



3D게임프로그래밍

-CHAPTER10-

SOULSEEK



목차

1. QuardTree



The background is a dark blue gradient with a large, faint, light blue circle in the center. In the four corners, there are decorative white line art elements resembling circuit boards or branching structures, with small circles at the end of the lines.

QUARDTREE

1. QUARDTREE

정의

- 사분트리, 자식노드가 4개인 트리
- 공간을 4개의 자식노드로 재귀적 분할하는 방법이다.
- 한 공간을 1/4로 계속 분할한다.
- 하나였던 지형이 1/4로 분할, 2단계에서 다시 또 1/4로 분할, 3단계에서 또 1/4로 분할..
- 이러다가 일정 크기 이하로 줄어들면 분할을 멈추는 것.

효과

- 쿼드트리를 사용하는 가장 큰 이유는 거대한 지형을 빠르게 검색할 수 있기 때문
- 큰 덩어리 단위로 필요 없는 데이터를 제거함으로써, 3D엔진이 처리해야 할 데이터양을 빠르게 줄일 수 있다.

1. QUADTREE

```
class ZQuadTree
```

```
{
```

```
    /// 쿼드트리에 보관되는 4개의 코너값에 대한 상수값
```

```
    enum CornerType { CORNER_TL, CORNER_TR, CORNER_BL, CORNER_BR };
```

```
private:
```

```
    ZQuadTree* m_pChild[4]; /// QuadTree의 4개의 자식노드
```

```
    int m_nCenter; /// QuadTree에 보관할 첫번째 값
```

```
    int m_nCorner[4]; /// QuadTree에 보관할 두번째 값
```

```
    /// TopLeft(TL)      TopRight(TR)
```

```
    ///      0-----1
```

```
    ///      |       |
```

```
    ///      |       |
```

```
    ///      2-----3
```

```
    /// BottomLeft(BL)   BottomRight(BR)
```

```
public:
```

```
    /// 최초 루트노드 생성자
```

```
    ZQuadTree( int cx, int cy );
```

```
    /// 하위 자식노드 생성자
```

```
    ZQuadTree( ZQuadTree* pParent );
```

```
    /// 소멸자
```

```
    ~ZQuadTree();
```

```
    /// QuadTree를 구축한다.
```

```
    BOOL Build();
```

```
    /// 삼각형의 인덱스를 만들고, 출력할 삼각형의 개수를 반환한다.
```

```
    int GenerateIndex( LPVOID pIB );
```

```
};
```

```
private:
```

```
    /// 자식 노드를 추가한다.
```

```
    ZQuadTree* _AddChild( int nCornerTL, int nCornerTR, int nCornerBL, int nCornerBR );
```

```
    /// 4개의 코너값을 셋팅한다.
```

```
    BOOL _SetCorners( int nCornerTL, int nCornerTR, int nCornerBL, int nCornerBR );
```

```
    /// Quadtree를 4개의 하위 트리로 부분분할(subdivide)한다.
```

```
    BOOL _SubDivide();
```

```
    /// 현재 노드가 출력이 가능한 노드인가?
```

```
    BOOL _IsVisible() { return ( m_nCorner[CORNER_TR] - m_nCorner[CORNER_TL] <= 1 ); }
```

```
    /// 출력할 폴리곤의 인덱스를 생성한다.
```

```
    int _GenTriIndex( int nTriangles, LPVOID pIndex );
```

```
    /// 메모리에서 쿼드트리를 삭제한다.
```

```
    void _Destroy();
```

1. QUARDTREE

- 쿼드트리 클래스는 쿼드트리가 점유할 지형의 크기를 생성자를 통해서 입력한 뒤 **Build()** 함수를 호출하면 내부적인 재귀호출을 통해서 쿼드트리를 생성하게 되어있다. 이 때, **생성할 지형의 크기는 반드시 2^{n+1}** 이어야만 한다, 그 외의 크기는 쿼드트리로 분할이 되지 않으므로 주의 해야 한다.

129 * 129 크기인 지형일 경우 노드에 저장되는 인덱스 값의 예를 들면..

