

게임엔진

## 제5강 씬 구성

이대현 교수



# 학습 안내

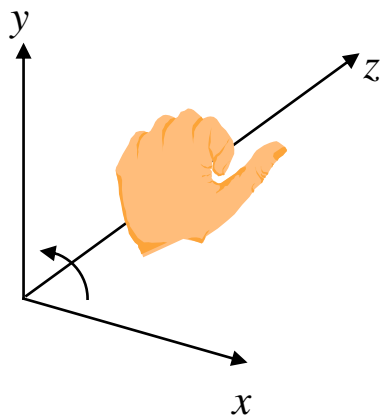
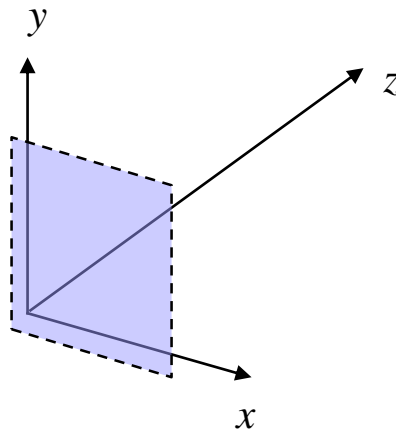
## ■ 학습 목표

- 3D 공간에서 캐릭터를 배치하여 3D 씬을 구성하는 방법을 이해한다.

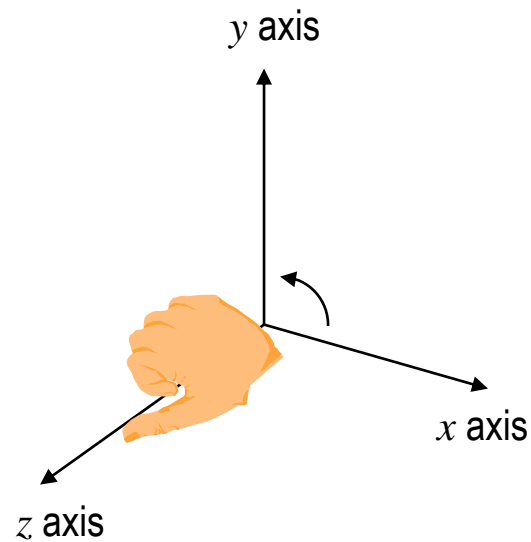
## ■ 학습 내용

- 3차원 좌표계와 벡터
- 씬노드의 위치 설정 및 이동
- 간단한 카메라워크의 구현

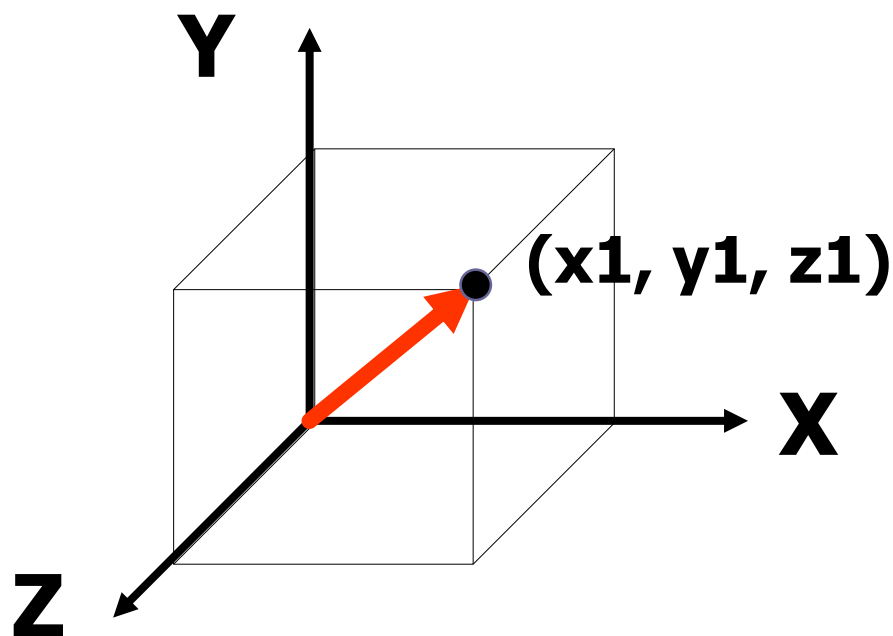
# 3차원 좌표계



왼손좌표계  
예) Direct3D



오른손좌표계  
예) OGRE



**Vector3(x1, y1, z1)**

- OGRE의 벡터 클래스.
- Vector2, Vector3, Vector4 가 있음.

Step 1: 광원의 설정(setAmbientLight)

Step 2: 엔터티의 생성(createEntity)

Step 3: 씬노드의 생성(createChildSceneNode)

Step 4: 씬노드에 엔터티를 배치(attachObject)

실습



## *Professor & Ninja* 두명의 캐릭터 렌더링



```
void go(void)
{
    // ... 전략

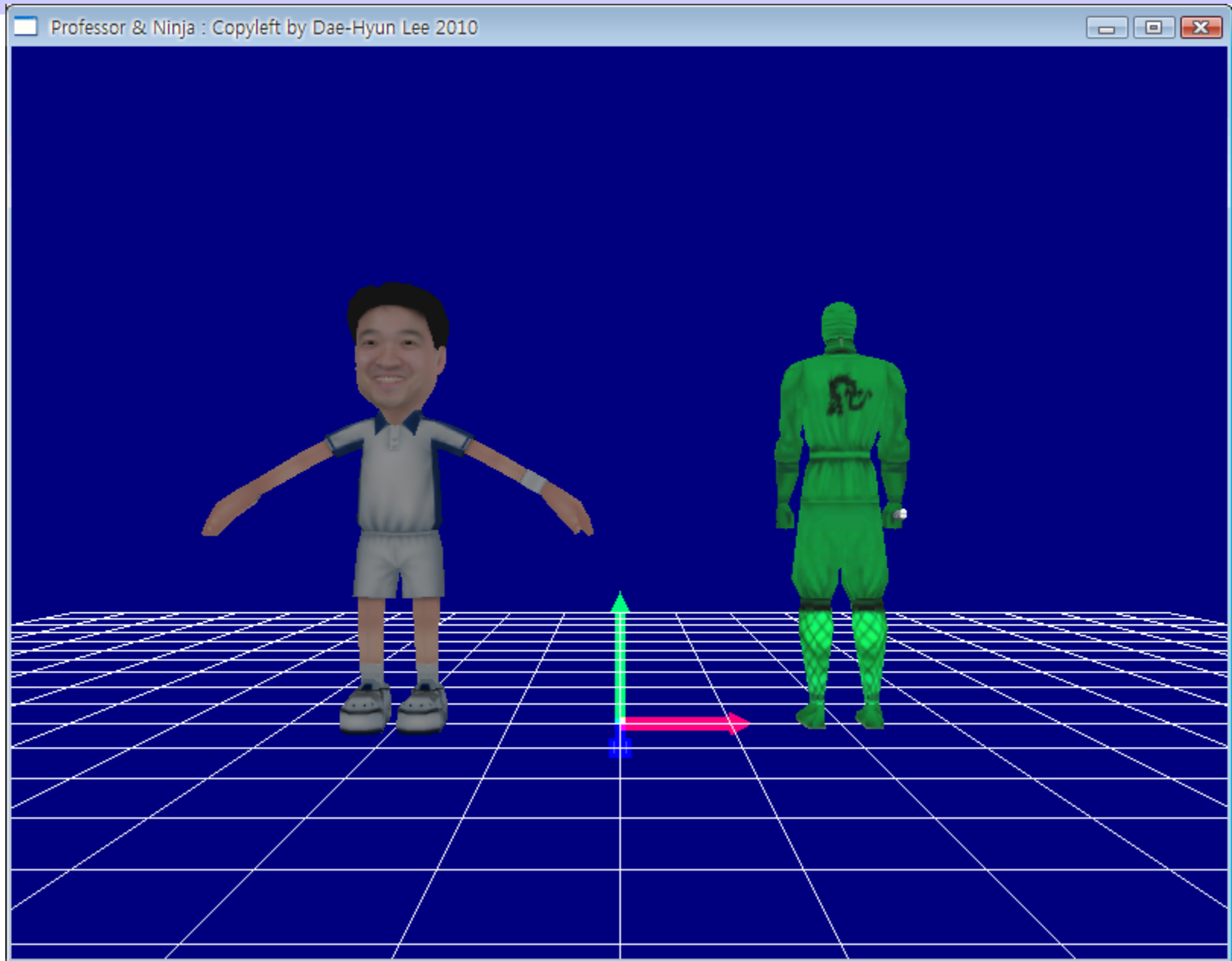
    Entity* entity1 = mSceneMgr->createEntity("Professor", "DustinBody.mesh");
    Entity* entity2 = mSceneMgr->createEntity("Ninja", "ninja.mesh");

    SceneNode* node1 = mSceneMgr->getRootSceneNode()->
        createChildSceneNode("Professor", Vector3(-100.0f, 0.0f, 0.0f));
    node1->attachObject(entity1);

    SceneNode* node2 = mSceneMgr->getRootSceneNode()->
        createChildSceneNode("Ninja", Vector3(100.0f, 0.0f, 0.0f));
    node2->attachObject(entity2);

    // ... 후략
}
```

# 실행 화면



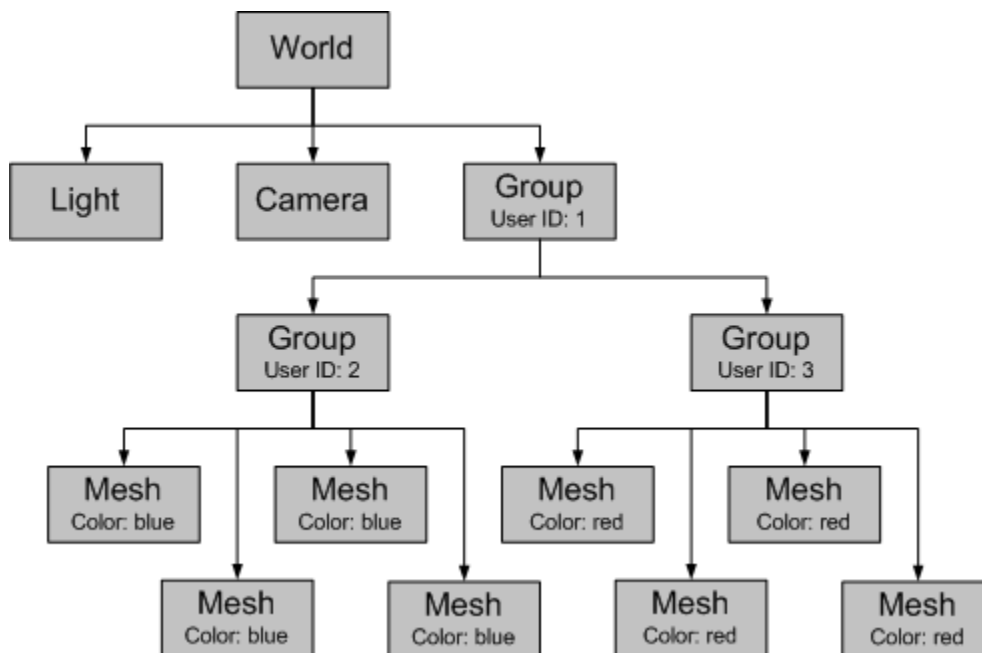


# 씬 그래프(Scene Graph)

- 오우거 엔진의 씬 구성은 씬 그래프의 구조를 만드는 과정임.

- 씬 그래프

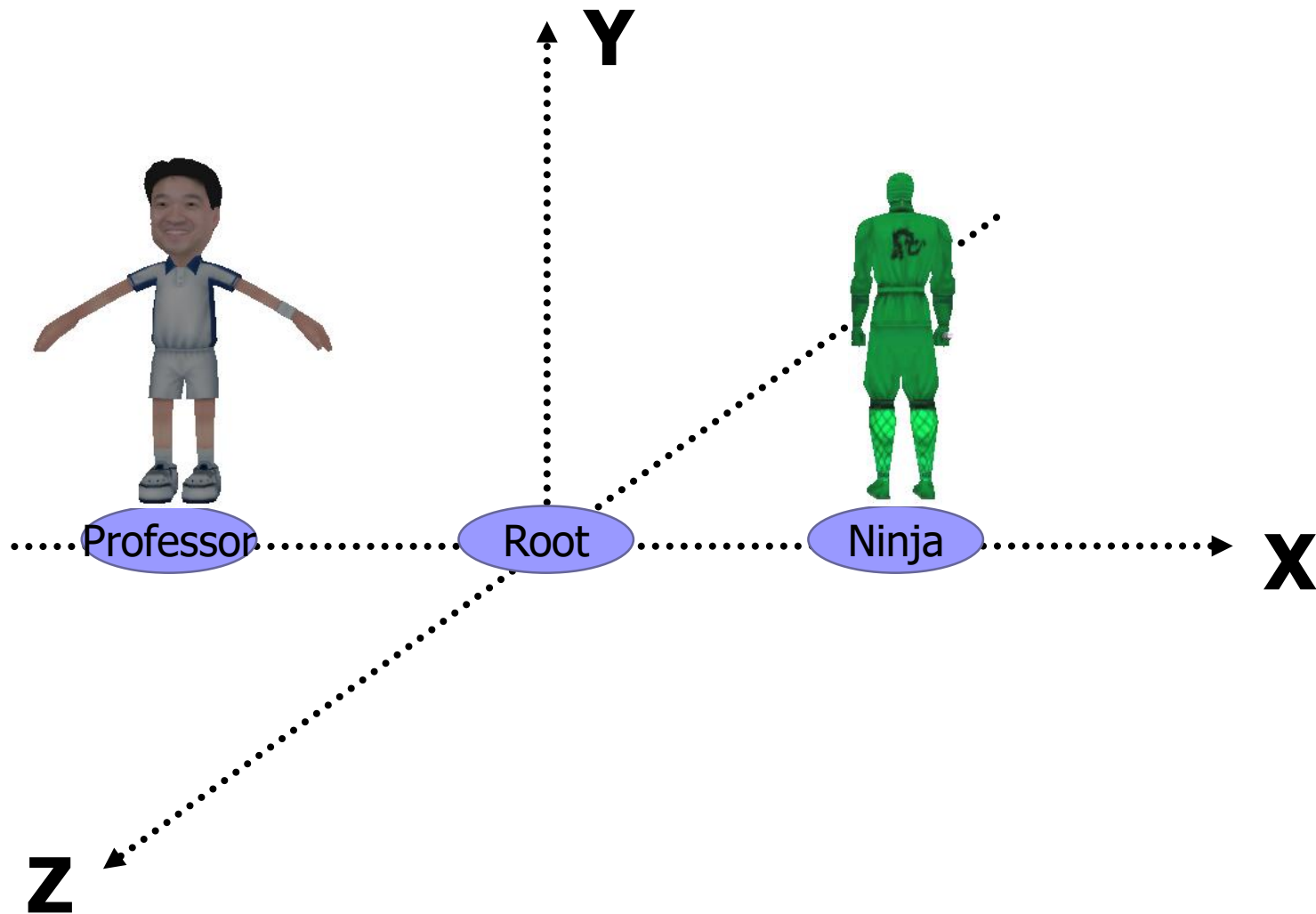
- 그래픽 씬을 나타내기 위한 표현 방식으로써, 일반적으로 공간적 정보를 담아 논리적으로 표현할 수 있는 구조를 지님.
- 씬 노드(Scene Node)들의 트리(Tree)
- 씬 노드는 엔터티(메쉬모델)를 담고 있으며, 수학적으로는 Local Axis의 역할을 하는 변환 행렬임.



Sample World Scene Graph

(Source: <http://www.ibm.com/developerworks/java/library/wi-mobile2/index.html>)

# 예제 씬 구성




# 개발자와 모델러와 사전에 합의해야 할 것들...

- 모델의 Pivot 위치
- 모델의 초기 방향
- 1 포인트의 실제 매핑 거리

# 씬 노드의 초기 위치 지정

```
SceneNode* node1 = mSceneMgr->getRootSceneNode()->  
    createChildSceneNode("Professor", Vector3(-100.0f, 0.0f, 0.0f));  
  
SceneNode* node2 = mSceneMgr->getRootSceneNode()->  
    createChildSceneNode("Ninja", Vector3(100.0f, 0.0f, 0.0f));
```

- 
- 초기 위치 지정.

# 씬 노드 생성 함수 자세히 알아 보기

```
SceneNode* Ogre::SceneNode::createChildSceneNode(  
    const String & name,  
    const Vector3 & translate = Vector3::ZERO,  
    const Quaternion & rotate = Quaternion::IDENTITY  
)
```

• 씬 노드 이름

- 초기 회전값.
- 사원수(Quaternion) 값을 이용하여 지정.
- default 값은 단위 행렬 → 회전 없음.

- 초기 위치.
- 벡터값을 이용하여 지정.
- default 값은 원점.

# SceneNode 클래스의 위치 설정 함수

- const Vector3 & getPosition (void)
  - 씬노드의 현재 위치를 얻는다.
- void setPosition (const Vector3 &pos)
  - 씬노드의 위치를 재설정.
  - 씬노드의 부모노드의 위치를 기준으로 하는 상대값임.

# Parent - Child 구성

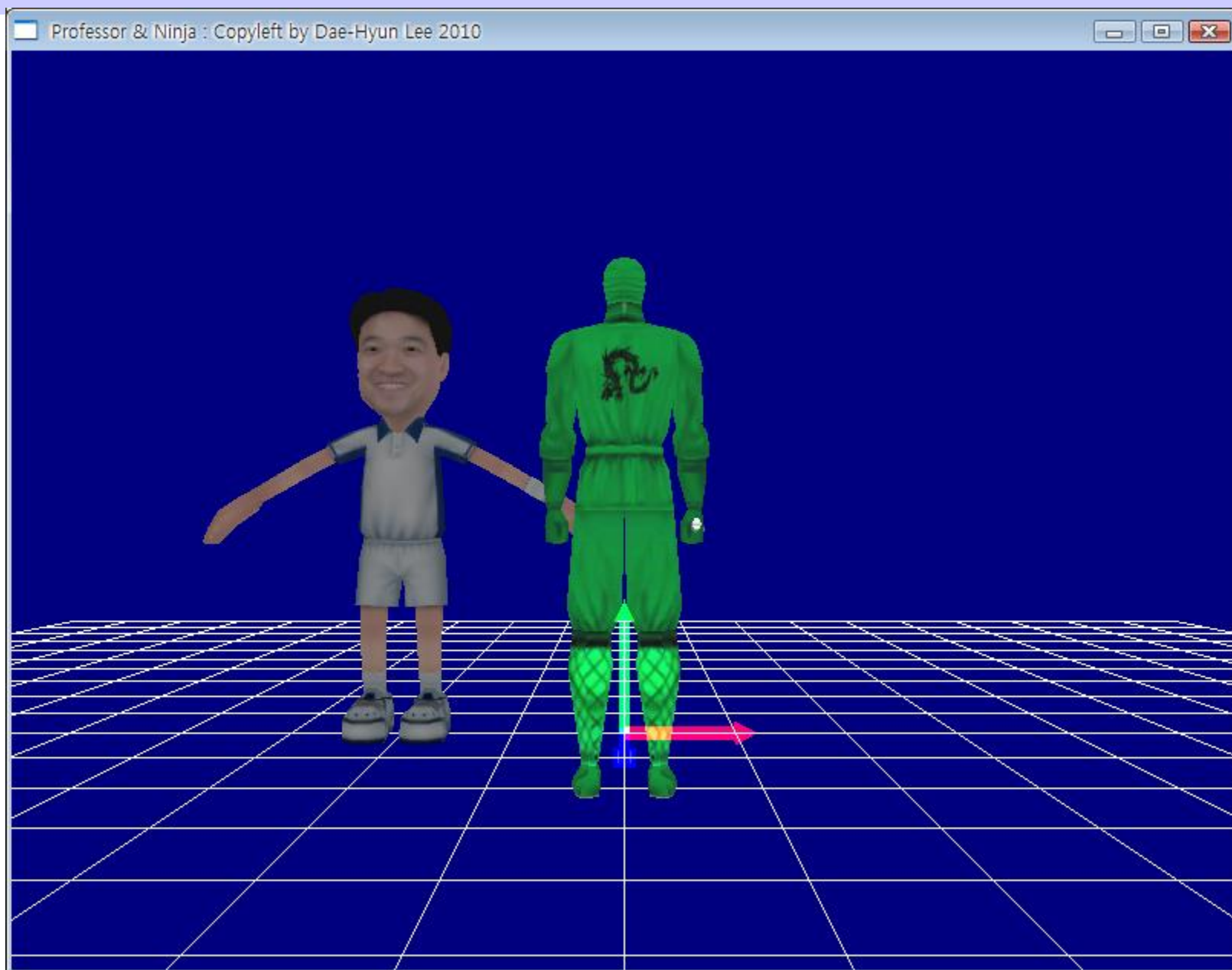


```
void go(void)
{
    // ... 전략

    SceneNode* node1 = mSceneMgr->getRootSceneNode()->
        createChildSceneNode("Professor", Vector3(-100.0f, 0.0f, 0.0f));
    node1->attachObject(entity1);

    SceneNode* node2 = node1->createChildSceneNode("Ninja", Vector3(100.0f, 0.0f, 100.0f));
    node2->attachObject(entity2);

    // ... 후략
}
```





# 씬노드의 이동(Translate)



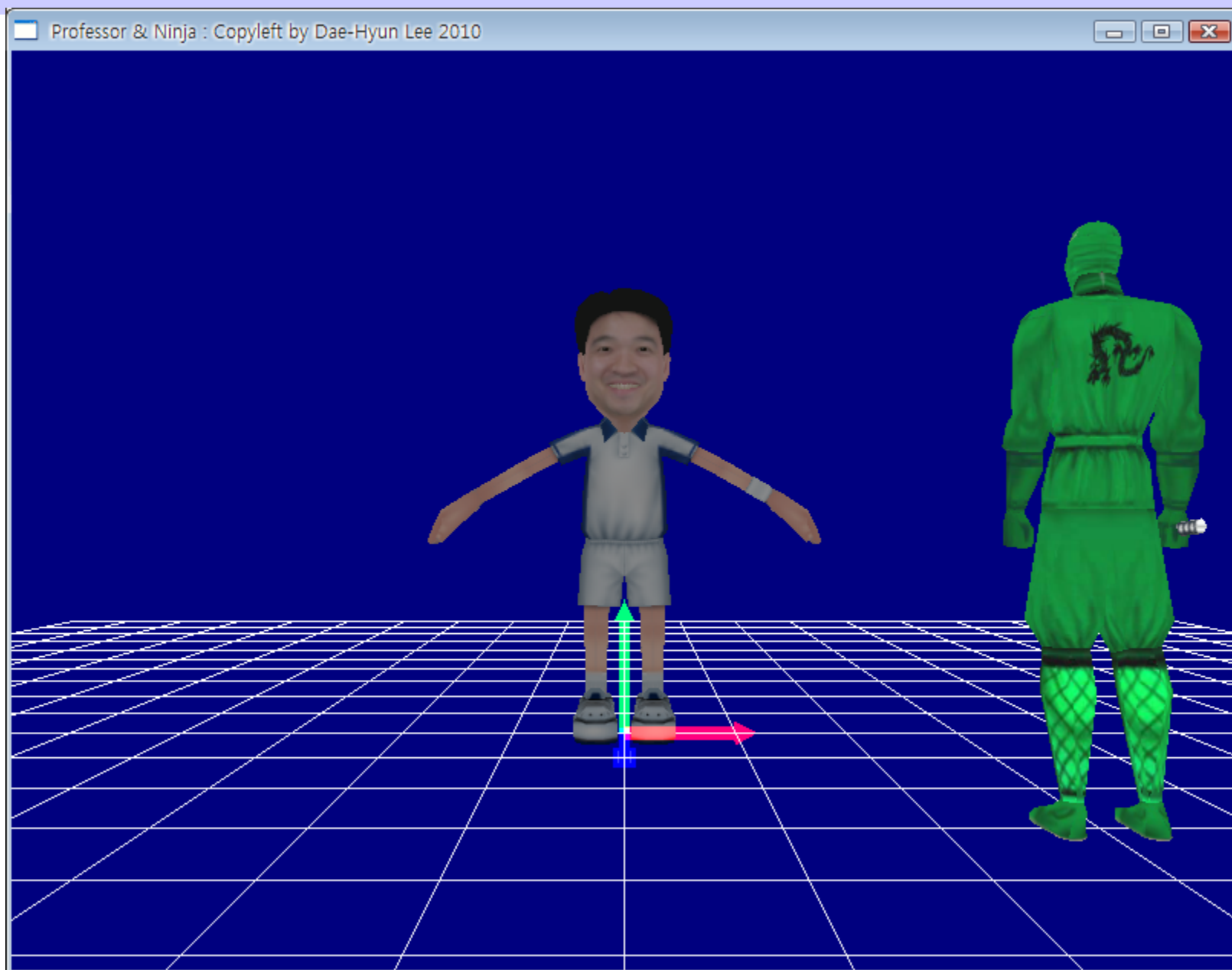
```
void go(void)
{
    // ... 전략

    SceneNode* node1 = mSceneMgr->getRootSceneNode()->
        createChildSceneNode("Professor", Vector3(-100.0f, 0.0f, 0.0f));
    node1->attachObject(entity1);

    SceneNode* node2 = node1->createChildSceneNode("Ninja", Vector3(100.0f, 0.0f, 100.0f));
    node2->attachObject(entity2);

    node2->translate(Vector3(50, 0, 50.0f));
    node1->translate(Vector3(100.0f, 0, 0));

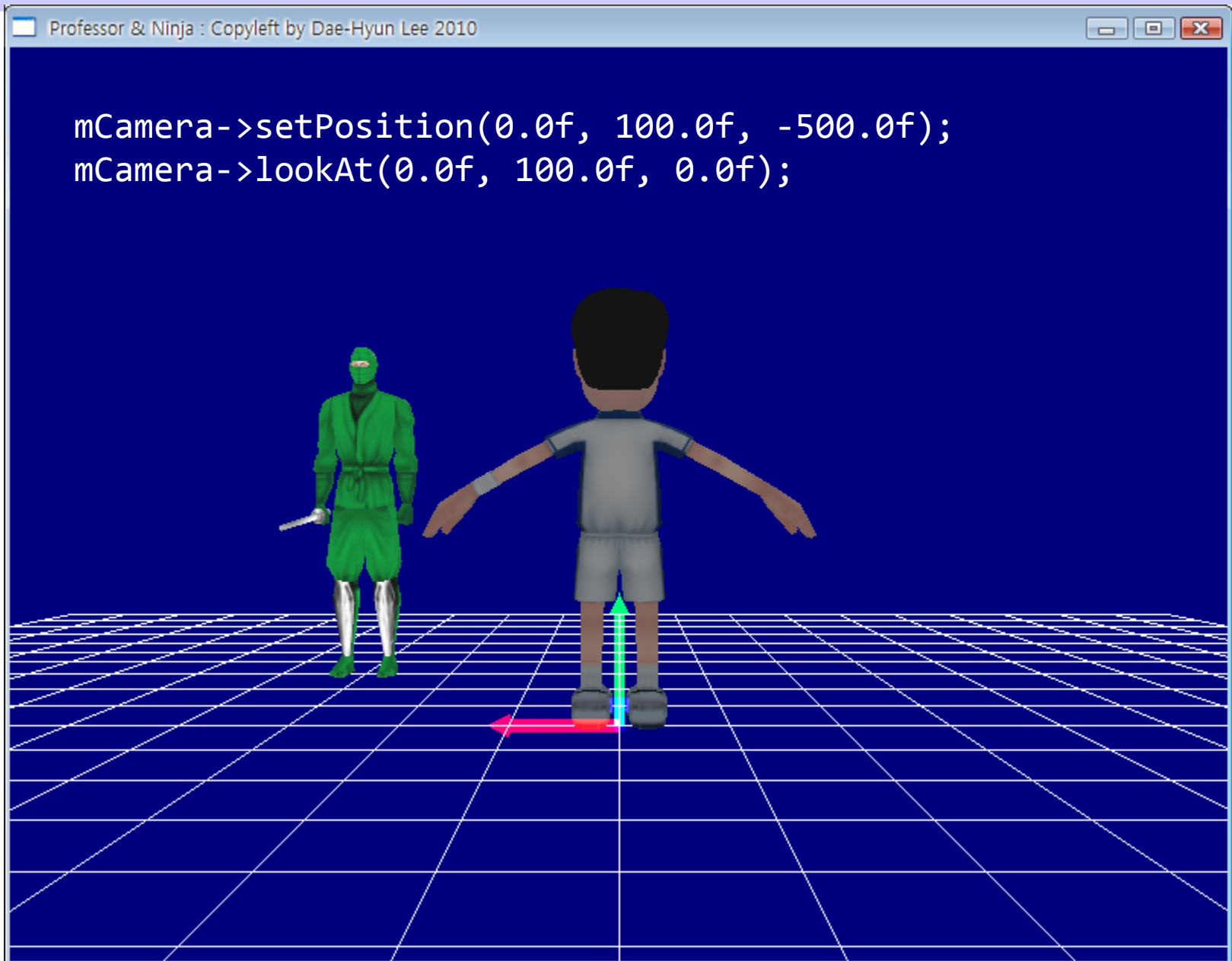
    // ... 후략
}
```



# SceneNode 클래스의 이동(translation) 함수

- void translate (const Vector3 &d, TransformSpace relativeTo=TS\_PARENT)
  - 장면노드를 지정된 벡터만큼 이동한다.
  - 디폴트로 부모노드의 transform space를 기준으로 하여 이동한다.

# 카메라의 위치 변경



# 씬노드와 엔터티의 획득 함수

- SceneNode\* Ogre::SceneManager::getSceneNode ( const String & *name* )
  - 이름을 가지고, 씬 노드를 얻는다.
- Entity\* Ogre::SceneManager::getEntity ( const String & *name* )
  - 이름을 가지고, 엔터티를 얻는다.

- 오우거 엔진은 오른손 좌표계를 사용.
- 씬 그래프 방식으로 씬 구성을 함.
- 씬 노드의 배치 및 이동 함수
  - setPosition() - 위치를 새로이 설정함.
  - translate() - 현 위치에서 지정된 벡터만큼 이동함.
  - 기준 원점은 부모 노드가 됨.
- 카메라의 위치 및 방향 변경 함수
  - setPosition()
  - lookAt()