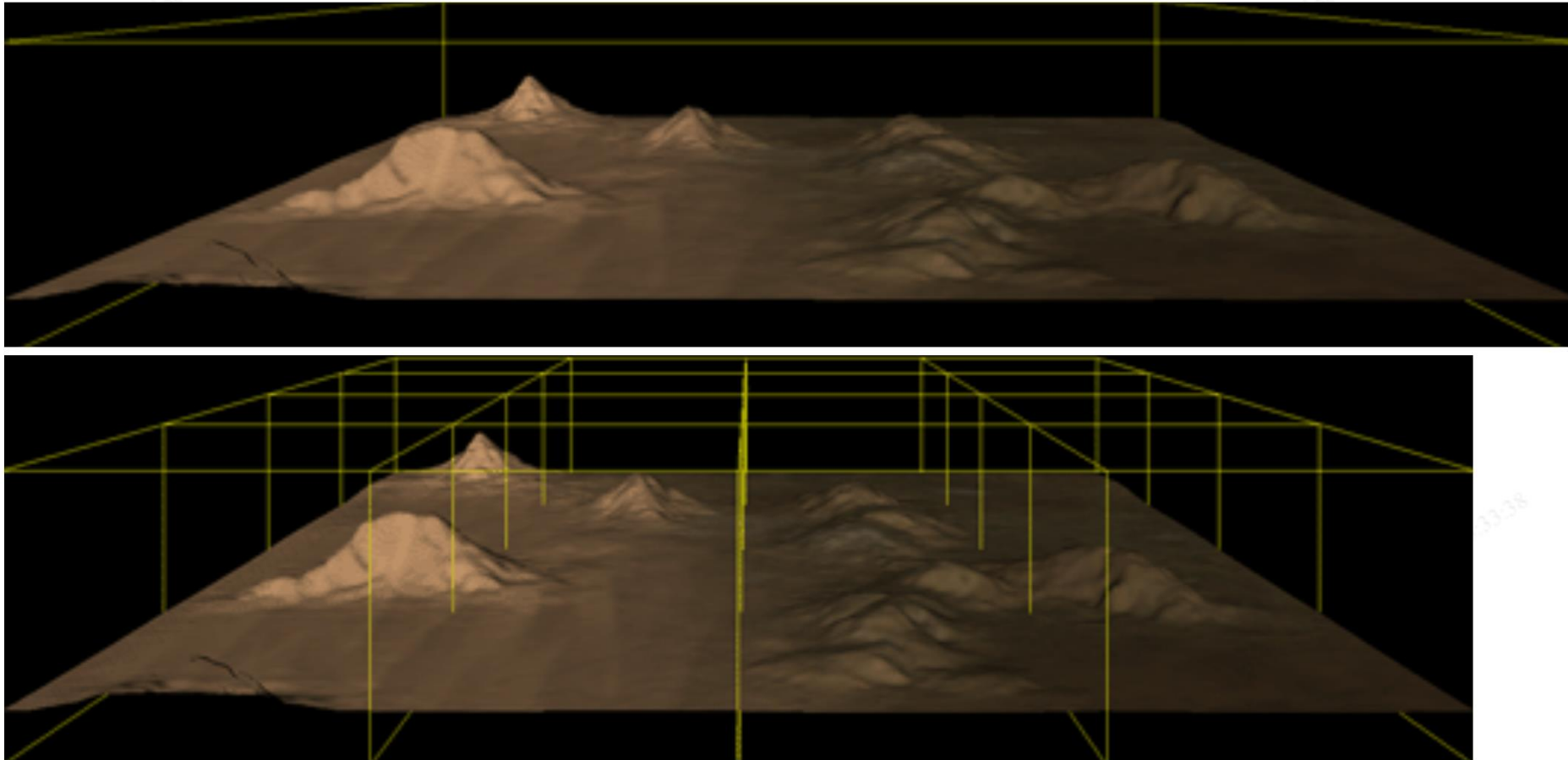
- 2차원 공간을 4개의 자식 노드로 재귀적 분할 방법(X,Y 방향으로 1/2씩 나눔).
- 알고리즘
 - (1) maximum recursion depth(최대 재귀 깊이) 지정
 - (2) scene의 maximum size 구하고, first cube size 설정
 - (3) cube에 포함 할 수 있고, children을 가지고 있지 않은 cube에 object를 삽입
 - (4) 최대 maximum recursion depth에 도달 하지 않았다면, 4개의 sub-cube로 분할
 - (5) sub-cube에 걸친 개수가 ($n > 1$)개이라면 분할 중지
 - (6) maximum recursion depth에 도달 할 때 까지 (3)으로 가서 반복
 - (7) 결론적으로, object에 딱 맞는(가장 작은) cube 도달

Quad Tree Example



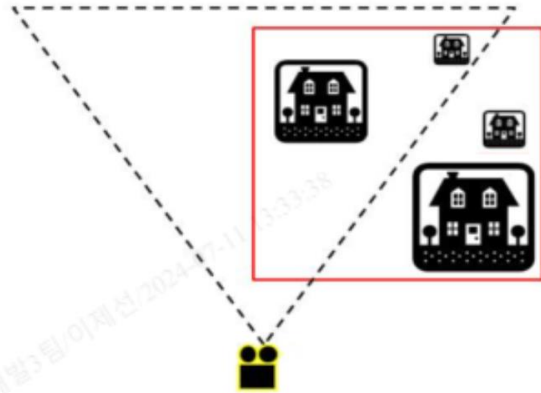
2. Quad Tree 알고리즘

- 결과 이미지 ([좌] Quad Tree적용 전, [우] Quad Tree적용 후)

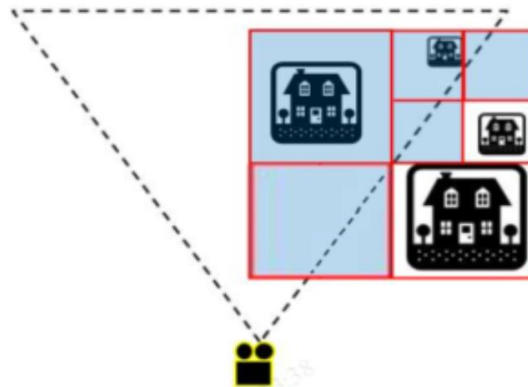


3. Quad Tree사용 예

- 1. Frustrum Culling의 경우



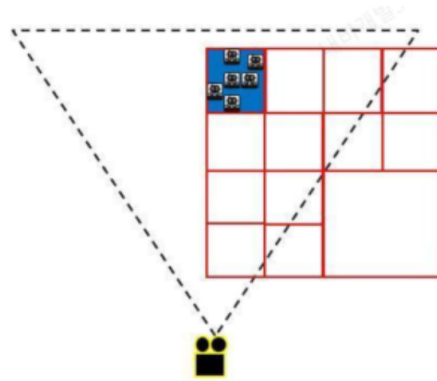
[공간분할 X] Object마다 모두 연산, 총 4번 (만약, object가 만 개라면 만 번 연산 => object와 의존성 O)



[공간분할 O] Node하고 연산(Object 개수에 의존성 X)

3. Quad Tree사용 예

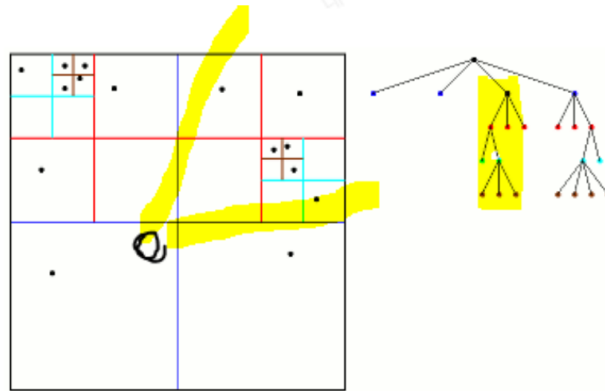
- 2. LOD(Level of Detail) 예시



[공간분할 X] Object 6개의 거리 연산

[공간분할 O] 카메라와 Node 거리 연산 1번

3. 충돌 알고리즘 예시



살아있는 Node 중에서도, 인접 영역의 object만 충돌처리로 최적화 가능(ex. 같은 Node 혹은 같은 children Node)

4. Quad Tree 문제점

- - 애매하게 걸쳐 있을 경우
- -> 충분히 작은데도 상위 공간(부모 Node)으로 들어감

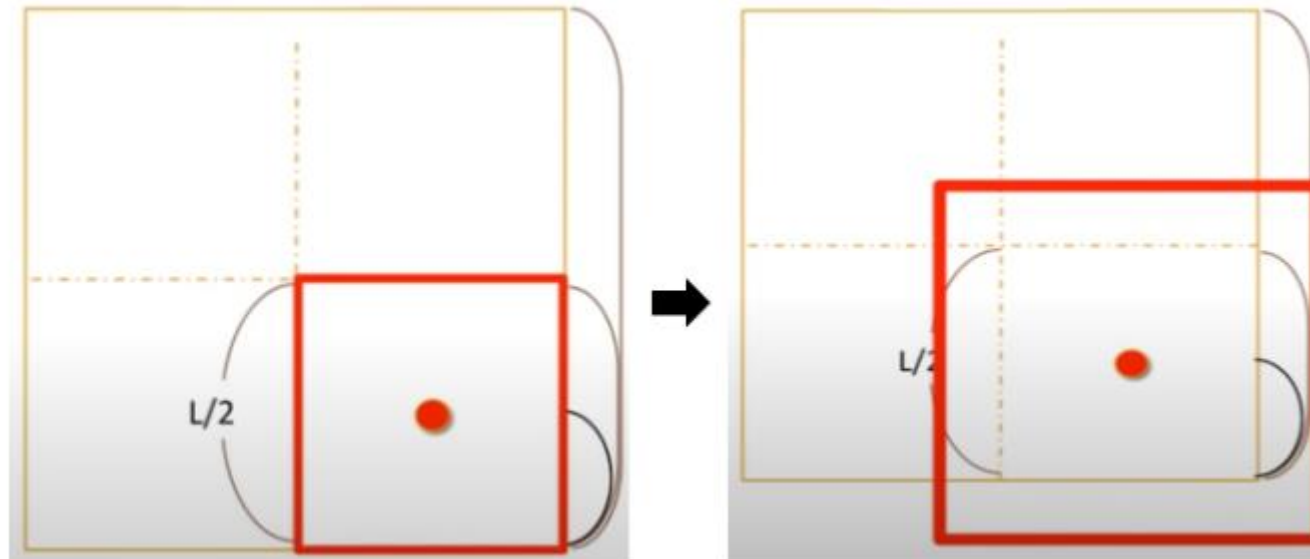


=> 공간을 너무 칼 같이($x, y, z \ 1/2$) 나눠서 발생하는 문제

4. Quad Tree 문제점

- 해결 방법

- 즉, "단위 사각형의 길이/2" => " $k * (\text{단위 사각형의 길이}/2)$ "



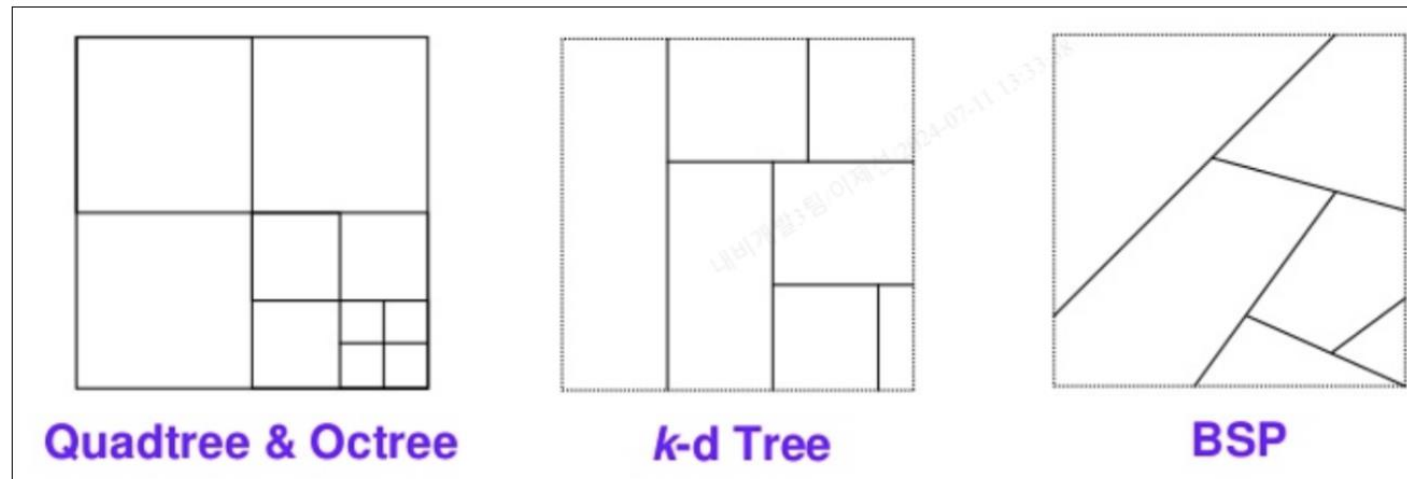
5. "QuadTree&Octree" vs "k-d tree" vs "BSP tree" 비교

- pros

- 쿼드 트리 이름 답게, 4개 분할은 더 얇은 tree를 생성하게함
- k-d tree/BSP tree는 이진 트리이기 때문에 depth가 깊어질 수 밖에 없음
- 불균형 tree이므로, 삽입/삭제 시 재 조정되는 overhead가 발생하지 않음. tree 생성 시 적은 비용
- k-d tree는 균형 이진 트리이므로 재 조정 연산 필요
- ★ 자식 단위로 분할 될 때, 분할 기준이 정해져 있으므로 연산이 적음
- k-d tree/BSP tree 분할 기준 예시: 10개 정점이 있다면 그 중간(5번 째) 점을 탐색하고 분할함. 즉, 기준을 잡으려 탐색하는 연산량이 큼

- cons

- 한 쪽으로 치우쳐지는 불균형 tree으로 인해 검색 시간 증가 될 수 있음. 즉, 순차 탐색처럼 될 수 있음.(균형 트리 보장x)



Q n A